EuroSPI 2010 Proceedings

Proceedings

The papers in this book comprise the industrial proceedings of the EuroSPI 2010 conference. They reflect the authors' opinions and, in the interests of timely dissemination, are published as presented and without change.

Their inclusion in this publication does not necessarily constitute endorsement by EuroSPI and the publisher.

DELTA Series about Process Improvement – 978-87-7398-152-8.

EuroSPI

EuroSPI is a partnership of large Scandinavian research companies and experience networks (SINTEF, DELTA, STTF), the ASQF as a large German quality association, the American Society for Quality, and ISCN as the co-ordinating partner.

EuroSPI conferences present and discuss practical results from improvement projects in industry, focussing on the benefits gained and the criteria for success. Leading European industry are contributing to and participating in this event. This year's event is the 17th of a series of conferences to which countries across Europe and from the rest of the world contributed their lessons learned and shared their knowledge to reach the next higher level of software management professionalism.

In EuroSPI 2010 we extended the scope of the conference from software process improvement to systems, software and service based process improvement. EMIRAcle is the institution for research in innovation in manufacturing and innovation, which came out as a result of the largest network of excellence for innovation in manufacturing in Europe. EMIRAcle key representatives joined the EuroSPI community and papers, case studies for process improvement on systems and product level will be included in future. EMIRAcle and Grenoble institute of technology are the hosts for of EuroSPI 2010.

EuroSPI Chairs

General Chair	Dr Richard Messnarz, ISCN, Austria/Ireland
EuroSPI Marketing Chair	Prof. Miklos Biro, Corvinus University, Hungary
Scientific Program Chair	Dr Andreas Riel, EMIRAcle - Grenoble Institute of Technology, France
Scientific Program Chair	Dr Rory O' Connor, Dublin City University, Ireland
Scientific Program Chair	Prof. Serge Tichkiewitch, Grenoble Institute of Technology, France
Industrial Program Chair	Risto Nevalainen FiSMA and Falconleader, Finland
Industrial Program Chair	Stephan Goericke, ISQI, Germany
Industrial Program Chair	Jørn Johansen, DELTA, Denmark
Industrial Program Chair	Mads Christiansen, DELTA, Denmark
Industrial Program Chair	Nils Brede Moe, SINTEF, Norway
Workshop Chair	Dr Richard Messnarz, ISCN, Austria/Ireland
Organizing Chair	Adrienne Clarke BA, ISCN, Ireland

Industrial Program Committee

Volker Ovi BACHMANN	SIBAC GmbH, Germany
Bo BALSTRUP	Center for Software Innovation, Denmark
Miklós BIRÓ	Corvinus University of Budapest, Hungary
Eva BRESKE	Robert Bosch GmbH, Germany
Mads CHRISTIANSEN	DELTA, Denmark
Taz DAUGHTREY	James Madison University, USA
Carol DEKKERS	Quality Plus Technologies, USA
Klaudia DUSSA-ZIEGER	Method Park Software AG, Germany
Damjan EKERT	ISCN GmbH, Austria
Detlef FEHRER	SICK AG, Germany
Stephan GOERICKE	iSQI GmbH, Germany
Philipp HAGENMEYER	ZF Friedrichshafen AG, Germany
Tim HIND	AXA, UK
Jørn JOHANSEN	DELTA, Denmark
Mika JOHANSSON	Finnish Software Measurement Association FiSMA, Finland
Gerhard LICHTENECKER	Magna Steyr, Austria
Richard MESSNARZ	ISCN, Austria/Ireland
Risto NEVALAINEN	FiSMA and Falconleader, Finland
Minna PIKKARAINEN	VTT Technical Research Centre of Finland, Finland
Alexander POTH	SQS Software Quality Systems AG, Germany
Samuel RENAULT	Centre de Recherche Public Henri Tudor, Luxembourg
Andreas RIEL	EMIRAcle - Grenoble Institute of Technology, France
Cristina ROMCEA	Conti Temic microelectronic GmbH, Germany
Tomas SCHWEIGERT	SQS Software Quality Systems AG, Germany
Gunther SPORK	Magna Powertrain, Austria
Maria STEFANOVA-PAVLOVA	Center for Innovation and Technology Transfer-Global, Bulgaria
Peter VON BRONK	Systemberatung Software-Qualität, Germany

EuroSPI Board Members

ASQ, http://www.asq.org

ASQF, http://www.asqf.de

DELTA, http://www.delta.dk

FiSMA, http://www.fisma.fi

ISCN, http://www.iscn.com

SINTEF, http://www.sintef.no

Editors of Proceedings

Richard Messnarz, ISCN, Austria/Ireland

Andreas Riel, EMIRAcle - Grenoble Institute of Technology, France

Damjan Ekert, ISCN, Austria

Mads Christiansen, DELTA, Denmark

Jørn Johansen, DELTA, Denmark

Sonja Koinig, ISCN, Austria

Welcome Address by the EuroSPI General Chair

Dr Richard Messnarz

EuroSPI is an initiative with 5 major goals (www.eurospi.net):

1. An annual EuroSPI conference supported by Software Process Improvement Networks from different EU countries.
2. EuroSPI supported the establishment of a world wide SPI Manifesto with SPI values and principles agreed among experts world wide. We build clusters of experts and knowledge libraries for these values and principles.
3. Establishing an Internet based knowledge library based on hundreds of experience reports contributed to EuroSPI since 1994.
4. Establishing a European Qualification Framework for a pool of professions related with SPI and management. This is supported by European certificates, exam systems, and online training platforms (European Certification and Qualification Association).
5. Establishing a world wide newsletter with articles from key industry and key European research associations helping to implement the SPI manifesto world-wide.

EuroSPI is a partnership of large Scandinavian research companies and experience networks (SINTEF, DELTA, STTF), the ASQF as a large German quality association, the American Society for Quality, and ISCN as the co-coordinating partner. EuroSPI collaborates with a large number of SPINs (Software Process Improvement Network) in Europe.

EuroSPI has established a newsletter series (newsletter.eurospi.net), the SPI Manifesto (SPI = **Systems, Software and Services Process Improvement**), an experience library (library.eurospi.net) which is continuously extended over the years and is made avail-able to all attendees, and a Europe wide certification for qualifications in the SPI area (www.ecqa.org, European Certification and Qualification Association).

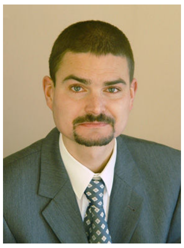
EuroSPI conferences present and discuss results from systems, software and services process improvement (SPI) projects in industry and research, focusing on the benefits gained and the criteria for success. This year's event is the 17th of a series of conferences to which international researchers contribute their lessons learned and share their knowledge as they work towards the next higher level of software management professionalism.

A typical characterization of EuroSPI was stated by a company using the following words: ”... the biggest value of EuroSPI lies in its function as a European knowledge and experience exchange mechanism for SPI and innovation”.

A cluster of European projects (supporting ECQA and EuroSPI) contribute knowledge to the initiative, including currently DEUCERT (EU Certificates Dissemination), iDesigner (integrated mechatronics designer), MONTIFIC (Financial SPICE Assessor), ELM (e-learning manager), LSSA (Six Sigma Related Qualification), ResEUr (Research to Entrepreneurship Strategies), etc. A pool of more than 20 qualifications has been set up.

Please join the community of cross company learning of good practices!

Welcome from the Organization Chair in France



Dr Andreas Riel

**Prof.
Serge Tichkiewitch**

The partnership between EuroSPI, EMIRAcle (the European Manufacturing and Innovation Research Association, www.emiracle.eu), and the ECQA (the European Certification and Qualification Association, www.ecqa.org) aims at putting complementary networked competences in Systems and Software Improvement at the disposal of members and partners from academia, industry and government. In a continuous effort to further enhance their scope of competences and influence on a European level, these organisations are always looking out for new members and partners.

Improvement of Products, Systems, Services and Processes has been the driver for the foundation of EMIRAcle in October 2007 with the mission to internationally promote, bundle and consolidate research and education in innovative product development. To this aim, the currently 23 EMIRAcle research institution members have intensive relationships with governmental organisations and fruitful, long-lasting collaborations with industrial and academic partners in numerous sectors. Among those are most notably automotive, aerospace, shipping, clothing and furniture. Key competences on the systems development level that are the core contribution of EMIRAcle in its strategic partnership with EuroSPI and the ECQA.

Acting as an international non-profit organisation based in Brussels (Belgium) and Grenoble (France), EMIRAcle provides its expertise in form of services in Co-Engineering, Co-Manufacturing, Co-Innovation, Co-Research and Co-Academy.
As one major outcome of the close collaboration with the EuroSPI and ECQA communities, EMIRAcle offers distance learning enabled training and European-wide certification for modern job roles in innovative product development. Current professions treated are Integrated Design Engineer, Researcher-Entrepreneur, and all levels of Lean Six Sigma Expert.

In the context of this successful relationship, we are very happy and honoured to host EuroSPI 2010 in the School of Industrial Engineering of Grenoble Institute of Technology, where the head of office of the EMIRAcle association is co-located with the G-SCOP Laboratory. This lab performs international top-level research in integrated product development, and the optimization of processes and logistics. Not only is it the first time in its long history that EuroSPI goes to France, but it is also the first big event that brings together communities which are highly influential in their domains. In this respect, EuroSPI 2010 in Grenoble marks a cornerstone in the building of a network of networks, which is ready to face the challenges of the development of modern products and services.

You are invited to join this highly international community in order to take an active part in shaping innovations!

Andreas Riel (andreas.riel@grenoble-inp.fr)
EMIRAcle Coordinator and Researcher at Grenoble Institute of Technology, France

Serge Tichkiewitch (serge.tichkiewitch@grenoble-inp.fr)
President of EMIRAcle and Professor at Grenoble Institute of Technology, France

Welcome by DELTA, Editors of the DELTA Improvement Series

Jørn Johansen

DELTA has been working with Software Process Improvement (SPI) for more than 16 years including maturity assessment according to BOOTSTRAP, SPICE and CMMI. DELTA has also been a partner in the EuroSPI conference from the very beginning 16 years ago. We are now for the 3rd time the publisher of the Industrial Proceedings from EuroSPI making it part of the DELTA series about Process Improvement.

Jørn Johansen is Manager of the DELTA Axiom department at DELTA. He has an M.Sc.E.E. from Ålborg University and more than 29 years experience in IT. He has worked in a Danish company with embedded and application software as a Developer and Project Manager for 15 years. Mr. Johansen has been involved in all aspects of software development: specification, analysis, design, coding, and quality assurance. Furthermore he has been involved in the company's implementation of an ISO 9001 Quality System and was educated to and functioned as Internal Auditor.

For the last 15 years he has worked at DELTA as a consultant and registered BOOTSTRAP, ISO 15504 Lead Assessor, CMMI Assessor and *ImprovAbility™* Assessor. He has participated in more than 40 assessments in Denmark and abroad for companies of all sizes. He was the Project Manager in the Danish Centre for Software Process Improvement project, a more than 25 person-year SPI project and Talent@IT, a 26 person-year project that involves 4 companies as well as the IT University in Copenhagen and DELTA. Currently Mr. Johansen is the Project Manager of SourceIT an 18 person-year project focusing on outsourcing and maturity. Mr. Johansen is also the co-ordinator of a Danish knowledge exchange group: Improving the Software Development Process, which is the Danish SPIN-group.

Contact: Jørn Johansen, DELTA, Denmark, e-mail: joj@delta.dk

Mads Christiansen

Mads Christiansen has an M.Sc.E.E. from DTU (Danish Technical University) and more than 31 years experience in product development and IT. He has worked for 19 years in a Danish company with embedded and application software as a Developer and Project Manager. Mr. Christiansen has been involved in all aspects of software development: specification, analysis, design, coding, and quality assurance and managing outsourced projects in Denmark and USA.

For the last 13 years he has worked at DELTA as a consultant in SPI (requirements specification, test, design of usable products and development models). Currently Mr. Christiansen works with eBusiness and as Innovation Agent. Mr. Christiansen is also *ImprovAbility™* Assessor and Trainer of *ImprovAbility™* project Assessors.

Contact: Mads Christiansen, DELTA, Denmark, e-mail: mc@delta.dk

Contents

Experience Session 3: SPI and People / Acceptance

Experience Session 4: SPI and Measurement

Experience Session 5: SPI and Testing

Experience Session 6: SPI and Agile

Experience Session 7: SPI and Systems Improvement

Experience Session 8: SPI and Management

Experience Session 9: SPI and Knowledge

Experience Session 10: SPI and GSD

Software Improvement Through Benchmarking: Case Study Results

Dr. Hans Sassenburg, Dr. Lucian Voinea, Peter Wijnhoven

Abstract

Since the early nineties in the previous century, many organizations have substantially invested into software process improvement. Starting in the military industry, the concept of process improvement has nowadays been widely adopted in many other industry segments. It is one the few initiatives that have sustained over time, this in contrast to many hypes. Available models and standards help to define improved processes not only on paper, but also to institutionalize them in the daily way of working. However, a justified and often raised question is what the payoff is. Does the quality of products increase? Has efficiency improved? Are products being brought to the market faster? And an overall question: compared to what? Benchmarking is a technique that makes use of external comparisons to better evaluate real capability and identify possible actions for the future. As such, it is an important instrument to drive improvement efforts. Using a best practice set of Key Performance Indicators to benchmark capability in several industrial case studies, no strong correlation could be found between real capability and maturity levels. Satisfying models or standards is no guarantee for real performance improvements. It is recommended to focus on a multi-dimensional assessment of the capability of an organization and derive improvements from benchmarked results.

Keywords

Benchmarking, software process improvement, Key Performance Indicator, metrics, capability, performance.

1 Introduction

We manage things "by the numbers" in many aspects of our lives. These numbers give us insight and help steer our actions. Software metrics extend the concept of "managing by the numbers" into the realm of software development. The software industry still isn't doing a very good job at managing by the numbers. Intuition prevails, where numbers should be used. Most projects are still managed by three indicators only: scheduled dead-line, overall budget and removal of critical defects towards the end of the project. This is a narrow view on a multi-dimensional problem. Compare it with a contesting in a Formula 1 race looking only at the fuel- and speedometer. Neglecting oil pressure, tyre pressure, fuel stop planning, weather conditions, and many other variables, will definitely cause you to lose the race.

Successful (software) organizations have found six measurement related objectives extremely valuable [Sassenburg 2006]:

- Knowing the capability of one's organization through the analysis of historical project data. In addition, one's own capability may be benchmarked against industry averages.
- Making credible commitments in terms of what will be delivered when against what cost. This involves project estimation based on known capability and analyzed requirements.
- Investigating ways to optimize project objectives (on dimensions like schedule or cost). This involves developing and comparing different project alternatives.
- Managing development once it starts. This involves project management, but more than generating simple PERT and Gantt charts.
- Deciding when a product can be released. This is a trade-off between an early release to capture the benefits of an earlier market introduction, and the deferral of product release to enhance functionality or improve quality.
- Analyzing the impact of new initiatives by assessing how capability is affected in which areas. This prevents organizations from chasing hypes.

Being able to meet these objectives requires an implemented measurement process, that converts measured process and product attributes to meaningful management information. Within a project or organization, it is often easy to get people enthused about metrics. But all too often, this enthusiasm does not translate into action. Even when it does, it is unlikely to be sustained and people might get lost in incomplete details. Getting too little or too much data is easy, identifying the relevant data and converting it to meaningful information for everyone is the challenge. Management needs the ability to step back from the details and see the bigger picture. Dashboards with the right information perform that function. They should support answering the following questions as listed in Figure 1.

Category	Typical questions
Project performance	How predictable is the performance of projects?
Process efficiency	How efficient is the development process?
Product scope	How large and stable is the scope of the planned effort?
Product quality	What is the quality of the resulting outcome/product?

Figure 1: Typical management questions to answer.

2 Best Practice KPIs

The critical success factor here is defining the appropriate Key Performance Indicators (KPIs) in each category. The goal of these KPIs is to foster greater visibility and faster reaction to opportunities and threats, hereby enabling informed decision-making. Based on research and industrial experience in many different organizations, a coherent set of KPIs has been selected that answers the questions of Figure 1. These KPIs represent current best practice in industry [Sassenburg 2009]. Efforts undertaken at improving development capability should have demonstrable effects on each of these KPIs, as indicated in the example in the most right column of Figure 2.

Category	Typical Key Performance Indicators	Effect
Project performance	Schedule Effort Staffing rate (manpower buildup profile [Putnam 1992, 1997]) Productivity (LOC/hour or other ratio)	- - + +
Process efficiency	Core activities (% of total effort) Support activities (% of total effort) Prevention activities (% of total effort) Appraisal/rework activities (% of total effort)	+ + + -
Product scope	Number of features Percentage of deferred features Size (in KLOC or other unit) Re-use level (percentage of size)	- - - +
Product quality	Complexity (architectural level, source code level) Test coverage (unit, integration, system testing) Defect density (released defects per KLOC or other unit) Cumulative defect removal efficiency	- + - +

Figure 2: Overview best practice KPI set [Sassenburg 2009].

3 *Software Benchmarking*

To manage process efficiently, software development organizations must focus on understanding how they perform. Key measures of performance include productivity rate, project time-to-market, and project deliverable quality. Assessing the results of software process is a starting point, but it does not provide context by itself – it is not sufficient for a complete understanding of status, how, and where to improve. Benchmarking is comparing one's own performance and operating methods with industry averages or best in class examples, with the goal of locating and improving one's own performance [Camp 1989]. As such, it is an important instrument to prioritize and drive improvement efforts. For many years the lack of readily available benchmark data prevented software managers from analyzing the real economics of software. Many (process) improvement initiatives resulted in satisfying standards/models instead of tangibly improving measured capability. Through the work of Capers Jones [2008] and others, now data on thousands of projects is available to the software industry. This enables making solid business decisions about software development practices and their results in terms of productivity and quality. It allows using economics as the basis of quality analysis [Boehm 2000, Wagner 2007] and balancing between cost versus productivity and quality.

In a series of assignments conducted by the authors in the period 2008-2010, the presented best practice KPI set was used to measure the performance capability of organizations. This implied the assessment of values for each indicator. Two important conclusions were drawn regarding the availability and quality of the data found [Sassenburg 2009]:

- Availability. Many low maturity organizations have the opinion that they lack quantitative data. In most cases, this is not true. Although not centrally stored, many sources of data can normally be identified. The challenge is to identify these data sources and analyze them in order to obtain useful information.
- Quality. Higher maturity organizations often have the opinion that they have access to rich sources of information. In most cases, the contrary is true. Despite many measurements and resulting data, the conversion of these data to useful management information is a weakness for many organizations. In addition, the availability of clearly defined measurement constructs and validation of measurement data before consolidation are exceptions. This leads to problems with respect to consistency, completeness and reliability [Maxwell 2001].

In case a software organization does not have sufficient, reliable benchmark data available, they can make use of the published data of Capers Jones [2008] and ISBSG [www.isbsg.org].

4 Case Study Results

The best practice KPI set was used in several benchmarking studies conducted by the authors in the period 2008-2010. Presented here are the case study results of two different organizations as an example. The results are representative for many other studies. Both organizations develop embedded systems for the business-to-business market. Although the systems are not consumer products, volumes are fairly high, varying from hundreds to many thousands. One system is considered safety critical, the other system is complicated due to regulations in the area of security of information which may not be manipulated in any way. In early discussions with both organisations, it was revealed that process improvement is institutionalized since many years and that CMMI Maturity Level 3 compliance is within reach. In both cases, some common issues had to be dealt with:

- So far, no strong benchmarking data has been published for feature deferral ratios, re-use levels, test coverage during different test types and complexity. Instead, we used our own data collected from previous studies to benchmark against.
- A common issue in embedded systems is the way feature size is calculated. Although function points are preferred as size measure, the only data available was lines of code. The backfiring technique was used to convert lines of code to function points [Jones 1995]. In both cases the programming language used was C, the resulting number of function points was close to 1‘000.

In the following paragraphs we highlight remarkable results from the studies that lead to further analysis and improvement efforts. Without benchmarking against industry data, these improvement opportunities would most likely have been unnoticed and left unaddressed.

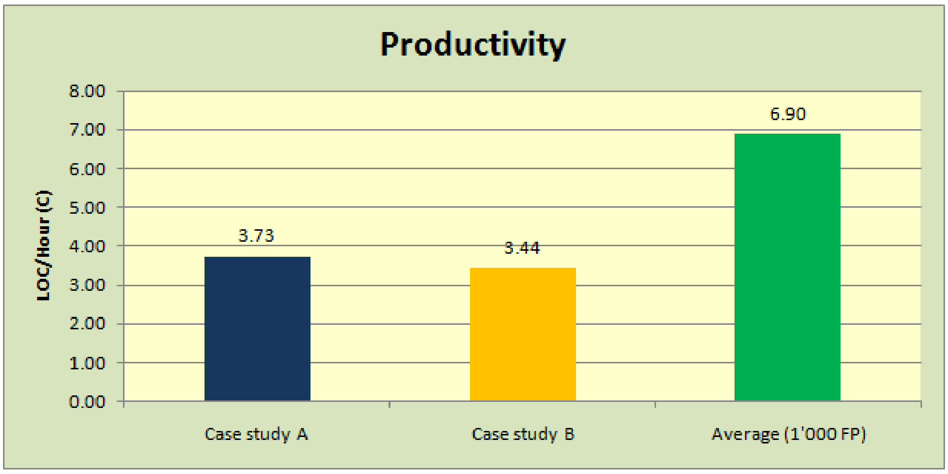


Figure 3: Productivity benchmarking.

Figure 3 shows how both case studies compare to benchmarking data regarding productivity in function points per staff month [Jones 2008]. It is obvious that both cases show a much lower productivity level than industry average. In a competitive market this is important to notice, analyse and improve. In these cases, the lower productivity was believed to be a consequence of safety requirements for case study A and security requirements for case study B.

Further remarkable results were found with respect to process efficiency. Regarding process efficiency, a Cost-of-Quality approach is used, based on work of Juran [1988] and Crosby [1979]. Distinction is made between four categories [Sassenburg 2010]:

- Core. Costs in this category are essential and bring direct value to a customer by changing the product in some way: requirements, architecture, coding.
- Support. Costs in this category are essential but do not bring direct value to a customer: project management, configuration management, administrative support.
- Prevention. These are costs incurred to prevent (keep failure and appraisal cost to a minimum) poor quality: quality planning, process improvement teams, reviews, inspections.

- Appraisal, rework. These are costs incurred to determine the degree of conformance to quality requirements: mainly testing and defect detection/removal.

Using these definitions, it will be obvious that improving efficiency will normally mean reducing the overall costs by reducing appraisal and rework costs. This can be achieved by increasing prevention costs. This Cost-of-Quality approach is relatively easy to implement and to use. From a project plan, all scheduled activities can be mapped to the four categories and the ratios can be calculated. Note that this enables management to validate the feasibility of a project plan if ratios from the past are known. If any of the projected ratios deviates substantially from values realized in the past, there should be assignable causes for this.

In Figure 4, the case studies results are compared to industry averages.[1] In both cases, a ratio for Appraisal/Rework of approximately 50% was found, which is very high, not only compared with industry averages but as an absolute figure as well. Also here, analysis led to the conclusion that this is a consequence of safety requirements for case study A and security requirements for case study B. However, still much higher ratios for prevention would be expected. This became one of the focus points for improvement activities.

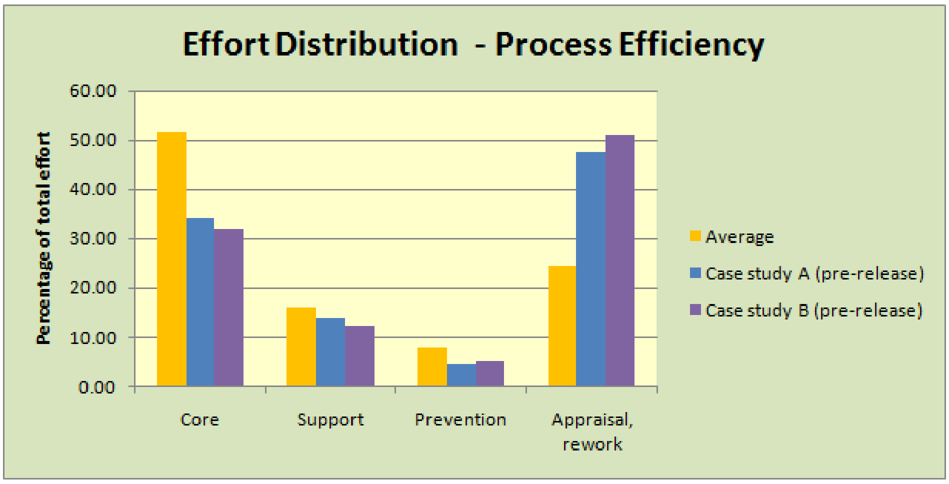


Figure 4: Process efficiency benchmarking.

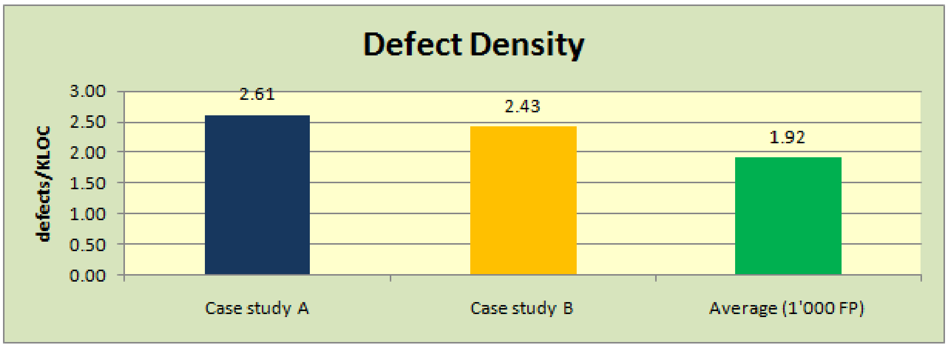


Figure 5: Defect density benchmarking.

In Figure 5, the case study results are compared to benchmarking data regarding defect density in defects per 1‘000 lines of code [Jones 2008]. While for software with safety requirements and security requirements one might expect having better figures than industry average, the contrary is the case here. The answer of management in both cases was that after releasing the software, many additional tests took place and delivery was to a limited number of users only. In other words, the defect density that finally reached the end-user was less high. On the other hand, they acknowledged that post-release maintenance and support costs were extremely high and should be reduced.

[1] Benchmarking ratios were obtained by mapping published project data [Jones 2008] to the four categories.

Figure 6 shows that the high defect density finds its origin in the low defect removal efficiency compared to industry average [Jones 2008]: too many defects remain uncovered during development and are detected post-release time. In-depth analysis revealed that the primary causes for low removal efficiency were highly complex architectures and code implementations. As a result, test coverage was very low.

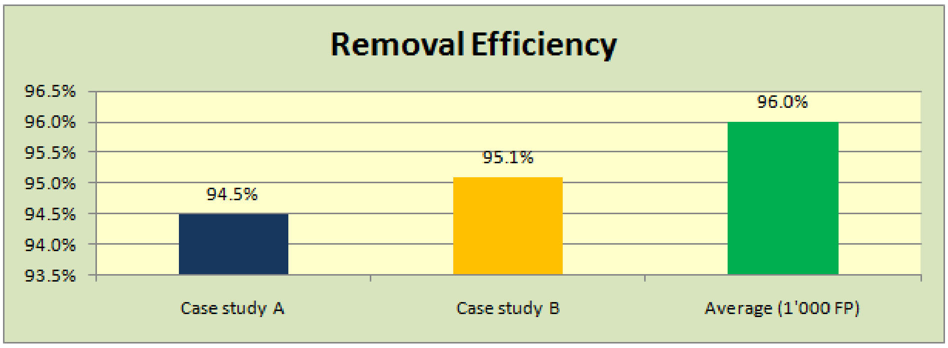


Figure 6: Removal efficiency benchmarking.

In both cases, it was very clear to all stakeholders that there are two main weak areas:

- The effort distribution revealed a very insufficient process, with a high ratio for Appraisal/rework. If post-release efforts for fixing defects would be included, the ratio would even become substantially higher.
- The architecture and code quality in both cases were low. At architectural level, high fan-out values were indicators for low cohesion and tight coupling resulting in a high level of change propagation. At code level, high cyclomatic complexity [McCabe 1976] values were found. As a result, problems arise regarding understandability, modifiability and verifiability.

These two areas were considered the primary causes for low overall capability and were used as the basis to define improvements. The availability of quantitative and benchmarked data helped both organizations to derive a solid business case for improvements.

5 Conclusions

Do higher maturity levels automatically lead to increased performance? In the studies performed, no strong correlation could be found between capability, expressed using the sixteen indicators, and maturity levels. On the other hand, process improvement makes sense, as it standardizes development processes. This creates transparency with respect to roles, responsibilities, activities and deliverables. However, standardization is no guarantee for real capability improvement. That is why aiming at for instance higher CMMI levels only is considered a wrong approach.

The recommendation is to focus on a multi-dimensional assessment of the capability of an organization and derive improvements from benchmarked results. A first step will be baselining the current capability using the best practice KPI set. In case measurements are not in place, the first improvement actions are identified. As a second step, realistic target values must be determined for a given period of time. The gap between target and actual values is the basis for deriving improvement steps. By focusing on the primary causes for low capability, the chances of sub optimization are reduced or even eliminated. The interesting fact is that improvements can only be achieved by changing the way of working. And of course, a validated improved way of working should be shared: considering standardization across projects becomes a logical process.

This brings us to the conclusion of this paper. Process improvement and standardization should not be a goal in itself. Real capability improvement is achieved by taking a quantitative view on processes and products, and setting realistic and quantified improvement targets. Using the presented KPI set in a benchmarking study reveals real capability by identifying strengths and weaknesses. This provides the basis for deriving improvements that make sense, whereas implementing and sustaining such improvements are structured by the use of process maturity models.

Literature

[Boehm 2000] Boehm, B.W., Sullivan, K.J., "Software Economics: A Roadmap", *ICSE, Proceedings of the Conference on The Future of Software Engineering*, 2010.

[Camp 1989] Camp, R.C., *Benchmarking: the search for industry best practices that lead to superior performance*. Milwaukee, Wisconsin: Quality press for the American society for quality control, 1989

[Crosby 1979] Crosby, P.B., "*Quality is Free",* New York: McGraw-Hill, 1979.

[Jones 1995] Jones, C.J., "Backfiring: converting lines of code to function points", *IEEE Computer, November 1995*.

[Jones 2008] Jones, C.J., *"Applied Software Measurement"*, McGraw-Hill, 2008.

[Juran 1988] Juran, J.M., Gryna, F.M., *"Juran's Quality Control Handbook"*, 4th ed., New York: McGraw-Hill Book Company, 1988.

[Maxwell 2001] Maxwell, K. D., "*Collecting Data for Comparability: Benchmarking Software Development Productivity*", IEEE Software, Sep/Oct 2001.

[McCabe 1976] McCabe, T.J., "A Complexity Measure", *IEEE Transactions on Software Engineering*, Vol. 2, pp. 308-320, 1976.

[Putnam 1992] Putnam, L.H., Myers, W., *"Measures for Excellence: Reliable Software On Time Within Budget"*, Yourdon Press Computing Series, 1992.

[Putnam 1997] Putnam, L.H., Myers, W., *"Industrial Strength Software: Effective Management Using Measurement"*, IEEE Computer Society, 1997.

[Sassenburg 2006] Sassenburg, H., *"Design of a Methodology to Support Software Release Decisions*", Doctoral thesis, University of Groningen, 2006.

[Sassenburg 2009] Sassenburg, H., Voinea, L., *"Standardization does not necessarily imply Performance Improvement"*, Automatisering Gids (in Dutch), Sept. 4th, 2009.

[Sassenburg 2010] Sassenburg, H., Wijnhoven, P., *"From Testing to Designing"*, Automatisering Gids (in Dutch), March 20th, 2010.

[Wagner 2007] Wagner, S., "Using Economics as Basis for Modelling and Evaluating Software Quality", *ICSE, Proceedings of the First International Workshop on The Economics of Software and Computation*, 2007.

Author CVs

Dr. Hans Sassenburg

Dr. Hans Sassenburg received a Master of Science degree in electrical engineering from the Eindhoven University of Technology (Netherlands) in 1986 and a PhD degree in economics from the University of Groningen (Netherlands) in 2006. He worked as an independent consultant until 1996, when he co-founded a consulting and training firm. This company specialized in software process improvement and software architecture and was sold in 2000. In 2001 he moved to Switzerland, where he founded a new consulting and training firm SE-CURE AG (www.se-cure.ch). In addition, he has been a visiting scientist at the Software Engineering Institute (www.sei.cmu.edu) since January of 2005. In 2009, he co-founded the Software Benchmarking Organization (www.sw-benchmarking.org), a consortium of international accredited partners. Dr. Sassenburg is an internationally published author on software engineering and economics.

Dr. Lucian Voinea

Dr. Lucian Voinea received a Professional Doctorate in Engineering degree (PDEng) from the Eindhoven University of Technology (Netherlands) in 2003 and a PhD degree in computer science from the same university in 2007. Starting from 1999, he worked as a freelance software engineer for companies in Europe and North America. In 2007 he co-founded SolidSource (www.solidsourceit.com), a company that provides solutions to support software development and maintenance. In 2009 he co-founded the Software Benchmarking Organization (www.sw-benchmarking.org), a consortium whose aim is to create a framework for benchmarking capability in the software development industry. Dr. Voinea is an internationally published author on software engineering and visualization topics.

Peter Wijnhoven

Peter Wijnhoven holds a BSc degree in mechanical engineering and a BSc degree in computer science. He has over 25 years of experience in embedded systems development. During this period he fulfilled a wide variety of functions, ranging from programmer to architect, from team to project leader, and from consultant to manager consulting group of Sioux Embedded Systems (www.sioux.eu). Mr. Wijnhoven is active as SPI- and business consultant since 1996, and is managing a group of consultants in the field of development process improvement, architecture, and integration and verification since 2000. During these years he has been actively involved in process improvement projects in multi-disciplinary R&D organizations in Europe, North America, and Asia.

Med-Adept: A Lightweight Assessment Method for the Irish Medical Device Software Industry

Fergal Mc Caffery, Valentine Casey

Regulated Software Research Group
Dundalk Institute of Technology
And member of
Lero- the Irish Software Engineering Research Centre,
fergal.mccaffery@dkit.ie, val.casey@dkit.ie

Abstract

In this paper we describe how a lightweight assessment method was developed to educate Irish software development organisations in relation to becoming medical device software suppliers.

Keywords

Medical Device Assessment method, Software Process Improvement (SPI)

1 Introduction

Two important sectors for the growth of the Irish economy are Medical Devices and Information and Communications Technology (ICT). The Medical Device (MD) and diagnostic industry is a cornerstone of the Irish economy. The sector has been identified, by both the Irish Industrial Development Authority and Enterprise Ireland, as a key growth area. Of the world's top 14 MD companies 11 have a base in Ireland; the indigenous base is also evolving rapidly; over 80% of the companies in this sector are involved in significant innovation.

At present the Irish MD industry is focused on manufacturing. The sector is particularly dependent on the continued use of stents in the treatment of cardiovascular disease. Therefore, either the discovery of an effective drug treatment for vascular plague or the general migration of the manufacture of medical devices to low labour cost locations could negatively affect the future growth of the Irish MD sector. The Expert Group of Future Skills Needs (an Irish Government advisory body) [1], has highlighted ICT forms a major part of the MD sector globally, however in Ireland, where a significant ICT sector exists, only a small part of the MD sector involves ICT. The report also highlighted that Ireland does not have a strong presence in the production of electronic based medical devices (which would include substantial software development).

There is therefore an opportunity to reduce the sectors dependence on stent manufacture by supporting the development of a software based MD industry in Ireland. This can be achieved through providing a range of services that will encourage existing indigenous MD companies to consider developing software and encourage multi-national MD companies to consider Ireland as a location for developing software. It will also provide opportunities for Irish software companies to develop software for the MD industry. However, to take up this opportunity they will need to demonstrate that their software development processes are both capable of producing high-quality software and achieving regulatory compliance. Thus, there is a need for a low resource software process assessment to determine the current state of their software development practices.

2 Software for the Medical Device Industry

Due to the safety-critical nature of medical devices, organisations developing MD software are expected to produce high-quality software through the use of defined processes. To tackle these issues, governments have put in place regulatory bodies to define regulatory systems for medical devices and to ensure that only safe medical devices are placed on the market. To aid the control of medical devices, regulatory bodies have adopted a classification scheme. The device manufacturer is obliged to establish and perform both pre-market and post-market duties as defined in the quality system regulations. The quality system requirements for Europe are defined in the ANSI/AAMI/ISO 13485 standard [2] and the requirements for the U.S.A. are defined in the 21 CFR Part 820 Quality System Regulations [3]. Applicable requirements are typically directly related to the class of the device. The regulatory or approved body, through its audits, checks conformance to the quality system requirements periodically.

In an attempt to address the vagueness of the regulatory requirements, the FDA published separate regulatory guidance documents for required software activities [4,5,6]. In Europe, many organisations rely on the regulatory guidance documents from the FDA due to insufficient guidance provided for the equivalent CE marking process. Additionally, there has been a steady progression in the MD software sector with the release of new and updated standards in an attempt to address the knowledge gap that exists between the high-level regulatory requirements and the low level detail and knowledge required to adequately satisfy those requirements.

Whenever we mention MD guidelines within this paper we refer to the following medical device standards and guidelines: ANSI/AAMI/IEC 62304 [7], FDA [4,5,6,8], European Council Guidelines [9], ISO 14971 [10], EN 60601-1-4 [11], GAMP 5 [12], TIR 32 [13], AAMI/IEC 61508 [14], and IEC 60812 [15].

3 Software Process for Medical Device Software Development

With the development of formal SPI models such as CMMI® [16] and ISO/IEC 15504-5 [17] researchers within regulated environments such as the Space and Automotive industries started to investigate how they could utilise these models to improve the practices within their industry domains. However, they discovered that although the existing models are comprehensive, neither CMMI® nor ISO/IEC 15504-5 addressed all of the regulatory needs and constraints of their specific industries. Researchers therefore sought to adopt the practices within these models while also expanding on them to account for regulatory requirements within their own domains. This resulted in the production of full SPI models tailored specifically for the Space domain [18] and Automobile domain - Automotive SPICE [19].

The authors are in the process of developing Medi SPICE which will be a comprehensive software process assessment model for the medical device industry [20] which is based on the AMMI/IEC 62304 standard, associated MD standards and guidelines (listed at the end of section 2) and ISO/IEC 15504-5. Medi SPICE like ISO/IEC 15504-5 and Automotive SPICE will contain both a Process Reference Model (PRM) and Process Assessment Model (PAM) that provide comprehensive coverage of the FDA and European Council guidelines, and associated standards (e.g. ISO 14971, IEC 60601-1-4,TIR 32 and GAMP) for the complete software development lifecycle. The overall objective of Medi SPICE is to provide a conformity assessment scheme to support first, second or third party assessment results that may be recognised by the regulatory bodies. The Medi SPICE PRM and PAM is being released in phases and once complete, will consist of a defined set of software processes that will contain base practices which when utilised will assist medical device software development organisations to fulfil the regulatory guidelines and standards of the medical device industry.

4 The Med-Adept Method

One of the main goals of the Regulated Software Research Group in Dundalk Institute of Technology is to support the growth of a MD software development industry within Ireland. The authors previously developed the Adept method [21] which was based upon the ISO/IEC 15504-5 and CMMI® models. Consequently, we based the Med-Adept method upon relevant process areas from the CMMI® and ISO/IEC 15504-5 models and included input from AAMI/IEC 62304. This therefore enabled the existing Adept questions to be established as the foundation for the new method and for additional questions to be added to enable coverage of relevant AAMI/IEC 62304 process areas. The Med-Adept method consists of an assessment component for each process that is deemed applicable for software development organisations wishing to become medical device software developers. However, even though each assessment component adopts a CMMI process area name, it provides coverage of CMMI®, ISO/IEC 15504-5 and AAMI/IEC 62304 practices through containing questions that relate to CMMI®, ISO/IEC 15504-5 and AAMI/IEC 62304

A key decision in the development of the Med-Adept method was to decide what process areas should be included. The process areas included in Med-Adept were chosen because:-

A. They have process area counterparts included within the AAMI/IEC 62304 standard;
B. They were previously included in the Adept method;

We then analysed each of the CMMI® process areas using the factors in Table 1 (see below).

Table 1. Suitability of CMMI® process areas for inclusion in Med-Adept method

CMMI® Process Area	Satisfies A	Satisfies B
Requirements Management	**Yes**	**Yes**
Project Planning	**Yes**	**Yes**
Project Monitoring & Control	**Yes**	**Yes**
Configuration Management	**Yes**	**Yes**
Measurement & Analysis		Yes
Process & Product QA	**Yes**	**Yes**
Supplier Agreement Management		
Requirements Development	**Yes**	**Yes**
Technical Solution	**Yes**	**Yes**
Verification	**Yes**	**Yes**
Product Integration	**Yes**	**Yes**
Validation	**Yes**	**Yes**
Organisational Process Focus		
Organisational Training		
Organisational Process Definition & IPPD		
Integrated Project Management & IPPD	Yes	
Risk Management	**Yes**	**Yes**
Decision Analysis & Resolution	Yes	
Organisational Process Performance		
Quantitative Project Management		
Organisational Innovation & Deployment		
Causal Analysis & Resolution	Yes	

Table 1, illustrates, that eleven of the twenty-two process areas from the CMMI® model satisfied both factors and should be included in Med-Adept.

4.1 What processes are included in Med-Adept?

In addition to the Med-Adept method enabling assessment against eleven CMMI® process areas it should also assess ISO/IEC 15504-5 and AAMI/IEC 62304 processes that are related to the eleven selected CMMI® process areas. The procedure for selecting the Med-Adept process areas was as follows:-

Step 1. Select one of the eleven CMMI® process areas (previously included in Adept – satisfies B in table 1);

Step 2. Serially scan this process area against the following list of 16 AAMI/IEC 62304 processes and select related ISO/IEC 15504-5 processes:- Risk Management, Configuration Management, Software Requirements Analysis, Software Development Planning, Software Architectural Design, Software Detailed Design, Software Integration, Software Unit Implementation and Verification, Integration Testing, Software System Testing, Quality Assurance, Software Release, Software Maintenance, Software Problem Resolution, Documentation, Software Safety Classification.

Step 3. Repeat Steps 1 and 2 for each of the eleven CMMI® process areas.

As a result of performing these steps the CMMI® to AAMI/IEC 62304 and ISO/IEC 15504-5 processes (software related) area linkages were determined (see table 2) and the Med-Adept method provides coverage of 11 CMMI® process areas, 12 ISO/IEC 15504-5 and 11 AAMI/IEC 62304 processes.

Table 2. CMMI® to AAMI/IEC 62304 and ISO/IEC 15504-5 process linkages

Med- Adept Processes		
Adept Processes		
Selected CMMI Process Area	**Selected ISO/IEC 15504-5 Process**	**AAMI/EC 62304 Process**
Risk Management	Risk Management	Risk Management
Configuration Management	Configuration Management	Configuration Management
Requirements Management Requirements Development	Requirements Elicitation Software Requirements Analysis	Software Requirements Analysis
Project Planning Project Monitoring & Control	Project Management	Software Development Planning
Technical Solution	Software Design Software Construction	Software Architectural Design Software Detailed Design
Product Integration	Software Integration	Software Integration
Validation Verification	Software Testing Verification Validation	Software Unit Implementation and Verification Integration Testing Software System Testing
Process and Product Quality Assurance	Quality Management System	Quality Assurance
		Software Release Software Maintenance Software Problem Resolution Documentation Software Safety Classification

It can also be observed that Med-Adept does not provide coverage of 5 AAMI/IEC 62304 processes. However, the main purpose of Med-Adept is to provide a low overhead assessment that will educate organisations in relation to generic SPI and in particular medical device software development process. Therefore, it is not intended to provide comprehensive coverage of CMMI, AAMI/IEC 62304 and ISO/IEC 15504-5 processes, but rather a starting point through focusing upon the processes that will provide the most benefit to organisations.

To encourage uptake of the Med-Adept assessment by Irish software organisations we wish to reduce the cost and time associated with the assessment. On-site interviewing is restricted to one day as this proved attractive to companies in relation to performing the Adept assessment [22]. Consequently, an Med-Adept assessment method will be limited to providing coverage of four selected CMMI® and (the related) AAMI/IEC 62304 and ISO/IEC 15504-5 processes as the aim of Medi-Adept is to introduce organisations to medical device software processes in a low overhead manner. Companies wishing to be assessed in more than 4 of these processes will then be able to extend the assessment across additional days.

4.2 The Stages of the Med-Adept Method

The Med-Adept method is divided into eight stages and the assessment team consists of two assessors who conduct the assessment between them.

Stage 1 (*Develop Assessment Schedule and Receive Site Briefing*) involves a preliminary meeting between the assessment team and the software company. The assessment team will discuss the main drivers for the company embarking upon a Med-Adept assessment During **stage 2** (*Conduct Overview Briefing*) the lead assessor provides an overview of the Med-Adept method for members of the organisation who will be involved in subsequent stages. **Stage 3** (*Analyse Key Documents*) provides a brief insight into project documentation. The primary source of data for the Med-Adept method is through a series of process area interviews conducted during stage 4. The main part of the Med-Adept method is **stage 4**. In this stage key staff members from the assessed organisation are interviewed. In an attempt to reduce the overhead of the assessment we restrict the scope of the assessment to 4 process areas (there are a maximum of 4 interviews). Each interview is scheduled to last approximately 1.5 hours. Each interview involves two assessors, and at least one representative from the company is present for each process area interview.

Table 3, illustrates (for example, the processes of risk management and configuration management) that the process interviews within an Med-Adept assessment includes additional questions to provide coverage of relevant AAMI/IEC 62304 and ISO/IEC 15504-5 processes in addition to the CMMI process areas. When developing the interview questions we mainly looked at the base practices and did not perform a detailed investigation into similarities and differences between CMMI®, AAMI/IEC 62304 and ISO/IEC 15504-5. Instead we checked the relevant interview questions from the Adept method to see if they covered their counterpart in AAMI/IEC 62304 and ISO/IEC 15504-5 processes.

Table 3. Breakdown of Med-Adept Questions

AHAA Interviews	**No. of Adept Questions**	**No. of New Questions**	**No. of Med-Adept Questions**
Risk Management	39	23	62
Configuration Management	39	2	41

Within Adept 39 questions were used to provide coverage of the specific goals of the CMMI® and the base practices of ISO/IEC 15504-5 for risk management. Med-Adept is more comprehensive and has 62 scripted questions for risk management (see Table 3). Med-Adept not only contains CMMI and ISO/IEC 15504-5 based questions, but also 23 additional questions that are specifically related to the risk management process of AMMI/IEC 62304 and other associated medical device standards and regulations. The configuration management process has 2 additional questions added to meet the specific requirements of Med-Adept.

On completion of a Med-Adept assessment companies will receive feedback regarding the current state of their practices in relation to CMMI, AAMI/IEC 62304 and ISO/IEC 15504-5 (unless a company specifies that they are only interested in one of the models).

Stage 5 (*Generate Assessment Results and Create the Findings Report*) is a collaborative exercise between the assessors which results in the development of the findings report. The resultant findings report consists of a list of strengths, issues and suggested actions for each of the process areas evaluated. The findings report is developed through reviewing the interview notes for each of the 4 assessed process areas. **Stage 6** (*Deliver the findings report*) involves presenting the findings report to the staff in the assessed organisation who participated in the interviews. **Stage 7** (*Develop a SPI Path with the Company*) involves collaborating with staff from the assessed company to develop a roadmap that will provide guidance to the company in relation to practices that will offer the greatest benefits in terms of the organisation's business goals. Companies wishing to become medical device software developers will be recommended to focus upon establishing working practices that will assist them to adhere to AAMI/IEC 62304 practices and to prepare them for future Medi SPICE assessments. **Stage 8** (*Re-assess the SPI Path and Produce a Final Report*) involves revisiting the assessed company approximately 3 months after the completion of stage 7 and reviewing progress against the SPI path that was developed in stage 7. The outcome of this stage will be an updated SPI path and a final report detailing the progress that has been accomplished along with additional recommendations. This stage is important as it provides feedback and assistance to the assessed company after a period of time. This stage also assists in compiling research material in terms of SPI experiences.

5 Observations from a Med-Adept assessment

The Med-Adept assessment was performed in an Irish MD organisation MedSoft (a pseudonym). MedSoft were aware of CMMI and ISO/IEC 15504-5, but they had not previously utilized these process improvement models. Before undertaking the assessment they were not familiar with the AAMI/IEC 62304 MD standard. On completion of the assessment the software development manager stated it had been beneficial to MedSoft in a number of ways. These included the provision of high-level training in relation to CMMI and ISO/IEC 15504-5 in the areas of risk management and configuration management. In addition they gained an insight into the MD regulations that are required to achieve compliance for these specific practices. Another important outcome was the assessment provided an introduction to the MD software development standard AAMI/IEC 62304. MedSoft also recognised the benefits of having an external auditor and in this situation receiving guidance in relation to improving their configuration management and risk management processes. Furthermore MedSoft identified the benefits of utilizing the Med-Adept lightweight assessment which did not require preparation on their part and took very little time to perform.

The results from the Med-Adept assessment highlighted that MedSoft's configuration management processes were very strong with regard to control, but they could be improved in terms of management. It also identified there was room for improvement in their risk management development practices. Therefore, if the recommendations in the Med-Adept findings report were implemented they would enable MedSoft to have both strong configuration and risk management practices.

The software development manager and the engineer both agreed that the strengths and weaknesses highlighted in the report were an accurate reflection of the company's risk management and configuration management practices. The management and staff of MedSoft also recognized that the recommendations were realistic and achievable and if implemented they would bring improvements and benefits to their organization. The software development manager stated that initially he intended championing these improvements in the location where the assessment took place. He then went on to state that he would oversee their implementation in other locations so that the overall organization could benefit from the assessment. MedSoft representatives met internally to discuss developing a SPI path after the Med-Adept findings report was presented. The objective of this meeting was to review and prioritise the report's recommendations and to plan how they will be implemented in a new project (which is stage 7 of Med-Adept).

Having discussed the assessment process with the management of MedSoft they outlined the benefits it offered. They also requested that within 6 months their software processes would be re-assessed (which is stage 8 of Med-Adept). We then focused on what lessons could be learned and identify where specific improvements could be made. MedSoft management highlighted that only 2 process areas had been assessed. Given the success of the initial assessment they considered it important that the number of process areas assessed be expanded. This was also in line with the research strategy. It was therefore agreed that on the release of the second edition of Med-Adept an additional assessment involving other software process areas will be undertaken.

6 Conclusions and Future Plans

The goal of the Med-Adept assessment is not certification but to provide a lightweight method for indicating to companies: the current state of their software processes; recommendations as to how they might improve; the status of their software processes both in terms of CMMI® and AAMI/IEC 62304 and ISO/IEC 15504-5; and their suitability to become medical device software developers. It is important to educate software development organisations in relation to how they may become medical device software developers and how they should improve their software development processes so that they may compete within this domain. This requires an appropriate approach that facilitates education and engages software development managers in a quality agenda. The application of the Med-Adept method will help raise the level of SPI knowledge within the assessed organisations. Also, the high-level findings report and the detailed SPI path will provide a road map for SPI within each assessed organisation. Furthermore, as the Med-Adept method requires little internal staff time, this should prove attractive to SMEs from a resource viewpoint.

From a research perspective the Med-Adept method: enables the Regulated Software Research Group (RSRG) at Dundalk Institute of Technology to gain an understanding as to whether existing software development practices within Irish companies are more CMMI, AAMI/IEC 62304 or ISO/IEC 15504-5 based. This will assists the RSRG in understanding areas that will present Irish software development companies with difficulties if they are to become medical device software development organisations – therefore this awareness will enable the RSRG to provide guidance within these areas; and will enable the RSRG to gain an understanding in relation to the strengths (profile) that Irish software companies possess particularly in relation to developing software for the medical device industry.

This paper has described the development of the Med-Adept method that provides coverage of 11 CMMI process areas, 12 ISO/IEC 15504-5 and 11 AAMI/IEC 62304 processes. It described a pilot release of the Med-Adept method, providing coverage of 2 processes. It also considered how a Med-Adept assessment was conducted in an Irish medical device software company. The company has since prioritised actions and are currently engaged in adopting a number of the recommendations as part of their software development practices. In the future we plan to extend the number of processes that may be assessed. We will extend the Med-Adept assessment to provide coverage of the remaining nine applicable processes that are displayed in table 1.

Acknowledgements

This research is supported by the Science Foundation Ireland (SFI) Stokes Lectureship Programme, grant number 07/SK/I1299, the SFI Principal Investigator Programme, grant number 08/IN.1/I2030 and supported in part by Lero - the Irish Software Engineering Research Centre (http://www.lero.ie)

References

[1] Expert Group of Future Skills Needs. 2008. "Future Skills Needs of the Irish Medical Device Sector", http://www.skillsireland.ie/press/reports/pdf/egfsn080205_medical_devices.pdf
[2] ISO: 13485, Medical devices -- Quality management systems -- Requirements for regulatory purposes, http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=36786
[3] 21 CFR Part 820 Quality System Regulations, http://www.gmp1st.com/mdreg.htm

[4] CDRH, General Principles of Software Validation; Final Guidance for Industry and medical device Staff. January 11, 2002

[5] CDRH, Guidance for the Content of Premarket Submissions for Software Contained in Medical Devices; Guidance for Industry and medical device Staff. May 11, 2005

[6] CDRH, Off-The-Shelf Software Use in Medical Devices; Guidance for Industry, medical device Reviewers and Compliance. Sept 9, 1999

[7] ANSI/AAMI/IEC 62304, Medical device software – Software life cycle processes, Association for the Advancement of Medical Instrumentation, 19-Jul-2006 (replacement for SW68)

[8] FDA's Mission Statement - http://www.fda.gov/opacom/morechoices/mission.html

[9] European Council, 1993. "Council Directive 93/42/EEC Concerning Medical Devices", 14 June 1993.

[10] ANSI/AAMI/ISO 14971, Medical devices – Application of risk management to medical devices, 2nd Edition, 2007

[11] BS EN 60601-1-4: Medical Electrical Equipment, Part 1 - General requirements for safety. (2000)

[12] GAMP 5, 2008. GAMP 5: International Society for Pharmaceutical Engineering (ISPE): A Risk-Based Approach to Compliant GxP Computerized Systemsan-2008. http://www.techstreet.com/cgi-bin/detail?product_id=1559506

[13] AAMI TIR32:2004, Medical device software risk management, 2005

[14] IEC 61508, 2006. IEC 61508 Overview Report, A Summary of the IEC 61508 Standard for Functional Safety of Electrical/Electronic/Programmable Electronic Safety-Related Systems, 2006, http://www.exida.com/articles/iec61508_overview.pdf. Last accessed August 2008.

[15] IEC 60812, Analysis technique for system reliability - Procedure for failure modes and effects analysis (FMEA), 2006.

[16] CMMI Product Team, Capability Maturity Model® Integration for Development, Version 1.2 (2006), http://www.sei.cmu.edu/publications/documents/06.reports/06tr008.html, Technical Report CMU/SEI-2006-TR-008

[17] ISO/IEC 15504-5: 2006 Information Technology – Process Assessment – Part 5: An exemplar Process Assessment Model , JTC 1/SC 7

[18] Cass A., and Volcker C. 2000 , SpiCE for SPACE: A method of Process Assessment for Space Projects, SPICE 2000 Conference Proceedings, http://www.synspace.com

[19] Automotive SIG, The SPICE User Group, Automotive SPICETM Process Reference Model, 2005, available from http://www.automotivespice.com

[20] F. McCaffery, A.Dorling, "Medi SPICE: An Overview", 9th International Conference on Software Process Improvement and Capability Determinations (SPICE 2009), pp. 34-41

[21] F. McCaffery, I. Richardson & G.Coleman, "Adept – A Software Process Appraisal Method for Small to Medium-sized Irish Software Development Organisations", European Software Process Improvement and Innovation Conference 2006, EuroSPI06, October, Finland

[22] Anacleto, A, von Wangenheim. C.G, Salviano. C.F, Savi. R. 2004, "Experiences gained from applying ISO/IEC 15504 to small software companies in Brazil", 4th International SPICE Conference on Process Assessment and Improvement, Lisbon, Portugal, pp.33-37 (April 2004).

Author CVs

Dr Fergal Mc Caffery

Dr Fergal Mc Caffery is a Lecturer with Dundalk Institute of Technology. He is leader of the Regulated Software Research Group in Dundalk Institute of Technology and a member of Lero. He has been awarded Science Foundation Ireland funding through the Stokes Lectureship and Principal Investigator Programmes to research the area of software process improvement for the medical device domain. Additionally, he has received EU FP7 research funding to improve the effectiveness of embedded software development environments for the medical device industry.

Dr Valentine Casey

Dr. Val Casey is a Senior Researcher with the Regulated Software Research Group in Dundalk Institute of Technology. His previous roles include Senior Lecturer and Research Area Leader at Bournemouth University, Researcher with Lero - the Irish Software Engineering Research Centre at the University of Limerick where he also lectured. He has over 20 years experience in the software industry. He has also provided consultancy services focusing on software process improvement and software testing in the financial and telecom sectors.

Integrated Approach for Automotive SPICE and ISO 26262 Assessments

Richard Messnarz[1], Hans-Leo Ross[2], Stephan Habel[2], Frank König[3], Abdelhadi Koundoussi[3], Jürgen Unterreitmayer[4], Damjan Ekert[1]

[1]ISCN GesmbH, Schieszstattgasse 4, A-8010 Graz, Austria
Tel: +43 316 811198, Fax: + 43 316 811312, Email: rmess@iscn.com
[2]Continental Automotive, Germany
[3]ZF Friedrichshafen AG, Friedrichshafen, Germany
[4]SQS AG, Cologne, Germany

Abstract

In 2005 Automotive SPICE (based on ISO 15504) has been published (see www.automotivespice.com) and used in major automotive firms world wide. In parallel the topic "Functional Safety" became important due to changes in liability law and the development of IEC 61508 as an application and branch independent standard for functional safety. As a result ISO 26262 for functional safety in automotive has been initiated classifying systems with ASIL (Automotive Safety Integrity Levels) levels and requiring additional processes, techniques, and methods to illustrate the competence for managing systems which have an impact on the loss of lives.

The working group which elaborated the methods and tools described in this paper are part of the SOQRATES initiative (www.soqrates.de) where more than 20 leading German firms collaborate in cross company task forces.

In the EuroSPI paper from 2008 we illustrated the IEC 61508 mapping and integrated assessment tools [4].

In this paper we present the most recent results of the ISO 26262 integration where assessment results can be wither an ISO 15504 based report or grouped into the ISO 26262 elements. Moreover we developed an online learning environment where all staff has access to an online multimedia safety training as integrated part of the assessment platforms.

An Automotive SPICE assessment usually takes (for the processes defined in the scope of the German automotive manufacturing association) 4 days per project. Adding the scope of a safety assessment this dramatically increases the number of hours used in assessments. In a working group of major automotive suppliers and assessment tool suppliers we developed from 2005 – 2010 an integrated assessment approach. Portals in the above mentioned automotive suppliers already use this environment.

1 History and Motivation

In 2003 the SOQRATES [7] (www.soqrates.de) initiative was formed and Automotive SPICE [5] was introduced into 16 firms in Germany. At the end of 2003 the firms decided to continue (without further state funding) in task forces to elaborate best practices to achieve a capability level 3 in certain core competence areas, such as system and SW design, requirements management related processes, and testing related processes. Since 2003 (for 7 years now) the teams elaborated annual knowledge releases which were transformed into training materials and distributed to the partnership. (= Cross company learning cycle).

In 2005 the partnership formed a further task force with Continental, ZF, IMBUS, SQS, and ISCN to elaborate a mapping between the Automotive SPICE and the new functional safety standard, to extend the content of SPICE assessments, and to elaborate tools for assessment which allow extending the assessment scope.

The following knowledge releases have been developed:

- Excel based mapping between Automotive SPICE and IEC 61508 [4], [6].
- Excel based mapping of ISO 26262 elements to SPICE practices (base and generic)
- A process model to perform an assessment where an integrated team of Automotive SPICE assessors and safety assessors perform an assessment.
- An integrated assessment tool set supporting the combined assessment. Reports can be generated either for ISO 15504 Automotive SPICE or for ISO 26262 (grouped in ISO 26262 elements).

2 Integrated Views - Automotive SPICE & Functional Safety

In [4] we described how we mapped Automotive SPICE towards IEC 61508 and ISO 26262.

During an assessment the “Functional Safety View” can be activated with the follwing effects:

- Base practices will have additional criteria.
- Generic practices include additional criteria.
- New safety practices will appear.
- A safety methods table per process can be opened to consider the use of methods when assessing the practices.

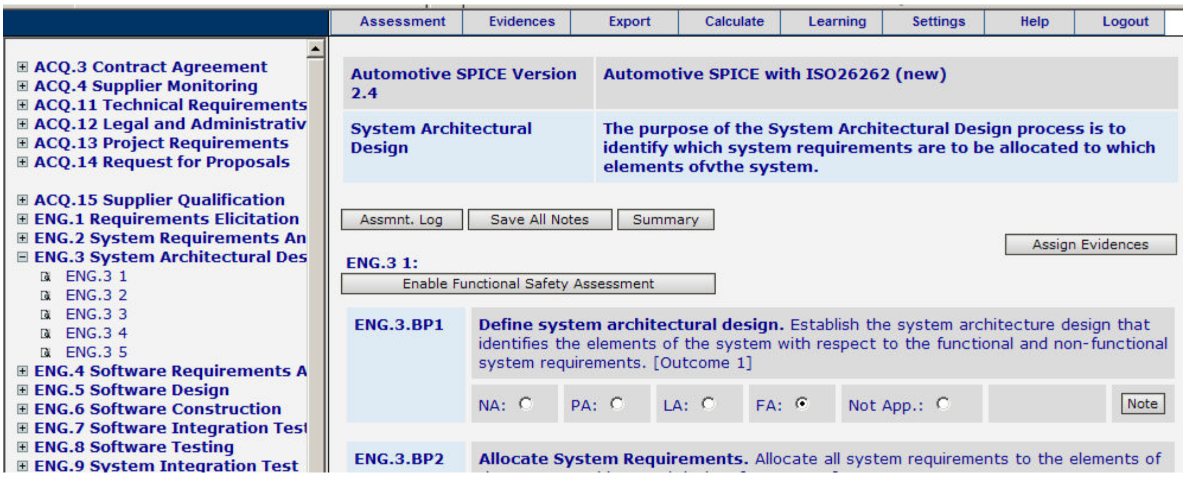


Figure 1: Assessment View Allowing Enabling Functional Safety

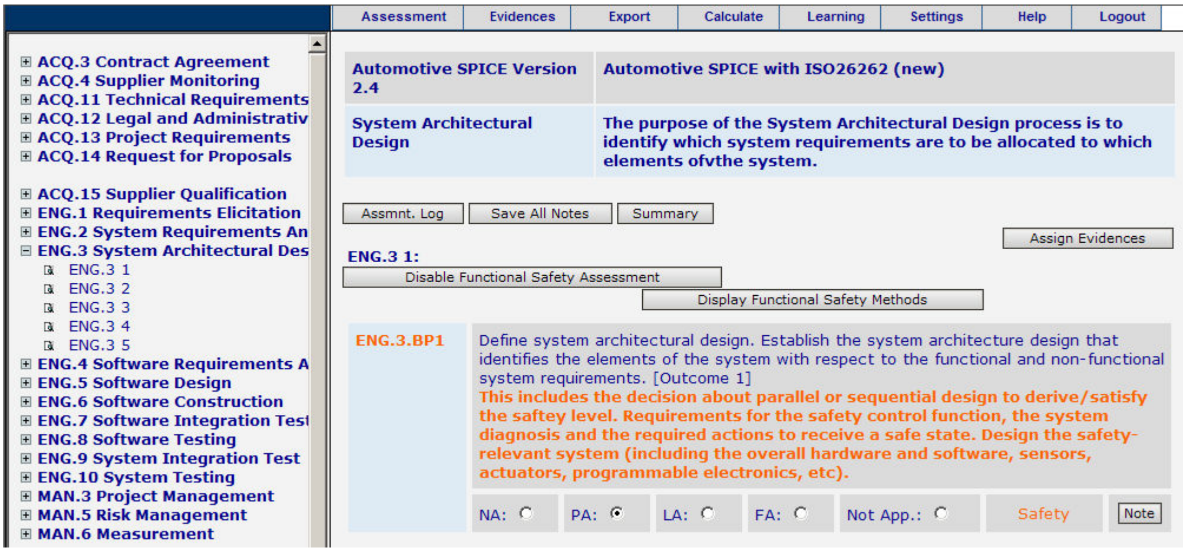


Figure 2: Activated Functional Safety Views – IEC 61508 Extensions

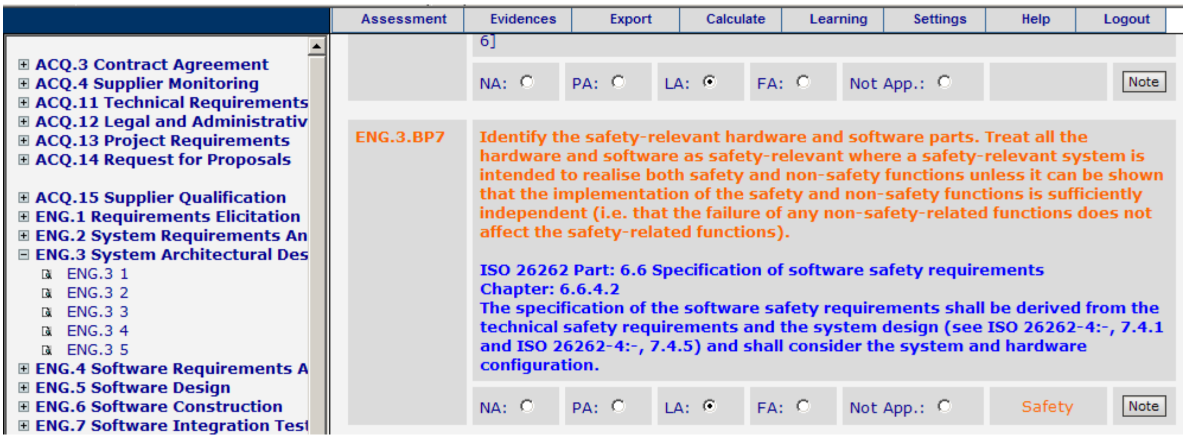


Figure 3: Activated Functional Safety Views – ISO 26262 Extensions

In Figure 2 the existing base practice ENG.3.BP1 has been extended by additional requirements originating from IEC 61508. The additional safety text has a different color.

In Figure 3 an additional safety practice has been included (not part of the SPICE model) and additional ISO 26262 element requirements were added. The additional safety text has a different color and also the ISO 26262 element notes have an extra color in the view.

It is also important to note that each view can be rated differently. This means that e.g. ENG.3.BP1 could be Fully rated in Automotive SPICE but have only a Partially rating in the extended safety view.

Compare Figures 4 and 5.

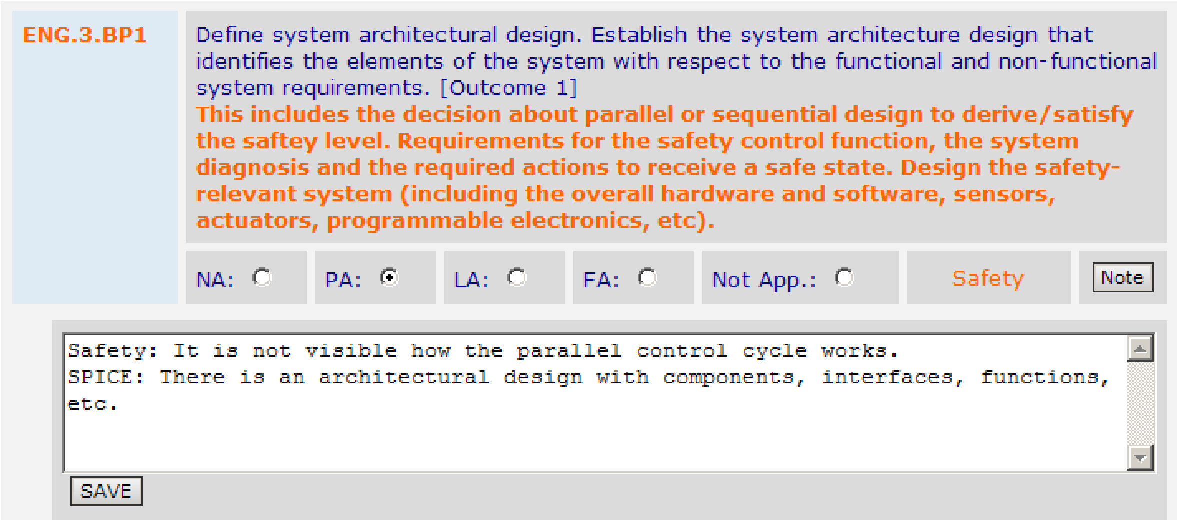


Figure 4: Activated Functional Safety Views – Rating

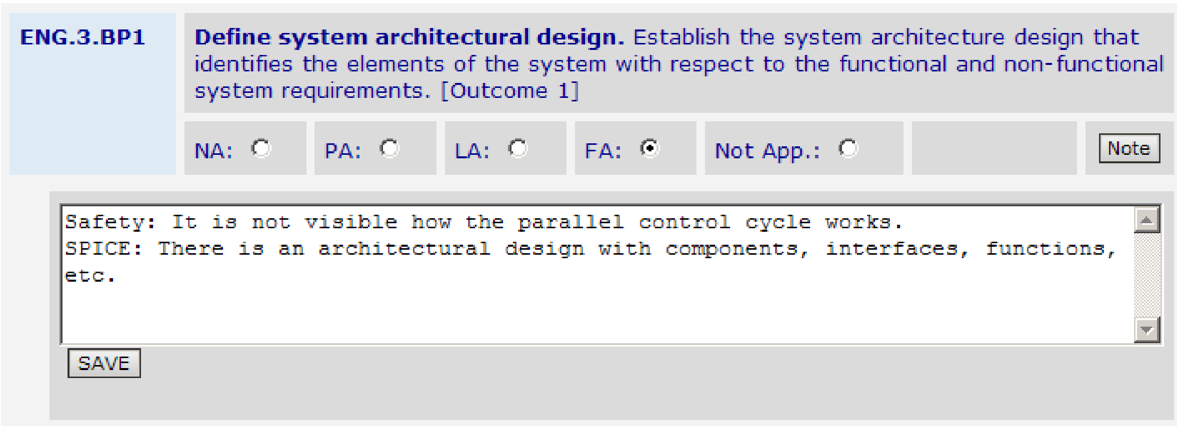


Figure 5: SPICE Rating (compare with Figure 4)

At the start of the assessment in the planning phase a target Capability Level and a target ASIL level are agreed. The target ASIL level (Safety Integrity Level [6]) defines what criteria from the methods table which is displayed during the assessment will be considered when rating the practices.

See Figure 6.

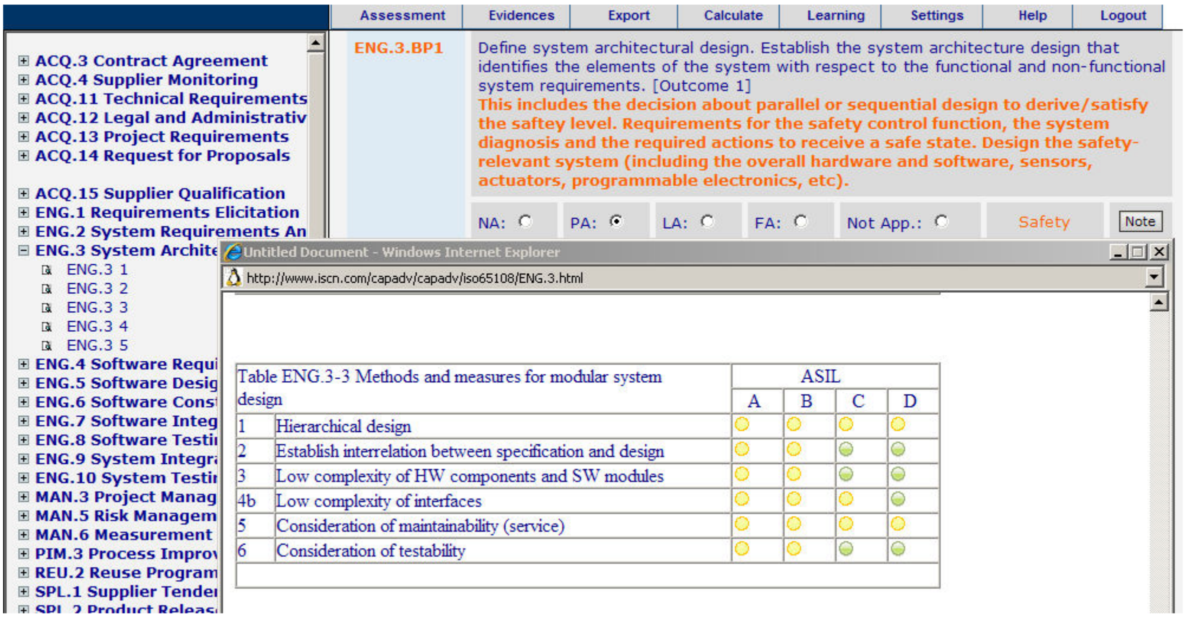


Figure 6: Integration of ISO 26262 Methods Tables

3 Assessment Results – Two Views

The satisfaction of capability levels and process attribute profiles can be displayed either with or without consideration of the safety view.

Figure 7: Display based on Automotive SPICE

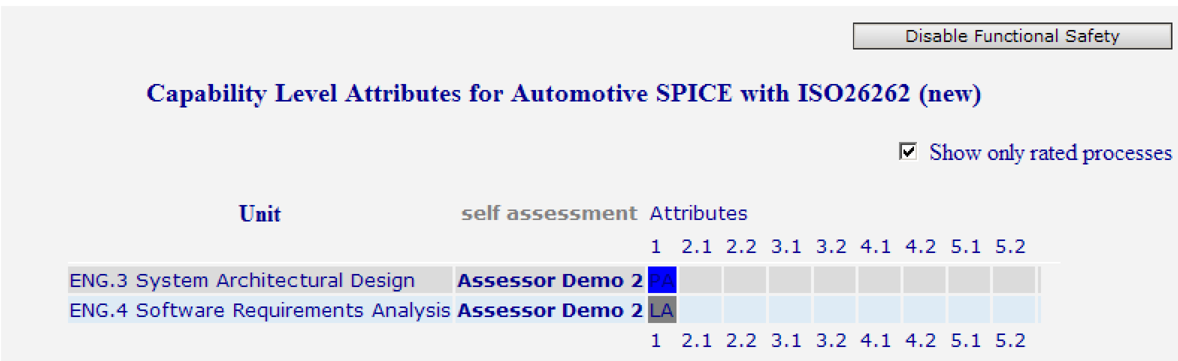


Figure 8: Display also considering safety extensions

In 2009 the manufacturers were not satisfied with these result presentations and requested a further analysis to be done to allow a direct mapping onto ISO 26262 elements.

1	ID	Road Vehicles - Functional Safety	SPICE Referenz
308	PART6_201	**6 Specification of software safety requirements**	...
309	PART6_202	**6.1 Objectives**	...
310	PART6_203	The first objective of this subphase is to specify the software safety requirements. They are derived from the technical safety concept and the system design specification.	ENG.4.BP1
311	PART6_204	The second objective is to detail the hardware-software interface requirements initiated in ISO 26262-4:-, Clause 6.	ENG.4.BP2, ENG.4.BP3
312	PART6_205	The third objective is to verify that the software safety requirements are consistent with the technical safety concept and the system design specification.	added ENG.4.BP9
313	PART6_206	**6.2 General**	...
314	PART6_207	The technical safety requirements are divided into hardware and software safety requirements. The specification of the software safety requirements considers constraints of the hardware and the impact of these constraints on the software.	ENG.4.BP2
315	PART6_208	**6.3 Inputs to this clause**	...
316	PART6_209	**6.3.1 Prerequisites**	...
317	PART6_210	The following information shall be available:	...
318 319	PART6_211	Technical safety concept (see ISO 26262-4:-, 7.5.1)	17-12

Figure 9: Extract of Mapping Work Done for ISO 26262

The partnership then established a list of ISO 26262 elements and their content and mapped this onto base practices, generic practices, and work products in Automotive SPICE.

This mapping was then integrated into the assessment portals (see Figure 3) and further Excel based reports have been developed to allow a report which can be grouped into ISO 26262 elements as well.

	A	B	C	D	E	F	G	H	I	J
1	Project	Assessme	Unit	Element	Performanc	Assessor	IEC61508	Comment	ISO 26262	ISO 26262 Chapt
7	Test Project	Automotive S	System Arch	ENG.3 1	ENG.3.ISO61	Assessor De	LA	Safety: It is n	Part: 6.6 Spe	Chapter: 6.6.4.2
8	Test Project	Automotive S	System Arch	ENG.3 1	ENG.3.ISO61	Assessor De	LA	Safety: It is n	Part: 6.6 Spe	Chapter: 6.6.4.2
9	Test Project	Automotive S	System Arch	ENG.3 1	ENG.3.ISO61	Assessor De	LA	Safety: It is n	Part: 6.6 Spe	Chapter: 6.6.4.2
10	Test Project	Automotive S	System Arch	ENG.3 1	ENG.3.ISO61	Assessor De	PA	Safety: It is n	Part: 6.6 Spe	Chapter: 6.6.4.2
25	Test Project	Automotive S	Software Req	ENG.4 1	ENG.4.BP7 E	Assessor De	PA	Safety: It is n	Part: 6.6 Spe	Chapter: 6.6.4.2

Figure 10: Deviation Report for ISO 26262 Element 6.6.4.2

4 *Knowledge Transfer Reduces Time and Effort*

After the assessment tools were set up and trial assessment could start the partnership still realized a further problem to be soled. It is very time consuming to explain the safety requirements and functional safety concepts during an assessment. Thus the Safety AK task force developed safety concepts training material which assessment attendees can attend online before the assessment.

These training environments can be accessed via the platform and include

- Multimedia lectures

- Notes
- Discussion area
- Background material and reference articles
- Exercises to be done

Figure 11: Open Related Courses / Knowledge

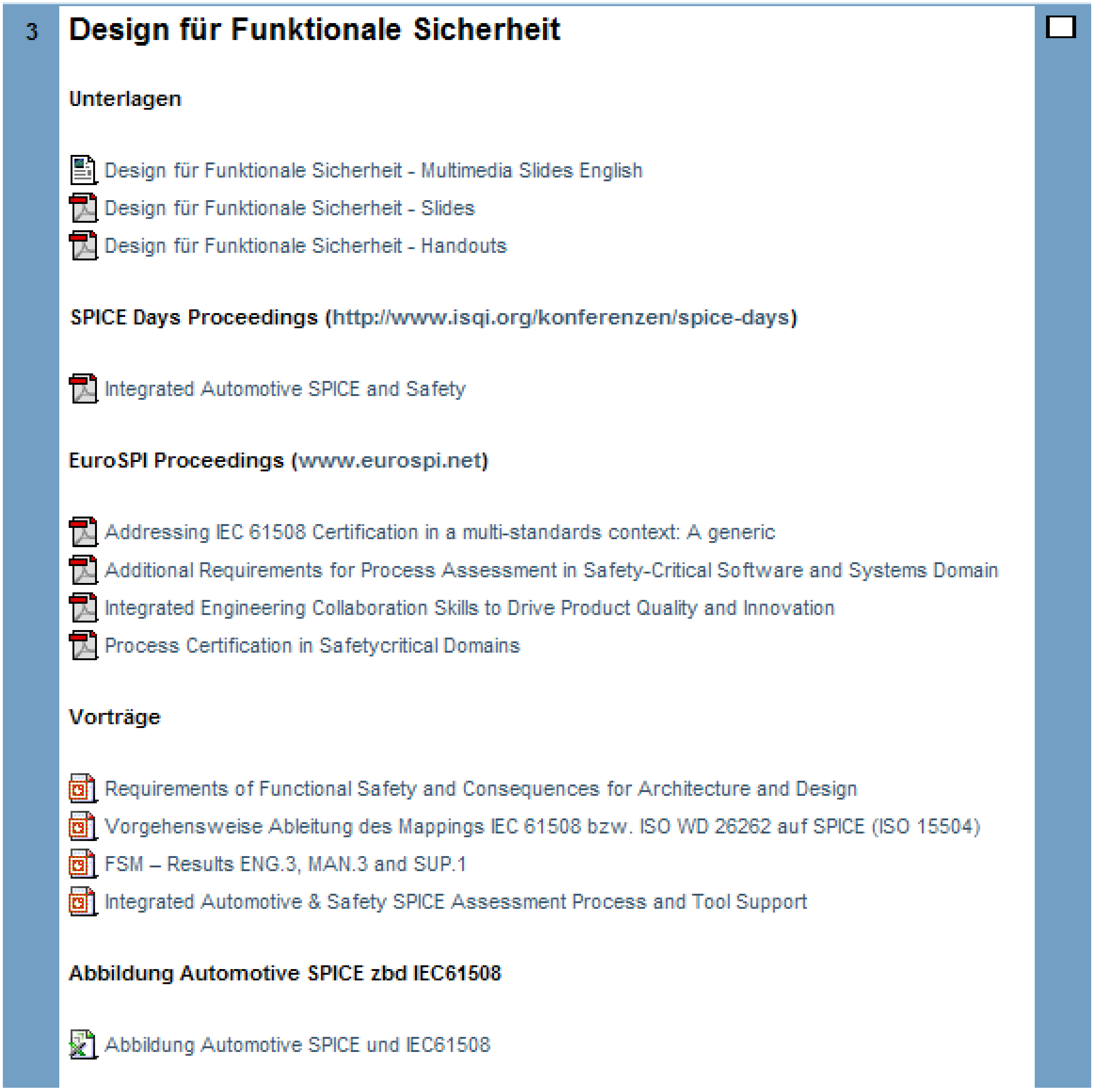


Figure 12: Online Training / Environment

Projects to be assessed can do a preparatory self assessment and attend the learning environment. This reduces significantly the assessment effort and avoids misunderstandings.

5 Lessons Learned

We had published our first experiences at EuroSPI 2008 in Dublin. Two years later we have learned in addition that

- Automotive SPICE and the integration into safety requires a mapping onto ISO 26262 (IEC 61508 is not enough!)
- Automotive manufacturers are not satisfied with just an Automotive SPICE like assessment reports, they expect an ISO 26262 element based deviation report as well.
- Integrated assessments require a high level of preparation and projects need an online training environment to understand safety concepts before they go to the assessment.

Currently we integrated fully part 4 and part 6 of ISO 26262. We continue to cover the remaining parts in the SOQRATES Safety AK task force in 2010.

References

[1] R. Messnarz, et. al, Assessment Based Learning Centers, in : Proceedings of the EuroSPI 2006 Conference, Joensuu, Finland, Oct 2006, also published in Wiley SPIP Journal (Software Process Improvement in Practice), Volume 12 Issue 6 , Pages 505 – 610, November/December 2007

[2] R. Messnarz, et.al, Better Software Practice for Business Benefit, IEEE Computer Society Press, 1999, Washington, Tokyo, Berlin

[3] R. Messnarz, From process improvement to learning organisations (p 287-294), Wiley SPIP Journal (Software Process Improvement in Practice), Volume 11 Issue 3 , Pages 213 - 335 (May/June 2006)

[4] SOQRATES Safety Team, Richard Messnarz, Hans-Leo Ross, Stephan Habel, Frank König, Abdelhadi Koundoussi, Jürgen Unterrreitmayer, Damjan Ekert, Integrated Automotive SPICE and safety assessments (p 279-288), in Wiley SPIP, Volume 14 Issue 5, September 2009

[5] Automotive SPICE, www.automotivespice.com

[6] ISO 26262, Road vehicles — Functional safety

[7] SOQRATES Initiative, www.sowrates.de

[8] HIS, www.his-automotive.de

Author CVs

MSc Leo Ross

Since autumn 2004, Hans-Leo Ross is in the Quality Business Process responsible for the implementation of the necessary measures for Functional Safety within Continental Automotive. Further more the general conformity of Engineering Processes to relevant standards (ASPICE, ISO TS 16949 etc.) as well as the coordination of the Conti Automotive Safety Manager belongs to the scope of responsibility.

Safety Managers in Conti Automotive are responsible for safety reviews and assessments in product area like chassis, brake, power train and interior electronic equipment.

In FAKRA (now Normenausschuss Automobil, German Automotive Standardization Body within VDA) and in the international standardization group ISO SC3/TC22/WG 16, Hans-Leo Ross is a nominated expert fort he coming standard ISO 26262.

At the University UNI-GH Paderborn, Hans-Leo Ross passed the Dipl.-Ing. In Communication and Electronic. During the years 1999 to 2003, Hans-Leo Ross graduated Master in Economy and Marketing and the Master of Advance Studies at the University of Basel (Switzerland).

The professional carrier passes System Engineering for Safety Concepts in the Oil and Gas industry as well as sales and Head of Product Management for Safety related control systems for plant automation and machinery applications. During this work, Hans-Leo Ross was frequently involved also in the development of safety standards such as IEC 61508.

Dr Richard Messnarz

Dr. Richard Messnarz (rmess@iscn.com) is the Executive Director of ISCN LTD. He studied at the University of Technology Graz and he worked as a researcher and lecturer at this University from 1991 - 1996. In 2 European mobility projects (1993 and 1994) he was involved in the foundation of ISCN, and he became the director of ISCN in 1997. He is/has been the technical director of many European projects:

- PICO - Process Improvement Combined Approach 1995 - 1998,
- Bestregit - Best Regional Technology Transfer, 1996 - 1999,
- TEAMWORK - Strategic E-Working Platform Development and Trial, 2001-2002,
- MediaISF – E-Working of media organisation for strategic collaboration on EU integration, 2001-2002
- ORGANIC – EU Innovation Manager, 2003 – 2006
- European Quality Network, 2005 – 2007
- European Certificates Association, 2008 - 2010

He is the editor of a book "Better Software Practice for Business Benefit", which has been published by IEEE (www.ieee.org) in 1999 (the leading research publisher in the USA). He is the chairman of the EuroSPI initiative and chair of the programme committee of the EuroSPI conference series.

He is author of many publications in e-working and new methods of work in conferences of the European Commission (E-2001 in Venice, E-2002 in Prague), and in the magazine for software quality (Software Quality Professional) of the ASQ (American Society for Quality).

He is a principal ISO 15504 and Automotive SPICE assessor. He has worked as a consultant for many automotive firms, such as BOSCH, ZF TE, ZF N, ZF SACHS, Continental TEMIC, T-Systems, Magna Powertrain, Giesecke & Devrient, etc. in the last 18 years. He is a founding member of the INTACS (International Assessor Certification Scheme) accreditation board, a founding member of the Austrian Testing Board, a founding member of the Configuration Management Board, and he is the technical moderator of the SOQRATES initiative (www.soqrates.de).

MSc Stephan Habel

Since early 2008, Stephan Habel has held the position Functional SafetyManager within the Business Unit Transmission. Besides, the general conformity of Engineering Processes to relevant standards (ASPICE, ISO TS 16949 etc.) as well as the coordination of the Project Safety Coordinator also belongs to his scope of responsibility.

Safety Managers at Conti Automotive are responsible for safety reviews and assessments in product areas such as Chassis, Brake, Powertrain and Interior Electronic Equipment.

He has been leading a team of electronic experts responsible for moderation of Hazard Analyze, FME(D)A and FTA.

From 1984 to 1989 Stephan Habel attended the Georg-Simon-Ohm University of Applied Sciences in Nuremberg and graduated as a Dipl.-Ing. (FH) in Electronics.

Since 1989 Stephan Habel has worked for Continental Temic. Starting out in a test center, he moved to the development department in 1993 where he was frequently involved in System Engineering Powertrain and Chassis Application for the automotive industry.

In 2003 Stephan Habel was Conti Temic's representative in the "first" Soqrates Group dealing with SPICE.

MSc Jürgen Unterreitmayer

Jürgen Unterreitmayer is an experienced test manager and project manager and participated in many development and testing projects in the telecommunications, automotive, logistics and transport industry. His main activity over the last years was the assessment and improvement of software development and testing processes.

Before specialising in software test and quality assurance Jürgen held various positions at the European Space Agency, Siemens, Viag Interkom and Sun Microsystems, mainly in hardware / software development, product management and business development.

Since October 2007 he is working as Delivery Manager for the Industry & Engineering business unit of SQS AG, Europe`s biggest independent consulting company focused on software quality management and assurance. Delivery Managers' responsibilities at SQS AG include team management, project management and consulting on senior level.

Before that he was head of the branch office in Munich for imbus AG, a company specialised mainly in soft-ware testing, also responsible for leading a team of consultants.

Jürgen passed the Dipl.-Ing. in electronic engineering at the Technical University of Munich.

Jürgen is certified as ISTQB Certified Tester and SPICE Assessor. Based on his experience he is executing trainings and supporting customers with coaching, mainly in Test Management.

Process Assessment Model Development towards Improvement Oriented Assessment in Space Domain

Yuko Miyamoto[1], Yumi Koishi[2], Motoko Takeuchi[2], Masafumi Katahira[1]

[1] Japanese Aerospace Exploration Agency, 2-1-1 Sengen, Tsukuba, Ibaraki, 305-8505 Japan
[2] Software Research Associates, Inc., 2-3-18 Minami-Ikebukuro, Toshima-ku, Tokyo, 171-8513 Japan
[1]{Miyamoto.Yuko, Katahira.masafumi}@jaxa.jp
[2] {koisi, motoko-t}@sra.co.jp

Abstract

A spacecraft needs to achieve extremely high quality and reliability. JAXA performs extensive product assurance to ensure the success of our space missions. For software process assurance, a process assessment has been introduced as an alternative to an audit. The purpose is to encourage improvement by suppliers, not to rate the capabilities of suppliers. By introducing the assessment, there are difficulties found for Japanese or for space development to use ISO/IEC 15504[1]. They are caused by high cost of performing improvement intended assessment and needs to take into account the requirement of JAXA software development standard. In order to eliminate such difficulty, JAXA is now developing a JAXA process assessment model (JAXA-PAM) and assessment procedures and techniques. In this paper, JAXA's original approach to implementing process assessments, its adaptability for Japanese space development, and its effectiveness are described.

Keywords

Software Process, Process Assessment Model (PAM), JAXA-PAM, Process Improvement, Process Assurance, Space Development

1 Introduction

The space domain places great emphasis on the quality of both hardware and software in order to satisfy rigorous mission success, safety, and reliability requirements. To ensure mission success, safety and reliability of space systems, JAXA requires suppliers to apply numerous standards and performs various assurance activities. Since hardware is a main part of space systems development, JAXA performs assurance activities in hardware, some of which need to be modified for software assessment purposes.

For on-board software, JAXA is now introducing a software process assessment as an alternative to an audit. The main objective of the process assessment is not to rate suppliers' capabilities, but to improve their software processes. The results can be used to motivate improvement. However, the rating result itself does not guarantee the success of a project, especially for highly reliable systems that are required in the space domain. The improvement-oriented assessment focuses on understand previously deployed processes, determining process strengths and weaknesses, and making suggestions for future improvements.

In 2008, JAXA began performing ISO/IEC 15504[1]-based improvement-oriented process assessments as a trial in cooperation with suppliers. In the trial assessment, engineers working for the supplier acted as assessors and interviewees. They are not professional assessors; however, they are the main players in development and improvement activities. JAXA believes this assessment is a valuable opportunity for suppliers to understand their development process and encourage improvement activities.

2 Implementation of improvement-oriented assessment in JAXA

2.1 Overview of Process Assessment framework in JAXA

2.1.1 JAXA Process Reference Model and JAXA Process Assessment Model

JAXA defined various development standards to ensure the development of space systems. The standards have been updated as needed, for example to accommodate evolving technologies and development environments. The old software development standards were developed with an eye to assuring compatibility with relevant standards, such as MIL-STD-498[1] for software development and ISO 9001[2] for quality management. The current version of the standard that complies with ISO/IEC 12207 [3] covers not only the on-board software but also all JAXA-developed space system products, including rockets, satellites, and their ground systems. Nonetheless, each product has very different features in terms of the development process. To deal with such differences, JAXA has developed a hierarchical software standards structure, as shown in Figure 2.1. In the lower layers of the standards structure, JAXA defines domain-specific software development standards for each domain considering characteristics of its software development processes. These standards are applied to all space system developments.

In the JAXA assessment framework, the top layer standard, i.e., the JAXA Software Development Standard, is treated as the Process Reference Model (JAXA-PRM).

The revised version of the ISO/IEC 15504 (IS) [4] allows to freely choose the surrounding PAM. With these changes, any process may now be assessed using the ISO/IEC 15504 part 5 standard (not only software life cycle processes). Process assessments have become a more flexible analysis tool, which helps to identify domain-specific, process-related risks more effectively. Therefore, JAXA has created its own PAM, called JAXA-PAM (by extending the ISO/IEC 15504 part 5), with the intention of introducing the particularities of spacecraft development in Japan identified during previous assessments. Some criteria in ISO/IEC 15504 part 5 do not match Japanese social behavior at either the inter-organizational and intra-organizational levels. JAXA modified the criteria and added their own interpretations to increase understanding. The main differences between the original ISO/IEC 15504 part 5 and JAXA's PAM are: a new management process (needed because of special supplier-customer relationships in Japan) and four space-specific processes taken from the Space for Space standard (S4S) [5]. These four processes are: Contract Maintenance, Safety and Dependability Assurance, Independent Verification of Software & Validation of Adequacy, and Information Management. Additionally, references have been made to the legacy JAXA standards in order to establish transparency and traceability for suppliers. These references will help suppliers to better understand what is meant in the new assessment model.

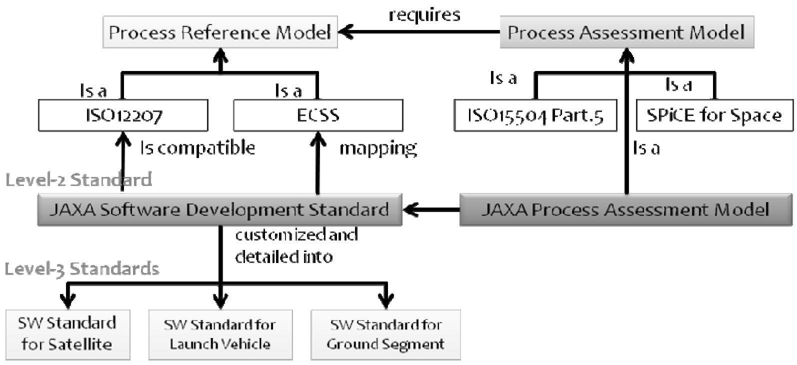


Figure 2.1: JAXA hierarchical standard structure

2.2 Trial SPiCE Assessment at JAXA

JAXA has performed assessments for satellite on-board software development as trial assessments. In this section, the organizational strategy of the assessment team and the assessment procedure are explained.

2.2.1 Organizational strategy of the assessment team

Since the purpose of assessment at JAXA is to trigger process improvements for software development, previously deployed processes must be thoroughly evaluated to find realistic and meaningful improvement opportunities. For this purpose, an assessment team was formed, consisting of JAXA, representatives, supplier representatives, and an independent assessment provider representatives.

It is important for the assessors to understand the philosophy, the rationale behind JAXA-PRM requirements, and the relationship between JAXA-PRM requirements and JAXA-PAM practices. Assessors investigate the processes in place in the assessed organization and the reasons why they implemented those processes.

To perform an effective assessment, it was concluded that there should be a trilateral assessment team (JAXA, a supplier, and an independent assessor). Assessors from JAXA convey the intentions of JAXA-PAM and explain JAXA PAM practices. Assessors from the supplier help the team understand the deployed processes and underlying reasoning. Independent assessors (a professional third party) maintain objectivity and offer a neutral viewpoint based on their experience.

The trial assessment team included several beginner assessors. Although large effort was made to coach them, this is one of the important points of our assessment: to educate people who will drive improvement activity in future. This assessment could be a good opportunity to understand and objectively review their development, particularly for the assessors representing the supplier.

2.2.2 Assessment procedure

- Preparation:

In the first step of the assessment, the team analyzes the relationship between JAXA-PAM base practices (BPs) and JAXA-PRM requirements, such as the JAXA Software Process Standard. Assessors then identify related deployed processes and work products in actual development conducted by the supplier based on their development plan. The results of these analyses are documented in a mapping table, called “Check List”. The differences between BPs and related JAXA-PRM requirements are identified and re-defined as “Check Points” in the “Check List”.

- Data collection (Review of Documents and Interviews):

In the document review, the deployed processes are investigated using a development plan, work products, any enabling artifacts of processes and other related supplier standards. In this task, some previously defined "Check Points" are confirmed and additional "Check Points" will be identified and confirmed in an interview process.

The strategy of the interview is not to cover everything, but to focus on the "Check Points" that were identified in during the preparation and document review. If the assessment team finds some weaknesses, the interview will focused on those points. Time allocation is also part of the strategy. In addition, interviewers discuss process-related issues that the interviewee (the development engineer) deals with when carrying out daily development tasks, even if these issues and the interviewee's opinions are not directly related to JAXA-PAM BPs or JAXA-PRM requirements.

- Results consolidation:

Assessment results, such as document review results and interview results, are consolidated and reviewed by all team members to verify data collection using the check list. The process itself is then evaluated. Assessment teams rate each BP and generic practice (GP), but these information are treated as just information of this task.

- Reporting:

The assessment team makes two reports—one for the assessed organization and one for the sponsor, i.e., JAXA. The report for the assessed organization includes all observed data, analysis results, and suggestions for improvement. However, the assessment report for JAXA only includes an abstract of observed data in order to report on process performance compliance.

2.3 Lessons learned from previous JAXA Trial Assessment

The trial assessment described in the previous chapter was successfully completed. However, some issues were identified. The amount of effort is one of the most important issues to be addressed. The assessment team has spent much time on the preparation task, especially to analyze the relationship between JAXA-PRM and JAXA-PAM. JAXA-PRM defines process requirements and JAXA-PAM describes goals that will be accomplished at the end of the process, i.e. process outcomes. It is not easy to compare JAXA-PRM with JAXA-PAM because the perspectives are different. Thus, more time is needed in which to determine the relationship between the JAXA-PRM requirements and JAXA-PAM BPs.

In the preparation task, assessors clarify their relationships and identify the differences. If there are some JAXA-PRM requirements that are not covered by JAXA-PAM BPs, "Check Points" are added to address them.

2.3.1 Lessons Learned from Software requirement analysis - ENG.4 assessment

For an example of the analysis of JAXA-PRM and JAXA-PAM, the ENG.4 case is explained in this section. In order to understand the structure of JAXA-PAM BPs, the relationship between BPs and GPs is analyzed by making a correlation chart. The relationship between JAXA-PRM requirements and JAXA-PAM BPs is then analyzed by mapping BPs and requirements (activities) defined in JAXA-PRM.

An example of detail analysis is shown below. The original description of ENG.4-BP1 is as follows and related requirements (activities) defined in JAXA-PRM are identified by "**#"

*Define**1 and prioritize functional and nonfunctional requirements of the software elements of the system**1-1 and their interfaces**1-2 and document**2 them in a software requirements specification. Analyze**3 the software requirements for correctness**3-1, completeness**3-2, consistency**4, feasibility**3-3, and testability**3-4. Identify any derived requirements**4.*

The BP1 included three activities: "*Define**1 software requirements*", "*Document**2 specification*" and "*Analyze**3 requirements*". There are some perspectives of "*Analyze**3 requirements*", including "*correctness**3-1*", "*completeness**3-2*", "*consistency**4*", "*feasibility**3-3*", and "*testability**3-4*". In addition, there is the "*BP4: Ensure consistency*" in ENG.4, which is very similar to "*consistency**4*" and "*Identify any derived requirements**4*".

In the JAXA-PRM, there is a requirement related to each activity (***1~**4*) and perspective(***1-1, **1-2, **3-1~**3-4*) that are mentioned above. In the results, it is found that BP1 is related to seven JAXA-PRM requirements (activities). These differences are identified in the "Check list" as seven "Check Points" for BP1.

The number of related JAXA-PRM requirements (activities) for each BP in ENG.4 is shown in Table 2.1.

Table 2.1: The number of JAXA-PRM requirements corresponding to each JAXA-PAM BP.

JAXA-PAM	Number of related activities in JAXA-PRM
BP1: Specify software requirements	7
BP2: Determine operating environment impact	1
BP3: Develop criteria for software testing	1
BP4: Ensure consistency	2
BP5: Evaluate and update software requirements	0
BP6:Communicate software requirements	2

Furthermore, there is some difficulty regarding data collection and evaluation. In the previous trial assessment, it was necessary to collect the data in accordance not with BPs, but with the seven "Check Points" to ensure the supplier correctly implemented JAXA's requirements in their development processes. This situation made it difficult to evaluate the strengths, weakness, and necessary improvements when evaluating BP1. Because BP1 includes a number of "Check Points", evaluation of the "Check Points" is merged into one BP1 evaluation result. Each "Check Point" is important for JAXA to ensure supplier conformity, but BP1 evaluation makes it ambiguous.

As described above, there are very similar perspective in ENG.4. One is "*consistency**4*" in BP1 and the other is "*BP4: Ensure consistency*". "*Consistency*" was also one of the most important perspectives of JAXA-PRM. However, only one requirement is defined in JAXA-PRM to ensure consistency. To avoided tautological evaluation, assessors evaluate "Consistency" as BP4, not as an perspective of BP1.

Moreover, comparing BP1 and other BPs in terms of the amount of "Check Points" or data to be collected for an assessment shows an unbalance of the weight of the evaluation result between BPs. In case of rating, such unbalance makes it difficult to evaluate the process based on the rating of BPs without adding weight.

3 Recognized Issues in PAM

3.1 Issues Discovered

Based on the experience with the past trial assessment, the issues are examined and categorized. In this chapter, JAXA-PAM related issues are reported.

3.1.1 Issues with model structure

From the analysis of lessons learned from the previous trial assessment, JAXA discovered three issues related to the JAXA-PAM BP structure.

- Uncertainties regarding the relationship between BPs in a process

There are various kinds and degrees of relationships between the BPs defined in each process. Therefore, the assessors need to analyze the relationship between BPs for each process themselves. The success of the analysis depends on the assessors' capability.

- Unbalances of weight of BP

As described in Section 2.3.1, some BPs include several activities and perspectives others do not. Such differences makes unbalances of weight of BPs..

- Duplication of BPs

As described in Section 2.3.1, some BPs have the same perspective, so that one single observed fact could influence the evaluation of several BPs simultaneously.

3.1.2 Less correspondence with JAXA-PRM

There is another difficulty in conducting an assessment using JAXA-PAM. As described in Section 2.3, JAXA-PRM and JAXA-PAM have different objectives; the two models do not corresponded. Therefore, it is necessary to expend much effort to compare JAXA-PRM requirements with BP descriptions during assessment preparation.

3.2 Modification in JAXA-PAM

3.2.1 Categorization and structure of BPs

To resolve the issues identified in Section 3.1.1, JAXA is considering categorizing and structuring BPs for each process to accord with the JAXA-PRM structure in order to facilitate understanding of JAXA-PAM.

Process requirements defined in JAXA-PRM can be grouped in several categories according to requirement objectives, such as "(requirement) identification", "(requirement) implementation" and "(requirement) evaluation". The information thus flows and the relationships between the categories in the process can also be identified. Based on the identified categories, the information flows, and the relationships, the structure of the process can be defined. The structure shows essential activity flow for conducting the process, which might be applicable to other processes with a similar purpose, such as engineering or management processes.

After the JAXA-PRM analysis, the identified structure is applied to JAXA-PAM BPs. This makes it easy to understand the structure of BPs in JAXA-PAM and the relationship between JAXA-PRM and JAXA-PAM, thus increasing the similarity between JAXA-PRM and JAXA-PAM.

3.2.2 Import JAXA-PRM perspectives into BPs

To improve the correspondence of JAXA-PRM to JAXA-PAM, JAXA-PRM requirements not covered by JAXA-PAM BPs are added to JAXA-PAM as BPs or as additional perspectives of existing BPs.

3.3 Examples of JAXA-PAM Modification

The modification of JAXA-PAM was performed based on the improvement concept described above. As an example of such modification, ENG.4 software requirements analysis of the engineering process group is described below.

3.3.1 Categorization and structure of BP

To determine the categorization and structure for the engineering process, we compare three engineering processes—system requirements analysis (ENG.2), system architecture design (ENG.3), and software requirements analysis (ENG.4). Five categories—"Identification", "Implementation", "Evaluation", "Authorization", and "Communication"—emerge and information flow between these categories is identified. A common structure is then defined and applied to the three JAXA-PAM processes. By categorizing BPs, we find that the BP1 crosses over three categories, “Identification”,” Implementation”, and” Evaluation”. Thus, BP1 is divided into several BPs.

3.3.2 Import JAXA-PRM perspectives into BPs

There are some requirements special to the space development in JAXA-PRM and JAXA-PAM does not cover them. .For instance, activity that related to the operational design is clearly defined in the JAXA-PRM software requirements analysis.

To correspond with this JAXA-PRM requirement, BPs were added, including considering the operation design (Category: Implementation-BP2 in Figure 3.2).

Because some expensive customized verification is required to verify the embedded satellite software, it is important to develop a verification plan early on and to examine the validation methodology. Thus, the idea of evaluating the impact on the development/verification environment was added to the existing BP, originally defined as "Evaluation of the influence on the operational environment, and a new BP (Category: Implementation -BP5 in Figure 3.2) was defined.

As a result of the categorization based on the above-mentioned concept, ENG.4 was revised as follows.

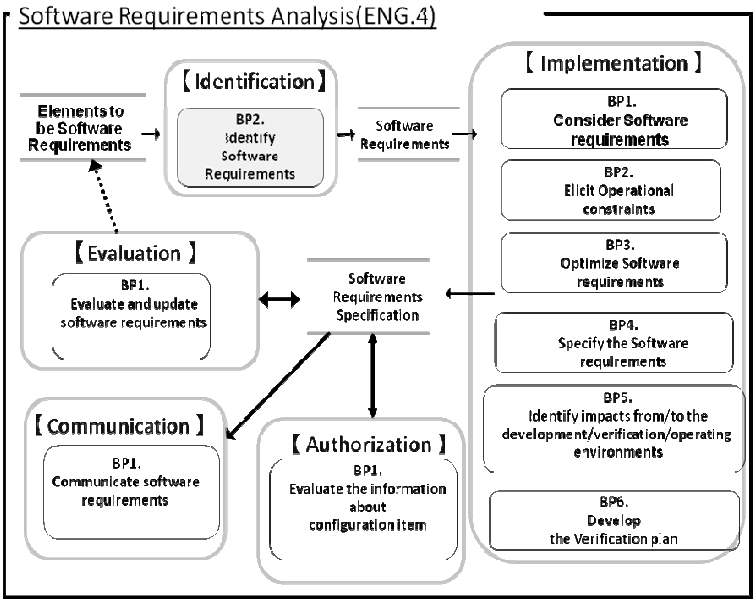


Figure 3.1: Structure of BP in ENG. 4

4 Lessons Learned from Trial Use of New JAXA-PAM

The assessment was tried to six processes including ENG4 by the technique similar to 2.2 by using reviewed JAXA-PAM. This chapter describes the feedback from the trial.

4.1 The result of the categorization of BPs

It was easy to understand the structure of BPs in the modified JAXA-PAM due to the categorization of BPs. It was easy to identify the checkpoints in each BP, though the number of BPs increased. Furthermore, it was easy for the assessment team to understand and the analysis and evaluation went smoothly.

4.2 The result of applying JAXA-PRM perspectives of JAXA-PRM into BPs

A new member familiar with CMMI[7] but with no knowledge of spacecraft development took charge of ENG.4. However, it was easy for him to understand JAXA-PRM by using the new JAXA-PAM. Thus, data was able to be collected. This resulted in an improved overall assessment process.

5 Conclusion and Future work

JAXA is now implementing a new improvement-oriented assessment framework. Some issues have been identified based on trial assessments. JAXA analyzed assessment results and identified issues in detail, and has begun to improve JAXA-PAM.

The main improvements to JAXA-PAM are 1) categorizing and structuring BPs, as described in Sections 3.2.1 and 2) introducing JAXA-PRM perspectives as described in Section 3.2.2. In the categorization task, BP categories are defined based on the relationship between BPs and standardized commonly in a process group. Then BPs are reconfigured based on the categories.

This improvement makes JAXA-PAM easy for assessors to understand the relationship and structure of BPs. In addition, by using common categories and a common structure, the weight of BPs are uniformed. In the trial of the new JAXA-PAM, the PRM and PAM analysis tasks become much easier and the time required is reduced. Furthermore, makes it easy for assessors to understand the relationship between PRM and PAM. by introducing perspectives of JAXA-PRM into JAXA-PAM. In the trial, even beginner assessors were able to collect data and evaluate BPs without omitting the perspective. Thus, the assessment team discovered what improvements were necessary and achieved the assessment objectives. Based on the trial assessment and results, the modification of JAXA-PAM improves assessment activities and contributes to achieve objective of JAXA, i.e. to perform improvement oriented assessment.

JAXA is now continuing the modification of JAXA-PAM step-by-step. A new JAXA-PAM will be evaluated its validity through assessments in the future.

Literature

1. MIL-STD-498: Software Development and Documentation. Canceled Doc Date: 27-MAY-1998. http://milstd.net/
2. ISO9001: Quality management systems. Available at: http://www.iso.org/iso/en/. Last checked 2008-05-22.
3. ISO/IEC 12207:1995. Information Technology - Software Life Cycle Processes International Organization for Standardization. Available at: http://www.iso.org/iso/en/. Last checked 2007-06-02.
4. ISO/IEC 15504:2003. Information Technology - Process assessment. Available at: http://www.iso.ch/iso/en/. Last checked 2007-06-02
5. SPiCE for SPACE: "SPiCE for SPACE: A Process Assessment and Improvement Method for Space Software Development", Ann Cass, Christian Völcker,L. Winzer, J.M. Carranza, A. Dorling, ESA Bulletin Number 107, August 2001.
6. “Software Process Definition for Multi-organizational Development in the Aerospace Domain”, Fabio Bella, Dirk Hamann, Jürgen Münch, Masa Katahira, Yuko Miyamoto, Makoto Shizunaga, International Conference on Software& Systems Engineering and their Applications, 2006
7. CMMI: Capability Maturity Model Integration. Available at: http://www.sei.cmu.edu/cmmi/.

Author CVs

Yuko MIYAMOTO

Ms. Miyamoto is an associate senior engineer in the Digital Innovation Center (JEDI) at the Japan Aerospace Exploration Agency (JAXA). She is in charge of software process improvement (SPI), including assessment and software Independent Verification and Validation (IV&V). She is an ISO/15504 certified provisional assessor and has experience with process assessments in the space domain.

Yumi KOISHI

Ms. Koishi is an engineer in the process improvement group at Software Research Associates Co (SRA). She has experience in software development of the enterprise system and software process improvement. Since 2007, she has been involved in SPI and assessment activities at JAXA. She is ISO/15504 certified provisional assessor and has experience with process assessments in the space domain.

Motoko TAKEUCHI

Ms. Takeuchi is project manager at Software Research Associates Co. (SRA). She has worked in the software engineering field for more than 15 years, and has recently worked in the area software process improvement. She is a registered ISO/15504 provisional assessor and PMP.

Masafumi KATAHIRA

Mr. Katahira is a senior engineer and Manager for Software Engineering for JEDI at JAXA. He worked on satellite operations and developing the JEM and H-IIA launch vehicle at NASDA, and complex systems at the Massachusetts Institute of Technology. He is engaged in software IV&V, software process improvement, and real-time operating systems for spacecraft. He is a member of the IEEE.

Improvement-Oriented Assessment for Space Development

Teiichiro Inoue[1], *Makoto Shizunagai*[2], *Tsutomu Kojima*[2], *Yuko Miyamoto*[3]

[1] Mitsubishi Electric Corporation, 325 Kamimachuya, Kamakura, Kanagawa, 247-8520 Japan
[2] Software Research Associates, Inc., 2-3-18 Minami-Ikebukuro, Toshima-ku, Tokyo, 171-8513 Japan
[3] Japanese Aerospace Exploration Agency, 2-1-1 Sengen, Tsukuba, Ibaraki, 305-8505 Japan

[1] Inoue.Teiichiro @aj.MitsubishiElectric.co.jp
[2] {sizunaga, t-kojima}@sra.co.jp
[3]Miyamoto.Yuko @jaxa.jp

Abstract

The Japan Aerospace Exploration Agency (JAXA) and Mitsubishi Electric Corporation are carrying out assessments for their development processes of spacecrafts and ground stations to increase the quality. These assessments complied with ISO/IEC 15504 Part 2, performing requirement, and JAXA Process Assessment Model which is based on a development standard for the spacecrafts. The purpose of these assessments is not the evaluation of development capability, but rather the detection of improvement tasks. This paper discusses our assessment process and challenges from the experience.

Keywords

Software Process Improvement, Assessment, ISO/IEC 15504, JAXA Process Assessment Model, JAXA Software Development Standard

1 Introduction

JAXA is the space agency in Japan which promotes the research and development programme in aerospace domain. Mitsubishi Electric Co. engages in the development of spacecrafts and ground stations as one of the prime contractors of JAXA. In such development projects, JAXA implements many tasks in order to improve quality, for example software process improvement, enactment of development standards, and independent verification and validation, and so on[1].

Traditionally, JAXA carried out quality audits as the approach to confirm the software quality. The result of the audits often turns out as the merely superficial reaction, following only the points reported in the audits. As there are rapid changes in the software technologies and environments, it is essential that organizations voluntarily detect improvement tasks and cultivate self-sustaining improvement capability. The importance of self-sustaining improvement activity can be recognized by the fact that CMM defines Optimizing Level, in which process improvement is continuously realized by optimizing software processes, as its highest maturity level[2], and that SPICE also defines Optimizing process as the highest level in its measurement framework[3]. Applying only audits inhibits not only the improvement of the development process itself, but also the improvement of such capability of the organizations.

With such concept in mind, JAXA considered introduction of process assessments instead of quality audits. Though JAXA investigated the assessment following SPICE, it turned out that simple application of SPICE is not enough for critical software in the space development domain, which requires high reliability and safety. SPICE defines its Process Reference Model as ISO/IEC 12207 amendments[4], which is not specialized for critical software. Since it is designed to be tailored as appropriate in order to apply to real processes, it is prepared to be able to apply to broadly wide field from stand-alone software to embedded software and firmware[5]. As a result, it lacks some processes and practices important for assuring high safety and reliability.

There are some examples for SPICE application in critical software domains, like application for automotive device software by European automotive industry, for software in space development field by European Space Agency, and for nuclear power plant device software by Finish research program. All of the above examples use specifically customized process assessment model based on the SPICE model[6][7][8].

In the case of the automotive industry, some OEM manufacturers require their suppliers to carry out SPICE based assessment and then share the assessment results by OEM manufacturers in order to evaluate the suppliers efficiently[9]. In European Space Agency case, one of the assessment purposes is defined to evaluate whether contract requirements to its suppliers are satisfied[10]. In the nuclear power plant case, certification mechanism on some kind of software processes like software development processes is introduced as a part of a method to evaluate and certify software safety introduce[11].

As described above, even in the domain of critical software, from the view point of supplier evaluation by acquirer SPICE application are attempted and promoted. However, the objectives of the JAXA assessments are not defined as mere surveillance and evaluation of supplier activities, which is called process capability rating, but defined as elicitation of improvement tasks which are considered to lead to process improvement activities. Therefore, we implemented some trial application to the suppliers with the intention of fostering organizational self-sustaining improvement capability of the suppliers.

In this paper, we report some knowledge and experience attained through our improvement-oriented process assessment activities, aiming to share such information with many organizations which are considering and planning the application of process assessment.

2 Outline of Assessment

JAXA developed the JAXA Software Development Standard for the development processes of rockets, spacecrafts and ground stations. JAXA Software Development Standard, the development guideline for its suppliers, has been considered by the investigative commission consist of engineers of JAXA and suppliers since 2005. This applicability and effect have proven through some trials.

JAXA needs to verify the observance of JAXA Software Development Standard in the suppliers' development processes. The audit, confirming Yes or No as it always has been in the past, is inadequate to check specific efforts. Therefore, we adopted the assessment and made a method to verify the observance of JAXA software development standard.

In recent years, there is a great change in the software development in the space industry as well as other industries. Complexity and size of software are increased rapidly. Process improvement for these developments is essential in order to improve efficiency. Therefore, we arrive at a solution that we call "Improvement-Oriented Assessment" to find improvement tasks.

We produced the JAXA Process Assessment Model (JAXA-PAM) associated with JAXA Software Development Standard and applied it to the assessment reported in this paper.

The summary of this assessment is shown in Table 2-1. The target project is the ground station system being developed at present in Mitsubishi Electric co. and target processes are System requirements analysis, Software requirements analysis and Quality management. The objectives are facilitating process improvement for software development, considering the better development process with different fields' engineer, and clearing the problems of software development.

Table 2-1: Summary of Assessment.

Project	*Ground station system*
Process	System requirements analysis Software requirements analysis Quality management
Objective	Facilitating process improvement for software development Considering the better development process with different fields' engineer Clearing the problems of software development and sharing them
Interviewee	Project Manager Systems Designer Software Designer Quality Control Engineer Quality Assurance Engineer

3 Improvement-Oriented Assessment

This chapter describes the concrete methods of Improvement-Oriented Assessment as shown by 2.

3.1 Overall Assessment Process

The flow of this assessment is shown in Table 3-1. In the advance preparation, we decided the schedule and roles of each assessors. We made and reviewed the check lists, summary of check points through the evidence check and the interview, based on the pre-interview of the project manager.

In On-site 1, we convened the opening session and had an opportunity to get to know each other with assessees. We checked the evidence with the check lists and organized unconfirmed check points of them. And then, we made the interview scripts in order to clear the points of the interviews.

We spent 3 days for interviews and agreement of interview results in the assessor team. We reconfirmed the interview results with interviewees to bridge the perception gap between assessors and assessees. On the basis of these results, assessor team rated the development process. The process attribute rating is only additional information in the Improvement-Oriented Assessment: what is important is to consider the problems and develop improvement policy. Assessees prioritize improvement tasks in response to the results of the reporting.

Table 3-1: Assessment Activities.

	Subject	*Activity*	*Term (day)*
1	Kick-off Meeting	Planning	1
2	Advance Preparation	deciding the schedule and role pre-interview of the project manager making the check lists	2
3	On-site 1	Opening (explanation to assessees) Evidence check Making interview scripts	3
4	On-site 2	Interview Agreement of interview results	3
5	Summarizing	Agreement of assessment results Rating	1
6	Output Review	Reviewing the output	1
7	Reporting	Reporting the result to assessees	1
8	Developing improvement policy	Developing a policy for improvement	1

3.2 Characteristic Activities of Improvement-Oriented Assessment

3.2.1 Check List

In this assessment, check lists were used so as to organize the check points. The original check list is made from the summary of JAXA-PAM's check points and modified by the information of pre-interview. Assessors check evidence with this check list and add the information to it. We can interview appropriately and efficiently with the interview script which is generated with the check list.

Improvement-Oriented Assessment takes longer time than that of general assessments which confirm Do or Not. To get details in short time we need to reach a topic efficiently. In this point check list was useful. A sample of the check list is shown in Table 3-2.

Table 3-2: Sample of Check List.

JAXA-PAM		Check Points			Feedback		
ID	Text	Check Points (to whom and what will we ask？What will we check in documents ?)	Target Documents	Plans for Interview	Results of Document Check	Results	Findings
BP1	Ensure consistency of system requirements analysis to software requirements analysis. Consistency is supported by establishing and maintaining traceability between system requirements and the software requirements when needed.	<Evidence Check> In the software requirements specification, 1) does traceability matrix exist? 2) what is division level of traceability matrix? <Interview> To software developer 1) how and who makes traceability matrix	- ABC-12345 : XXX S/W requirements specification (software requirements specification)	1) How to maintain traceability with base documents? 2) Tell me division level of the traceability matrix.	In ABC-12345, traceability matrix exists, but its division level is defferent depending on the function.	There is the process of ensuring consistency (making traceability matrix). They don't have a rule of segmentation.	check point 2) doesn't comply with JAXA requirement.

3.2.2 Efforts for Information Gathering

At the interview session, we don't do Yes/No question in order to find problems lead to our process improvement. We change level of questions in accordance with the topics. For example, "what improves your effectiveness?" in a topic of strength. "what causes the problems?" in a topic of weakness.

3.2.3 Assessor Team

In a short time assessment it is difficult to find improvement tasks of assessees of such domain as space development which have a characteristic development pattern. Engineers who have domain knowledge become the assessors because they facilitate broad-ranging discussion. Engineers from other company or research laboratory also join the assessor team to prevent one-sided view from those who are brought up in the same working environment. The members and positions of assessor team are shown in Table 3-3.

According to Paul Byrnes et al., type of the assessment, like supplier evaluation by acquirer or internal evaluation for improvement purpose, is heuristically characterised by asking the following points: who is the sponsor, who participates the assessment team, how the assessment result is used[12]. In supplier evaluation case, for example, the sponsor will be acquirer organization, and the assessment team will consist of entirely external staff (either a customer or third party team). In internal evaluation case, the sponsor will be management of the organization to be assessed, members of the organization will be on the team, and the assessment result will be used within the organization to improve processes by providing input to action plan. In the assessment in this paper, the sponsor is the acquirer (JAXA), however the assessment result is intended to be used for encouraging the improvement activity within the supplier, and the assessment team consists of members from all of the acquirer organization, supplier organization, and third party organization.

By the diversity of the assessor team, we achieve some effects as below.

- Understanding the organizational structure and the process of the target project correctly
- advancing concrete discussions on weaknesses and challenges

Table 3-3: Composition of the Assessor Team.

	position	Category
A	Mitsubishi Electric	project manager of the target project
B	Mitsubishi Electric	on-board software engineer for satellites
C	Mitsubishi Electric	Quality assurance engineer
D	Mitsubishi Electric	in-company software researcher
E	JAXA	software improvement engineer
F	SRA	software improvement consultant

3.2.4 Pre-Interview

It is difficult that an interviewer who doesn’t have domain knowledge find out specific problems from assessees. In a short time interview sparing engineer's valuable time to understand their background is inefficient. Therefore the pre-interview, explanation of the organizational configuration or the technical vocabularies, helps interviewers to comprehend them.

4 Analysis on Assessment Data

As explained beforehand, this assessment was performed with the intention of detecting tasks which lead to improvement activity. As shown in chapter 3, some unique approaches are adopted in order to attain the assessment intention. This chapter explains benefits and issues of our approaches.

Table 4-1 shows the relationship between the check points used in this assessment and the process assessment model from which the check points are derived.

Table 4-1: Check Points and PAM.

Process	BP/GP	Practices	Check Points	Answers in Interview	Interview Time (minutes)
System requirements analysis	Base Practice	9	28	228	90
	Generic Practice	21	29		
Software requirements analysis	Base Practice	9	23	153	90
	Generic Practice	21	29		
Quality management	Base Practice	9	24	307	180
	Generic Practice	21	32		

Without comparable data, unfortunately, it is not possible to prove the quality of the assessment only by these data, however we think the result is successful one in the point that abundant of information are collected.

On the other hand, from the viewpoints of the quality of collected information, we unfortunately judge that some interview sessions resulted only to gather relatively, even if broadly, shallow information with time limitation, and were not successful to collect profound information as originally expected. Considering that the purpose of the check points is to collect essential information in the focused area where needed to improve, it may be that the preparation of the check points resulted as counterproductive to the original intention.

However, it is expected that by focusing prioritized points at first and then building up check points carefully, more high quality information will be elicited without reducing the amount of the collected information.

For example, when trying to collect profound information in focused area and also keep coverage of the target process, prepare questionnaire carefully in advance and collect some information for fundamental questions before interview sessions. And then focus on the prioritized area in interview sessions. With such approach it is expected that in the assessment target process will be covered efficiently and make more time for interview sessions to focus on prioritized area.

Table 4-2 shows the working hours and the number of engaged team members for the activities performed as the team task during the assessment.

Table 4-2: Record of Assessment Work

Tasks	Working Hours	Number of Staff
Kick-Off Meeting (1st assessment team meeting)	2.5h	12
Briefing for Assessment Team	2.0h	9
Assessment Team Meeting (2)	4.0h	9
Assessment Team Meeting (3)	4.0h	10
Assessment Preparation Meeting	1.5h	5
Document Review	13.0h	9
Assessment On-Site Opening Ceremony	1.5h	9
Interview Script Development	7.0h	9
Interview	8.5h	9
Verification of Interview Records and Consensus Building	6.0h	9
Observation Result Review	12.0h	9
Development and Review on the quick report on Assessment Observation Result	3.0h	4
Presentation of Assessment Observation Result	2.0h	9
Total	65.0h (9 Days)	-

Table 4-3 Working Hours Given to the Process of Building Team Consensus.

Tasks	Efforts
Building Consensus on the Method of Check List Establishment.	4.0h
Check Points Review (consensus building)	4.0h
Interview Points Review (consensus building)	7.0h
Consensus Building on Interview Records	6.0h
Consensus Building on Observation Result	10.0h
Consensus Building on Report	3.0h
Consensus Building Total	34.0h
Total	65.0h
Ratio of Consensus Building Working Hours	52.0%

As shown by these tables, building team consensus was challenging task and needed amount of working hours in this assessment. Taking into account that this assessment was performed with the aim of process improvement as its primary goal, these data can be interpreted as the result of a number of discussions between the members with different standpoints, which are considered to be fruitful activities according to the assessment goal. A huge variety of aspects and themes generated through the discussions during the consensus building helped identifying beneficial information which facilitate the deeper understanding of target organization and target process areas. They also helped to get balanced judgments on the assessment observations by covering various viewpoints and interpretation through the discussions and analyses with the members of different organization.

However, working hours for the assessment activities are the factor directly affects the cost and there is no doubt they are hard to ignore. In order to internalize the assessment activities in the organizations as the process improvement approach, it is essential to leverage the cost and effect of the assessment and then get commitment from management layers and organization members.

In this case the assessment was performed with the policy of focusing intensively on prioritized points and asks such points in the interview. However, in processes which had many points to confirm during the interview, interviewers were tend to be fully occupied to cover whole points addressed in the process of the assessment model, and then resulted to ask only wide and shallow questions. Therefore, it is needed to make more time for interviews, by such approaches as preparing questionnaire carefully, and handling fundamental issues before interview, etc.

However, with the approach of focusing intensively on partial areas, the assessment result will be significantly affected by the leader's judgments and assessment approach. Even the assessment will be carried out through the approach explained in chapter 3, more rules should be established.

5 Result of Improvement-Oriented Assessment

As I said earlier, the building team consensus needed amount of working hours because the assessment goal was to clarify the improvement tasks. After the presentation of assessment observation result, surprisingly, assessees agreed with almost all the results of the reporting. In the improvement-oriented assessment, we are checking not only ensuring that the objective evidence is sufficient to cover the scope but also evaluating their level and effect. We reported the improvement tasks and points that they need more efforts. The assessees investigated those problems and summarized how to improve them. Since assessors understood developing environment and background sufficiently and then rated, the report must have been accurate and acceptable to assessees.

After this assessment, many positive opinions were gleaned from a questionnaire to assessees. For instance, an assessee was able to find their challenges through this assessment. We reached the object of the improvement-oriented assessment which leads to improvement activity, because assessees understood that assessments were the important activities to recognize the current issues facing their departments or divisions objectively.

6 Conclusion

As stated above, we showed the concrete assessment process and results from the experience of the Improvement-Oriented Assessment. The purpose of this assessment is not evaluation of development capability, but rather detection of improvement tasks. The characteristic activities as referred to in chapter 3 made it possible to find out the details (strength and weakness) of the target organization. The continuing operation of this assessment can help its improvement.

Some problems of the Improvement-Oriented Assessment remain as a matter to be discussed further. It requires time for interviews or agreements in details compared to general assessments. Besides an encompassing assessment, we have to lower assessment costs with some methods. For instance, we can reduce load to check the points confirmed by repeat assessments. From now on, low-cost and short-time assessment process or methods will be needed to conduct assessments repeatedly. As a measure of effectiveness of an assessment, we need to develop the ways as referred to in chapter 4.

Acknowledgement

We would like to thank Professor Kumiyo Nakakojim from Research Center for Advanced Science and Technology (RCAST), University of Tokyo, Japan for proofreading this paper.

Literature

1. Yuko Miyamoto , Masa Katahira , Jürgen Münch , Alexis Ocampo ,"Integrated Framework of Development and Assessment Standards for Space System," Proceedings of PROFES 2008, 2008.

2. Paulk. M.C., et al, "The Capability Maturity Model: Guidelines for Improving the Software Process", Addison-Wesley, 1995.

3. ISO/IEC 15504-2, Software engineering – Process assessment – Part2: Performing an assessment, 2003

4. ISO/IEC 15504-5, Software engineering – Process assessment – Part5: An examplar Process Assessment Model, 2006

5. ISO/IEC 12207, Information Technology – Software life cycle processes, 1995

6. Automotive SIG, Automotive SPICE Process Assessment Model v2.4, Available from, http://www.automotivespice.com/

7. The European Cooperation for Space Standardization. ECSS-Q-80-02 Part 2A Draft 1, Software process assessment and improvement - Part 2: Assessor instruments, 2005

8. Halminen, J. and Nevalainen, R., "Qualification of safety-critical systems in TVO nuclear power plants." Softw. Process 12, 6 (Nov. 2007), 559-567.

9. HIS - Working Group Assessment, HIS-WG-Assessments v.31, Available from http://portal.automotive-his.de/index.php?option=com_content&task=view&id=22&Itemid=31

10. The European Cooperation for Space Standardization. ECSS-Q-80-02 Part 1A Draft 1, Space product assurance - Software process assessment and improvement – Part 1: Framework, 2005

11. Risto Nevalainen, Mika Johansson, Hannu Harju. "Process Certification in Safety-critical Domains," In Proceedings of EuroSPI 2008, 2008.

12. Byrnes P, Phillips M., Software Capability Evaluation Version 3.0 Method Description, Technical Report CMU/SEI-96-TR-002, 1996

Author CVs

Teiichiro Inoue

He obtained a Master in Aerospace Engineering in March 2003. He then started working at Mitsubishi Electric Co. During three years from 2003 to 2005, he is working at the design system engineering centre to carry out a project to improve software processes to whole company. Since 2006 he has been working for Mitsubishi Electric Co. Kamakura Works as a software engineer for satellites. He obtained iNTACS ISO/IEC 15504 assessor certification in 2008.

Makoto Shizunaga

He earned an academic degree in historical science in 1998.He then worked in a software vendor of electronics company affiliate before joined SRA Inc. in 2002. From 2004 he was on loan to JAXA for three years and engaged in software process improvement programmes during the stay in JAXA. He joined back to SRA Inc. in 2007 and has been working as consultant in the software process improvement field since then. He obtained iNTACS ISO/IEC 15504 assessor certification in 2007.

Tsutomu Kojima

He is one of the directors of the Japanese Software Process Improvement Consortium (JASPIC;http://www.jaspic.jp/). Since 1998, he has participated in an internal software process improvement activity at Software Research Associates, Inc. (SRA; http://www.sra.co.jp/index-en.shtml) that aims to implement CMM-based software process improvement (SPI) and ultimately to stem profit loss due to deficit projects. He has been working as consultant in the software process improvement field since 2004. He obtained iNTACS ISO/IEC 15504 assessor certification in 2009.

Yuko Miyamoto

She is an engineer in the Software Engineering Team, JEDI Center in JAXA. She is working on software process improvement and software IV&V. She received B.S., M.S. from University of Tsukuba. She obtained iNTACS ISO/IEC 15504 assessor certification in 2008.

The First Achievement of SPICE Level 2: a Case Study from Turkey

Burcu Büyükkağnıcı, Ebru Yazıcı, Özgür Tüfekçi

iNNOVA IT Solutions, Ankara-Turkey
{bbuyukkagnici;eyazici;otufekci}@innova.com.tr

Abstract

This paper presents a software process improvement and capability determination project where ISO/IEC 15504 Part 7 is used to determine organizational maturity level of iNNOVA IT Solutions. The assessment was critical for being the first ISO 15504 Class A Type 1 assessment in Turkey. The authors outline the software process improvement project, methodologies and tools used as well as the assessment method, findings and recommendations from the assessment, lessons learned and the planned actions to be taken by iNNOVA as the next step.

Keywords

ISO/IEC 15504, Software Process Improvement Project, Organizational Maturity, Process Assessment

1 Introduction

iNNOVA IT Solutions Inc. (iNNOVA) has been acquired by Turk Telekom, one of Europe's top five and the world's top ten fixed line telephone operators in 2007. Since then the number of engineers working at iNNOVA has been quadrupled and the company has doubled its revenues. As an inevitable consequence of this fast growth, iNNOVA now has to deal with new challenges as well as the existing ones which have now increased in magnitude.

With the rapid growth, it is now harder to use a common language in terms of processes in all the projects of iNNOVA, leading to the need for more comprehensive institutionalization. On the other hand environmental challenges that existed before are now harder to deal with the size the company has reached. Turkish software market is still immature and in the stabilization period. In addition to these market changes, public sector and Telco solutions have characteristic difficulties such as complexity and time critical delivery necessities with thousands of end users, time to market business goals and legislative obligations. Public sector solutions have further needs such as effective change management and scope management as the customer requirements are generally subject to change even after the analysis period. Under these circumstances, iNNOVA has to deliver high quality products despite all uncertainties and time stress, and should try to keep customer satisfaction high.

iNNOVA has, therefore, initiated a software process improvement project mainly to achieve better product quality, greater schedule predictability, productivity and consequently increased competitiveness. As İNNOVA, we have committed ourselves to improving the way we do our business.

One prominent approach to doing this is to integrate process modeling with assessments, which additionally is known to provide more accurate process ratings and higher quality process models [1]. Following the famous quote of Watts Humphrey [2], "If you don't know where you are, a map won't help" iNNOVA decided to have an assessment to see its position in the process improvement journey.

In addition to İNNOVA's commitment to improve the way it does business, driven by the above mentioned problems, there has been compulsory changes that has lead the company to invest in process assessment which may be listed as:

- Turkish Prime Ministry started to convince the public sector of the importance of implementing either SPICE L2, CMMI-3 or AQAP-160 in large scale e-government projects in 2006.
- Also Turkish State Planning Organization issued recommendations for standardization in IT sector and advised that ISO 15504, CMMI and AQAP 160 standards should be utilized for software quality in public sector projects.

These legislations will show their results in public sector IT procurements, and IT companies will have to represent their quality management capabilities through a certification in one of these standards.

In addition to that, Turkish Standards Institution (TSI) published TS ISO/IEC 15504 standards and started assessment facilities in 2009.

As a result such an initiative became inevitable and INNOVA began to implement ISO/IEC 15504 standards in its software development projects leading to SPICE certification in this field.

2 Background

2.1 About iNNOVA

iNNOVA provides consultancy, application development, technical support and training services in a wide range of fields including determining corporate information system strategies correct architectural construction, fortification with backbone applications; to integrating new technological solutions required by constant change with previous investments and to operational support which will lead to the optimum functioning of systems. Our solutions in diverse fields, such as portal applications, Telco OSS/BSS application development and integration; electronic payment infrastructures, electronic bill issuance and collection, enterprise resource planning, operational systems, corporate security and kiosk systems carry our business partners forward to e-business processes through new business models. iNNOVA is also the solution partner of international companies such as EMC, Microsoft, IBM, HP, SUN, Oracle, Juniper, SCALA, VMWARE, Symantec and Blue Coat.

Deriving its strength from its work ethic and a team of experienced IT engineers, who have proven their success, iNNOVA believes that every good service it provides will come back in the form of success and determines its priorities by taking the interests of its customers into consideration.

iNNOVA foresees the changing needs of its customers and constantly learns and works to create new products and services to meet those needs; places great emphasis on teamwork within the organizational structure and supports personal creativity on one hand, while diligently carrying out teamwork, which is the seal of quality for projects, on the other.

2.2 History of Quality Assurance Activities

Customer satisfaction and intrinsic product quality has been the two critical success factors for iNNOVA in the operating markets since it has been founded in 1999. In these circumstances, iNNOVA has defined its quality management system structure and achieved ISO 9000:2000 certification in 2006. iNNOVA has established its engineering, management and support process definitions based on ISO 12207 software development life cycle model. IEEE Software Engineering Standards were included as guides and templates to iNNOVA Quality Management System. Since document management, process automation and integration of CASE tools with development environments were seen as important factors as adaptable process definitions, investment was made to development infrastructure and key architecture team was established.

2.3 Moving to SPICE

In compliance with ISO 9001: 2008 standard, İNNOVA Quality Management System requires managerial reviews to be carried out regularly in the company. In the managerial review meeting of 2009, after examining the results of quality management system performance report, iNNOVA has decided to move its software development capability across organizational level and started a software process improvement project.

After this management target was set in the company, it was necessary to make decision on the process and evaluation model providing benchmarking options across most of the software development industry.

There are two generally accepted software process best practice models worldwide. One of them is ISO 12207 and the other one is CMMI for development. The two evaluation corresponding models are ISO/IEC 15504 and SCAMPI respectively. Both ISO 15504 and CMMI are accepted models by Turkish government.

iNNOVA has chosen ISO 15504 (SPICE) due to the fact that ISO 15504 is an international standard with emphasis on engineering processes. Since the process management system has already been established based on ISO 12207; ISO 15504 as an assessment model seemed to be more reasonable and suitable for the company’s culture. In addition to these factors, Turkish Standardization Institution has started to make ISO 15504 assessments. Considering technical as well as social/financial benefits for the country ISO/IEC 15504 was justified by board of the directors as the chosen assessment model.

It was decided that assessment covering mainly engineering, project's management and support processes would be of at most help in the short term. As we wanted a model that approves our organizational maturity, we wanted to execute the assessment according to ISO/IEC 15504 part 7 [3].

3 SPI Project Overview

3.1 Methodology

Software Process Improvement (SPI) activities, managed as projects, have always been ongoing efforts in iNNOVA. Derived from the SPI program model, IDEAL [4], iNNOVA has defined its own SPI cycle. Similar to IDEAL it is formed of a five step continual process. The steps are listed below:

1. Diagnostics & Definition of the SPI Scope
2. Planning the Project
3. Implementation of SPI
    a. Update of Process Assets
    b. Deployment of Processes
4. Assessment and Analysis
5. Closure and Next Cycle

3.2 Diagnostics and Definition of the SPI Scope

iNNOVA SPI projects are started, finalized or revised at the yearly managerial review meetings. The main inputs for the scope of the SPI project are the business needs of the organization and the status derived from the yearly Quality Management System Performance Report (QPR). QPR contains detailed information on planned versus actual results of the projects for predefined project goals, such as project schedule, problem resolution statements, bug criticalities, customer satisfaction ratings, quality audit results, review results, lessons-learned statements after the closure of the projects, and similar. These findings present an insight on the processes which the SPI effort should focus on.

The QPR findings together with the updated 2009 business plan led us to focus our efforts on a process improvement plan with the scope of engineering, project management and other project support processes, in order to get better results from the projects. With this particular goal in mind, the staged model of ISO/IEC 15504 was examined and processes of Level 2 were selected as the best practice set and the "to-be" baseline.

3.3 Planning the Project

A project plan was prepared by the Project Manager. Budget was determined.

The Project Management Plan consisted of:

- Process definitions,
- Preparation of templates and guidelines,
- Improvement of the documentation after revisions,
- Internal assessments
- Rework for corrective actions
- Trainings

The Project Plan, along with all the above mentioned documentation was published on iNNOVA's corporate a portal so that all the employees could easily access the information.

3.4 Implementation of SPI

3.4.1 Update of Process Assets

With the SPICE Level 2 target in mind, it was decided to identify and list the non-conformities or differences between our as-is software development processes and ISO 15504 level 2 requirements. Since iNNOVA is a software development company, ISO 12207 exemplar model and the minimum set of level 2 processes have been found appropriate to be selected for the assessment; therefore the gap analysis has been conducted for the following process list:

- ENG-1 Requirements Elicitation
- ENG-4 Software Requirements Analysis
- ENG-5 Software Design
- ENG-6 Software Construction
- ENG-7 Software Integration
- ENG-8 Software Testing
- SPL-2 Product Release
- ENG-11 Software Installation
- SUP-1 Quality Assurance
- SUP-2 Verification
- SUP-3 Validation
- SUP-4 Joint Review
- SUP-7 Documentation
- SUP-8 Configuration Management
- SUP-9 Problem Resolution Management
- SUP-10 Change Request Management
- MAN-3 Project Management
- MAN-5 Risk Management

This gap analysis included the steps of examining the defined processes, templates and form templates of iNNOVA, outcomes/products of projects and the implemented infrastructure in the Company. The results of this preliminary analysis have presented us the fundamental foundation for the improvement of iNNOVA's Software Improvement Project Plan. After gap analysis the plan and budget were revised and updated for the details.

Template and form definitions were the priory selected assets to be updated and made available for use in projects. The project audits were then conducted according to these new process definitions. During this period; a total of 20 processes, 60 templates and approximately 20 guides (mainly engineering, support and managerial) have been revised.

3.4.2 Deployment of Processes

An Application Lifecycle Management (ALM) tool integrated with the development environment were used in the projects throughout the process improvement activities, this tool has also been customized with respect to the needs of the new process.

In order to inform all personnel about the new processes, seminars have been organized within the Company. Meanwhile, iNNOVA has also regularly contacted its customers to present detailed information on these new processes.

The SPI Project lasted for about one year. All the data produced during the implementation phase were collected in the corporate portal and also in ALM database.

The two very important challenges within this period were process automation and execution of processes in a workflow-enabled system integrated with development environment; due to the proven experiences showing that the engineering and other managerial/support processes must be executed in an integrated environment with workflow automation in order to get, the most accurate real-life implementation results.

Helping us to deal with these two challenges were a powerful Application Life Cycle Management (ALM) tool already in use, providing process and workflow definitions and keeping all documents under version control. Based on the broad previous experience within iNNOVA, we strongly believe that ALM tools that especially integrate project management and engineering activities together as well as provide easier project and product tracking; significantly reduce process improvement perception delays and resistance against organizational change. By the help of this ALM tool, a total of 19 process workflows along with 16 templates were automated during software process improvement project.

Upon the final review and completion of process asset definitions and determination of application life cycle management tool; the project audits were planned. These audits were conducted as ISO 9000 internal audits and corrective/preventive actions were taken against non-conformances. All internal audits and product audits were conducted and recorded, with all corrective/preventive workflows being executed on this ALM tool where non-conformances were immediately fixed and closed.

3.5 Assessment and Analysis

Upon the completion of internal assessments, iNNOVA decided to be assessed by independent assessors to certify its international competency.

The independent assessment was conducted by one competent and two provisional assessors working within an independent assessment body. The assessment was CLASS 1 TYPE A assessment; lasted for 5 days.
A staged level part 7 assessment requires the selection of projects which are representative for the organization and where the level achieved in these projects reflects the level of the organization.
Selected projects, representative for the organization were e.g.: "Management Information System and Decision Support System Project of Turkish Privatization Administration" (OYBS) and "Management Information System Project of Agriculture and Rural Development Support Institution" (TKDK).

Thus two software development projects were selected as sample projects to be assessed, based on a rationale of selecting the projects those could best represent other projects of iNNOVA, with the selection criteria being product domain, project and product size, duration of the projects and the customer similarity. Selected sample projects, representative of the whole, were: “Management Information System and Decision Support System Project of Turkish Privatization Administration” and “Management Information System Project of Agriculture and Rural Development Support Institution”.
As the objective and scope of the assessment were defined, the assessment plan, sent previously by the competent assessor was updated. Then the interviewees - project team members in charge – were selected and the durations of each process interview sessions were planned.

The main focus of the assessment has been on the reviews of work products (outcomes) of the projects with the interviewees. Prior to the assessments the interviewees had prepared presentations about their work considering process attributes of SPICE, as a method to shorten the assessment time. Also helping to achieve results faster was a web based assessment tool used by the assessors to consolidate their ratings at the end of each assessment day.

On 5th and final day of the assessment; the assessors had prepared the final report along with a presentation of their findings, ratings and weaknesses, addressed to the Board of Directors; interviewees and the Software Process Improvement team. As a result, iNNOVA has been officially certificated for compliance with ISO/IEC 15504 maturity level 2.

3.6 Closure and Next Cycle

Now that software process improvement activities have been compared against an international standard framework, resulting in a positive outcome as INNOVA has been certified at organizational maturity level 2, the company is eager to go for more.

Even after this successful result, it is still a fact that the certification itself is not just a happy ending, nor a static stance for iNNOVA to rely on forever. It is only a benchmarking point, on which we had been standing on the final day, or a snapshot of the very moment. Improvement is a never-ending process in nature, and requires a dynamic approach to deal with changes, challenges and results. Moreover, this improvement process has once more proven that using a well-known process improvement framework will eventually provide a guide to determine a starting point and measure the progress. In this context, iNNOVA is determined and will continue its software process improvement efforts to higher levels, with quantitative management and innovation always in mind and in practice.

So a SWOT Analysis is performed to see the big picture. Summary of the findings is listed in the Table 1 below. The next cycle will go through this analysis and focus on how to eliminate weaknesses, overcome threats, keep strengths and reap the benefits of opportunities. The methodology will be repeated as explained in Section 3.1. The closure of this episode included evaluation of the lessons learned as explained in the next Section.

Table 1: SWOT Analysis

STRENGTHS	WEAKNESSES
• Proven success during the past decade, surviving two major economical crisis • Strong management sponsorship • Rapidly-growing organization • Turk Telekom group company • High team cohesion, high team motivation, high team technical capability • Strong application lifecycle management infrastructure • Open communication and democratic involvement of employees in decision making processes. • Very rich process asset library • Traceability is managed in a highly effective way	• Fast growing organization, changes in organizational culture • Improvement need in CM: Current branching strategy should be updated with a more sophisticated one. • For problem analysis; trend charts should be empowered and used for project reporting purposes. • Application lifecycle management tool should be customized in order to match project management's status reporting needs. • Quality management system should be improved in order to enable quantitative management needs. • Trainings should be extended to become organization-wide. • Process guides should be detailed (visual guiding is necessary, i.e: for workflows). Especially for document reviewing activities, the teams should be guided by detailed checklists. • To enable distributed development, communication tools should be improved such as ALM integrated messaging solutions.
OPPORTUNITIES	**THREATHS**
• Growing telecommunications market • Shortened software development lifecycle by elimination of rework and integrated application life cycle management environment • Cheaper global telecommunication costs open new markets as people connect to the Internet • Popularity among people for Internet access and rich internet applications • The demand for social networks, enterprise and mobile applications, content management systems, telecommunications sector-related software is growing.	• Global financial crisis • Fierce competition, many companies entering the market. • Technology life cycle is getting shorter.

3.7 Lessons Learned

Just like all project closures, the SPI closure also included an evaluation step. The following findings have been reported by the project team members:

- Process improvement activities need executive commitment and awareness of the teams participating in projects. Team members have to learn the processes, be eager to implement them and question the way they work to find better solutions to automate the processes and work more efficiently. Executive commitment has to be continuous for organizational changes.

- For this purpose, motivation of the team is crucial. Teams will be highly motivated when they are trained about the processes and find out how the improved processes save the lives of their projects, and how they can achieve their business goals while satisfying their customers.

- Periodical ISO 15504 trainings should be conducted within the orientation trainings for new employees. Quality assurance team should be strengthened both for periodical trainings and also for supporting the people having questions about the implementation.

- In addition to the ISO 15504 processes trainings, application lifecycle management tool trainings should also be conducted. For faster and valid data collection, each team member should use the tools without any doubt.

- A new SPI project will be started for improving weak points defined above where the same methodological cycle will be repeated.

4 Conclusion

As Humprey points out, it should now be clear to just about everyone in the software business that the current testing-based quality strategy has reached a dead end [5]. In order to continue to deliver high quality products in an environment that is getting more complex and as the number of customers and the product diversity increase; process improvement and dissemination of this culture is crucial for our company. Therefore, iNNOVA has dedicated its resources to continuous process improvement, which resulted in being awarded the SPICE Level 2 compliance certificate according to ISO/IEC 15504 part 7: Assessment of Organizational Maturity.

The first assessment identified the strengths and weaknesses of iNNOVA. Under the light of the recommendations for process improvement made by the independent assessors, iNNOVA will now establish and implement an action plan in accordance with these recommendations, since SPICE L2 certification is not a final goal for iNNOVA, where process improvement will continue with the next step which is to achieve SPICE L3 in the forthcoming year.

Literature

1. Barafort, B., Jezek, D., Mäkinen, T., Stolfa, S., Varkoi, T., Vondrak, I.: Modeling and Assessment in IT Service Process Improvement, EuroSPI 2008, Springer-Verlag Berlin Heidelberg, (2008)
2. Humphrey, W. Managing the Software Process, Addison-Wesley, (1989)
3. ISO/IEC 15504:2008 Part 7 Organizational Maturity
4. McFeeley, B., IDEAL: A User's Guide for Software Process Improvement, Handbook CMU/SEI-96-HB-001, Carnegie Mellon University, (1996)
5. Humphrey, W. Reflections on Magement, Addison-Wesley, (2010)

Author's CV

Ebru Yazıcı

Mrs Yazici was born in 1971. She has graduated from the Electrical and Electronics Engineering Department of Middle East Technical University in 1993. She has received her M.Sc. degree in 1997 from the same department. She worked for various IT companies and in year 2008 she joined iNNOVA IT Solutions as Program Manager of Public Sector Solutions

Burcu Büyükkağnıcı

Miss Büyükkağnıcı was born in 1982. She has graduated from Industrial Engineering Department at Middle East Technical University in 2003. After working for various IT companies including defense sector, in year 2009 she joined iNNOVA IT Solutions as a quality and process engineer.

Özgür Tüfekçi

Mrs. Tufekci was born in 1973. She has graduated from Industrial Engineering Department of Middle East Technical University in 1995. She has received her M.Sc. degree in 1997 from the Industrial Engineering Department at Bilkent University while working as a research assistant in the same department. Later, she moved to industry and worked in several IT companies and joined iNNOVA IT Solutions in 2010 as Quality Assurance and R&D Manager. Mrs. Tufekci is a Ph.D. candidate in Informatics Institute at Middle East Technical University.

A Cluster Approach to Security Improvement according to ISO/IEC 27001

Nicolas Mayer
Centre de Recherche Public Henri Tudor
29 av. John F. Kennedy, L-1855 Luxembourg
nicolas.mayer@tudor.lu

Abstract

ISO/IEC 27001 is currently the standard approach to Information System (IS) security. It explains how to establish an Information Security Management System (ISMS) which objective is a continual improvement of information security. The associated certification is an evidence for the stakeholders of the organisation that security risks are assessed and treated. However, this standard is still considered as difficult to implement by SMEs, mainly due to their limited financial and human resources. It is generally a costly process until being certified and a deep knowledge of the standard and its principles is required. In order to consider this issue, we developed within a research project an implementation guide, templates and software tools to assist SMEs in ISMS establishment. This paper presents the validation of these results through industrial experimentations in three different organisations.

Keywords

Information security, ISO/IEC 27001, compliance, SME

1 Introduction and problem statement

Information security is at the heart of Information Systems (IS). For example, in the past two years, 52% of businesses have experienced an unforeseen interruption, and the vast majority (81%) of these interruptions has caused the business to be closed for one or more days [1]. A challenge of today remains that security should be adapted to each organisation. Each organisation should select security measures that are suited to its security needs, instead of trying to target an unreachable level of security. In this context, the ISO/IEC 27001 standard [2] is a suited answer. The objective of the standard is a continuous improvement of information security. This continuous improvement is provided through the establishment and management of an Information Security Management System (ISMS). An ISMS involves especially that a risk management approach has been used to assess the security of the organisation, and thus that relevant measures have been selected and implemented. An ISMS of an organisation can be certified, in order to give some confidence to stakeholders that the ISMS is compliant with the standard and, therefore, that a continuous improvement of information security is guaranteed.

A key characteristic of the ISO/IEC 27001 standard is that it covers all types of organisations, irrespective of its size, origin or activity. Based on our previous work [3], an SME has some strength regarding the adoption of the principles of the standard, like flexibility and reactivity. However, they have also a main issue regarding the establishment of an ISMS: their limited financial and human resources. The establishment of an ISMS is most often a long and costly process, and it needs to involve people with a good knowledge of the standard.

The objective of our research work is to help SMEs to adopt the ISO/IEC 27001 standard, through the improvement of the ISMS establishment process. This paper is about the experiments of our research results in this field with different SMEs. Section 2 of the paper briefly presents the ISO/IEC 27001 standard. Then, Section 3 summarises our research results for ISMS establishment. Section 4 is about the two experiments performed and it reports the lessons learned. Finally, Section 5 draws the conclusion of these experiments and describes the future work.

2 The ISO/IEC 27001 standard

The outcome of ISO/IEC 27001 [2] is the effective establishment and management of an ISMS. The purpose is a continual improvement of information security. Relying upon quality management and ISO 9001 [4] principles, the standard is built around a PDCA (Plan-Do-Check-Act) cycle. It is necessary to note that the standard does not require nor induce an absolute level of security to reach. The objective is to ensure a constant alignment to the organisation security needs and to improve security over time. This objective is reached through the use of a risk management approach. It aims at selecting and implementing security measures that are suited to the security risks of the organisation.

The standard contains a set of normative requirements one must comply with to obtain the certification. They are expressed from Section 4 to Section 8 of the standard (see Figure 1) and also include Appendix A. The other sections are considered as informative, and thus are not mandatory for the certification.

It is necessary to establish and manage the ISMS by following the PDCA cycle composed of four iterative steps (described from Section 4.2.1 to Section 4.2.4 in the standard). The whole ISMS must be supported by a specific documentation, whose requirements are explained in Section 4.3. Additionally, some requirements are especially developed in a dedicated section, because of their importance or complexity. Thus, the standard includes sections regarding management responsibility (Section 5), internal ISMS audits (Section 6), management review of the ISMS (Section 7) and ISMS improvement (Section 8).

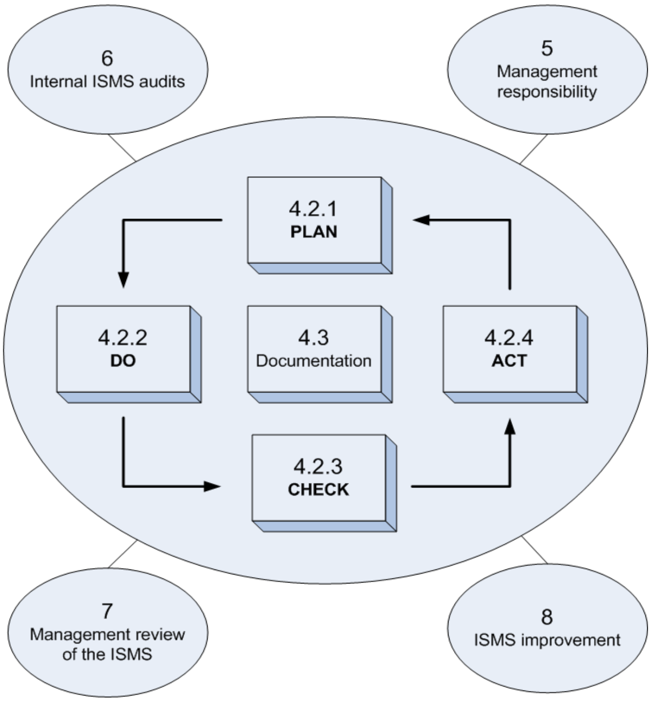


Figure 1: Representation of the ISO/IEC 27001 standard

Moreover, the standard requires to assess the selection of the security measures listed in Appendix A. This selection is performed and justified in a document called “Statement of Applicability”. Appendix A consists of a list of 133 security controls, based on ISO/IEC 27002 [5]. Those controls cover the complete scope of information security, by providing IT technical measures (e.g., system acceptance, protection against malicious code), management measures (e.g., security policy, business continuity planning), measures on physical security (e.g., secure areas, equipment security) and human resources security (e.g., security awareness, termination or change of employment).

3 Research results for ISMS establishment

The industrial experience report described in this paper is based on a previous research project [3]. The objective is now to assess and validate the results in an industrial context. The above mentioned project aims at identifying what are the specific needs of SMEs regarding ISMS, and how to establish an ISMS in an SME context. The outcomes of this project are 1) an implementation guide taking into account the specificities of SMEs, 2) templates and software tools supporting the implementation process described in the guide. In order to develop these artefacts in a structured way, we propose a research method following an action research approach [6]. The research method, presented in Figure 2, consists of three steps:

- **Step 1** – *Initial experiment*: An initial experiment of ISMS implementation in an SME context is performed, in order to identify the related issues.
- **Step 2** – *Building the guide and the templates*: The guide and the supporting templates and tools are developed based on the conclusions drawn during step 1. They are then reviewed by experts, in order to have a first level of validation.
- **Step 3** – *Experimenting the guide*: In order to strengthen the validation, industrial experiments are conducted.

It is necessary to note that Step 2 and 3 are performed iteratively, with incremental updates of the results. Step 1 and 2 are currently finished. Two expert review processes have been performed in Step 2. The two experiments planned for Step 3 are also finished and are the topic of this paper.

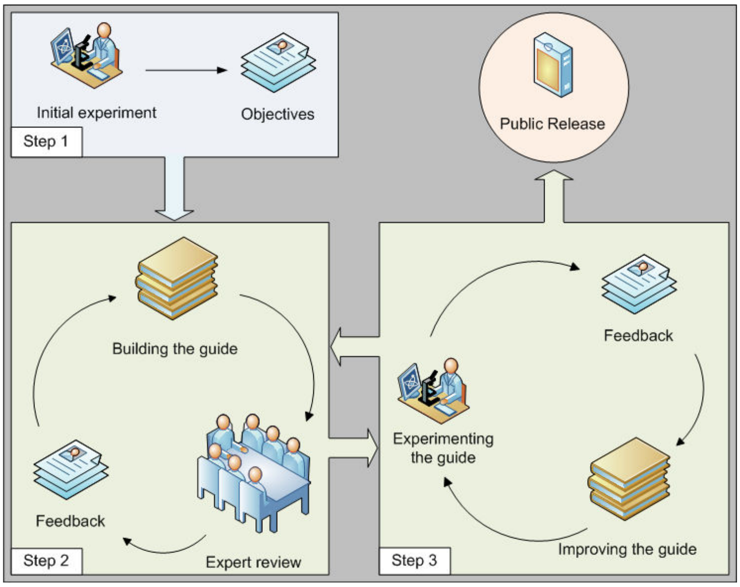


Figure 2: Research method of the project

4 *Experiments*

Two experiments were conducted in order to assess our research results. The first one was an experiment on a single organisation. The objective was to test our results (the guide, the templates and the tools) in a classical frame, through the establishment of an ISMS. The second one was more innovative, because we tested our results on a cluster composed of two organisations. After assessing the effectiveness of our research results in the first experiment, the objective was then to assess their efficiency in the second, by constraining the time and effort needed to establish the ISMS in both organisations.

4.1 Assessment of our research results with a single company

The first experiment took place in the Luxembourg's Ministry of the Economy and Foreign Trade (MEFT). The scope was not the whole MEFT, but only two departments. The first one was the Human Resources (HR) department and the second one was the department in charge of IT management and management of the national website about information security (http://www.cases.lu). The experiment started in April 2009 and ended in December 2009. The project team of the MEFT was composed of the leader of the HR department, two persons involved in the IT management and two persons involved in the website management. The objective of establishing an ISMS for the MEFT was to improve security management in these two key departments.

This experiment had an interesting initial context. The team had already a good knowledge about information security and the ISO/IEC 27001 standard. A risk assessment had already been done in 2006 with the EBIOS method [7]. Moreover, a security policy and some of the key procedures necessary for an ISMS were already defined, like incident management or control of documents. Finally, the security level of the organisation was already good and many security measures recommended in the appendix of the standard were already implemented.

The experiment work plan was composed of the following steps:

- Initial training about the ISO/IEC 27001 standard and our implementation guide;
- Gap analysis between the current state and the state-to-be;

- Update of existing procedures, based on our templates and tools;
- Definition of missing procedures, based on our templates and tools;
- ISMS validation;
- Management review of the ISMS.

The experiment is currently finished. The positive aspects in this experiment were the competence of people involved in the project. There were also some weaknesses mainly related to the availability of the team. This lack of availability implied that we were not able to completely perform our work plan. However, it was an interesting experiment, giving some interesting feedback to validate our results, as described in Section 4.3. It helped to strengthen our implementation guide and the associated templates. It was also a good way to validate the usefulness of the software tools we used during the gap analysis and the risk management steps.

4.2 A cluster approach to ISMS establishment

In this experiment, the innovation is in the cluster approach to ISMS establishment. In this context, a cluster is defined as the composition of a group of organisations wanting to implement the ISO/IEC 27001 standard. In our context, two companies form the cluster. Our assumption is that an efficient cluster should be composed from three to five organisations (see Section 5). However, in order to mitigate the risk related to this cluster experiment, and in order to respect the time and budget constraints of our research project, we restricted the cluster to two organisations for this experiment. The first interest of a cluster approach to ISMS establishment is the reduction of the associated cost. The cost of the trainings are supported by the whole cluster and thus divided by the number of organisations within the cluster. Another strength of the cluster approach is that the organisations can share their feedback about ISMS establishment. In each training session, some time is dedicated to this activity.

Some criteria are defined to well scope the organisations that can join the cluster:

- The organisation must be a small or very small enterprise. The total length of our approach and the different steps are sized for SMEs.
- The management commitment for ISMS establishment is a fundamental prerequisite. Without such a commitment prior to the beginning of the project, the risk of giving up during the project remains too high.
- The availability of enough human and financial resources must also be clear prior to the beginning of the project.

The cluster approach is composed of three parts. The first one is a gap analysis of half a day on site and half a day of result analysis. The second part is a training of 6,5 days, including an initial training of 1,5 days. The training is divided in sessions, most often on a half day basis. Finally, the third part of the approach is the individual coaching of each organisation. The coaching part consists of half-days of help to implementation, review and validation of the work done. The number of necessary coaching is different from one organisation to another. It depends on the competencies of the organisation, and on the work already done before the beginning of the cluster we can reuse for the ISMS. A coarse-grained estimation of this charge can be done through the gap analysis. We evaluate the coaching ceiling at 14 days.

The work plan of this experiment is:

- Planning definition;
- Initial training about the ISO/IEC 27001 standard and our implementation guide;
- Gap analysis between the current state and the state-to-be;
- Succession of “training / associated coaching” cycles on a 3-4 weeks basis.

For the training part, we need to divide the standard in coherent sets of requirements. The objective is to structure the trainings on a half day basis, in order to avoid to give too much information during each training session to the attendees. During trainings, each attendee shall learn a limited number of knowledge, he shall be able to apply during the weeks following the trainings. Moreover, the focus of these trainings is put on know-how needed by the attendees, like how to use our tools or templates. Most of the theoretical knowledge is learned during the initial training. It is highly recommended for each organisation to have two persons following the trainings, in order to have a backup for the different steps of the ISMS establishment. The trainings are:

- Scope, ISMS policy and assets (0.5d);
- Risk assessment (0.5d);
- Risk treatment and treatment plan (0.5d);
- Document management (0.5d);
- Security policy, security measures and effectiveness (2 x 0.5d);
- Human resources management (0.5d);
- Incident management and continual improvement (0.5d);
- Management review of the ISMS (0.5d);
- Audit (0.5d).

The first company in the cluster is IfOnline. IfOnline manages the IS of a building in Luxembourg, shared mainly by financial organisations. Its second activity is service provider for a HR software. IfOnline is a very small enterprise of four employees. The IS security of IfOnline is regularly audited. The objective of IfOnline is to be certified in order to reduce the length and scope of these audits. The project team is composed of a manager of IfOnline and a second employee. The scope of the certification is the whole organisation of IfOnline. The second organisation of the cluster is the Luxembourg Airport Authority (LAA). The LAA has many missions like the air traffic control under Luxembourg jurisdiction, the management of aeronautical telecommunications exchanges, the meteorological assistance to air traffic as well as international cooperation concerning climatology, etc. The LAA is an organisation of about 150 employees. Four persons composed the project team. A project manager was in charge of the project, and he was supported by three persons: two from the IT department and the physical security manager of the LAA. The LAA must comply with the European commission regulation No 2096/2005 laying down common requirements for the provision of air navigation services. A requirement of this regulation is to have a security management system established, especially in order to ensure “the security of operational data it receives or produces [...], so that access to it is restricted only to those authorised”. An ISO/IEC 27001 certification is thus a suited answer to satisfy this requirement. The scope for LAA is initially the administrative department, and will be extended in a second step to the whole organisation. The experiment started with the two organisations in October 2009 and will finish in April 2010.

The initial context of IfOnline is a good security and awareness level, because security is at the core of their business. Many recommended security controls are already in place. However, there is a lack at the documentation level and most of the controls are very few formalised. For the LAA, the initial context is different. They are already ISO/IEC 9001 certified [3] and thus many mandatory procedures are already existing (audit, management review, document management, etc.). A risk assessment focused on physical security has already been performed. The LAA shall mainly improve its information security, based on a complete risk assessment.

The experiment is currently finished, and both organisations are preparing the certification. For IfOnline, the ISMS is currently established in the whole organisation and the certification audit is planned for the last quarter of 2010. Regarding the LAA, some security measures in the administrative department are still missing. They are currently implemented in order to complete the ISMS establishment. From the third quarter of 2010, they will start to extend the ISMS scope to the whole LAA. They expect to be certified in 2011.

4.3 Lessons learned

For the first experiment with the MEFT, the guide and the associated tools were a reference helping to highlight the strength and weaknesses of the procedures and practices in use. For example, after presenting the documentation requirements of the standard and our implementation proposal, the conclusion was that our approach is more suited to the MEFT than their existing procedure. A second example is about the risk assessment approach. The review of the risk assessment was performed based on the results of the previous one. However, the method used was modified, and the data were gathered thanks to our software tool. This feedback reinforces the need to keep simple an ISMS and to propose examples that are suitable to SMEs. The second main observation of this experiment was the key motivation provided by the certification constraint. This point was missing in this experiment, because the testing of this new approach was the main objective for the MEFT, and not the certification. The certification encompasses some pressure coming especially from the deadlines imposed by the audits. When the certification is not an objective for the organisation, it is difficult to keep the pressure on the team. As a conclusion, even the experiment was not fully completed, the feedback from the MEFT is good, because the result is 1) a better understanding of the standard, its requirements and its setting up by the whole team, 2) an improvement of the procedures already in place, and 3) the update of their risk assessment, showing some remaining gaps in security and which new procedure and/or practice shall be put in place.

For the cluster approach, the difference between both organisations was very interesting for validation purpose (e.g., public/private, number of employees, activities, etc.). It showed that our package is suited to different SME's contexts. It was first interesting to observe that both organisations did not have any problem to speak about their feedback all along the experiment. Although we had some doubts at the beginning of the project about the opportunity to have open discussions between the organisations about their ISMS, they were not impervious to information exchange. This point was reinforced by the fact that the organisations were not rivals, working on the same market. Their feedback about this aspect of experience sharing was very positive. Then, the planning definition is a key step for the cluster approach. It is necessary to define at the beginning of the project a planning taking into account the potential unavailability or overloaded periods for the different organisation, in order to avoid that an organisation has some late. A continual progress of each organisation all along the project is necessary. Our initial planning took into account these unavailability periods and it was a success factor. Regarding the tools and templates, the feedback was first that they have more added-value with a low-maturity organisation. Naturally, an organisation already ISO/IEC 9001 certified will not use every templates of the package. However, even in this context, the efficiency of establishing an ISMS with our package is better than without any support. Time consumed for establishing an ISMS in this experiment was about a third of time consumed in the initial experiment of the project [8]. Finally, based on the feedback gathered, the most interesting aspect of this experiment was the model of the training part. The division of the training in half-day sessions and the mix between theoretical aspects, mainly learned during the initial training, and practical ones, during training sessions, was the key aspect of our model. The LAA and IfOnline confirmed that it is a cornerstone for an efficient progress all along the project.

At least, both experiments were the input for the next step of our project that is the transfer of the research results to SMEs and consulting firms. The objective is now to train professionals to establish an ISMS with our tools. A labelling scheme is currently defined. The label shall guarantee that its owner is able to provide an accompaniment to a SME all along its ISMS establishment. Each person wanting to obtain the label must first follow a training, then pass the associated exam and finally be evaluated by a coach during its first ISMS establishment mission. For the last part about evaluation by a coach during a concrete mission, the experience collected during our experiments provides us most of the requirements. The requirements shall reflect the different skills one must have to be effective during the mission. These requirements are organised into four categories reflecting the different parts of the accompaniment:

- Gap analysis
- Risk management
- Management system procedures
- Security policies and procedures

5 Conclusion and future work

This paper reports about the two experiments conducted in the frame of a research project [3] about the ISO/IEC 27001 standard in an SME context. The experiments aim at validating the research results of the project: a guide, templates and software tools that form a package for supporting the ISMS establishment in SMEs. The first experiment was the use of this package for establishing an ISMS in two departments of the MEFT. The second experiment was a cluster approach to ISMS establishment, performed with two organisations: IfOnline and the LAA.

The conclusion drawn from the experiments are first that our package is relevant for an SME. As explained in Section 4.3, the time needed for ISMS establishment is shorter with our tools than without. Moreover, the collective aspects of the cluster approach have shown their interests. More interesting, some SMEs acknowledge than without our package and our cluster approach, they would not be able to target ISO/IEC 27001 certification. It is necessary to note that a new cluster of 4 organisations shall start in the second quarter of 2010. The global objective of demonstrating to SMEs that they can establish an ISMS in a simple and efficient manner and that they can target the certification is reached.

Current work is about the transfer of our tools. A strategy is currently defined, in order to transfer to consulting firm and SMEs our package. A way of transfer we also consider is to start new clusters in which we deliver the training part and some consulting firms provide the coaching part. Finally, feedbacks in general on ISO/IEC 27001 in SMEs are positive and spur ourselves on going on in this way. The first private SME ISO/IEC 27001 certified in Luxembourg announced recently that their break-even point for the certification will occur three years after their certification.

Literature

1. Agility Recovery Solutions, Hughes Marketing Group. 2009. Disaster Recovery & Business Continuity Survey.
2. ISO/IEC 27001. Information technology - Security techniques - Information security management systems - Requirements. International Organization for Standardization, Geneva, 2005.
3. Thierry Valdevit, Nicolas Mayer, and Béatrix Barafort. Tailoring ISO/IEC 27001 for SMEs : A Guide to Implement an Information Security Management System in Small Settings. In Springer, editor, Proceedings of the 16th European Systems & Software Process Improvement and Innovation Conference (EUROSPI'09), 2009.
4. ISO 9001. Quality management systems - Requirements. International Organization for Standardization, Geneva, 2000.
5. ISO/IEC 27002. Information technology - Security techniques - Code of practice for information security management. International Organization for Standardization, Geneva, 2005.
6. David E. Avison, Francis Lau, Michael D. Myers, and Peter Axel Nielsen. Action research. Commun. ACM, 42(1):94-97, 1999.
7. DCSSI. EBIOS - Expression of Needs and Identification of Security Objectives. http://www.ssi.gouv.fr/en/confidence/ebiospresentation.html, France, 2004.
8. Nicolas Mayer. Model-based Management of Information System Security Risk. PhD thesis, University of Namur, 2009.

Author CVs

Nicolas Mayer received his M.Sc. degree from the University Henri Poincaré, Nancy, France, in 2004, and his Ph.D. degree from the University of Namur, Belgium, in 2009. He is currently product manager at the CRP Henri Tudor, Luxembourg, for the business line "Security & continuity management". His research interests are mainly in the fields of risk management, IS security and requirements engineering.

Enhancement of Information Systems Success using a Generic Framework for Business Process Requirements Elicitation

Albert Murungi[1], Mercy Amiyo[1], Josephine Nabukenya[1]

[1]Faculty of Computing and Information Technology, Makerere University, P.O.Box 7062, Kampala, Uganda
{amurungi, mamiyo, josephine}@cit.mak.ac.ug

Abstract

A number of organizations today have adopted Process Aware Information Systems (PAIS) to support their business processes. The success of such information systems is greatly dependant on the correctness of the business process model according to which it is configured. The correctness of a business process model is majorly affected by the quality of the requirements which depends on the elicitation process. However, there is lack of a generalized and systematic means to carrying out requirements elicitation that emphasizes the constant communication and prioritization. We therefore design a generic framework that entails a model which considers the two constructs, more so a set of guidelines that provide systematic steps/activities to be performed in order to achieve a comprehensive and satisfactory set of business process requirements. To design the framework, we followed the design science approach.

Keywords

Requirements Elicitation, Process Aware Information Systems, Business Processes

1 Introduction

A number of organisations today have adopted Process Aware Information Systems (PAIS) to support their business processes. A *Process Aware Information System* can be defined as “a software system that manages and executes operational processes involving people, applications, and/or information sources on the basis of business process models" [9]. A *business process model* is defined by Weske [29] as a representation of a collection of logically related activities that are performed to achieve a given business goal/outcome. To achieve a good business process model, it is vital for one to understand an organisation’s business processes and environment so as to establish the right requirements for business process modelling and the development of process aware information systems [29]. A *business process* as defined by Davenport [6] consists of a number of tasks that need to be carried out and a set of conditions that determine the order of the tasks.

Requirements elicitation (RE) is “concerned with learning and understanding the needs of users and project sponsors with the ultimate aim of communicating these needs to system developers” [40]. This concept has also been defined by several other researchers [3, 13, 23]. Didar and Chad [40] do highlight the fact that a number of process models have been proposed over the years for requirements elicitation and these do provide a generic roadmap of the process with the flexibility to adapt to contextual differences amongst individual projects. Didar and Chad [40], however, further point out that these models do lack definitive guidelines for requirements elicitation. Tsumaki [28] also brings into light the cost implication of failed systems and also goes on to state that similarly, fixing mistakes made at requirements elicitation stage accounts for 75% of all error removal costs. It is actually indicative that requirements elicitation is one of the critical stages [32] in systems development projects. The Standish group [37] survey reveals that requirements related factors are amongst those that greatly challenge project successes. Improvement of requirements elicitation requires us to first understand the underlying problems [14, 15], and as such Tsumaki [27] identified some of the problems with requirements which included; incomplete requirements, incomplete understanding of needs, un-testable terms, too many requestors, unorganized bulky information sources, unnecessary design considerations, ambiguous requirements, ill-defined system boundaries, overlooking tacit assumptions, poor users’ collaboration amongst others. It is therefore quite likely that improving how the industry performs elicitation could have a dramatic effect on the success record of the industry [16]. To this end we seek to place the correlation between requirements elicitation, business processes and information systems and as a result go ahead to describe a generic framework for requirements elicitation especially suited for developing countries.

Through proper requirements elicitation, we can achieve best process models that depict the series of activities or tasks performed to accomplish a pre-defined goal [8]. Such process models are used to configure information systems that support the activities. It is thus important to understand an organisation’s business needs before coming up with a supporting process model and information system. Davenport [5] concurs that companies actually fail to reconcile the technological imperatives of some of the information systems with the business needs of the enterprise itself and as a result this could contribute to the total or partial failure of the systems. More over when systems analysts fail to elicit requirements effectively, there is an overall negative cascading effect on the success of an information systems development project [41, 42].

Thus, in this paper we present a generic framework that entails a model with a set of guidelines for requirements elicitation to enhance the requirements elicitation process and the success of information systems in turn. In the section that follows, we explore requirements elicitation models with a focus on the state of practice as well as the challenges involved. Section 3 describes the research approach that we follow to design the generic framework. In section 4 we present a generic framework that entails first a model that takes into consideration the constructs of communication (feedback) and prioritization amongst the various processes and activities that should take place to derive requirements; secondly a generic set of guidelines that can be used to elicit the requirements. Lastly we conclude with a means of how we intend to validate the designed framework.

2 Requirements Elicitation Models: State of Practice and Challenges

The *requirements elicitation* (RE) process as has been defined in the preceding section involves "eliciting requirements and not just capturing or collecting them and hence the need for discovery, emergence, and development elements in the elicitation process" [3]. RE is performed incrementally over multiple sessions, and iteratively to increasing levels of detail and often partially in parallel with other systems development activities [40]. The result of this activity is usually a "detailed set of requirements in natural language text and simple diagrammatic representations with additional information including description of sources, priorities and rationale."[40].

Most requirements elicitation models focus on specific methodologies or techniques [15]. For example, Sommerville et al. [25] describe approaches for using viewpoints to elicit requirements. The Robertsons' Volere requirements methodology [23] presents a template-driven documentation generation approach including its activities with inputs, outputs and recommended techniques for every activity. Other researchers have developed specific process models that define how to use scenarios for requirements elicitation [30, 31]. Sutcliffe and Ryan [26] also do present a model of elicitation that combines scenarios, prototypes, and design rationale. Another model [38] does present a more collaborative approach involving Joint Application Designs. However, very few general models of elicitation do exist [15]. According to Aurum and Wohlin [3], the requirements elicitation process consists of six activities, that is; a) Understanding the application domain, b) identifying the sources of requirements, c) Analyzing the stakeholders, d) Selecting the techniques, e) approaches and tools to use, and f) eliciting the requirements from stakeholders and other sources. Merwe and Kotze [19] also suggest a five step approach that could be followed by developers during identification of business processes and process models, of which requirements elicitation is a vital part. The steps include a) definition of scope whereby role players and key persons for the development team are identified, b) Identification of procedure, c) data gathering, d) comparison of the acquired results, and e) verification of the results. These researchers provide techniques for requirements elicitation such as interviews, questionnaires, task analysis, domain analysis, introspection, card sorting, repertory grids, laddering, group work, brain storming, Joint Application Design, ethnography, observation, protocol analysis, prototyping, goal based approaches, scenarios, viewpoints to mention but a few. Technique selection is dependent upon factors such as the context of the system, time, and availability of resources such as human, financial and hardware resources, level of iteration, scope limitations, safety criticality of the system, and legal or regulatory constraints [4]. In general, requirements elicitation models either capture a specific methodology or technique whereas others simply model elicitation [15].

From the preceding discussion, we observe that the requirements elicitation models ignore the role of knowledge about the current problem, solution, project characteristics, known requirements, and missing requirements, in performing both elicitation and also selection of an elicitation technique [15]. Additionally there is need to achieve a highly interactive elicitation process that provides for strong foundations for the emergence, discovery and invention of requirements [3]. The other factors that are crucial for the success of requirements elicitation include the management of available resources (for example time, finances and human resources), analytical consideration of organizational business processes and prioritization of requirements identified by stakeholders [16]. Often times, system development teams struggle with fluctuating requirements, communication breakdowns and also face difficulties in prioritizing requirements [16]. This in turn greatly affects the requirements elicitation process leading to the scoping of inaccurate business process models, according to which information systems are designed and configured. This ultimately hinders information systems success.

To overcome some of the above mentioned weaknesses, Hickey and Davis [15] add a new elicitation technique selection process along with its driving characteristics to the existing general elicitation methodologies (see figure 1). They also do explicitly highlight the role of knowledge, provide a unified framework for understanding the purpose and role of requirements elicitation in software development, state the underlying assumptions, and also define the situational characteristics. Hickey and Davis [15] formulated a mathematical function to explain the operation of their model, i.e., $elicit_i\ (R_i, S_i, t_i) \rightarrow R_{i+1}, S_{i+1}$, whereby R_i are the currently known Requirements, S_i is the current Situation and t_i is the technique selected for elicitation. These three constructs are parameters for the $elicit_i$ function leading to identification of the new Requirements R_{i+1}, and redefinition of the Situational context S_{i+1}. However, because t_i is a member of T (the set of all known techniques), Hickey and Davis [15] apply the selector function $\sigma\ (R_i, S_i, \chi\ (T)) \rightarrow \{t\ \varepsilon\ T\}$ to choose all possible elicitation techniques for the current elicitation methodology and based on personal preference P, a researcher could choose to apply one of the possible techniques based on the selector function $\pi(\{t\}, P) \rightarrow t_i \in \{t\}$. $\chi(T)$ represents the characteristics of all elicitation techniques which captures some innate aspects such as reduction of ambiguity, effectiveness at helping people converge to a solution, conflict resolution, and raising new issues amongst others. The resulting elicitation function for an elicitation methodology M_j then becomes $elicit_i(R_i, S_i, \pi(\sigma(R_i, S_i, \chi(T)), P)) \rightarrow R_{i+1}, S_{i+1}$. This function can be repeated n times until all requirements have been identified for any methodology M_{jn}. The $elicit_i$ function and selector function $\sigma(R_i, S_i, \chi(T)) \rightarrow \{t \in T\}$ represent the two ovals in their model.

The Hickey and Davis model [15] provides a great insight into the need for requirements elicitation in software development projects. However, the model does not address the need for constant communication amongst the stakeholders, the need to prioritize and verify candidate requirements vis-à-vis the previously established goals, and also the need to have systematic steps to guide the entire process of requirements elicitation.

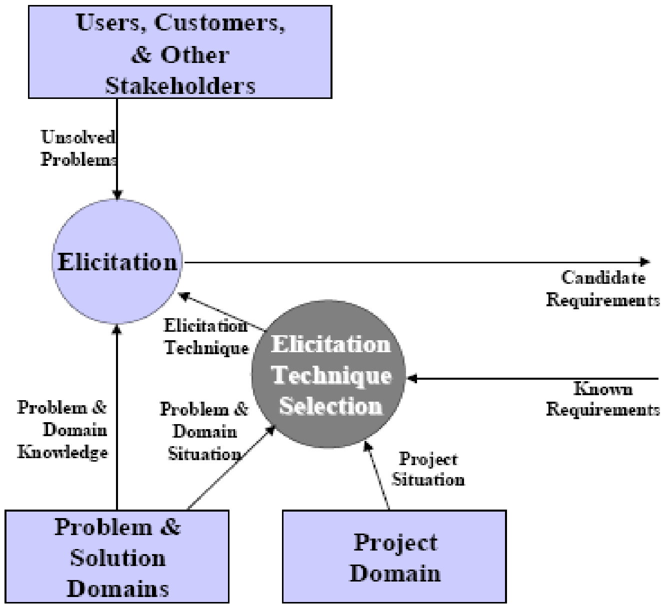


Figure 1: Elicitation activities (Source: Hickey and Davis [15])

Thus far, there is need for a generic requirements elicitation process model that emphasizes the need for constant communication (feedback), verification, and prioritization in addition to a generalized and systematic set of guidelines for carrying out requirements elicitation. To this end, we design a framework that takes into account the factors observed in the preceding paragraph. To achieve this, we build onto the model designed by Hickey and Davis [15] that represents a generalization of all known elicitation methodologies and techniques. We incorporate the constructs of constant communication and prioritization into our model as summarized in Table 1. To aid the selection of a relevant technique to use for the elicitation process, the choice should be dependent on the project context and this is also often critical in the success of the elicitation process [20].

Table 1. Matrix showing Reviewed Requirements Elicitation Models Vs Designed Model

Variable \ Model/approach	Merwe and Kotze [19]	Robertson's Volere model [26]	Hickey and Davis Model [15]	Designed Model
Generic guidelines	N	Y	N	Y
Communication (Feedback)	N	N	N	Y
Collaboration	N	N	Y	Y

Application of problem and solution knowledge	N	N	Y	Y
Technique selection	Y	Y	Y	Y
Prioritization	N	N	N	Y

Y- Yes (The model takes into account the variable)
N- No (The model does not take variable into account)

From the discussions above , we observe that the existing models for requirements elicitation do mention some of the major activities to consider for requirements elicitation such as stakeholder analysis, comprehension of the application domain, technique selection, and data gathering amongst others. Whereas some of these activities are mentioned by the models, the aim of these activities is not explicitly stated. Furthermore, some of the models do actually attempt to provide a generalized methodology for requirements elicitation as well. However, the existent models do experience some level of communication breakdowns, ignore collaboration amongst the stakeholders, and do not offer systematic generic instructions for requirements elicitation that are easily adaptable to most application domains. They also fail to address the issue of constantly changing requirements and inadequately use the existing knowledge about both the problem and solution domain.

3 Research Approach

In order to design the generic framework for requirements elicitation for business processes, we followed the design science research method. This method consists of three cycles namely, the relevance cycle, the design cycle and the rigor cycle [36]. We chose this method because it aims at producing artefacts that contribute to the body of knowledge and are relevant to the community [33, 34, [35, 36]. In this research, design science was used to highlight the relationships between the environment, the existing knowledge base and our research project. A review of multiple “state-of-the-art” [36] methods and practises in the requirements elicitation, business process management, and information systems application domains was done with an aim of positioning our research. Therefore past knowledge has been provided and also referenced in this research project to ensure its innovation [36]. Additionally design science has been successfully used in related researches such as [39] [34]. We designed a framework that entails a model for the requirements elicitation process that puts into consideration the communication (feedback) and prioritization constructs, and a set of guidelines on how to elicit requirements.

In conducting this research, existing literature was thoroughly explored to understand gaps amongst the existing business process requirements elicitation models from which the constructs used in the framework were derived. A model and set of guidelines were designed to guide the requirements elicitation process.

4 Generic Framework for Requirements Elicitation

In this section we present the a generic framework that is comprised of a model and a set of guidelines that consists systematic steps or activities that need to be performed in order to achieve a comprehensive and satisfactory set of business process requirements. The designed framework takes into consideration communication (feedback) amongst the various processes and activities that should take place during the requirements elicitation process, as described in the following section. The framework also emphasizes prioritizing and verifying the candidate requirements based on the existing goals before the final requirements are specified which also offers a solution to the emergent conflicting requirements.

4.1 The Generic Requirements Elicitation Model

Based on the unified model proposed by Hickey and Davis [15], we integrate the role of constant communication (feedback) amongst the processes, plus prioritization and verification of the candidate requirements based on previously established goals as well. We chose this specific model because it attempts to provide a general methodology for requirements elicitation while taking into account existing knowledge about the problem and solution domain, elicitation technique selection, as well as the application context (see section 2). The proposed model emphasizes the need to prioritize and verify elicited requirements whereby candidate requirements are compared against previously established goals before specification of the final requirements. This also helps in ensuring that conflicting requirements are identified to avoid duplication and reinvention of requirements. The rectangles in the model (see figure 2) represent the elicitation environment, that is, problem, solution, application/project domain and stakeholders, whereas the ovals represent elicitation sub-processes, that is, elicitation, elicited requirements and technique selection. The double ended arrows illustrate the feedback amongst the different constructs.

The designed model is functionally related to the generic guidelines that we describe in section 4.2. The sub processes closely interact with the environment so as to elicit the relevant requirements. The new construct of prioritizing and verifying candidate requirements seeks to test these candidate requirements based on set criteria before specification of the final requirements. It is from the environment that we find the unsolved problem and also the available knowledge about the problem. This knowledge interaction eventually leads to a solution domain that can eventually inform our elicitation process of the necessary requirements to be elicited. Furthermore, the selection of the elicitation technique is dependent upon the environmental constructs such as the situation and the already known requirements. The requirements elicitation process as a whole seeks to address the unsolved problem and also to establish a better understanding of the problem and domain knowledge. The model ensures that all the activities and sub-processes provide constant feedback to one another and are closely tied together to enhance constant communication as illustrated by the double ended arrows in Figure 2. We anticipate that the model will lead to specification of a complete set of requirements that are fully representative of the organizational business processes. This in turn will contribute to the design of process-aware information systems.

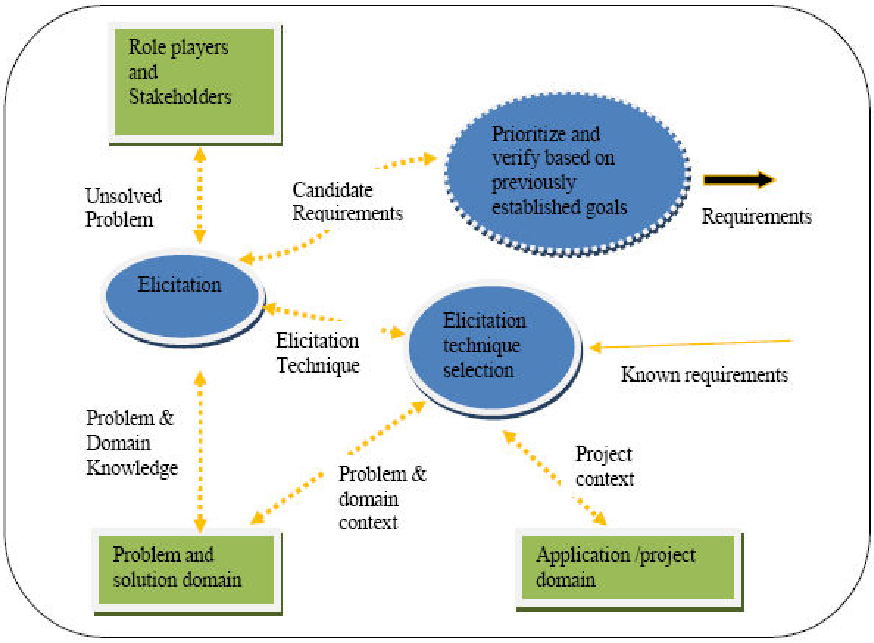


Figure 2: The requirements elicitation model

4.2 Generic Guidelines for Requirements Elicitation

To ensure proper use and the success of the model above, we put forward a set of guidelines that provides a generic and systematic procedure on how to carry out requirements elicitation. In the first step, in order to achieve a comprehensive and satisfactory set of business process requirements, it is important to understand the application domain in order to define the scope of the problem domain thus the second step. This forms the basis of the whole requirements elicitation process. With a well defined problem and solution domain, we move on to the third step whereby a proper analysis of identified stakeholders is made. Iteration between the third and second steps can be done until the right stakeholders have been identified. The next step involves identification of the sources from which requirements will be elicited from. Using the outputs from the previous step, selection of an appropriate data collection tool/technique/approach is then done. This is a vital step since it greatly contributes to the success direction of the data collection phase. It is after this step that data is then collected from the identified sources by applying the selected tool/technique/approach in the problem/application domain. Thereafter, the collected data is analyzed and business process requirements are established. To avoid emergent conflicting, unclear and incomplete requirements, the next step is aimed at prioritizing and verifying the established requirements against the previously established goals. This prioritization ensures that requirements are ordered according to their importance and urgency. In the next step the resulting list of requirements are verified by the stakeholders to make sure they are right and complete. If need be an iteration is done between the previous step and the first step to ensure that a comprehensive and satisfactory set of business process requirements has been achieved. The next step refines the candidate requirements and finally all of them are presented in form of a final requirements specification document. The step-wise flow of the guidelines is presented in Figure 3 followed by its description:

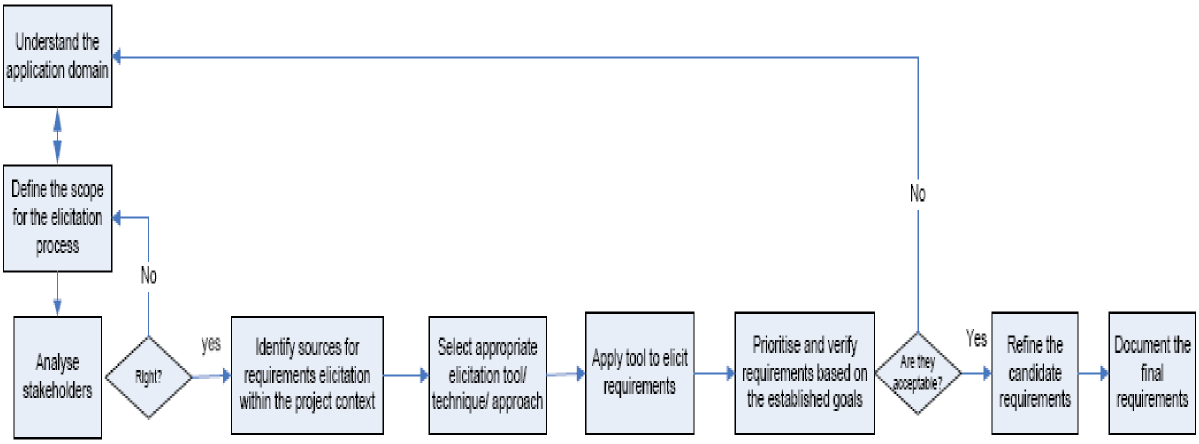


Figure 3: Flow chart of guidelines

1. *Understand the application domain* – This is aimed at ensuring that there is reconciliation and a detailed understanding of the relationship between business processes, organizational and technical environment. This initial activity forms the foundation for the whole requirements elicitation process. Some prompts that may be used to gain this knowledge are;

   a. What are the existing work processes?
   b. What are the problems to be solved by the system?
   c. How is the structure and maturity of the organization?

2. *Define the scope of the requirements elicitation process* – Having understood the application domain, we then define the scope of the elicitation process. This way, we are able to define the boundaries and constraints within which to restrict the requirements elicitation process. During this activity, the following aspects should be done;

   a. Define a high level mission statement for the project.
   b. Define the project context.
   c. Identify role players.
   d. Identify key persons.

e. Evaluate available resources, for example, Time, Finances, and Human resources.
f. Approval from management to undertake the project.

3. *Analyze the stakeholders* – Stakeholder analysis is aimed at ensuring that we do have the right individuals and role players for the requirements elicitation process based on the project context. During the analysis, a matrix for stakeholder comparison could be used or suitable and applicable criteria could be chosen to guide the process based on the project context. Stakeholders may be composed of either a) Internal or external groups of individuals, b) Customer/ project sponsor/ Actual user of the system and c) subject matter experts amongst others.

4. Having done the stakeholder analysis, we *check to see that the identified stakeholders fulfill the needs of the project* and if they do not then we iterate back to previous step (scope definition for the project). We continue to the identification of requirements sources if the identified stakeholders are acceptable.

5. *Identify the sources of requirements* – The possible sources of requirements are identified based on the already defined project context, as well as problem domain. These may include; stakeholders, users, documentation, subject matter experts, existing systems and processes. The identified sources of requirements will have an effect on the next step, that is, technique selection.

6. *Select technique/tool/approach* – Based on the defined application and project context, in addition to the problem and solution domain, appropriate techniques/tools/approaches are identified to perform the actual eliciting of requirements. The use of more than one technique/tool/approach is recommended. This step is also dependent on the results of the previous step since different sources of requirements will require different elicitation techniques, tools, or approaches.

7. *Apply tool to elicit requirements* – The techniques, tools or approaches selected at the step above are then used to elicit the requirements. Some key attributes to consider during this step are;

   a. Follow scope restrictions
   b. Detailed investigation of needs and wants of all stakeholders
   c. Determine future system processes
   d. Document all the requirements elicited

8. *Prioritize and verify requirements based on previously established goals* – This step is highly reliant on the results of the preceding step. Prioritize and verify the requirements based on previously established goals for the system. During prioritization, emphasis is put on ordering of the requirements based on the level of importance. The verification exercise should then endeavor to answer some of the following questions;

   a. Do the requirements meet the desired features of the system?
   b. Do the requirements provide all fundamental functionalities that fulfill the system objectives?

9. If the requirements are still not acceptable after the step above, *iterate back to the first step*. Relooking at the application domain and gaining its deeper understanding is bound to inform the elicitation process even more. However, if the candidate requirements are satisfactory, continue to the refinement stage.

10. *Refine elicited requirements* – Refine the elicited requirements in consultation with the various stakeholders through ways such as peer reviews and walk-through, interviews amongst others.

11. *Document the final requirements* into a complete requirements specification document.

If followed as prescribed, it is envisaged that the above model and guidelines should enable proper identification of business process requirements, leading to successful development of process aware information systems.

5 Conclusion and Future Research

In this research we have looked at a number of approaches that support requirements elicitation. Notwithstanding their efforts, we observed that the requirements elicitation process experiences communication breakdowns, lacks prioritization of requirements, and collaboration amongst the various stakeholders within the problem, solution and application domain. To address this gap, we designed a generic framework that comprises of a model and a set of systematic guidelines for carrying out requirements elicitation. The framework emphasizes the need for constant communication (feedback), collaboration and prioritization. Specifically, the model pretests the candidate requirements against previously determined goals and also encompasses the aspects of the application/project domain, problem and solution domain, requirements elicitation, and elicitation technique selection. We also set out generic guidelines that systematically state the activities to consider during requirements elicitation in addition to some guiding questions to ask ourselves during the same process of requirements elicitation. This will achieve a highly interactive elicitation process that provides for strong foundations for the emergence, discovery and invention of requirements and also helps to resolve the emergent conflicting requirements.

As a next step, we propose to conduct case studies to empirically validate the framework design to check if it indeed improves the business process requirements elicitation process and thus their effect on the success of process aware information systems.

References

1. Aalst, W.M.P., & Kees V.H.: Workflow management, Models, methods and systems. Massachusetts London, England: The MIT Press Cambridge (2002)
2. Argyris, C.: Management information systems: the challenge to rationality and emotionality, Management Science, 291 (1991)
3. Aurum, A., Wohlin, C.: Engineering and Managing Software Requirements. Springer Berlin Heidelberg (2005)
4. Avison, D., Fitzgerald, G.: Information Systems Development: Methodologies, Techniques and Tools. Berkshire: McGraw-Hill (2003)
5. Davenport, T.H.: Putting the Enterprise in the Enterprise System. Harvard Business Review (1998)
6. Davenport, T.H.: Process Innovation: Reengineering Work Through Information Technology. USA: Harvard Business School Press (1993)
7. Dumas, M., van der Aalst, W.M.P., ter Hofstede, A.H.M.: Process-Aware Information Systems: Bridging People and Soft-ware through Process Technology. Wiley-Interscience, Hoboken, NJ, USA (2005)
8. Curtis, B., Kellner, M., Over, J.: Process Modeling. Communication of the ACM 35(9): 75-90. (1992)
9. Davey, B., Cope, C.: Requirements elicitation- what's missing? Issues in Informing Science and Information Technology 5 (2008)
10. DeLone, W. H., McLean, E. R.: Information Systems Success: The Quest for the Dependent Variable. Information Systems Research, 3(1), 60–95 (1992)
11. DeLone, W. H., McLean, E. R.: Information systems success revisited. Paper presented at the Proceedings of the 35th Hawaii International Conference on System Sciences (2002)
12. DeLone, W. H., McLean, E. R.: The DeLone and McLean model of information systems success: a ten-year update. J. Manage. Inform. Syst., 19(4), 9-30 (2003)
13. Gottesdeiner, E.: Requirements by collaboration, Addison- Wesley (2002)
14. Hickey, A., Davis, A.: Elicitation technique selection: How do experts do it? 11th IEEE International requirements engineering conference, Monterey Bay (2003)
15. Hickey, A., Davis, A.: Requirements Elicitation and Elicitation Technique Selection: A model for Two Knowledge-Intensive Software Development Processes. Proceedings of the 36th Hawaii International Conference on System Sciences. IEEE Computer Society Press (2003)
16. Hofmann, H., Lehner, F.: Requirements engineering as a Success Factor in Software projects, IEEE Software, 18, pp 58-66. (2001)
17. Lindquist, C.: Required: Fixing the requirements mess: The requirements process, literally, deciding what should be included in software, is destroying projects in ways that aren't evident until it's too late. Some CIOs are stepping in to rewrite the rules. CIO, 19(4), 1 (2005)
18. Merwe, A., Kotze, P., Cronje, J.: The Functionality of a Requirements Elicitation Procedure Developed for Process Modelling within the Higher Education Application Domain. SACLA (2004)
19. Merwe, A. v., Kotze, P.: A systematic approach for the identification of process reference models. IASTED International conference. Innsbruck, Australia (2009)

20. Nuseibeh, B., Easterbrook, S.: Requirements engineering: A roadmap. In: Conference on the future of software engineering, pp. 35-46. Limerick (2000)
21. O'Brien, J.A.: Management Information systems: Managing Information Technology in the E-Business Enterprise. McGraw_Hill Higher Education (2002)
22. Rai, A., Lang, S. S., Welker, R. B.: Assessing the validity of IS success models: An empirical test and theoretical analysis. Information Systems Research, 13(1), 50–69 (2002)
23. Robertson, S., Robertson, J.: Mastering the requirements process. Addison Wesley (1999)
24. Seddon, P. B., Staples, S., Patnayakuni, R., Bowtell, M.: Dimensions of information systems success. Communications of the Association for Information Systems, 2(20) (1999)
25. Sommerville, I., et al., "Viewpoints for Requirements Elicitation: A Practical Approach," Third International Conference on Requirements Engineering, IEEE Computer Society Press, pp. 74-81 (1998)
26. Sutcliffe, A., and M. Ryan, "Experience with SCRAM, a Scenario Requirements Analysis Method. In: 3rd International Conference on Requirements Engineering, pp. 74--81. IEEE Computer Society Press (1998)
27. Tsumaki, T., Tamai, T.: Framework for matching requirements elicitation techniques to project characteristics. Software Process Improvement and Practice, 11, 505-519 (2006)
28. Urquhart, C.: Themes in early requirements gathering: The case of the analyst, the client and the student assistance scheme. Information Technology & People, 12(1), 44 (1999)
29. Weske, M.: Business Process Management: Concepts, Languages, Architectures. Springer Berlin Heidelberg, New York (2007)
30. Hsia,P., et al., "Formal Approach to Scenario analysis," IEEE Software, 11, 2 pp. 33-41 (March 1994)
31. Holbrook, H.: "A Scenario-Based Methodology for Conducting Requirements Elicitation," ACM SIGSOFT Software Engineering Notes, pp. 95-104 (1990)
32. Marakas, G., Uretta, M.A.: Requirements Elicitation Technique Selection: A Theory-based Contingency Model. Paper presented at the 1997 Americas Conference on Information systems (2007)
33. Winter R..: Design Science Research in Europe. European Journal of Information Systems, 17 (2008)
34. Carlsson A. S.: Towards an Information System Design Research Framework: A Critical Realist Perspective, DESRIST (2006)
35. Hevner, R. A., March T. S., Park J., Ram S.: Design Science in Information Systems Research. Management Information Systems Quarterly, 28(1) (2004)
36. Hevner, R. A.: A Three Cycle View of Design Science Research. Scandinavian Journal of Information Systems, 19(2) (2007).
37. The Standish Group Report,www.projectsmart.co.uk/docs/chaos-report.pdf (1995)
38. Liou, Y.I., Chen, M.: Using group support systems and joint application development for requirements specification. Journal of Management Information Systems, 10(3), pg. 25 (1994)
39. Becker, J., Weiß, B., Winkelmann, A.: Developing a Business Process Modeling Language for the Banking Sector – A Design Science approach. 15th Americas Conference on Information systems (2009)
40. Didar, Z., Chad, C.: Requirements Elicitation: A Survey of Techniques, Approaches and Tools. Engineering and managing software requirements. Springer Berlin Heidelberg, 19-46 (2005)
41. Pitts, M., Browne, G.J. :Stopping Behavior of Systems Analysts During Information Requirements Elicitation, Journal of Management Information Systems (21)1, pp. 203 (2004)
42. Chakraborty, S., Sarker, S.: Experimental Learning for the future systems analysts: Effective use of role-play in the classroom. In: Proceedings of the AIS SIG-ED IAIM 2007 Conference (2007)

Author CVs

Albert Murungi

Mr. Albert Murungi is an accomplished and seasoned IT professional currently working with the Infectious Diseases Institute; a public-private partnership with Makerere University. He graduated with a First Class Honors degree of Bachelor of science in Computer science from Makerere University and is currently pursuing a Masters degree in Information Systems from Makerere University. His research interests include requirements engineering, business process management, information security and systems engineering.

Mercy Amiyo

Ms. Amiyo graduated in October 2005 with Honors in BSc. Computer Science from Makerere University. She then attained a Masters degree of Internet and Database Systems from

London South Bank University in July 2007. Today she works with the Department of Information Systems, Faculty of Computing & IT, Makerere University as an Assistant Lecturer. She is also currently pursuing her PHD studies in the research area of Decision Enhancement Studio for business process Agility at Makerere Univerisity.

Dr. Josephine Nabukenya

Dr. Josephine Nabukenya has a PhD from Radboud University Nijmegen, The Netherlands, and is a Senior Lecturer and Head of the Information Technology Department, Faculty of Computing and Information, at Makerere University, Uganda. Her research focuses on issues related to designing of, and field applications of collaboration processes (Collaboration Engineering); Collaborative Decision Making; Analysis and Design of Information Flow; Process Modeling; and Requirements Engineering. Her articles have appeared in journals such as the Group Decision and Negotiation, Journal of Information Systems Security

Introducing IT Cost Services Management in a small enterprise*

Filipe Albero[1], June Amillo[1], Magdalena Arcilla[2], Jose A. Calvo-Manzano[1]
[1]Facultad de Informática, Universidad Politécnica de Madrid, Campus de Montengancedo s/n, 28660 Boadilla del Monte, Madrid, Spain
[2]ETSI Informática, Universidad Nacional de Educación a Distancia (UNED), 28040 Madrid, Spain
filipe.albero.pomar@alumnos.fi.upm.es; amillo@fi.upm.es; marcilla@issi.uned.es; joseantonio.calvomanzano@upm.es

**Authors are in alphabetical order*

Abstract

This paper shows a case study on the implementation of a financial management system related to IT services in a small enterprise whose core business is not IT-related. Process models (ITIL is the most popular) describe "what to do" but not "how to do". Moreover, small non-IT enterprises do not know the value of IT services provided by their providers, the value of the assets underlying the provisioning of those services, and the qualification of operational forecasting. Introducing Financial Management is the key to connect the business and IT, and in this way, getting operational visibility, and insight and superior decision making. A case study is presented describing the first steps given in order to introduce IT cost services management, enhancing decision making and obtaining operational control.

Keywords

Financial management, ITIL, non-IT small enterprise, IT costs, budgeting.

1 Introduction

The past decades have seen a dramatic transformation in the use of computer technology; from early 1980s to date, big and costly data centers have been replaced by personal computers that are affordable even to the smallest enterprises. The current ubiquitous presence of computers in corporations seems to support the common believe that they facilitate productivity growth, moreover such relationship has been proven true independently of company's size or sector [1].

Therefore it is not surprising that investments in Information Technology (IT) are an ever increasing part of corporate expenditures. According to a survey of corporate United States, the majority of CEOs and CIOs consider IT as a strategic advantage and roughly 25%, out of 297 executives consulted, believe that IT investments should increase during economical downturn [2].

However, companies rank IT importance differently depending on their size and how technology relates to their core business. Some compelling information [3] regarding the use of IT in enterprises indicates that 13% of the consulted companies consider IT at the core of their businesses and invest aggressively on technology to stay ahead of competitors. Another 34% of enterprises consider IT an important investment albeit not being their core business - such companies tend to be of medium or small size. What is surprising is that more than half of the companies, that is 53%, were not assertively investing on IT - such companies tend to be large companies with stable computing infrastructure in which upgrades are likely to be complex and daunting activities.

Therefore it would be logical to assume that IT expenditures are under control; after all, they are essential to productivity growth and considered an important asset, especially to medium and small companies. Though, that is not what happens. According to Schneider [4], as much as 86% of north American senior financial executives reported that their IT spending is not under adequate control. That is alarming news to the corporate world; but it is especially alarming to small and medium enterprises, which account for 99% of all companies in Europe – equivalent to 23 million businesses [5]. The representativeness of small and medium enterprises in the United States is also similar to the European reality; there, an estimate shows that there exist roughly 21 million enterprises with 20 or fewer employees [6].

Furthermore, it is a common practice for accounting departments to include all IT infrastructure costs in their reports without making any kind of distinction from the rest of expenditures.

This management approach may not aid the understanding of the real costs associated to the different IT services, due to their complexities, and give a false impression to the internal or external clients with respect to price and quality. In fact, one of the main problems in identifying the costs of IT services is that - from a user point of view - they are considered as a tool for carrying out routine activities; such a frame of mind often leads to the use of technology without much thought to its adequate use. Ignoring that the use of IT services incurs in costs leads to users not to be concerned for the correct use of those services and they cannot objectively evaluate if the services are of the expected quality. Thus, possible end-user complaints will unlikely be clear and based on objective issues, which unfortunately also lead to a distancing between the enterprise's core business and the IT services provided.

How could costs be controlled and assigned to the IT services provided by the IT department? How could IT budgets be forecasted for a given period of time and aligned to the real needs of the business? How could IT investments be evaluated with regard to the achievement of their goals?

The aforementioned questions can be answered if enterprises achieve adequate levels of IT services management that also acknowledges financial aspects. This way, organizations can follow best practices process models as reference.

2 Research methodology

The research methodology for implementing financial management in non-IT small enterprises is composed of the following activities.

1) Review of process models with respect to their guidance on financial management.
2) Review of financial principles with regard to budgeting and accounting.
3) Define the IT services (service catalogue) provided to non-IT enterprises.
4) Assign the IT cost to the IT services defined previously for a period of time (usually annual), in terms of fixed costs and effort. In this way, the total costs and effort are calculated.
5) Define IT budget for next time period. The budget is calculated by departments or units in terms of costs and effort taking into account current period data. The infrastructure's age is taken into account also for forecasting the purchase of hardware in the new time period.

3 Process Models

The main process models related to IT are ITIL v3 (Information Technology Infrastructure Library version 3) [7-11] and CMM-SVC v1.2 [6]. The financial management of IT services is mentioned explicitly in ITIL, but not in CMMI-SVC. For this reason, only a general view of ITIL v3 is presented along with a more detailed description of the financial management of this model.

3.1 ITIL v3

ITIL v3 is a collection of guidelines aimed at helping private business and government agencies to conduct IT services. The guidelines are maintained by many collaborators under supervision of the United Kingdom's Office of Government Commerce (OGC), which published the 3rd version of the standard composed of five publications: Service Strategy [7], Service Design [8], Service Transition [9], Service Operation [10] and Continual Service Improvement [11]. The Service Strategy volume provides guidance on how organizations can improve and develop their IT services relying in a market-driven approach. It is located at the core of the ITIL Lifecycle and it is the most relevant publication for financial management. It includes the following processes: Financial Management, Demand Management and Management of Services Portfolio.

ITIL v3 cites what activities have to be done in each process, though it does not explain how. A brief review of the volume Service Strategy is presented next because it contains the information most relevant to assess IT services finance in enterprises.

3.2 Additional Models

Conceptually, process models describe "what" has to be done, but not explicitly state "how" to achieve it. IT process models provide guidance on how services can be managed effectively throughout their life cycles - from conception to retirement - in a repeatable manner. However, financial management activities are left out of the main standard models to be implemented at business administrator's discretion; such characteristic provides an opportunity for further research on how financial management can be refined to improve process models. In this direction Table 1, compares the process models that have been reviewed and it presents the gaps in knowledge that this paper addresses.

Table 1. Comparison of Process Models regarding IT Financial Management

	Proposed framework	ITIL	CMMISVC	COBIT	eSCMSP	eTOM	ISO 20000	ITSCMM	OPM3	PMBOK	PRINCE2
Provides guidance on assessing IT expenses?	●	⊙	○	⊙	⊙	–	–	⊙	–	⊙	○
Provides guidance on identifying the Total Cost of Ownership of IT?	●	○	○	○	○	–	–	○	–	○	○
Suggests the use of techniques for Budgeting?	●	○	○	⊙	⊙	–	–	⊙	–	⊙	○
Financial guidance can be used in SME?	●	●	●	●	●	–	–	●	–	●	●
Defines IT cost factors?	●	○	○	○	○	–	–	○	–	○	○
Legend: ● yes, ○ no, ⊙ partially[1], – data not available											

4 *Financial Management*

The financial management process of IT services is implemented through three main activities: budget elaboration and control, accounting and billing (also known as charging) [12]. Next a brief description of each is provided.

4.1 Budget Elaboration and Control

The goals to be achieved by budgeting are [7] [12]:

- Predict and estimate the money necessary to deliver the IT services during a predefined period,
- Ensure that the business has enough funds to cover the needs of the IT services,
- Ensure that the IT service level agreements can be delivered during the determined period,
- Ensure that the real expenses can be compared with the predicted expenses at any time,
- Allow for early warnings of excess or lack of resource use, and
- Ensure that income will be able to cover for the forecasted expenses - in cases where charging applies.

Among the many different methods used to draw up a budget, the most usual are [7] [12]:

- Zero-Based Budgeting: this type of budgeting does not take into account any historical data. It is needed to detail and explain all the outcomes, associating the expenses and costs with existing resources and services provided. It is a costly activity and it can be used a single time to define the first year's budget and then use the Incremental Budgeting technique for following years, and
- Incremental Budgeting – consists of using the previous year budget and modifying it according to the expenses forecasted by the IT department and the business management.

Once a method for budget implementation has been chosen, it is necessary to define the categorization for each cost element (associated to each service); this service categorization should be maintained over time to allow for follow-ups in future budget periods. Among the categories to be defined, it is needed to take into account all future costs that can be forecasted as well as existing contracts (i.e. internet connection), real state rent, and staff salary, among others. Moreover, as much as posible, it is also needed to estimate all the known costs based on previous periods (i.e. overtime worked by the engineer in the previous fiscal year).

---

[1] Means that some information is available, albeit not necessarily detailed or comprehensive

4.2 Accounting

The most relevant goals of this activity are [7] [12]:

- Assess the money spent providing IT services and compare it with the forecasted budget,
- Identify and categorize the different cost elements (associated to services),
- Calculate the costs incurred by providing the different IT services to both internal or external clients,
- Aid the business decisions based on the costs of IT services,
- Identify the costs associated to changes. Make daily decisions with full understanding of the implication regarding costs and risks,
- Offer detailed information regarding the sources of IT service expenses,
- Define an adequate profit strategy for acknowledging the benefits and costs provided by the use of technology and identify different working scenarios, and
- Perform a cost analysis and Return on Investment (ROI).

From the business point of view, the approach and definition of accounting and charging should consider the IT infrastructure strategy.

As it has been discussed, among the main goals of accounting, it can be highlighted the tasks of defining the elements of cost and their association to IT services provided.

Once the complete structure of the services has all costs associated by element, it is possible to determine the costs per service. Having this information, and depending on the accounting method defined by managers, it is possible to define the prices of IT services and finally to realize the billing of attribution of costs.

The main goal to apply an efficient cost control is to take into account their nature and to be able to assign, to each element of the IT infrastructure, the corresponding costs, following the definition of cost model defined by the enterprise.

4.3 Billing or Cost Attribution

The fact that the organization of IT implements a billing process or attribution of charges depends entirely on a business decision, according to its strategy and operation. If senior management decides that the organization of IT services should function as an independent business unit, receiving its full support, it would be posible to implement billing activities to:

- Recover all costs of the IT services provided, in a fair and accurate manner, from the customers that use such services, and
- Influence the behavior of customers and users, making them aware of the cost associated to the IT services provided by the organization and encouraging them to make a good use of such services.

A billing or charging process should have the following characteristics [7] [12]:

- Simple. The benefit of implementing the billing should begin with simple procedures, not complex ones. The complexity can lead to distort the current roles defined in the organization, prioritizing more administrative tasks that those of IT,
- Fair. There should be a balance between cost and charge, which will help in the search for efficiency, and

- Realistic. We must never lose sight of the strategy of the business. If the business loses, the IT organization also loses. IT organization should not be inefficient and a drag for the business or the company runs the risk of becoming less competitive.

Moreover, there are certain factors that should govern the requirements for charging policies to be implemented:

- Determine the required level of return from expenditures. If the organization of IT chooses to recover all costs, it may choose to function as an autonomous unit,
- Determine the desired degree of influence on the behavior of customers and users; this should promote more efficient use of resources without making the business less competitive, and
- Design the policy of charges to facilitate the recovery of the costs on the basis of use.

5 Case Study

THE COMPANY 91 SL (fictitious name for confidentiality reasons) is an organization with headquarters in Madrid founded in 1991. It currently employs 17 staff members and is organized around the following departments:

- Department of Accounting and Finance (DAF); is responsible for providing fiscal consultancy and the accounting of the company's clients, as well as its own accounting.
- Department of Labor (DL): is responsible for carrying out payrolls, TCs, update the Social Security status of the company's clients, as well as all the activities related to its own human resources.
- Department of Mortgage Management (DMM): is responsible for everything related to the management of real state certificates, deeds and processing of inheritance of the company's clients,
- Department of Legal (DC): is responsible for carrying out legal advices, issuing criminal certificates, wills, and everything related to the efforts related to the Division of Motor Vehicles (as, for example, renewal of a driving license).
- Department of IT (DIT): responsible for providing support to the rest of departments. In addition it is also responsible for ensuring compliance to ISO Quality 9000 and LOPD (Organic Law of Data Protection).

With respect to the infrastructure, THE COMPANY has a midrange server for hosting data and applications, 16 personal computers and 2 laser printers (all connected through the internal LAN). Each department only can have access (through licenses) to the applications needed for performing their daily work which are installed in the server. There are 4 matrix printers needed for printing official documents. Table 2 summarizes the infrastructure of the company.

The company also maintains a web site that provides all the services of the company, on the company's web page. The web site is hosted at a third-party ISP server, which also provides storage and security services.

Table 2. Infrastructure of the company by department

Departments	Accounting	Labor	Legal	Mortgage	DIT
Specific Applications (number of licenses)	Contab (6)	Lab (5)	GestorC In-house (4), Traffic (4)	GestorM In-house (3)	
Number of PCs (oldness)	5 (3-2006, 1-2007, 1-2008)	4 (1-2007, 2-2008, 1-2009)	4 (1-2005, 1-2006, 1-2007, 1-2008)	3 (1-2007, 1-2008, 1-2009)	
Number of Laptops (oldness)	1 (2009)	1 (2009)			1 (2007)
Matrix Printers		1	2	1	
Base Applications	Windows 2000, Office 2003				

5.1 IT Services

The Department of IT consists of a single person (an engineer in computer science). The engineer is responsible for maintaining the infrastructure of the company, as well as solving the incidents and problems of the users of both staff and web users. So, the Department of IT provides the following services to the rest of departments of the company (see Table 3).

Table 3. Infrastructure of the company by department

Category	Services
HARDWARE	Installation (HI), Updating (HU), Maintenance (HM)
OPERATING SYSTEM	Installation (i.e Windows, Linux) (OSI), Updating (OSU), Maintenance (OSM)
BROWSER	Installation (i.e. Internet Explorer, Firefox) (BI), Updating (BU), Maintenance (BM)
DEPARTMENT SPECIFIC SOFTWARE	Installation (DSSI), Updating (DSSU), Maintenance (DSSM)
COMMON SOFTWARE	Installation (i.e. Microsoft Office, printer drivers) (CSI), Updating (CSU), Maintenance (CSM)
E-MAIL CLIENT	Installation (i.e. Microsoft Outlook) (ECI), Updating (ECU), Maintenance (ECM)
TELECOMMUNICATIONS INFRASTRUCTURE	Maintenance (monitoring of the external provider services), as well as the functioning of the broadband internet connection (TIM)
WEB SITE	Maintenance (WSM), Backup (WSB)
BACK UP	Backup of the company data hosted in the main server (BD)
QUALITY MANAGAMENT	Inspection and maintenance of quality attributes, including ISO and LOPD guidelines (QM)

The difference between updating and maintaining services lies in the fact that the updating installs new versions of the services, while the maintenance refers to resolution of issues.

5.2 Financial Management of IT Services

The company currently works in an ad-hoc manner with regard to IT and only accounts for the overall costs of the different services and the staff of IT; also, it has no control of IT costs per department. Therefore, in the next sections the accounting of IT for the year 2009 is carried out, and based on it, an incremental budget is proposed for the year 2010.

5.2.1 IT Costs, Financial Year 2009

In 2009, the total costs incurred by the company with regard to IT were of €53,010.42 out of which €12,210.42 was related to payments to third-parties and €42,000 was related to the salary of IT staff. Facilities rent for external services such as electricity or water bills were not included in these figures.

Table 4 shows the costs of external service providers by each of the IT services during the year 2009. The column Service indicates the service in question, the column Provider indicates the name of the external supplier (fictitious name) of the service, the column Fixed Costs indicates the corresponding fixed costs for the payments to external suppliers, the column Effort indicates the effort in hours devoted by IT staff to carry out a given activity (it is the effort by the engineer of the IT Department to carry out updating and maintenance of relevant services).

In the case of installation costs, the only costs taken into account were the ones related to the 2 computers bought in 2009.

The efforts of the IT staff were estimated, because there is not an institutionalized practice of reporting worked hours.

Table 4. IT Costs, FY 2009

Services	Provider	Fixed Costs	Effort
HI	FD	€1,200.00	
HU	FD	€255.62	2 h
HM			30 h
OSI			
OSU	Microsoft		120 h

Services	Provider	Fixed Costs	Effort
OSM			30 h
BI			
BU	IE, Mozilla		120 h
BM			
DSSI			
DSSU	SS, GI, Traf, CC, S	€2,614.59	22 h
DSSM			60 h
CSI			
CSU	Microsoft /Adobe/		120 h
CSM			
ECI			6 h
ECU			
ECM			6 h
TIM	CT	€1,494.00	72 h
WSM	A, Google	€6,646.21	100 h
WSB			24 h
BD			12 h
QM			200 h
TOTAL		€12,210.42	923 h

If the effort made by the staff of the IT Department and the cost per hour that could be incurred by a third-party provider (for example, 60 €/hour) are taken into account, it is clear that keeping the in-house IT Department is saving €13.380,00 yearly (€55.380,00 that would be paid to the external supplier company compared to €42.000,00 related to the worker of THE COMPANY).

5.2.2 *IT Budget, Financial Year 2010*

The budget of IT for the year 2010 is going to be based on the incurred costs of 2009. In addition, it will be considered also an increase in the Consume Price Index of 0.8%. The budget of IT for the year 2010 will be based on the various departments of the company. The cost model chosen will be based on the number of computers (PC and laptops that each department have) and licenses.

Table 5 shows THE COMPANY's IT budget for 2010. The column Service indicates the identifier of the service, the column Accounting indicates the Department of Financial Accounting, the column Labor indicates the Department of Labor, the column Legal corresponds to the Department of Legal and the column Mortgage to the Department of Mortgage.

THE COMPANY will buy 5 PCs (3 for the Accounting Department - €1,800.00, and 2 for the Legal Department -€ 1,200.00) and a new server with its operating system (estimate cost is divided between the 5 departments - € 1,700.00 the server and € 1,000.00 the base software). Also, it is going to install Windows 7 in 10 PCs (5 in the Accounting Department, 3 in the Labor Department and 2 in the Legal Department).

The budget of IT will correspond to the sum of the budgets for all departments plus the cost of staff related to the IT Department (€ 42,336.00); each cell indicates the budget for the estimated effort spent in each department (in minutes) by the IT Department.

Table 5. IT Budget, Financial Year 2010

Service	Accounting	Labor	Legal	Mortgage	IT
HI	€2,140.00 120'Añadir	€340.00 120'	€1,540,00 120'	€340.00 120'	€340.00 120'
HA	€93.80 22'	€67.00 16'	€53.60 13'	€40.20 9'	
HM	663'	474'	379'	284'	
OSI	€825.00 210'	€575.00 150'	€450.00 120'	€200,00 90'	€200.00 240'
OSU	2.653'	1.895'	1.516'	1.137'	
OSM	663'	474'	379'	284'	
BI					
BU	2.653'	1.895'	1.516'	1.137'	
BM					
DSSI					
DSSU	€1,400.33 486'	€877.50 347'	€357.33 278'	208'	
DSSM	1.326'	947'	758'	568'	

Service	Accounting	Labor	Legal	Mortgage	IT
CSI					
CSU	2.653’	1.895’	1.516’	1.137’	
CSM					
ECI	133’	95’	76’	57’	
ECU					
ECM	133’	95’	76’	57’	
TIM	€554.82 1.592’	€396.30 1.137’	€317.04 909’	€237.78 682’	
WSM	€2,468.20 1.500’	€1,763.00 1.500’	€1,410.40 1.500’	€1,057.80 1.500’	
WSB	531’	379’	303’	227’	
BD	265’	189’	152’	114’	
QM	4.421’	3.158’	2.526’	1.895’	
TOTAL	€5,342.15	€3,951.80	€4,128.37	€1,875.78	
	346 h	247 h	198 h	149 h	6h

6 Conclusions

ITIL provides companies of all sizes with a general guideline for managing IT services costs. However, the model provides with little information on the activities that are needed to properly carry out the management of costs. In this direction, this paper presents a concise process for financial management of IT services and its application to a small company whose core business is not focused on IT.

Taking into account the above results, we can conclude that a small business can benefit from IT financial management, and have greater control over its IT expenditures. Cost of provision of each IT service or business unit has been understood and accounted, and future expenditures have been forecasted.

Future work could: a) explore the mapping between IT services and their IT Cost Factors - which are the smallest IT units that incur cost; b) account for the Total Cost of Ownership of IT assets; c) accommodate a comparison of forecasted costs against actual costs; d) integrate closely to accounting scorecards, and e) also develop a robust application to be used by small enterprises.

Acknowledgements

This work has been sponsored by the *everis Foundation* and the *Technical University of Madrid* through the “Cátedra de Mejora del Proceso Software en el Espacio Iberoamericano”.

Literature

1. E. Brynjolfsson and L. M. Hitt, Computing Productivity: Firm-Level Evidence, Cambridge, MA: MIT Sloan School of Management, (2000).
2. L. Cosgrove and W. a. B. Worthen, Findings from CXO's IT Spending and Lessons Learned Survey, in CIO Magazine, (2001).
3. K. Huang, Towards An Information Technology Infrastructure Cost Model, in Center for Technology, Policy, and Industrial Development. vol. Master of Science in Engineering and Management Cambridge, Massachusetts: Massachusetts Institute of Technology, (2007).
4. C. Schneider, IT: Hold on to Your Wallet!, in CFO Magazine, (2000).
5. Commission Recommendation 2003/361/EC, Official Journal of the European Unit, (2003).
6. CMMI Product Team, CMMI for Services, Version 1.2, Carnegie Mellon University CMU/SEI-2009-TR-001, ESC-TR-2009-001.
7. Office of Government Commerce, Service Strategy, TSO, (2007).
8. Office of Government Commerce, Service Design, TSO, (2007).
9. Office of Government Commerce, Service Transition, TSO, (2007).
10. Office of Government Commerce, Service Operation, TSO, (2007).
11. Office of Government Commerce, Continual Service Improvement, TSO, (2007).
12. Office of Government Commerce, Service Delivery, TSO (2004)

Author CVs

Filipe Albero

He has attained a BSc in Computer Science in Brazil at the Pontifícia Universidade Católica do RS (PUCRS). The nightly lectures of his undergraduate course allowed him to work during the day for major multinational companies such as Hewlett-Packard and Telefónica where he exercised his skills as a software developer and tech leader. He has further enhanced his technical abilities attaining a number of certifications, namely SCEA (Sun Certified Enterprise Architect), OCUP (OMG Certified UML Professional), SCWCD (Sun Certified Web Component Developer) and SCJP (Sun Certified Java Programmer). Currently Filipe Albero Pomar is finishing his graduate studies of MSc in Software Engineering at the Universidad Politécnica de Madrid. Filipe can be reached by e-mail at the following address filipe.albero.pomar@alumnos.upm.es.

June Amillo

Dr. Amillo received a Doctorate Degree in Electrical Engineering from Polytechnic University of Madrid (UPM) and a Ph.D. in Mathematics from Virginia Tech. She returned to UPM as an Assistant Professor in 1981. Currently, she is a Professor in the Computer Science Department at UPM where she teaches Mathematics, Software Engineering Economics and Computational Finance.

Her research interests are primarily in the analysis of embedded options in software projects and in the use of innovation and technology in Computer Science Education. In addition to teaching and research she has been consulting on Project Management for over ten years.

Magdalena Arcilla

BSc. in Computer Science, Facultad de Informática at the Universidad Politécnica de Madrid. She Advanced Studies Diploma in the Languages & Computer Science Department of the UNED (Spanish Open University) She is assistant professor in the Languages & Computer Science Department of the UNED (Spanish Open University). She is teaching in the area of Software Engineering. She has also worked in the Computer Science Department of the Carlos III University of Madrid and she has 8 years of experience as developer software for CustomWare Company. She is author of several international papers related to the software engineering area. She is author of a book in the software engineering topics.

Jose Antonio Calvo-Manzano

He has a Ph.D. in Computer Science. He is assistant professor in the Faculty of Computer Sciences at the Universidad Politécnica de Madrid. He is teaching in the area of Software Engineering. He is author of several international papers related to the software engineering area. He is author of books in the software process improvement and software engineering topics. He has collaborated in several European projects oriented to software risk management, software process improvement in requirements management. He has participated in several projects with enterprises related to manufacturing process improvement and also software process improvement. He is a member of the team that has translated CMMI-DEV v1.2 to Spanish.

CONEVTO: Contract Evaluation Tool for Software and Services Acquisition Organizations*

Calvo-Manzano Jose A.[1], *Cuevas Gonzalo*[1], *Mejia Jezreel*[1], *Muñoz Mirna*[1], *Rocha Álvaro*[2], *San Feliu Tomás*[1]
[1]*Faculty of Computer Sciences, Technical University of Madrid*
{joseantonio.calvomanzano, gonzalo.cuevas, ma.munoz, tomas.sanfeliu}@upm.es, jmejia@mpsei.fi.upm.es
[1]*Faculty of Computer Sciences, Technical University of Madrid*
[2]*Faculty of Science and Technology, University Fernando Pessoa, Porto*
amrocha@ufp.edu.pt

*****Authors listed in alphabetical order.

Abstract

Contracts play an important role during the period of time that the outsourcing relationship is in effect. If an organization decides to acquire software and services products, the contract is a fundamental mechanism to ensure that expectations are realized. This paper describes a contract evaluation tool for Software and Services Acquisition Organizations, to achieve this, a contract model and an evaluation method through the tool have been developed. This tool allows the acquirer to know the coverage level related to the clauses of a contract in order to select or reject it. Besides, a case study is presented.

Keywords

Contracts, Contract model, Evaluation method, Contract evaluation tool, Acquisition.

1 Introduction

Outsourcing as a concept was accepted in the 1980s and is still used today to describe “a contractual relationship with a specialized outside service provider for work traditionally done in-house” [7]. Since then the Information Technology (IT) services outsourcing market has grown rapidly every year [4, 10, 14, 15, 20, 21, 28].

However, while the outsourcing is experiencing a considerable growth, the number of reported cases of failure is also increasing [14]. According to a study from the Software Engineering Institute (SEI) [25], 20 to 25 percent of IT acquisition projects fail within two years and 50 percent fail within five years.

Mismanagement, inability to articulate customer needs, poor requirements definition, inadequate provider selection and contracting processes, uncontrolled requirements changes and important gaps in the contracts are some of the factors that contribute to project failure.

The majority of project failures could be avoided if the acquirer learns how to prepare or evaluate properly the contracts [25].

Contracts are [22] “a framework which almost never accurately indicates real working relations, but which affords: 1) a rough indication around which such relations vary, 2) an occasional guide in cases of doubt, and 3) a norm of ultimate appeal when the relations cease in fact to work”. Moreover, a contract is considered to be the only means to guarantee the expected achievement, and also the primary means to explain the acquirer to provider relationship [12, 16].

Usually the product or service to be exchanged is specified in a contract, in a way that the acquirer and the provider know what they can expect and what is expected of them, the disagree resolution mechanisms, the rules of the engagement, the financial exchanges and the change management procedures [13,18].

An important principle for IT outsourcing was formulated by Beulen and Ribers [3]: “If a company decides to outsource, the contract is the only mechanism to ensure that expectations are achieved. It constitutes the foundation for transferring responsibility and includes the agreements that form the basis for executing the IT service”.

The purpose of this paper is to present a contract evaluation tool (CONEVTO) that allows the acquirer to know the coverage level related to the clauses in order to select or reject it.

This paper is organized as follows. Section 2 shows a brief description of the framework tool, this is the contract model (structure, categories and its main clauses and the evaluation method). Section 3 describes the tool for evaluating contracts called CONEVTO. Section 4 addresses a case study. And finally, Section 5 covers the conclusions.

2 Framework Tool

A contract model for software and services acquisition and a method for evaluating contracts in accordance to the proposed model have been established. A tool to automate the method has been developed.

2.1 Contract Model

This model establishes the main components and clauses to be included in a contract grouped in 7 categories. Each category contains the clauses which are related to the same subject (see Figure 1).

A clause is a set of components that are related among them. A component is a basic activity that must be included within a contract. To establish the contract model, the following steps were developed:

- **Execute a systematic review:** The systematic reviews apply an explicit and rigorous method to identify, critically appraise, and then synthesize relevant studies in the published research, using quantitative methods to assess work fields from different studies [13].

  The search for issues related to outsourcing contracts was based: 1) on the work of Biochini et al., and Kitchenham et al., [4, 19] that proposed a protocol for systematic review, 2) on the guidelines proposed by Biochini et al., and Goo et al., [4, 12] and 3) on the forms of extracting information from software engineering papers developed by Jens et al., [15] and other similar systematic reviews. As a result of the systematic review, 31 primary studies were found. These studies refer or make important considerations about Outsourcing Contracts, but not all of them refer to a definition of the structure or clauses in an Outsourcing Contract.
- **Discover the similarity among clauses:** The similarity is the correspondence among the clauses or information provided by the authors into the 31 primary studies. The similarity allows us to establish what clauses and information within the clauses should be considered in the definition of an acquisition contract. 20 clauses were obtained.
- **Identify the Categories:** Further analysis was conducted to gain an in-depth understanding of the twenty obtained clauses. This analysis allowed us to see that some clauses are related to the same subject and it was possible to group them into 7 categories.

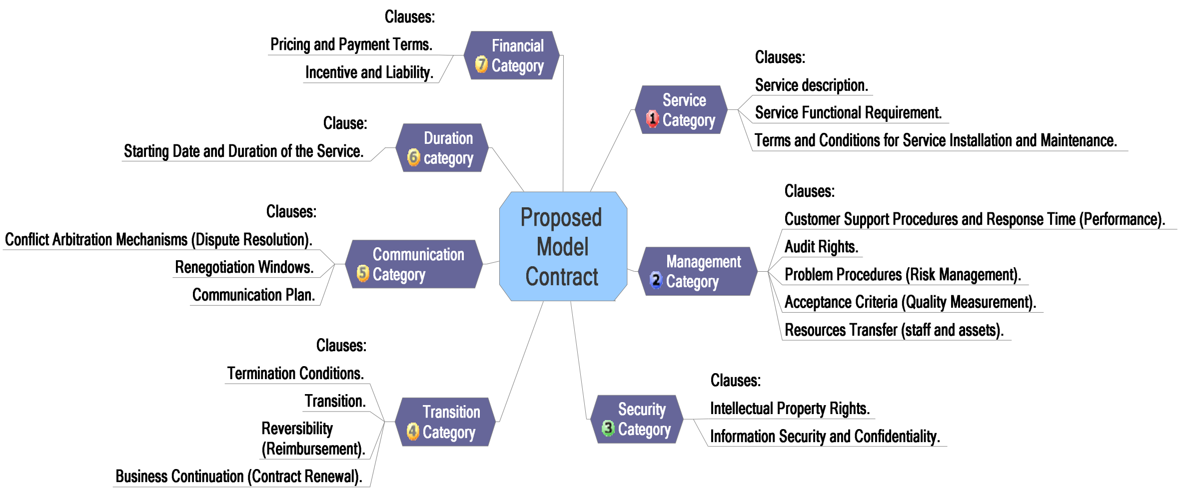


Figure 1. The main clauses and categories in proposed contract model

The legal regulations were not considered in establishing these categories because they are different in each country and proposed sector.

Figure 2 shows the contract model, structured into seven categories (Figure 1), and each category has several clauses which contain a number of components. The components are the elementary information found in the clauses.

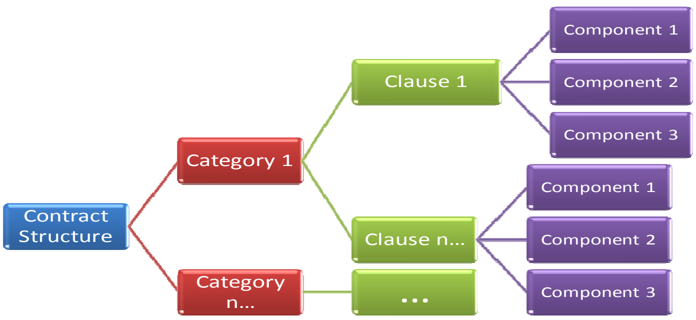


Figure 2. The structure of the contract

2.2 Evaluation Method

This method seeks to obtain the coverage level of the clauses and categories of a contract. Two evaluation criteria have been defined to obtain the coverage level:

- Evaluation Criterion 1 based on the contract model.
- Evaluation Criteria 2 based on the business goals.

2.2.1 Evaluation Criterion 1 based on the contract model

This evaluation criterion seeks to calculate the coverage level of each clause, category and contract based on the sum of the percentages achieved by each component in the proposed model contract. To achieve this, the following steps are established.

2.2.1.1 Calculate the Percentage

The same weight is assigned to all the categories, the clauses in these categories and components within these clauses to obtain the percentage value for each component, clause and category in the proposed contract model. For the overall contract, 100% is divided by the total number of categories, and this is the weight assigned to each. Then, 100% is divided by the total of clauses in the category, and this is the weight assigned to each clause. Finally, 100% is divided by the total components in the clause, and this is the weight assigned to each component.

Formula 1:

Porcentaje $X_i = 100/nX_i$

Where nX_i*: is the number of components, clauses or categories of the element, component, clause or category* X_i *in a contract*

2.2.1.2 Identification the total percentage

Once the percentage of each component, clause and category is assigned, the total percentage by each clause, category and contract is calculated. Formula 2 is used, which calculate the percentage value of each component covered and the value obtained is multiplied by the percentage assigned in formula 1 to obtain the clause coverage. Formula 3 is used, which calculate the percentage value of each clause covered and the value obtained is multiplied by the percentage assigned in formula 1 to obtain the category and finally, the formula 4 is used to obtain the contract coverage. By adding the results obtained and multiplying each clause coverage by its weight corresponding to the clauses of that category, we obtain the category coverage. For the categories, we obtain the coverage of the contract in a similar manner.

Formula 2:

$$X = \%\ Coverage\ by\ Clause = \left(\sum_{i=component}^{ComponentsbyClause} i = yes\right)\left(Clause = Fomula1\right)$$

Formula 3:

$$Y = \%\ Coverage\ by\ Category = \left(\sum_{i=clause}^{ClausesbyCategory} X\right)\left(Category = Fomula\,1\right)$$

Formula 4:

$$Z = \%\ Coverage\ by\ contract = \sum_{i=category}^{CategoriesbyContract} Y$$

2.2.2 Evaluation Criteria 2 based on the business goals

This method seeks to identify the coverage level of the clause, category and contract based on the weight given to each contract model component, clause and category according to the business goals. In other words, the acquirer will assign the percentage that reflects the degree of importance of each component, clause and category.

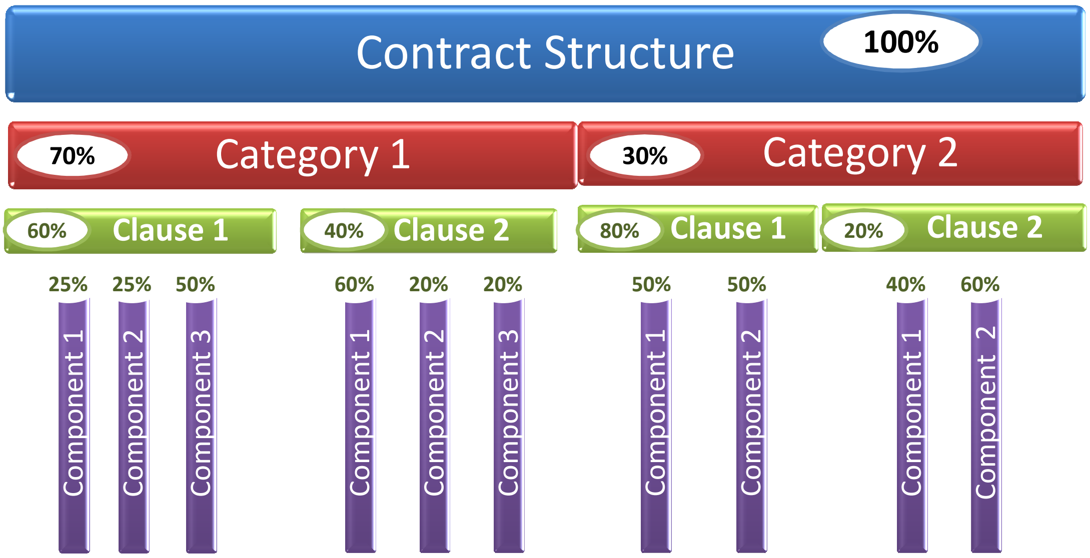


Figure 3. Assignment of percentage based on business goals

As Figure 3 shows, the percentage established in each category represents 100% distributed among the clauses. As in the category, the percentage established in each clause represents 100% distributed among the components in each clause.

2.2.3 Coverage Criteria

To evaluate the obtained values by using formulas 1 and 2, the criteria were created to define the type of coverage. If the value is equal to 100, the coverage (clause, category or contract) is considered complete. If the value is greater than 75 and less than 100, the coverage (clause, category or contract) is considered large and so on, as Figure 4 shows.

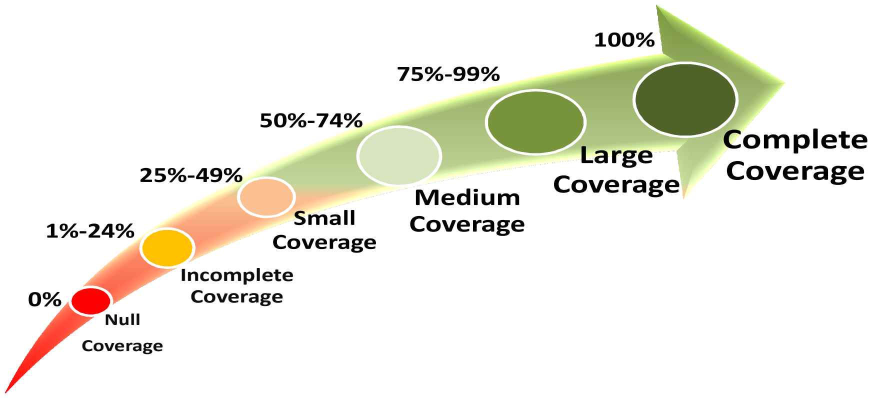


Figure 4. Coverage Criteria

3 Contract Evaluation Tool (CONEVTO)

This section describes the Contract Evaluation Tool (CONEVTO) which allows recording and getting automatically the contract coverage level. This tool has been developed taking into account the previous contract model and evaluation method. This tool has been developed in a Microsoft excel sheet.

The contract is analyzed manually to check if it contains the components defined in the contract model. If a component is found, it is highlighted and a sticky note is allocated for writing the component name (see Figure 5).

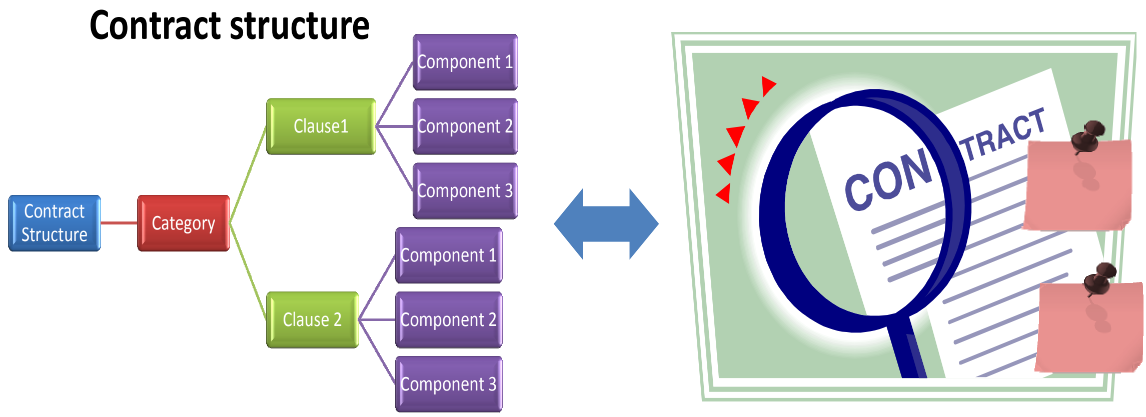


Figure 5. Example of Contract Analysis

Once components are manually detected within the contract content, the following steps in the tool are performed: information recording, percentage assigning and finally getting the results.

3.1 Information recording

In order to evaluate contracts, the contract elements (categories, clauses and components) are organized in a table according to the contract structure defined in section 2.1.

Table 1. Organization of Contract elements

Category	Clause	Component	Contract 1 Found Component
Services	Services Description	Functional requirements	YES
		Schedule	YES
		Roles	YES
		Assignments roles	YES
		Technical information on functionality	YES
		Technical information on capacity	NO
		Technical information on quality	NO
		Technical information on the interfaces	YES
		Capacity Criteria.	NO
		Quality Criteria	NO

As Table 1 shows, in order to record the components that were found in the contract content, the "Found Component" column is selected (choosing YES/NO). In this way, the clause, category or overall contract coverage level percentage based on the evaluation criterion 1 (*based on the model contract*) is obtained. Besides, to carry out the evaluation criterion 2 *(based on business objectives)* these recorded components are used.

3.2 Percentage assingnig

In order to perform the evaluation criterion 2 (*based on business objectives*) the percentage (weight) that reflects the importance degree for each category, clause and component is established (see Table 2).

Table 2. Organization of Contract elements

Service Category	Clause	Component	Contract 1
Service Category			10,00%
	Clause 1		40,00%
		Component 1	20,00%
		Component 2	5,00%
		Component 3	10,00%
		Component 4	5,00%
		Component 5	10,00%
		Component 6	20,00%
		Component 7	10,00%
		Component 8	5,00%
		Component 9	10,00%
		Component 10	5,00%
			100,00%
	Clause 2		30,00%
		Component 1	30,00%
		Component 2	30,00%
		Component 3	10,00%
		Component 4	20,00%
		Component 5	10,00%
			100,00%
	Clause 3		30,00%
		Component 1	50,00%
		Component 2	50,00%
			100,00%
			100,00%

Table 2 shows the options of the Excel sheet to assign the weight given for each model category, clause and components.

3.3 Getting results

Once the information is recorded as shown in Tables 1 and 2 is recorded, the tool shows the coverage level results of the evaluation criteria 1 and 2 through charts. According to these criteria, the charts show the percentages each clause, category and overall contract achieved (see Figure 6).

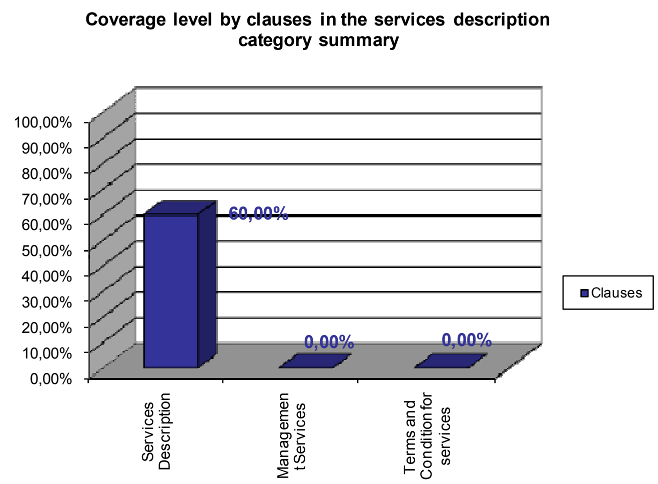


Figure 6. Example of the coverage level by clauses

4 Case study

In order to confirm the feasibility of the contract evaluation tool four contracts were evaluated.

The outsourcing contracts analyzed are from spanish companies related to the IT area. The company related to contract 1 is a company with over 20 years of experience in the market for computers and telecommunications. Their commitment is to provide their acquirers a comprehensive and timely solution and support of a real added value.

The company related to contract 3, is a small company that develop software projects (applications to web, mobiles and Windows systems) and provide assistance support (process and product audit, business analysis and requirements specification).

The company related to contracts 2 and 4 is a multinational consulting firm that offers its acquirers comprehensive business solutions covering all aspects of the value chain, from business strategy to systems implementation. They are active in the sectors of Banking, Healthcare, Industry, Insurance, Media, Public Sector, Telecom and Utilities.

The contracts are analyzed manually to check if they included the components for each clause then the found components are recorded in the contract evaluation tool described in section 4.

Figure 7 shows the overall coverage of the contracts.

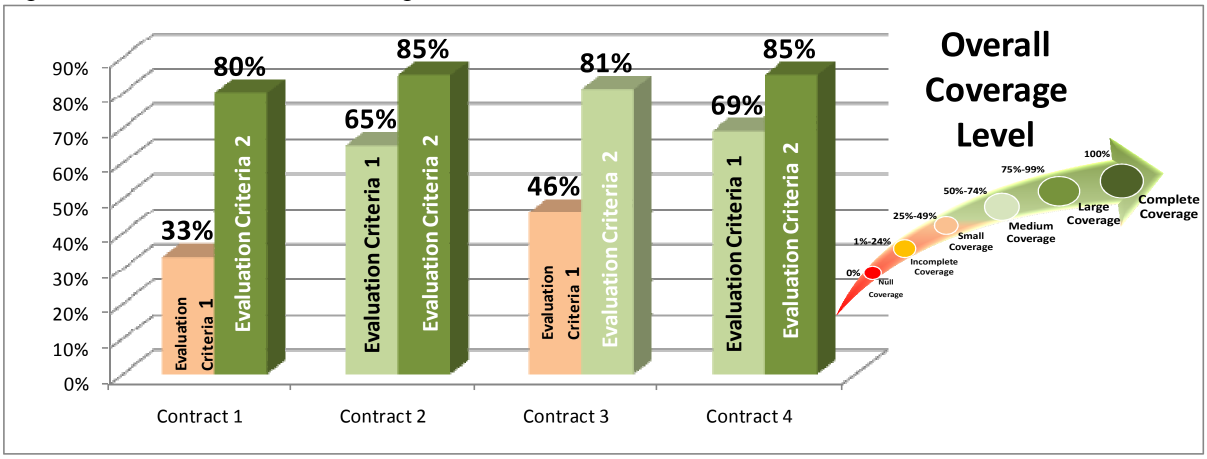


Figure 7. Overall coverage by each contract

On the one hand, according to the evaluation criterion 1, the Contracts 1 and 3 has small coverage, with 33% and 46%, respectively of overall coverage level. Moreover they lack information in most of the clauses defined in the proposed contract model. Contracts 2 and 4 have a medium coverage level (65% and 69%, respectively). Contracts 2 and 4 have higher coverage level in the 4 categories (services, management, financial and duration) than the contracts 1 and 3. They can be considered well defined according to the proposed contract model.

On the other hand, according to the evaluation criterion 2 (based on business goals), the Contracts 1, 2, 3 and 4 have a large coverage (80%, 85%, 81% and 85% respectively).

The results obtained with the evaluation criterion 1 for the contracts 1 and 3 confirmed: a) poor clauses definition, b) future problems between the acquirer and provider related to the contract 1 which was confirmed later by a lawsuit between them.

The results obtained with the evaluation criterion 2 indicated that the most of the business goals of the acquirer organization are covered.

The results obtained with the evaluation criteria 1 and 2 in the contracts 2 and 4 indicated a large coverage. However, the acquirer should make a contract review to include clauses in order to improve the overall contract, which was confirmed later in the case of the contract 2 having a renegotiation with the provider.

5 Conclusions

CONEVTO allows acquirers to evaluate the contract coverage level in order to select the best contract for a future acquisition. With the established evaluation criteria in the evaluation method, it is possible to determine the coverage in terms of the contract model and the weighting given for each contract model component. Moreover, the contract model provides a guide, in order to establish which clauses must be included in the contract proposal submitted to potential acquirers.

The evaluation method, the contract model, as well as the contract evaluation tool are confirmed by the case study in section 4. This case study is an evidence of the adequacy of the CONEVTO.

Acknowledges. This work is sponsored by everis Foundation, and Technical University of Madrid through the Research Group of Software Process Improvement for Spain and Latin American Region.

Literature

1. Aubert, B., Houde, J. F. et al. Category of IT outsourcing contracts. System Sciences, 2003. Proceedings of the 36th Annual Hawaii International Conference. (2003).
2. Barua, A., Susarla, A. et al. "Making the most out of an ASP relationship." IT Professional 3(6): 63-67. (2001).
3. Beulen, E. and Ribbers P. (2003). IT outsourcing contracts: practical implications of the incomplete contract theory. System Sciences, 2003. Proceedings of the 36th Annual Hawaii International Conference. (2003).
4. Biolchini J., Mian P., Natali A., and Travassos G., "Systematic Review in Software Engineering," COPPE/UFRJ, Rio de Janeiro May. (2005).
5. Bryson, K. M. and Sullivan W. E. Designing effective incentive-oriented outsourcing contracts for ERP systems. System Sciences, 2002. HICSS. Proceedings of the 35th Annual Hawaii International Conference. (2002).
6. Buco, M., Chang, R. et al. A Generic SLA Semantic Model for the Execution Management of e-Business Outsourcing Contracts. E-Commerce and Web Technologies: 249-257. (2002)
7. Corbet and Associates, "Outsourcing's Next Wave," (2002).
8. Chin, W. and Goles T. "Information systems outsourcing relationship factors: detailed conceptualization and initial evidence." SIGMIS Database 36(4): 47-67. (2005).
9. Craze D., Mendonca M., Basili V., Shull F., and Jino M., "Extracting Information from Experimental Software Engineering Papers," in Chilean Society of Computer Science, 2007. SCCC '07. XXVI International Conference of the, pp. 105-114, (2007).
10. Gellings, C. "Outsourcing Relationships: The Contract as IT Governance Tool," in System Sciences. HICSS. 40th Annual Hawaii International Conference on, 2007, pp. 236c-236c. (2007).
11. Giunipero, L. C., Harris, A. et al. "Impact of organizational and contract flexibility on outsourcing contracts." Industrial Marketing Management 27(5): 373-384. (1998).
12. Goo, J. and Nam K. Contract as a Source of Trust--Commitment in Successful IT Outsourcing Relationship: An Empirical Study. System Sciences, 2007. HICSS 2007. 40th Annual Hawaii International Conference. (2007).
13. Grefen, P., Koetsier, M. et al. Contracts for Cross-Organizational Workflow Management. Electronic Commerce and Web Technologies: 110-121. (2000).
14. Huai, J. "An Incentive Model of IS Outsourcing Contract," in Wireless Communications, Networking and Mobile Computing, 2007. WiCom 2007. International Conference on, pp. 6588-6592. (2007).

15. Jens, D., Tim, G., Rudy, H., and Bandula, J. "Information systems outsourcing: a survey and analysis of the literature" SIGMIS Database, vol. 35, pp. 6-102, (2004).
16. Junchao, X. and Qing W. Contract-driven cooperation development process for software outsourcing. Computer and Information Technology, CIT. The Fifth International Conference. (2005).
17. Kern, T. The Gestalt of an information technology outsourcing relationship: an exploratory analysis. Proceedings of the eighteenth international conference on Information systems. Atlanta, Georgia, United States, Association for Information Systems. (1997).
18. Kern, T. and Willcocks L. "Exploring information technology outsourcing relationships: theory and practice." Journal of Strategic Information Systems 9(4): 321-350. (2000)
19. Kitchenham, B. A., Pfleeger S. L., et al, "Preliminary Guidelines for Emprical Research in Software Enginerring," National Research Council of Canada, (2001).
20. Lee, M. "IT outsourcing contracts: Practical issues for management." Industrial Management & Data Systems, vol 96, pp 15. (1996).
21. Liston, P., Byrne, J., Byrne, P., and Heavey, C. "Contract costing in outsourcing enterprises: Exploring the benefits of discrete-event simulation," International Journal of Production Economics, vol. 110, pp. 97-114, (2007).
22. Nellore, R. "Validating specifications: a contract-based approach." Engineering Management, IEEE Transactions on 48(4): 491-504. (2001).
23. Pino F., Garcia F., and Piattini M., "Software process improvement in small and medium software enterprises: a systematic review," Software Quality Control, vol. 16, pp. 237-261, (2008).
24. Schmidt, H. Service Contracts Based on Workflow Modeling. Services Management in Intelligent Networks: 132-145. (2000).
25. S. E. Institute, CMMI for Acquisition, Version 1.2, (2007).
26. Shein, E. "Railroad wields penalties to keep its service contract on track. (Southern Pacific Transportation Co)," PC Week, vol. v12, pp.14(1), (1995).
27. Stevens K. R., "Systematic reviews: the heart of evidence-based practice," AACN Clinical Issues: Advanced Practice in Acute & Critical Care, vol. 12, pp. 529-538, (2001).
28. Zhang P., Zeng Z.-x., and Huang C.-p., "Study on Critical Success Factors for IT Outsourcing Lifecycle," in Wireless Communications, Networking and Mobile Computing, 2007. WiCom 2007, pp. 4379-4382, (2007).
29. Calvo-Manzano, J. Cuevas, G. et al.. Outsourcing Contracts: A Systematic Review. In International Book Series “Information Science & Computing”, Number 11. Bulgaria and International Journal "Information Technologies and Knowledge" Volume 3. (2009).
30. Calvo-Manzano, J. Cuevas, G. et al. Evaluación de Contratos de Adquisición de Productos y Servicios de Software en Outsourcing“.

Author CVs

Calvo-Manzano Jose A.

He is PhD in Computer Science. He is assistant professor in the Computer Science School at the Polytechnic University of Madrid. He is teaching in the area of Software Engineering, specifically in the domain of software process management and improvement. He has participated in more than 20 research projects (European and Spanish Public Administration). He is author of more than 50 international papers. He is author of books related to software process improvement and software engineering topics also. He is a member of SOMEPRO research group, where his main research field is process improvement, focusing in the topics of requirements management, project management, process deployment, and solicitation and supplier agreement development.

Cuevas Gonzalo

He received an Engineering degree in Telecommunications in 1965 and a PhD in Telecommunications in 1974. He also received an MS in Computer Science from the Polytechnic University of Madrid in 1972. He has been vice dean of the Computer Science faculty at the Polytechnic University of Madrid, where he is a full professor since 1970. He worked for Iberia Airlines as Software Development Director, supervising over 200 technicians, Data Processing Centre Director, Data Transmission Software Development Director, and the person in charge of strategic planning and software process improvement. He led European projects on software best practices from 1991 to 1995. His main research field is software engineering, including both technology (methods, techniques, and formalisms) and management. His current research interest is process models and methods, and transition packages. He is member of ACM, senior member of IEEE, member of the Telecommunication Engineering Association and member of the Computer Sciences Association.

Mejia Jezreel

He is a Master in Computer Science (2003-2005) and Computer Science Engineer (1996-2000) from the Institute Technological of Orizaba University of México. He is actually Software Engineering PhD student from the Computer Science faculty at the Polytechnic University of Madrid. His research field is Software Engineering. His current research interest consists on outsourcing process improvement. He has published several technical papers on acquisition process improvement. He is colaboring with an everis Foundation research project.

Muñoz Mirna

She is a Master in Computer Science (2002-2004) and Computer Science Engineer (1996-2000) from the Institute Technological of Orizaba University of México. She is actually Software Engineering PhD student from the Computer Science faculty at the Polytechnic University of Madrid. Her research field is Software Engineering. Her current research interest consists of the development of a methodology whose goal is to incorporate process improvement focusing on diminishing change resistance. She has published several technical papers on project management and process improvement. She is colaboring with an everis consultants research project.

Rocha Álvaro

He has a PhD in Information Systems and Technologies, specialization in Engineering and Management of Information Systems, from University of Minho, Portugal. He is Associate Professor at University Fernando Pessoa – Porto. He is teaching subjects about Software Engineering, Management Information Systems and Health Information Systems. He is co-founder and member of GIMED (Medical Informatics Research Group) at University Fernando Pessoa and co-founder and President of AISTI (Iberian Association for Information Systems and Technologies).

San Feliu Tomás

He is PhD in Computer Science. He has been working 15 years to Software Engineering field as programmer, associate and assistant professor and consultant. He has been involved in CMM appraisals in Spain and Latin America. As Software Engineering assistant professor at the Polytechnic University of Madrid, his main research field is software process and software process improvement. His current research interest includes project management and risk management. He has published several books on process improvement and he has published technical papers in software engineering and software process improvement.

NQA2 - Quality Management in Role Based Teamwork Environments

Mihail Milev[1], Damjan Ekert[1], Dr. Richard Messnarz[1]
[1]ISCN GesmbH, Austria
mmilev@iscn.com, dekert@iscn.com, rmess@iscn.com

Abstract

In the last three years, the European Commission was giving around €60 billion for sustainable growth in form of European Projects. These projects are carried out by groups of organisations with common beliefs, goals and intentions, which necessitate good teamwork approach and structure.

Still up to now only about 10% of the projects funded are still visible on the market some years after they finished. Good teamwork, high quality of results and a joint exploitation strategy are a must to succeed long term on the market.

International standards like ISO 15504 provide a framework of which processes, practices and roles you must consider when implementing projects. R.M. Belbin discussed role based teamworking on an international scale already in 1993 and his work has meanwhile been implemented in a number of role based team-working systems.

In 1998 a first NQA (Network Quality Assurance) was developed together with a knowledge system provider Hyperwave. This system was used in a field study 2001 – 2003 in an EU project TEAMWORK to analysis typical behaviour of innovation teams, involving above 50 organisations and above 40 distributed projects.

Meanwhile above 30 EU consortia used this approach. In 2007 it was decided to move this role and team-working based quality assurance to an open source strategy which can serve hundreds of distributed EU initiatives.

In this paper we show the results of this recent initiative.

Keywords

Role definitions, team-working, quality assurance in distributed teams

1 Introduction

To be able to compete on the global markets companies must constantly evolve [4], [6]. Quite often it is impossible to achieve placed goals due to overcrowded marketplaces and the only possible solution in such cases is building stable, reliable and effective partnerships. Two of the biggest benefits of such commercial-collaborations are the significantly reduced risk – partners may “absorb” critical situations, and effort sharing, which will result in discovering much larger and income-bringing marketplaces [2], 3], [7].

A lot of problems arise when talking about bigger markets, shared responsibilities and distributed collaboration [8], [9]. To reach full effectiveness such corporative structures need stable coordination of the different partners and control over communication, shared knowledge and document flow. In our modern times is impossible to think, that direct supervision, heavily centralized structure and traditional – old-fashioned, ways of communication like simple telephone conversations, will be enough. A modern approach would be distributing rights and tasks over decentralized teams, usage of online conferencing methods and systems for controlling the knowledge [5].

The field study in the EU Teamwork project and recent experiences 3], [4]with supporting Eu initiatives with more than 20 partners per project (EU Cert) show that the success of a project directly relates to the ability of knowledge control which helps to integrate the sustainable products at the end of the project. Partners in such collaborations need an effective way for creation, management and archiving of documents, project configuration possibilities, network support and control over information flow, reliable security mechanisms, which assure the privacy of the documents.

2 Concepts of NQA

Before we can define what NQA is, we must carefully observe, what a partnership needs in order to maintain stability, control and prosperity. Online teamwork software has four main tasks (fig.1):

1. construct stable infrastructure allowing mechanisms for organizations to collaborate on certain projects, based on the server concept;
2. develop processes and quality criterias based and working on the infrastructure, which help to control the work, performance, quality and development of the distributed teams;
3. support the communication techniques, which build the distributed work;
4. build feedback mechanisms - such techniques are needed to constantly have an overview and control over the project and the knowledge flow.

What very important in such structure is, that it leads to building knowledge. In order to keep this information as real knowledge, the partnership must differentiate what kind of software it may use. Normal information systems, although developed to fulfill some of the points stated above, store information only as a collection of data – raw mass of bits and bytes, without any particular representation or concrete structure between them. NQA on the other side, as teamwork knowledge software, works like the brain – links are being built between the different elements of the organization and throughout knowledge is created. This is not a simple document repository – it brings mass of competence, thus offering powerful mechanisms for solving already encountered, scope-specific problems.

As a simple example for the importance of knowledge instead of simple repository, we can give the NQA system itself. It has been developed over its own knowledge for the last fifteen years. The NQA project started in the mid 90s and has been field-tested in many companies and also in projects funded by the European Union. This resulted in several re-developments and further improvements. All of these were stored in the system itself, having a specific structural organization, developed by Dr.Richard Messnarz. Through this knowledge the further developments of the system can be easily based on the field-test results and overcome already solved problems.

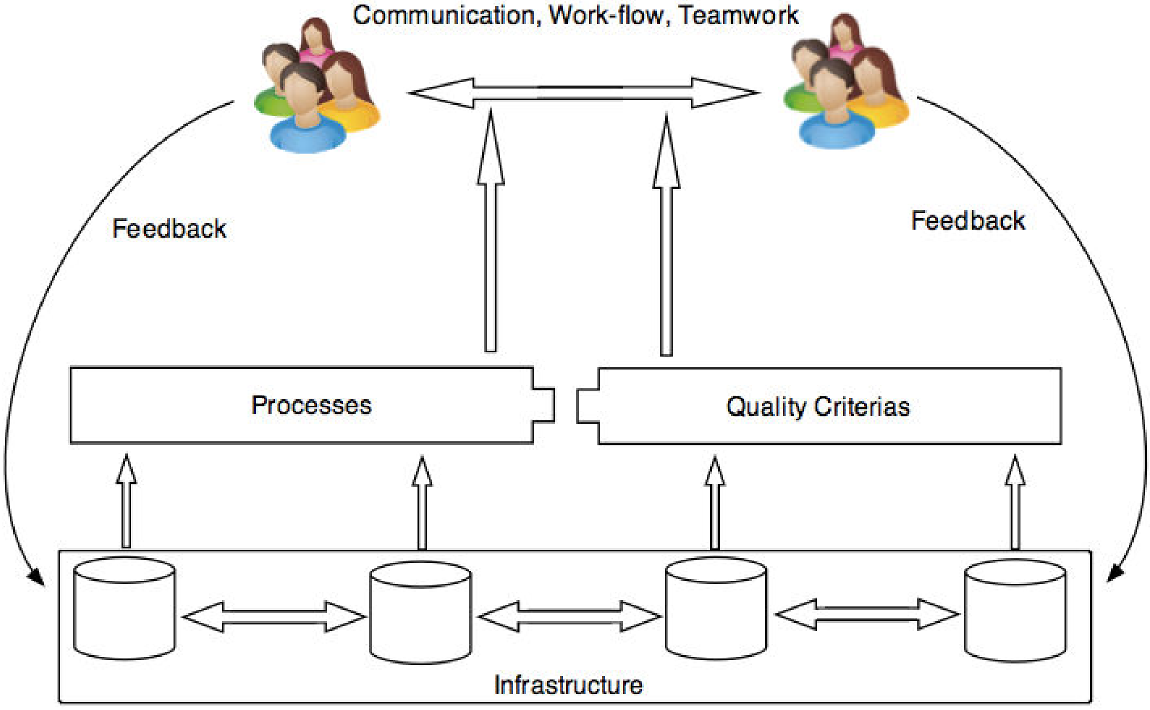


Figure 1: Basic Teamwork Software Concepts

In order to properly understand why a differentiation of knowledge from raw document data is necessary, one must understand the principle for structuring data in the NQA system.

The processes are not treated as sequence of different tasks, which output some kind of product, but they are result of integrated teamwork. The building unit of the organization are scenarios – management scenario, design scenario, etc., and each scenario is built by (fig.2):

1. roles with defined responsibilities;
2. working steps to which the roles and resources are assigned;
3. work-flow built from the working steps;
4. deliverables produced by the roles, when they do their assigned steps from the workflow.

The idea behind this concept is to concentrate on roles [1], 3], [5], [7] and the main plan behind them, rather than on the output products. This way organization can think carefully about the staff working on the process and the scenarios will produce the end product. On the other hand, the people can concentrate on their responsibilities. For them is enough to know, who they should communicate with and what should they do. Another benefit of this approach is the possibility to easily integrate new staff, because of the lower training and adaptation time. The fourth task of the teamwork software as stated above is feedback mechanisms. Since the users and communication between them, stay underneath NQA, it is very easy for managers, partners and organizations to receive feedback – for the process development, document flow and results produced.

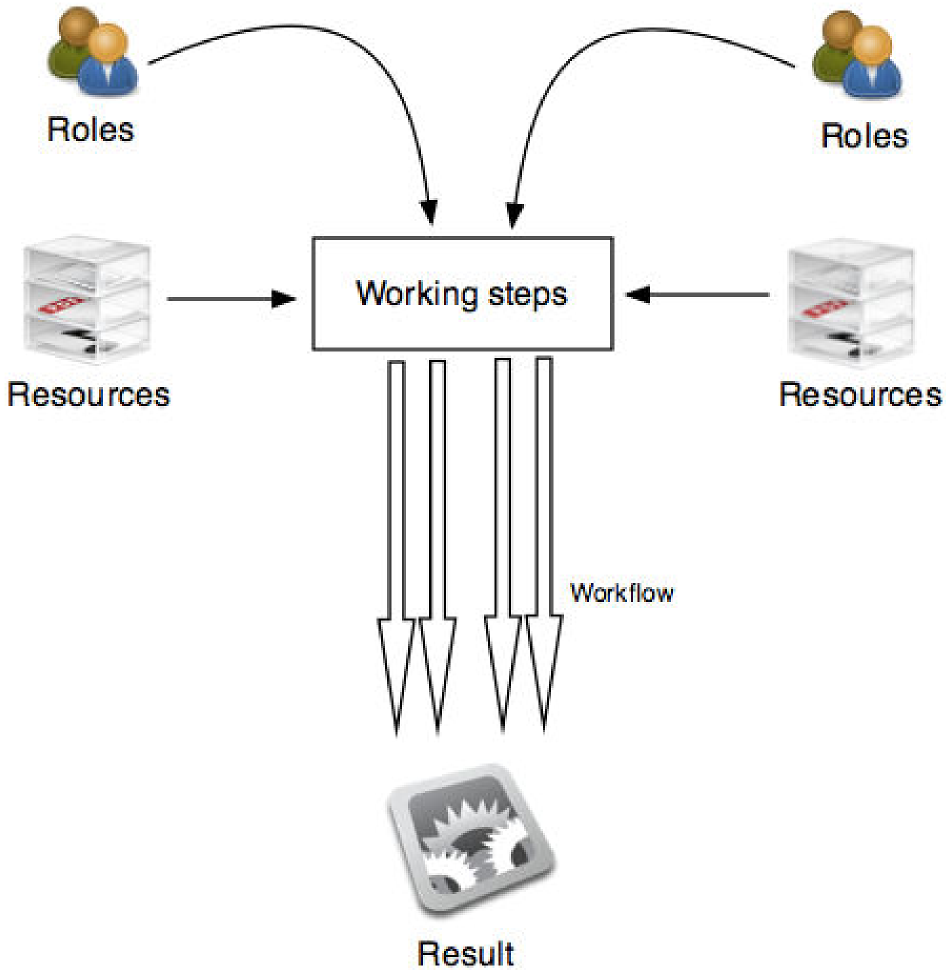


Figure 2: The NQA Concept

As a second very important concept behind the NQA system is the separation between functionality and data. [2], [3] The functionality of the system and how it impacts the workflow in a partnership is defined by a special configuration mechanism. The output of this mechanism is defined by the type of project people will be working on, needed roles, communication connections. This configuration assigns the roles into working steps and builds from them the whole working flow. Data on the other side is controlled only by users – it is the product of their work. This data is structured inside the created configuration, but since it is not connected to or dependent from it, it can be reused in other projects, thus other structures, without disturbing the workflow. More over dynamic links between documents can be created for merging different workflows or sharing information.

Very often in working situations documents must be edited, reviewed, discussed or just archived. This yields the need of control and management mechanisms – revision control [2], [3] . Documents are defined not only by their data, but also version and status. Editing a document – either uploading a new file from user's computer, or creating it directly online in the "what you see is what you get" editor, does not delete the previous document data, It is archived as an older version, but still remains accessible. Users may review the new version, continue editing it or revert back to the old one. After a document gets reviewed, its status can be changed, choosing from several status possibilities – Draft, Reviewed, Accepted, etc., and according to the project and partnership needs.

Access to the virtual office must be easy and work with it should be as intuitive as possible. The NQA teamwork software may be used either on a separated, partnership-dedicated server, or on corporate intranet server, or on a global Internet server with shared rights. It is based on web technologies and users access their projects via one of the existing web browsers.

3 Experience Pool

The NQA Teamwork system has been used by more than 30 research projects mostly co-funded by the European Commission [3], [4], [5], [7] and in various software development projects. The high demand for such a network based project and quality management systems raised from the fact that all of the EU projects included partners, distributed among several countries, collaborating together on a topic. Therefore it was crucial to use a system which was easily accessible over the internet, does not require any special client software, plug-ins or add-ons and no special technical knowledge. The system and its functionality have grown over the past years, features and approaches were stepwise introduced to the users. An Example: In the starting phase (2000) the version control functions were rejected by the users as it was too *complex* for them. They were used to save their files by adding a version numbers to the filename. After several months in the project, participants started to ask on which version of the document they can work, who changed what in the last version and where my changes considered and integrated. Additional effort was required by the users and the project management to integrate all content and establish a clean version controlling. The version controlling became one of the key elements for managing (EU) projects. Similar experiences were made with the introduction of other features like document status, discussion forums, notifications etc. In the current projects where more than 20 partners collaborate, a system like the NQA Teamwork becomes indispensable.

However, the main disadvantage with the existing system was that the Hyperwave Information Server used as a server platform for the NQA, is not an open source solution.

We have meanwhile 2007 – 2010 redesigned the system as a component of a content management system JUMLA which we is available as a free platform on the market. However, companies registering for this will be provide with an annual service agreement (annual service cost) to maintain the services / upgrades. This way we combine content management, open source and future development strategies in the market.

Currently we roll out the system to larger networked projects in the EU context.

The development of the new NQA started, with the goal to meet the specific requirements and needs of companies which accomplish European wide projects and to preserve the current functionality and GUI. The following basic requirements have been considered in the new platform:

- Access from the web browser with no additional plug-in, add-in required
- Easily manageable and user friendly
- Highly customized
- Re-usable templates and project environments to ensure a common quality
- Offer non-formal ways of communication (chat, discussion forum)
- Support version control mechanisms (status, owner, date created, date modified, modified by…), version history and document status (draft, reviewed, approved)
- Define access rights, change control and editing rights
- Notifications on events (new documents, new version of a document)
- Open source environment, no license fees
- Bidirectional linking

4 Challenges for NQA 2010

Following Moore's law, technologies have evolved a lot in the last ten years. Nowadays small nettops and mobile devices, which offer full computer-alike user interface, are all over us. The virtual office must make advantage of these devices and offer the partners the possibility to have their work in the pocket, at one hand reach. The users on the other side expect simplicity, clear and defined working paths, so they can fulfill their tasks as fast as possible.

Almost every company today has at least portfolio web page, some of them base their whole business and internal structure on second web services. Integrating new virtual office software must be without any risks for the developed infrastructure and it must also offer significant and cost-worthy new features. Integration process should be as automated as possible, should offer its functionalities right after the installation and with as few further configurations as possible. Teamwork solutions must be easy and intuitive to work with, without the need of extra, much costing staff trainings.

Today's partnership development as stated above is dependant on data flow – documents, calendars, e-mails, etc. To ensure reliability and security, every virtual office software must purpose an easy to do, secure and steady backup mechanism. A partnership based on insecure and unbackupped data is exactly as instable and unsafe as it. The use of backups allows fast, stable and reliable data regeneration in case of hardware problems, software crashes and hack attacks.

As we can conclude from the Experience Pool section, a big problem of NQA was, that it was based on closed-source, proprietary software, called Hyperwave. It offered a clear, but resource-hungry way to divide functionality and data. Using templates, objects were generated after the configuration process. These objects were then used to build a wrapper around each document, according to the user role. This process was long and hard to understand by the user and only few administrators were able to change the project configuration, if needed. Integration with other web services was also complicated, because of the proprietary platform, language, port and architecture used. Due to these facts possible firewall problems could arise – the security of the software is totally reliable on its coding and no third-party, specialized security software was able to detect and prevent hacker attacks. Since the company developing Hyperwave is not more to be seen on the market, partnerships were left with no support. The rapidly changing server market, constantly updated the software platforms and soon Hyperwave had problems to run on newer systems. Either teamwork had to be based on old, security unreliable and unstable platforms, or users had to change to new teamwork systems.

The biggest challenge for NQA 2010 is to fulfill the users' requirements and remove the above stated problems, but without loosing any of the functionalities developed until this moment.

5 Implementation

It is clear, that the software must be rewritten from scratch. Doing it this way, we can remove the dependencies from closed, proprietary code. This also allows the developers to offer a new approach to the development process and internal software structure and benefit as much as possible from modern technologies. In the last years the software developer communities have concentrated on open source coding – a revolutionary approach, allowing every developer to contribute for improvement of software. The code behind the software is freely accessible, downloadable and editable. Changes made can be sent back to the community for review and merging. NQA 2010 is also based on open-source software (OSS). It is developed as a component in the Joomla! Content Management System (fig.3). Since it is OSS, experienced developers from the whole world may contribute to it and constantly improve it. Another benefit of the OSS approach is, that there are no hidden hacks. This offers a further level of security for corporate users – knowing what happens and what could go wrong is the best level of security possible.

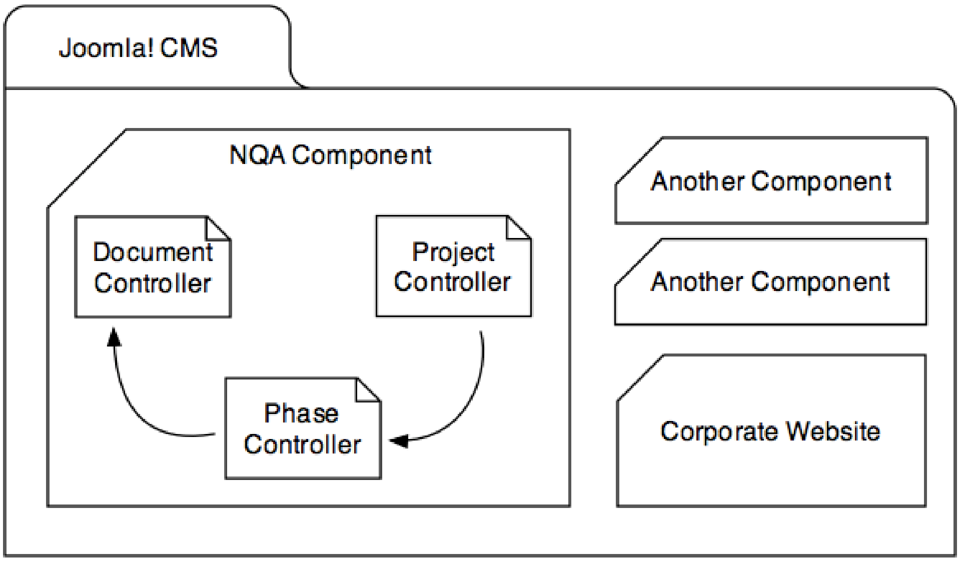


Figure 3: NQA as Joomla! CMS Component

Joomla! is written in PHP, HTML, CSS web languages, which have strict standardization, control and bug checking by W3C – non-profit consortium staying behind each web standard. Because of these standards, software written in these languages is reliable, secure and with as few as possible errors. NQA is divided into sections, which consists of different controllers. This modularized architecture assures, that an error in one working path of the teamwork solution will not affect another and cause an avalanche effect. Since this content management system is very popular, a lot of companies use it as a corporate web page infrastructure. The development of NQA as a component for it fulfills another user requirement – easy and risk-free integration of the teamwork system into existing infrastructure. From our experience with the NQA 2010 installing it in an existing working environment will not take more than an hour and role users can start their work right after the setup process has finished.

Second important point in the implementation of NQA 2010 is to remove every overwhelming and complex part of the software. Instead of compiling templates for each document instance, a much faster approach is to have dynamic templates (fig.4). With today's powerful servers and fast computational development languages, generating templates for documents "on the fly" is not a problem. What we have learned from our development experience is that such systems need only pre-defined configuration how to resolve the documents in the process tree. The templates themselves contain placeholders. On runtime, important document information and system functionalities are placed in these holders. For further speed improvements, templates are cached and pages of the same type use one and the same template. This also allows faster re-development of the system, in case needed.

Data security and integrity is a core task of every teamwork system. Backuping documents from a workflow is the key for making a system reliable and trust-worthy. NQA teamwork software relies on the MySQL database system, which is currently the most used, open source and having the biggest community database products. Furthermore is a standard packet of every popular server distribution. It offers tested, market-proved backup possibilities – replication, to-file archiving, to-package archiving, etc. According to our experience, virtual office software should not concentrate on self-backup-ing mechanisms, but rather use proven solutions, which offer wide-range of functionalities and are also developed to encrypt backups or secure them by other means.

Since NQA is a Joomla! component, which is based on web technologies, usually running on the most famous server – Apache, the software can rely also on other security mechanisms to prevent brute force, denial of service or other hack attacks. This way the teamwork platform offers full security for the partnership – secure collaboration, which could be easily integrated in existing corporative environments and allow role users to easily fulfill their all-day tasks.

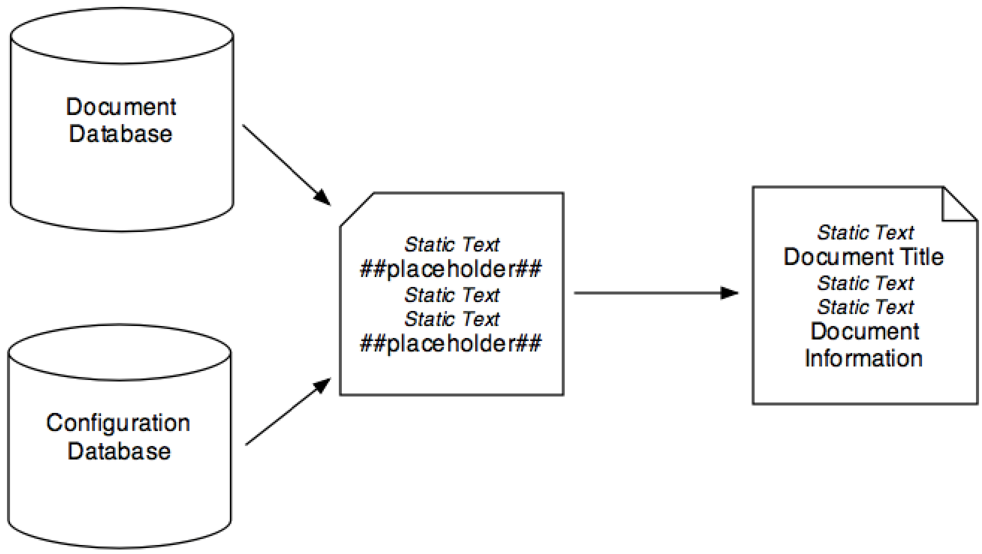


Figure 4: NQA Template Structure

6 Conclusion and Outlook

With the globalization and further development of Internet, borders not only between countries fall apart, but also such in corporate world. With each day newer opportunities arise and to challenge and explore them companies need partners. That is why the field of teamwork solutions is currently very wide and important in the informatics. Concepts behind NQA teamwork system have proven to be functional and helping to keep everything under control in collaborations, without the need of specialized hardware or software. The system is currently being integrated for the running EU consortia and distributed research teams in the EuroSPI and ISCN networks. Feedback is collected constantly.

The goal is to improve the team-working capability in distributed European developments.

With feedback from these projects future specifications and responsibilities of collaboration software can be built.

Also it is planned to structure the team models for ISO standards including ISO 15504, ISO 26262, and ISO 9001 processes.

Literature

[1] R. M. Belbin, Team Roles at Work, Elsevier Science, 1993

[2] R. Messnarz, R. Stubenrauch, M. Melcher, R. Bernhard, Network Based Quality Assurance, in: Proceedings of the 6th European Conference on Quality Assurance, 10-12 April 1999, Vienna, Austria

[3] R. Messnarz, G. Nadasi, E. O'Leary, B. Foley, Experience with Teamwork in Distributed Work Environments, in: Proceedings of the E2001 Conference, E-Work and E-commerce, Novel solutions for a global networked economy, eds. Brian Stanford Smith, Enrica Chiozza, IOS Press, Amsterdam, Berlin, Oxford, Tokyo, Wash-ington, 2001

[4] R. Messnarz, R. V. Horvat, K. Harej, E. Feuer, ORGANIC - Continuous Organisational Learning in Innovation and Companies , in: Proceedings of the E2004 Conference in Vienna, E-Work and E-commerce, eds. Brian Stanford Smith, Enrica Chiozza, IOS Press, Amsterdam, Berlin, Oxford, Tokyo, Washington, 2004

[5] R. Messnarz, G. O'Suilleabhain, R. Coughlan, From process improvement to learning organisations (p 287-294), in: Wiley Interscience Journal, SPIP Software Process Improvement in Practice Series, Volume 11 Issue 3 , Pages 213 - 335 (May/June 2006), 2006

[6] R. Messnarz, D. Ekert, Assessment-based learning systems - learning from best pro-jects (p 569-577), in: Wiley Interscience Journal, SPIP Software Process Improvement in Practice Series, Volume 12 Issue 6 , Pages 505 - 610 (November/December 2007), 2007

[7] R. Messnarz, D. Ekert, M. Reiner, G. O'Suilleabhain, Human resources based improvement strategies - the learning factor (p 355-362), in: Wiley Interscience Journal, SPIP Software Process Improvement in Practice Series, Volume 13 Issue 4 , Pages 297 - 382 (July/August 2008), Jul 2008

[8] G. Spork, U. Pichler, Establishment of a performance driven improvement programme (p 371-382), in: Wiley Interscience Journal, SPIP Software Process Improvement in Practice Series, Volume 13 Issue 4 , Pages 297 - 382 (July/August 2008), Jul 2008

[9] A. Riel, S. Tichkiewitch, R. Messnarz, Integrated Engineering Skills: Improving your System Competence Level. In: EuroSPI 2008 Industrial Proceedings, Dublin, Delta Series about Process Improvement, ISBN 978-87-7398-150-4pp. 8.9-8.20.

Author CVs

Dr Richard Messnarz

Dr. Richard Messnarz (rmess@iscn.com) is the Executive Director of ISCN LTD. He studied at the University of Technology Graz and he worked as a researcher and lecturer at this University from 1991 - 1996. In 2 European mobility projects (1993 and 1994) he was involved in the foundation of ISCN, and he became the director of ISCN in 1997. He is/has been the technical director of many European projects.
He is the editor of a book "Better Software Practice for Business Benefit", which has been published by IEEE (www.ieee.org) in 1999 (the leading research publisher in the USA). He is the chairman of the EuroSPI initiative and chair of the programme committee of the EuroSPI conference series. He is author of many publications in Wiley SPIP, Springer LNCS/CCIS, and journals.
He is a lead ISO 15504 assessor. He has worked as a consultant for many automotive firms, such as BOSCH, ZF, Continental, Magna, T-Systems, etc. He is a founding member of the INTACS (International Assessor Certification Scheme) accreditation board, a founding member and vice president of the ECQA (European Certification and Qualification Association), and he is the technical moderator of task forces to which above 20 German companies collaborate, the SOQRATES initiative (www.soqrates.de).

Dipl.-Ing. Damjan Ekert

Dipl.-Ing. Damjan Ekert studied Telematics at the University of Technology, Graz, Austria. He finished his studies with distinction. Since 2001 he works for ISCN as SW project manager and SW integrator. He is a certified e-Security Manager and European Project Manager Trainer. For the last three year he has been working as a consultant in the field of Software Process Improvement, he is member of the S2QI Initiative, certified ISO15504 Assessor, member of the European Certification and Qualification Association and vice- president of the ISECMA e.V. (International IT Security Management Board).

Mihail Milev

Mihail Milev currently studies Telematics at the Technical University in Graz. During his more than ten years long development experience, he has specialized in web and mobile development and also in software optimization for the mobile. He is professional Java, PHP and Objective-C developer and his skills cover not only traditional operating systems, but also a variety of mobile platforms like Android© and iOS©. Since October 2008 Mihail Milev works for ISCN as software developer and server administrator. In 2009 his primary target was the re-development of the NQA Teamwork platform.

ProjectIT-Enterprise: a Software Process Improvement Framework

Paula Ventura Martins[1,2], Alberto Rodrigues da Silva[2]

[1] Universidade do Algarve, Campus de Gambelas,
8005-139 Faro, Portugal
[2] INESC-ID /Instituto Superior Técnico, Rua Alves Redol, nº 9
1000-029 Lisboa, Portugal
{pventura@ualg.pt,alberto.silva@acm.org}

Abstract

Process descriptions represent high-level plans and tends to not provide information for real world software projects. Misalignments between processes and projects are caused by processes that are unrelated to daily practices or hardly mapped to project practices. We argue that software processes should emerge and evolve collaboratively within an organization. In this paper we present and discuss a collaborative tool for process authoring and project management based on the vision of agile software process improvement.

Keywords

Software Process Improvement, Process Management, Project Management

1 Introduction

Standard "Software Process Improvement" (SPI) models impose that SPI practitioners focus on specific process problems and ignore other problems more important to organizations, such as, not explaining the mechanisms of team's collaboration and how to react when facing existing problems. Commonly small and medium organizations, that have strong budget and schedule constraints, tend to reject traditional SPI initiatives such as those based on CMMI [1], ISO 9000 [2] or ISO/IEC 15504 [3], because the challenge to successfully carry out programs with these reference standards entails a substantial overhead and costs.

Although standard SPI models have been highly publicised and marketed, they are not being widely adopted. Therefore their influence in software organizations remains more at a theoretical than practical level [4]. Some surveys and studies [5,6] have emphasized that the majority of small organisations are not adopting standards such as CMMI. Another case is observed in Brazil where Brazilian software industry and universities are working cooperatively in implementing a successful SPI strategy that take into account software engineering best practices and aligned to Brazilian software organizations context [7]. We argue in our research that the emphasis in SPI should be stressed on communication, coordination, and collaboration within and among project teams in daily project activities, and consequently the effort in process improvement should be minimized and performed as natural as possible. Little attention had been paid to the effective implementation of SPI models which has resulted in limited success for many SPI programs. SPI managers want guidance on how to implement SPI activities, rather than what SPI activities do actually implement. Limited research has been carried out on exploring new approaches to effectively implement SPI programs. On this basis, we propose in this paper an agile methodology, and a complementary tool, to describe software processes based on projects experience.

This paper focus on the description of ProjectIT-Enterprise, which is a Web based tool for process and project management. Section 2 introduces the ProjectIT initiative, giving in particular an overview of the set of tools that together compose the ProjectIT workbench. Section 3 describes how ProPAM (Process and Project Alignment Methodology) and ProjectIT-Enterprise tool are combined to work together in order to support process improvement. Finally, section 4 presents the related work concerning other initiatives. Finally, section 5 concludes this work, justifying our perception that this proposal has innovative contributions for the community.

2 The ProjectIT Initiative

The Information Systems Group of INESC-ID has been involved for some time in the area of software engineering research. ProjectIT is a research initiative that reflects on this area and it has the main goal to analyze, integrate and support best practices for managing and implementing IT projects [8]. The research, reported in this paper, has been performed within the context of this ProjectIT initiative.

The initial definition of the ProjectIT initiative considered a set of guiding principles linked to information systems development processes, namely: process, project and business alignment; customers and users involvement; plan and control projects at the minimum effort; facilitate communication based in visual models; architecture-centric development; reuse at the most situations; software development based on models; and finally, agility and simplicity at all levels [9].

The underlying investigation to this initiative has been rendered and validated through different tools, in particular: ProjectIT-Enterprise and ProjectIT-Studio (see Figure 1). **ProjectIT-Enterprise** is an integrated environment to support projects management and collaborative work, being classified as a Web-based collaborative tool.On the other hand, **ProjectIT-Studio** is a CASE tool, with a rich-client desktop interface, focused on the accomplishment of high productivity activities, associated to requirements and tests specification and management, system design, automatic code generation and software development [9].

This paper addresses some of the major research issues that had been integrated into ProjectIT-Enterprise, namely: process definition, process-project alignment and project management.

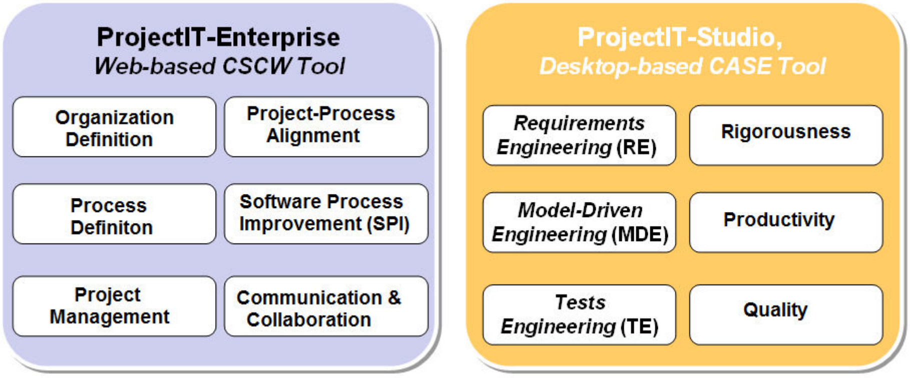


Figure 1: ProjectIT-Enterprise and ProjectIT-Studio

3 ProPAM – Process and Project Alignment Methodology

The "Process and project alignment methodology" (ProPAM) is an agile approach focused on organizations' needs for communication, coordination, and collaboration within and among project teams. This methodology helps its users to describe: (1) how processes and projects can be defined and managed; (2) how project teams acquire and use knowledge to improve their work. A key feature of ProPAM is the integration of SPI activities with software development activities. This way, we considered that project teams and projects themselves as the baseline for improvement. The following subsections present a brief overview of ProPAM's most important aspects and how ProjectIT-Enterprise supports these SPI activities. Further details can be found in [10,11].

As Figure 2 illustrates, ProPAM methodology includes activities that intend to implement, as well as improve, an organizational software process (process level). Nevertheless, these SPI activities also include interaction between project managers and process managers as key to the organizational success of SPI initiatives (project level). At project level, the methodology proposes to assist organizations in their daily effort to assess and manage problematic situations of specific projects, develop and implement solutions to help managing these problems. Project level covers project information needed to systematically support or reject many process decisions. At process level, project feedback leads to process reviews and iterative process improvement. The dynamic interplay between these two levels (project and process levels) shows the synergy between the activities performed by project roles (project manager and team members) and the activities performed by process roles (process manager and process engineers) involved in SPI. These activities are identified in Table 1.

Table 1: Process and project alignment activities

Stage	Process Level	Project Level
Stage 1: Process definition	• Ensure alignment with organization's goals • Establish a shared vision of process improvement	• Initial project meeting • Get project data and measures
Stage 2: Apply process to project	• Formalize project control and monitoring • Formalize application of new practices	• Define project plan • Action items • Change project plans • Project meetings
Stage 3: Process assessment and refinement	• Process assessment • Formalize new process version	• Get project data • Final feedback meeting

However, to manage the inherent complexity of these levels, ProPAM provides a method for formal process definition (stage 1) and improvement based on project experience (stage 2). Alignment lays the foundation for successful process definition efforts, as well as it ensures that resulting improvements are synchronized with organizational goals. Data flow synchronization between these two levels minimizes uncertainty and the amount of unused data between working groups. Consequently, the

ability to strategically relate information flows between these levels is essential to the success of these endeavours.
ProPAM is an iterative process improvement methodology organized into three stages: (1) process definition; (2) apply process to project(s) and monitoring; and (3) process assessment and refinement. The following subsections present an outline of these three stages.

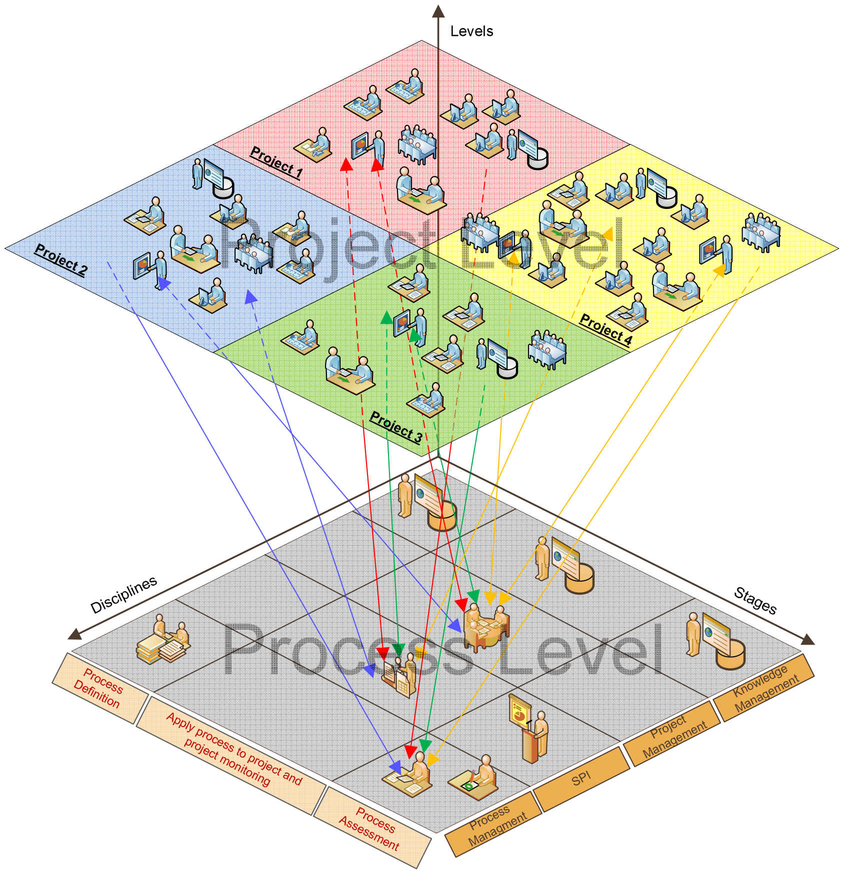


Figure 2: Overview of process and project alignment methodology (ProPAM)

3.1 Stage 1: Process Definition

The objective of this stage is to find out organization's needs regarding SPI, analyze and understand problems with senior managers, and define the first version of the software development process. The initial process specification is performed through the application of ProjectIT-Process metamodel shown in Figure 3. In this stage, the process manager finds current process practices to include in an initial process. Initial meetings with senior managers mainly covered questions about the organization's mission and goals. Also important is the information retrieved through interviews and questionnaires. The organization also delivers several documents relevant for an initial development process definition. A kick-off meeting mainly covered a description of the collaboration process and a presentation of a process draft.
The process definition stage ends when its milestone is crossed, which means develop a clear vision of the process and prepare environment to initiate an SPI program. It should be understood that crossing the milestone is not a matter of certainty. It is just an agreement about the process actually applied in projects of the organization. During this stage, the process team do the following:

- Define the scope of the SPI program with senior managers;
- Create the first process version and obtain agreement on the proposed process;
- Create a short description of all process disciplines;
- Create a more detailed description of critical activities of those disciplines;
- Define the SPI program plan;
- Prepared the environment: tools and infrastructure needed for the SPI program.

In this domain, ProjectIT-Process Metamodel shows formal process elements that can be applied to construct specific development processes. As Figure 3 illustrates, PIT-Process models different kinds of process elements and the most basic relationships between them. This model is just a conceptual description of these main perspectives. Presented diagram intends to give an overview of the proposed metamodel and its views. ProjectIT-Process Metamodel includes two complementary views: the static view and the dynamic view, where each view shows the discipline and the temporal perspectives, respectively. The **static view** shows the way how process concepts are related (i. e., their structure) and their characteristics. The **dynamic view** specifies a temporal pattern to systematically perform the most common activities.

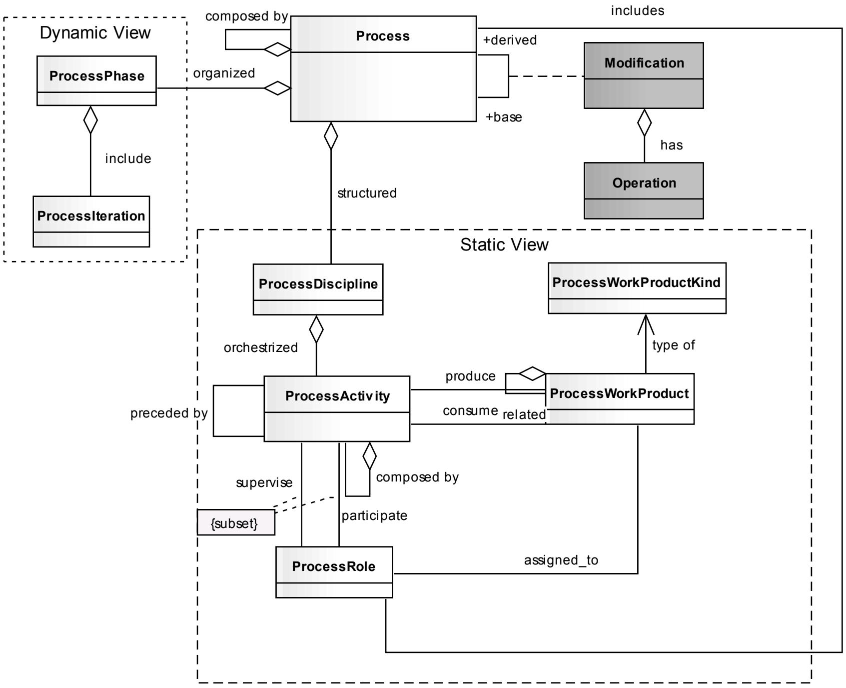


Figure 3: ProjectIT-Process Metamodel

ProjectIT-Enterprise integrates process definition to support process manager in this initial phase dedicated to analyse and identify core activities performed by project teams. It allows describing supporting disciplines (in terms of activities, work products and roles) and supporting phases (in terms of number and size of iterations). Figure 4 shows the Rational Unified Process (RUP) defined within this environment.

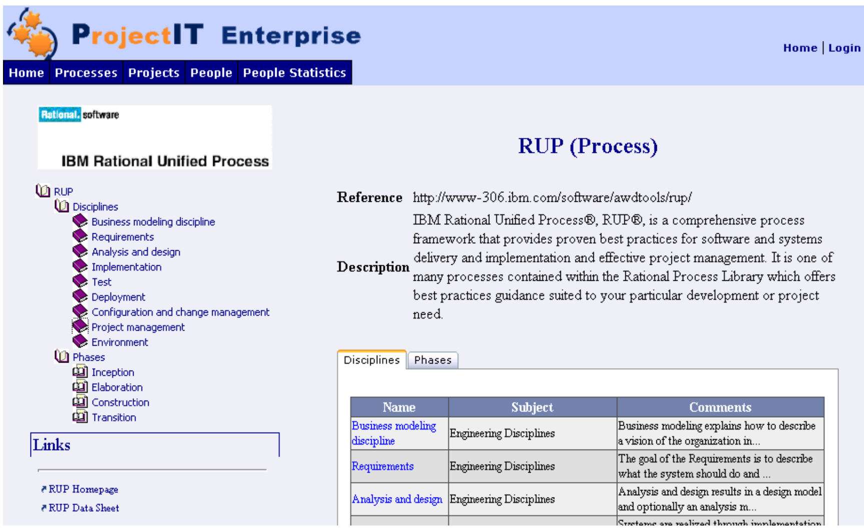


Figure 4: Defining a process (e.g., RUP)

3.2 Stage 2: Apply Process to Projects

Apply process to project(s) stage involves planning and executing the project within the base process best practices. It also provides assurance that the project is progressing according to the base process or it reveals the need to take SPI actions since activities performed by team members are different from those specified in the process. The goal of this stage is to identify and solve problems with existing procedures, propose new practices to address those problems, and observe new practices in action. During this stage, project teams and process manager will test and validate improvements against old project practices.
Several activities are performed at the process and project level. At project level, several projects will be under inspection to detect, introduce, and validate new software development practices. Then, these practices will be analyzed, at process level, as candidates for future improvements in the base process. This stage follows a pattern of cyclic iterations that allows detecting and validating new software engineering practices. At each iteration end, an SPI feedback meeting is held to review changes proposed by team members and to notify about changes that will be applied from that point forward.

In the apply process to project(s) and monitoring stage, involved roles will:

- Specify project plans based on the process model;
- Refine the SPI program plan;
- Eliminate risking software engineering practices;
- Propose new software practices to improve the process;
- Test, validate and reject proposed practices.

The most important SPI work products that are produced during this stage are: metrics data, SPI iteration reports, software engineering practices and SPI meeting presentations. The milestone that marks the end of this stage and the beginning of the Process Assessment and Refinement stage is crossed when the process manager and project teams agree that: (1) software engineering practices describe the detailed behaviour that address projects needs; (2) major problems have been solved; (3) process practices provides some useful value to the organization and these practices are stable enough to be implement a new and improved process version.

Initially, ProjectIT-Enterprise allows specification of projects based on process models: monitoring projects; eliminating risk-prone software practices; proposing new software practices; testing, validating, and rejecting proposed practices; and preparing and coordinating iterations. Considering a process that has a different set of process items (disciplines, activities, phases, work products, roles), the tool guide and shape a project using that template. A process template defines a set of activities that comprise best practice of how to achieve a certain goal.
Figure 5 presents a new project plan, generated after select a process type between several process templates. ProjectIT-Enterprise is flexible and allows the project plan to be modified by removing or adding activities, change activities order, planned man-hours, duration, roles, work products, etc.

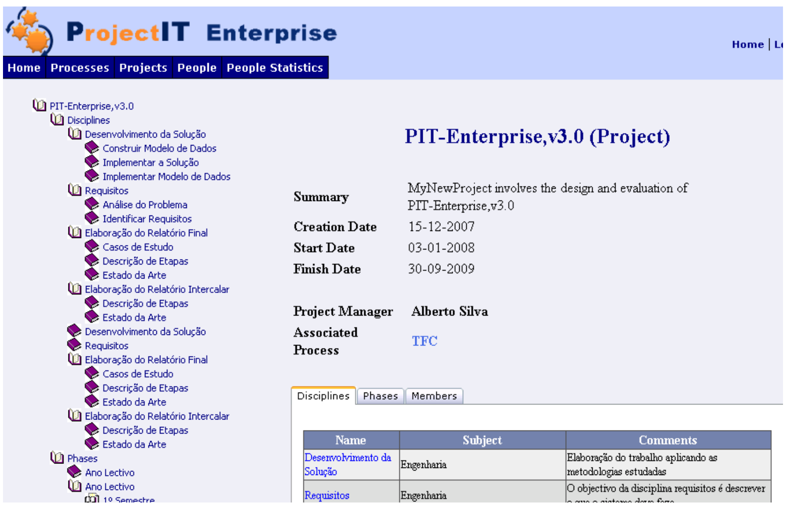


Figure 5: Project definition

3.3 Stage 3: Process Assessment and Refinement

The main goal of process assessment and refinement stage is to analyze the improvement opportunities identified in projects and validate the SPI actions accepted in previous meetings. This stage intends to analyze process data and assess results to support final decisions or corrective actions in the software development process. At project level, collected data is analyzed, interpreted, and used by the project itself. At process level, process manager uses measurement data from projects to make conclusions about proposed changes. This stage includes a final feedback meeting to discuss and validate the new process version, identifying and comparing these new patterns with previous process versions.

The most important work products produced in this stage are: project(s) assessment report(s), a SPI assessment report and an improved process version. The new process version constitutes the final milestone of this stage. This milestone is crossed when all the involved roles agree that: the objectives set during the process definition stage (and modified throughout the second phase) have been met; and especially if all intervenient are satisfied with the new process version.

When refining the process, outcomes and results described in the assessment report should be made concrete through a set of activities with the main goal to refine the organizational base process. Several activities are performed to update the used software development process and create a new process version. Although knowledge storage is the only concrete activity from this stage, other activities performed in this stage are related to knowledge acquisition and knowledge transfer.

Although not yet implemented in ProjectIT-Enterprise, our idea is to allow process version management. Initial focus, concerning this feature, is on tool support to better handle and manage versions of evolving process definitions. However, the purpose of this paper is not to discuss an innovative solution for process version management. The problem of evolving process management is related to the facility of comparing best practices proposed by each solution. The tool will operate on data that evolves over time and whose history is recorded through a version management tool. The versioning approach allows process innovation to be captured, assessed and reusable for future projects. In this research context, we conceive a model and respective mechanisms that allowed the alignment of projects and processes definition.

However, existing process metamodels only allow to define processes without giving appropriate support to analyse process that are improved versions of older ones. ProjectIT-Process Metamodel (Fig. 3) introduces process versioning. A process, which is either a composite process or a set of disciplines, includes the definition of its unique name and a process tree. A process is a root version, a revision or a variant. This means that a process must have a version number and a state (Transient, Released or Obsolete). Our approach allows represent two process types: (1) simple process and (2) composite process. A simple process comprises several disciplines and a composite process includes one or more other process (simple or composite). A new process version can be derived from an older version of the process by applying one or more modification operations (update, delete, referenced by a composite process, derived versions allowed, create projects and associated projects). Fig. 3 presents the main elements of the extended metamodel (modification and operation metaclasses).

4 Related Work

The idea of providing a complete SPI support tool, throughout the entire life cycle is not new. Some of the most popular SPI tools are: Eclipse Process Framework Composer (EPC) [12], Rational Method Composer (RMC) [13], IRIS Process Author (IPA) [14] and Visual Studio Team System Process Editor (VSTSPE) [15]. However, many of these tools follow a traditional process principle, where processes are designed in a separated tool by a process team and then exported to a different tool to be used by project teams. It is a single directional form of communication, contradictory to principles presented in ProPAM. From the referred tools, IPA has the best support for collaboration through its Wiki-based features. However, these changes are not automatically propagated for the process repository. Concerning the other tools, EPC, RMC and VSTSPE only collect user feedback from a single entity, and so collaboration is not supported.

To improve process descriptions, it should be possible to comment and annotate proposed changes by project teams. Agile communities emphasize that would be naive and inefficient to always start

project plans from scratch. For this reason, project teams today often consists of tailoring the contents of an existing process repository to the needs of a specific project settings.
Following the principles of ProPAM, ProjectIT-Enterprise adopts a bi-directional form of communication, between process team and project team. The lack of alignment between process and project(s) results from processes unrelated to project activities and failure in detecting project changes that improve processes.
Main differences between ProjectIT-Enterprise supporting ProPAM and other evaluated tools relays on the strong concept of “process and project alignment”. Changes are detected in projects and previously are propagated to process descriptions creating a new process version. Actually, ProjectIT-Enterprise supports mainly process and project definition. More features need to be included in the near future, particularly a process version management infrastructure to evaluate SPI success.
Another problem in SPI tools is poor support to version management. When documenting improved process descriptions, a good version management system is important to trace differences between process versions. IPA handles version tracing through exporting process descriptions to XML and previous store in external version control systems. XML descriptions can be re-imported into IPA as necessary. However, versions comparison is not possible.

5 Conclusions and Future Work

In this paper we introduce the ProjectIT research initiative, its main issues and challenges, and the importance of a collaborative framework for SPI. To validate the proposed ideas and contributions, we have developed the ProjectIT-Enterprise tool. ProjectIT-Enterprise provides collaborative features for process definition, project management as well as process and project alignment. ProjectIT-Enterprise currently supports the two most relevant stages of ProPAM methodology: (1) process definition and (2) apply process to projects. Results achieved until now show that it is possible to specify projects based on process data in a more productive way, by adapting and integrating techniques such as modelling and models transformation. ProjectIT-Enterprise is a tool that integrates process definition and project definition (based on previously defined processes). As opposite to evaluated tools, such as EPC, RMC, IPA and VSTSPE, where processes are designed in a specific environment and then exported to be used by other project management tools. Different tools for different teams prevent capturing new practices suggested by project teams and propagation to process descriptions after being accepted by process teams. In order to react to an industry that requires agility, quality and efficiency, it is imperative to design tools that provide collaborative approaches.
Another common problem in SPI is the assessment of proposed changes (new process version) to accept as a good solution. Although not yet implemented in ProjectIT-Enterprise, ProPAM methodology includes a stage dedicated to process assessment and refinement that provides some ideas on how to address process version management to evaluate SPI success. In conclusion, we believe that software process tools should move towards a new direction that includes deeper interaction between process and project teams to create and to improve real world processes adopted by the software industry.

Literature

1. SEI: Capability Maturity Model Integration (CMMI), Version 1.2. SEI: CMU/SEI-2002-TR-029. Software Engineering Institute, USA (2002)
2. ISO: ISO 9000-3: Quality management and quality assurance standards - Part 3. International Organization for Standardization (1990)
3. ISO/IEC: 15504-2 Information technology - Software process assessment – Part 2: A reference model for processes and process capability. July (1998)
4. Coleman, G., O'Connor, R.: Investigating Software Process in Practice: A Grounded Theory Perspective. Journal of Systems and Software (2007)
5. Staples, M., Niazia, M., Jefferya, R., Abrahamsd, A., Byatte, P., Murphyf, R.: An exploratory study of why organizations do not adopt CMMI. Journal of Systems and Software **80** (2007) 883-895
6. Staples, M., Niazi, M.: Systematic review of organizational motivations for adopting CMM-based SPI. National ICT Australia (2006)
7. Weber, K., Araujo, E., Scalet, D., Andrade, E., Rocha, A., Montoni, M.: MPS Model-Based Software Acquisition Process Improvement in Brazil. In: R. Machado, F. Abreu, Cunha, P. (eds.): 6th Quality of Information and Communications Technology (QUATIC 2007). IEEE Computer Society, Lisboa, Portugal (2007) 110-122
8. ProjectIT: ProjectIT web site. In: INESC-ID, ISG, http://isg.inesc-id.pt/pit-enterprise/Home@1.aspx (2010)
9. Silva, A.R., Videira, C.: UML Metodologias e Ferramentas CASE, Vol. Volume II (2008)
10. Martins, P.V., Silva, A.R.: PIT-P2M: ProjectIT Process and Project Metamodel. In: Lecture Notes in Computer Science, Vol. 3762. Springer Berlin / Heidelberg, Agia Napa, Cyprus (2005) 516-525
11. Martins, P.V., Silva, A.R.: ProPAM: SPI based on Process and Project Alignment. In: Mehdi Khosrow-Pour, D.B.A. (ed.): 2007 IRMA International Conference. IGI Publishing, Vancouver (2007)
12. Eclipse Process Framework Web site, http://www.eclipse.org/epf/.
13. Rational Method Composer Web site, http://www-01.ibm.com/software/awdtools/rmc/.
14. IRIS Process Author Web site, http://www.osellus.com/IRIS-PA.
15. Microsoft: Visual Studio Team System Process Editor, http://visualstudiogallery.msdn.microsoft.com/en-us/0e69a28f-020c-488b-80b3-f4c89a20621d.

Author CVs

Paula Cristina Negrão Ventura Martins

Paula Ventura Martins obtained a doctorate in computers from the IST of the Technical University of Lisbon. She is an auxiliary professor in the Electronics and Computers Engineering Department of Faculty of Sciences and Technology, University of Algarve, and she is also an associated researcher of the Information Systems Group at INESC-ID. From August 1992 until October 1993 she was a systems analyst at Instituto de Soldadura e Qualidade (ISQ). She is member of the Portuguese Ordem dos Engenheiros. Her current interests are software process improvement,software development processes, workflows, modeling languages, and project management.

Alberto Rodrigues da Silva

Alberto Rodrigues da Silva is an associate professor of information systems in the Computer and Information Systems Department at the Technical University of Lisbon (Instituto Superior Técnico -IST/UTL), Lisboa, Portugal. He is also a senior researcher at INESC-ID Lisboa, and a partner at the SIQuant—Engenharia do Território e Sistemas de Informação company. He has professional and research interests in the following areas: modeling and metamodeling, model-driven engineering, requirement engineering, enterprise knowledge-based platforms, and CSCW and CASE tools. He also has interests in enterprise architectures, software project management, geographic information systems, and social networks. He is author or coauthor of three technical books and more than 140 peer-reviewed scientific communications, and he has served on the program committees of several international conferences and workshops in different areas of computer science. He is member of the ACM and the Portuguese Ordem dos Engenheiros.

Teaching Software Process Improvement and Assessment

Christiane Gresse von Wangenheim[1], Jean Carlo R. Hauck[1, 2]
[1]Federal University of Santa Catarina (UFSC)
Florianópolis/SC – Brazil
gresse@gmail.com
[2] DkIT – Dundalk Institute of Technology
Dundalk/Co. Louth, Ireland
jeanhauck@gmail.com

Abstract

In order to prepare students for careers as software process engineers, software engineering education needs to adopt innovative instructional designs to support effectively the learning of required knowledge and skills. In this paper, we propose a cross-course design for teaching software process improvement and assessment in a graduate course in combination with an undergraduate capstone project course adopting a constructivist approach. We applied the proposed course design and investigated its impact on learning effects, its adequacy and strengths and weaknesses by administering a pre-and post-test and applying a questionnaire at the end of the course. First evaluation results indicate a positive learning effect on students to develop competencies required for software process engineers as well as it successfully engaged both graduate and undergraduate students while providing a beneficial experience through their interactions.

Keywords

Software Process Improvement, CMMI, Software Process Assessment, Education, Software Engineering

1 Introduction

Software process improvement (SPI) is becoming more important each year in order to meet the challenge of complex software systems and an increasing demand for more reliable systems. By now a large number of software organizations have established software process improvement initiatives and many of them have been formally assessed [1]. Yet, given the broad range of approaches, methods and tools for SPI, organizations are struggling to find competent professionals, who are able to effectively engineer software processes around the organization [2]. In this context, an additional, distinct role is required, the Software Process Engineer (SPE) [3], who is responsible for the definition, assessment, establishment and maintenance of software processes, analysis of quality problems and support for the implementation of improvement suggestions [4]. These responsibilities are distinct enough from other software development or management tasks that responsible need to have specific knowledge on SPI concepts and processes, software process capability/maturity models and standards, software process improvement and assessment methods as well as good interpersonal skills [2, 4]. And, although there have been significant advances during the last year in software engineering education, in general, e.g., through the development of the SWEBOK [5] and Software Engineering (SE) curriculum guidelines [6], less emphasis on SPI can be observed even on the graduate level [7, 8]. SPI content is more typically taught in professional training courses and/or formal preparation courses for professional certifications, such as, e.g., SEI's SEPM Certificate Programs (http://www.sei.cmu.edu/credentials/sepm.html#sepm) or the International Software Process Improvement Certification (ISPIC) (http://www.spinstitute.org/certification.htm). Therefore, it becomes important to provide opportunities for students to learn these required knowledge and skills also as part of formal education [7].

Another issue is the way in which SE courses are typically taught. Expository lessons are still the dominant instructional technique in, basically, all sectors of education and training [9]. While they are adequate to present abstract concepts and factual information, they are not the most suitable for higher-cognitive objectives aiming at the application and transfer of knowledge to real-life situations [10]. Thus, in order to improve SE education, a general trend is to emphasize “hands on” experience for the students related either to industry or a simulated environment [11. Yet, so far, there have been made only very few proposals for teaching SPI effectively recreating an authentic context in which software processes are engineered in the classroom.

At the Master Program in Applied Computer Science at the UNIVALI – Universidade do Vale do Itajaí in Brazil, a SPI course is being held for master students since 2006. In the beginning, the course concentrated on theoretical topics. A classical educational method had been adopted using expositive lectures, discussions and case study reviews. However, it seemed that the course did not successfully teach the content on the application level and did not motivate the students sufficiently. As a result students acquired a surface knowledge of basic topics, but had problems to apply them as well as to achieve higher cognitive levels as a basis for their research projects.

Thus, in 2009, the SPI course has been re-designed with the objective to increase the learning effect, specifically, on the application level and reinforcing the understanding of relevant concepts. A constructivist approach has been adopted, using situated learning and problem solving in an authentic context through group work and the nurturance of reflexivity and learning in an ill-structured domain. As, due to confidentiality reasons and unavailability of staff, no access to a real software organization was given in order to apply software process assessment (SPA) and improvement, the graduate SPI course has been combined with a capstone project in the undergraduate Computer Science course at the UNIVALI. This recreated a realistic context, which provided the opportunity for the graduation students to apply SPI concepts on the software projects being run in the undergraduate course. This cross-course design successfully engaged both graduate and undergraduate students, while providing a beneficial practical experience, which contributed to learning on the cognitive level of application.

2 Related Work

SE education has been received increased attention recently, and much progress has been made, principally by the development of the SWEBOK [5] as well as curriculum guidelines for SE courses [6]. And, although, the most common approach to teaching SE is the use of lectures, supplemented by laboratory sessions, tutorials, etc. [6], there have various articles published on educational strategies for SE education. Recent trends reflect a shift from objectivist learning, which views learning as the transmission of knowledge from the teacher to the learner, to constructivist learning, regarding learning less as the product of transmission than a process of active construction [6,13]. In this setting, diverse instructional designs and experiences on SE education have been published [11, 14], but, only very few focus specifically on SPI education. An exception is the experience reported by T. Dingsøyr et al. [15] on teaching SPI around an industrial case study based on lectures and group exercises. Another example is a graduate SE course to educate students on the basic concepts of SPE proposed by Hawker [3]. This course design is based on the OMG Software Process Engineering Metamodel and the IEEE Standard for Developing a Software Project Life Cycle Process as ways to model and compare process design alternatives and to provide mechanisms to assemble reusable process components into enactable processes. Other courses use the Personal Software Process (PSP) to teach software process improvement [16]. At the Ecole Polytechnique de Montreal a SE course held [17], carries out a project where students use a simplified version of the Trillium model to assess their project. Another example is the Real World Lab course at the Georgia Institute of Technology [18], where undergraduate students are involved with real industry projects and take part in performing a CMM assessment on local industry by interviews.

In order to offer an environment in which students can have hands-on experiences, most of these experiences are based on an industry partnership in which students participate in the companies' SPI projects. Yet, often software organizations are reluctant to share their quality and process issues with students and/or do not have the capacity to assign staff to those activities [8] and, therefore, such a partnership may not always be possible. In this context, an alternative for providing an environment in which students can learn to apply SPI concepts may be the combination of courses in a cross-course design. In other SE areas, the adoption of such a cross-course design has shown positive results [19, 20, 21]. Cross-course designs seem to be especially indicated when using the advantage of capstone projects being executed, which allows to apply SE concepts on larger and more complex projects within the time and resource restriction of each of the individual courses. In addition, they also can offer a more stimulating environment for teaching relevant skills, such as, communication and help to motivate SE better. Yet, so far, no experiences on such instructional designs for teaching SPI have been encountered.

3 Proposal for a Cross-Course SPI Course

One of the main research areas at the Master Program in Applied Computer Science at the UNIVALI is SPI and SPA. Therefore, master students need to acquire knowledge and skills relevant to SPEs. Students who enter the course are Bachelors in Computer Science, with basic SE knowledge and practical software development experiences. They attend at least two SE related courses – one providing a general overview on SE and one focusing, specifically, on SPI. The SPI course is offered with 4-hour lectures during 15 weeks. The objective of the course is to teach basic knowledge on SPI, mainly on defining and documenting software processes, assessing software process capability and/or maturity as well as on selected SE topics, such as, project management, on the cognitive levels of remembering, understanding and applying in accordance to Bloom's revised taxonomy [12] as well as skills, such as, communication, team work, leadership, and problem solving.

In order to achieve these objectives, we propose an educational strategy based on the constructivist learning theory through the integration of practical course work within a simulated software project being run in parallel as part of an undergraduate capstone project at the UNIVALI. Within this capstone project, undergraduate students work in teams to plan and monitor, analyze, design, implement and test a software system. This cross-course design enables the students of the SPI course to assess and define a software process for an authentic environment. For a better understanding, first the design of the graduate SPI course is explained in detail and, then, a summary of the undergraduate capstone project discipline is provided.

3.1 SPI course design

The learning objective of the graduate SPI course is that students remember and understand software process improvement and assessment concepts, models and approaches and acquire the competency to apply them with assistance in practice. The students should also reinforce their knowledge on project management and SE in general. Table 1 summarizes the lecture plan.

Table 1. Lecture plan

Unit	No. of hours		Contents	Teaching method	Evaluation
	theory	practice			
0	4	0	Course presentation (and pre-test)	- Expositive lecture with discussion - Multiple-choice exam	
1	2	0	SPI- basic concepts and approaches (IDEAL, ISO/IEC 15504-4)	- Expositive lecture with discussion	Questions in final exam
2	2	0	Software process reference models (CMMI, ISO/IEC 12207, ISO/IEC 15504-5, MPS.BR)	- Expositive lecture with discussion	Questions in final exam
3	2	16	SPA – concepts and process (based on SCAMPI and ISO/IEC 15504)	- Expositive lecture with discussion - Assessment project	- Work project 1: Assessment project - Questions in final exam
4	4	12	Software process definition and	- Expositive lecture with	- Work product 2:

			documentation -concepts and process	discussion - Process modeling project	Process definition project - Questions in final exam
5	4	0	Course debriefing (and post-test)	- Discussion - Multiple-choice exam - Lecture evaluation questionnaire	

The educational strategy of the course is based on a constructivist approach providing a hands-on experience to the students to enable them to learn how to apply those concepts and approaches in practice. Expositive lectures are reduced to a minimum, just to provide an introduction and a general overview on SPI concepts as a basis for the work projects to be done during the course. The main focus of the course is on two practical work projects to be done in groups. These work projects simulate authentic SPI situations for students to learn and exercise the application of SPA and process definition. The work projects take place in the software project being run as part of an undergraduate capstone project (see section 3.2). As input to the work projects, students receive a detailed instruction by the teacher and a set of relevant material (including, context descriptions, e.g., of the software organization and its process used in the capstone project, a definition of a SPA method and background material, such as, the CMMI model, etc.). This information as well as the artifacts being created by the students is managed on a google site (Figure 1).

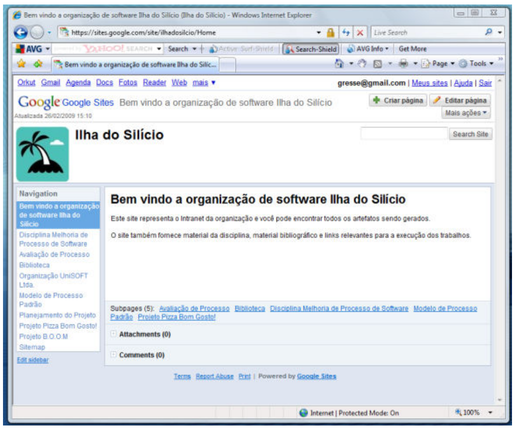
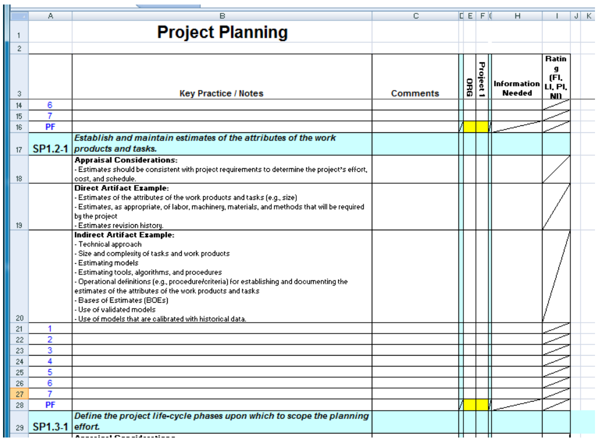


Figure 1. Google site overview of the SPI course

A large part of the work projects is done in the classroom in order to allow the students to meet as a group as well as to allow the teacher to function as a tutor, who keeps active learning going, answers questions, indicates additional material and encourages reflections. In the end of each of the work projects, the student groups present the results in the classroom and discuss and compare their results and experiences with their colleagues.

Work project 1: Process assessment. The objective of the assessment project is to teach the remembering and understanding of SPA concepts and approaches and the application of SPA in practice. As a basis for the realization of the assessment a simplified version of the SCAMPI method [22] has been predefined, including a high-level process description and document templates. During this exercise, the student groups plan, execute and analyze a SPA focusing on capability level 2 of the Project Planning (PP) process in accordance to the CMMI-DEV v1.2 model [23]. Here we focus the assessment on the Project Planning process area, as this is the first step of the capstone project in the undergraduate course, and, therefore, results of this step are available in time for the realization of the assessment. The assessment is realized in a cross-course way in the organization simulated in the undergraduate capstone project course. The students plan the assessment based on a characterization of the organization and the competencies and roles assigned to the group members of the capstone project. As further input to the assessment, the project plan produced in the capstone project is provided. Based on this information, the assessment group initiates the identification and documentation of direct and indirect evidences. All evidences collected during the assessment are documented in an EXCEL sheet (Figure 2), indicating the evidences for each of the specific and generic practices of the selected process area.



	A	B	C	ORG	Project 1	Information Needed	Rating (FI, LI, PI, NI)
1		Project Planning					
2							
3		Key Practice / Notes	Comments	ORG	Project 1	Information Needed	Rating (FI, LI, PI, NI)
14	6						
15	7						
16	PF						
17	SP1.2-1	*Establish and maintain estimates of the attributes of the work products and tasks.*					
18		**Appraisal Considerations:** - Estimates should be consistent with project requirements to determine the project's effort, cost, and schedule.					
19		**Direct Artifact Example:** - Estimates of the attributes of the work products and tasks (e.g., size) - Estimates, as appropriate, of labor, machinery, materials, and methods that will be required by the project - Estimates revision history.					
20		**Indirect Artifact Example:** - Technical approach - Size and complexity of tasks and work products - Estimating models - Estimating tools, algorithms, and procedures - Operational definitions (e.g., procedure/criteria) for establishing and documenting the estimates of the attributes of the work products and tasks - Bases of Estimates (BOEs) - Use of validated models - Use of models that are calibrated with historical data.					
21	1						
22	2						
23	3						
24	4						
25	5						
26	6						
27	7						
28	PF						
29	SP1.3-1	*Define the project life-cycle phases upon which to scope the planning effort.*					

Figure 2. Example extract of the evidence collection sheet

Then, following the assessment plan, the assessment team realizes interviews with members of the capstone project in order to collect also affirmations from the project members (Figure 3).

Based on the obtained affirmations, the data collection is completed and the obtained information is analyzed. As result of this activity, each assessment group prepares and presents a report on the assessment results indicating improvement opportunities. The assessment results are also provided as feedback to the students of the capstone project. The cross-course integration of this work project is detailed in Table 2.

Table 2. Cross-course integration in the assessment work project

Graduate SPI course		Undergraduate capstone project course
Plan for assessment	<-	Characterization of "organization"
	<-	Characterization of member's competencies and roles assignment
Obtain objective evidence and document in data collection plan	<-	Project plan
Examine objective evidence		
Realize interviews and examine evidence from interviews	<->	Interviews with members of the project team
Generate appraisal results	->	

Figure 3. Assessment interview

Work project 2: Software process definition. Based on the results obtained during the assessment, each group of the SPI course defines a software process in the context of the organization of the capstone project, in alignment with the objectives and practices required with respect to capability level 1 of the CMMI-DEV v1.2 for the Project Planning process. The process definition is done in a hybrid way, combining descriptive definition (based on the elicited informal process being executed in the capstone project) and prescriptive definition (improving identified weaknesses with respect to the practices as required by CMMI). The students document the process, describing objectives, activities, methods, techniques and tools to be used as well as work products to be consumed and generated (including, the definition of templates for all work products to be generated) and the identification of roles and responsibilities. The process definitions are documented in a demo version of the Enterprise Architect tool (http://www.sparxsystems.com.au/) (Figure 4). In addition, the students explicitly track the compliance of the defined process to all required specific practices of the PP process of the CMMI-DEV v1.2.

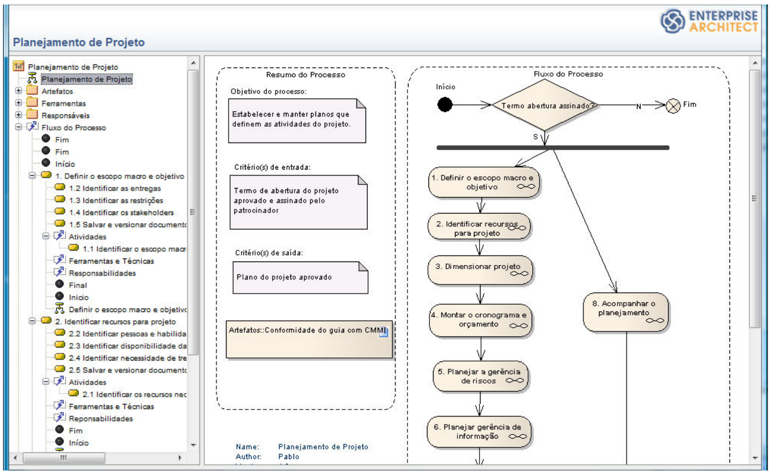


Figure 4. Example extract of a developed process definition

Evaluation: The achievement of the expected learning outcomes is assessed based on the evaluation of both work projects (SPA and process definition) and the result of an exam in the end of the course. The work projects are evaluated with regard to completeness, correctness, clarity, consistency of the produced results and the organization and knowledge shown during the presentation. The exam is a multiple choice test including questions on the cognitive levels of remembering, understanding and application.

3.2 Capstone Project Course

At the Computer Science undergraduate program at the UNIVALI, software engineering concepts are introduced in two software engineering courses (SE 1 and SE 2) covering basic concepts and the software process focusing on requirements development, design, testing, project and quality management. Two subsequent courses (APS1 and APS2), focus on the application of this theoretical knowledge and provide “hands-on” experience. In the APS2 course, students realize a semester-long capstone project in groups of 4 to 6 students. They plan and execute a software project executing requirements analysis, design, implementation and testing.

The project starts with a planning phase, in which the students plan the project using a pre-defined project plan template. Then, they start the technical activities following a predefined waterfall process model consisting of 4 phases: requirements development, design, implementation and testing. During the execution of the project the students collect data (basically, on the effort spent and start and end dates of activities) and in the end a project post-mortem is realized. To enable close accompanying of the student work and the provision of early feedback, each team has to deliver and present its results at the completion of each phase of the project.

The role of the customer is played by the teacher. Each student is assigned to a specific role (e.g., project manager, requirement analyst) indicating his/her primary responsibility. Yet, due to the learning objectives, all students participate in all phases of the capstone project.

4 Preliminary Evaluation

As part of the change of the course design, a preliminary evaluation of the new design has been performed in the first semester of 2009. The objectives of the evaluation are to analyze:

O1. If a positive learning effect on the cognitive levels of remembering, understanding and applying level and/or skills can be observed;

O2. If the course design is considered appropriate in terms of teaching method, adequacy of work project, cross-course integration and utility in practice; and

O3. What are the course strengths and weaknesses?

The objective of this preliminary evaluation is rather to obtain a first subjective evaluation of these aspects from the student's point of view.

These research questions have been analyzed based on the Kirkpatrick's four-level model for evaluation [24], a popular and widely used model for the evaluation of training and learning. In accordance to Kirkpatrick's four-level model for evaluation, we investigate all objectives on level one: reacting, which focuses on how the participants feel about the learning experience by collecting data via satisfaction questionnaires. We investigate objective 1 also on level two: learning, which focuses on the evaluation of the increase in knowledge by administering a pre-and post-test. On level 2, we evaluate the learning effect separately for each of the knowledge levels (remembering, understanding, applying) by comparing the average scores between pre-test and post-test (relative learning effect).

Different kinds of data were collected, including the realization of a pre-test exam in the beginning of the course as benchmark and a post-test exam in the end of the course. Both exams were multiple choice tests with similar content and degree of difficulty. In addition, subjective data has been collected via questionnaire from the students in the end of the course.

4.1 Results

The proposed course design in the SPI course has been applied during the 1. Semester 2009 at the Master Program on Applied Computer Science at the UNIVALI. In total, 5 students attended the course. In general, we obtained a very positive feedback with respect to the new course design.

O1. Can a positive learning effect on the cognitive levels of remembering, understanding and applying level and/or skills be observed?

This question has been analyzed by comparing the results in the pre-test and post-test. In general, the average difference is 17.2 points (with a total number of 80 points per test) varying from a difference of 7 to 33 points, indicating that the knowledge of the students increased. It can also be observed that the greatest knowledge increase took place on the cognitive level of application (Figure 5), as intended in the learning objective of the course.

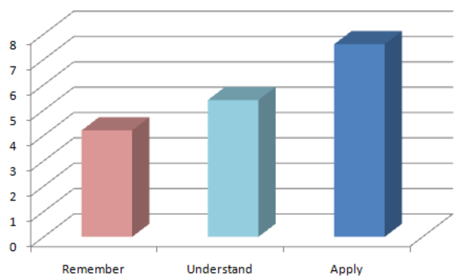


Figure 5. Average difference of knowledge per cognitive level

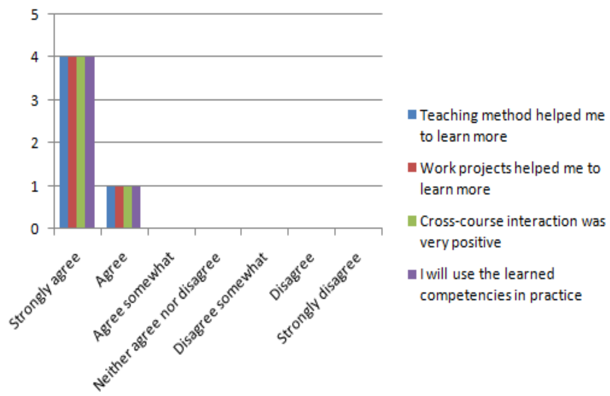


Figure 6. Distribution of number of student responses

The subjective evaluation by the students also indicated that the students believe that the course helped them to evolve relevant skills, such as, team work, problem solving, communication, etc.

O2. Is the course design considered appropriate in terms of teaching method, adequacy of work project, cross-course integration and utility in practice?

These aspects have been analyzed based on subjective data collected from the students at the end of the course. On a 7-point likert scale, all students either agreed or strongly agreed that the teaching method, the work projects and the cross-course integration are appropriate and they pretend to use the acquired competencies in practice (Figure 6).

O3. What are the course strengths and weaknesses?

Based on the feedback obtained by the students, the principal strengths of the course design are:

- Presentation of theoretical concepts constantly in relation with practical application;
- Strong emphasis on practical work;
- Practical work in combination and interaction with the undergraduate course;
- Presentation and discussion of the results of the practical work projects in the classroom; and
- Organization of the two practical work projects as a sequence (second work project building upon the results of the first one).

As principal weaknesses the students cited:

- Small number of students attending the course and, consequently, the formation of only very small groups;
- Lack of a complete working example; and
- Ineffective usage of the time reserved for practical work in classroom.

4.2 Discussion

Although, the results of this preliminary evaluation have to be interpreted with extreme caution due to the very small sample size and its restriction to only one application, the results may provide a first indication that the new course design has a positive impact on the learning effectiveness. It seems that the constructivist approach, in which learning is defined as an active process for knowledge building rather than a knowledge acquisition process, contributes positively to the learning of knowledge on the application level as well as relevant skills. Yet, we also observed that just providing a learning environment, literature and a general introduction may not be sufficient. Students (maybe, due to the fact, that they are more used to traditional classroom teaching) expect a more guided approach. We therefore, intend to include more expositive lectures substituting independent literature study by the students on their own. Another alternative is also the integration of more diverse teaching methods, including, for example, games and case studies.

The cross-course design of the course was considered a very positive aspect of the course ensuring a richer learning experience. The students of both, the graduate and the undergraduate course, liked the experience very much. It turned the assessment into a “real” experience applied to a project and people outside their own course, which was executed with great care and in a professional way.

Another issue emphasized by the students was the presentation and discussion of the results of the work projects in the classroom. The students expressed that these offered them an additional opportunity for learning by examples, especially, as no “golden solution” for the work projects was available.

The course design also seems to be able to deal with varying levels of background, as within this application, the students background varied from professionals with PMP certification to students who finished graduation ten years ago and were just starting to learn about software engineering.

5 Conclusions

Teaching the application of SPI concepts is challenging. In this paper, a cross-course design is proposed combining a graduate SPI course and an undergraduate capstone project in order to recreate an authentic environment, which allows students to acquire practical knowledge and experience. First experiences in applying this design provide a preliminary indication for a positive impact on learning. Implementing the improvement opportunities identified, we intend to repeat the application of this course design in the SPI course, collecting also feedback on a larger scale.

Acknowledgements

We would like to thank the students of the SPI and the APS2 course at the UNIVALI during the 1. Semester 2009. This work was supported by the CNPq (*Conselho Nacional de Desenvolvimento Científico e Tecnológico*) and CAPES (*Coordenação de Aperfeiçoamento de Pessoal de Nível Superior*), entities of the Brazilian government focused on scientific and technological development.

Literature

1. SEI. Process Maturity Profile CMMI for Development SCAMPI Class A Appraisal Results 2008 End-Year Update. Software Engineering Institute, Pittsburgh, March 2009.
2. IISP. Software Process Improvement Body of Knowledge (SPIBOK) http://www.spinstitute.org/spibok.htm
3. J. S. Hawker. A Software Process Engineering Course. American Society for Engineering Education. Proceedings of the 2009 American Society for Engineering Education Annual Conference, Austin, TX, 2009.
4. R. McFeeley. IDEAL: A User's Guide for Software Process Improvement. Handbook CMU/SEI-96-HB-001, Carnegie Mellon University/Software Engineering Institute, Pittsburgh, 1996.
5. IEEE Computer Society. SWEBOK – Guide to the Software Engineering Body of Knowledge, 2004 version, IEEE Computer Society, 2004.
6. The Joint Task Force on Computing Curricula IEEE CS/ACM. Software Engineering 2004 - Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering, 2004.
7. E. Biberoglu & H. Haddad. Survey of Industrial Experiences with CMM and the Teaching of CMM Practices. Journal of Computing Sciences in Colleges, vol. 18, no. 2, pp 143 – 152, December 2002.
8. M. L. Jaccheri. Software Quality and Software Process Improvement Course based on Interaction with the Local Software Industry. Computer Applications in Engineering Education, John Wiley & Sons, Volume 9, Issue 4, pp 265 – 272, Mar 2002.
9. F. Percival et al. Handbook of Educational Technology, 3. ed. Kogan Page, London, 1993.
10. J. Choi, M. Hannafin. Situated Cognition and Learning Environments: Roles, Structures and Implications for Design. Educational Technology Research Development, vol. 43, no. 2, pp. 53–69, 1995.
11. M. L. Jaccheri & P. Lago. How Project-based Courses face the Challenge of educating Software Engineers. Proc. of the Joint World Multiconference on Systemics, Cybernetics and Informatics (SCI'98) and the 4th International Conference on Information Systems Analysis and Synthesis, Orlando, USA, pp. 377-385, 1998.
12. L. W. Anderson, D. R. Krathwohl (Eds.). A Taxonomy for Learning, Teaching, and Assessing: A Revision of Bloom's Taxonomy of Educational Objectives. New York: Longman, 2001.
13. S. Hadjerrouit. Toward a Constructivist Approach to Web-based Instruction in Software Engineering. Proc. of Norwegian Computer Science Conference. Oslo/Norway, November 2003.
14. E. Bareiša et al. Software Engineering Process and its Improvement in the Academy. Information Technology and Control, Vol.34, No.1, 2005.
15. T. Dingsøyr et al. Teaching Software Process Improvement through a Case Study. Computer Applications in Engineering Education Computer Applications in Engineering Education, Volume 8, Issue 3-4, pp. 229 – 234, Nov 2000.

16. G. W. Hislop. Teaching Process Improvement in a Graduate Software Engineering Course. Proc. of 29th ASEWIEEE Frontiers in Education Conference, San Juan/Puerto Rico, 1999.

17. P. N. Robillard, J. Mayrand, J.-N. Drouin. Process Self-Assessment in an Educational Context. Software Engineering Education, Springer Lecture Notes in Computer Science, vol. 750/1994, pp. 211-225, 1994.

18. M. M. Moore & T. Brennan. Process Improvement in the Classroom. Software Engineering Education, Springer Lecture Notes on Computer Science, vol. 895/1995, pp. 123-130, 1995.

19. C. Brown & R. Pastel. Combining Distinct Graduate and Undergraduate HCI Courses: an Experiential and Interactive Approach. Proc. of the 40th ACM Technical Symposium on Computer Science Education, Chattanooga/USA. pp. 392-396, 2009.

20. G. Sindre et al. The Cross-Course Software Engineering Project at the NTNU: Four Years of Experience. Proc. of the 16th Conference on Software Engineering Education and Training, IEEE Computer Society, Washington/USA, 2003.

21. R. J. Fornaro et al. Cross-functional Teams Used in Computer Science Senior Design Capstone Courses. 30th Annual Frontiers in Education Conference, 2000.

22. SCAMPI Upgrade Team. Standard CMMI Appraisal Method for Process Improvement (SCAMPI) A, Version 1.2: Method Definition Document. Handbook CMU/SEI-2006-HB-002. Carnegie Mellon University/ Software Engineering Institute, Pittsburg, 2006.

23. CMMI Product Team. CMMI for Development, Version 1.2. Technical Report CMU/SEI-2006-TR-008, Carnegie Mellon University/ Software Engineering Institute, Pittsburg, 2006.

24. D. L. Kirkpatrick, J. D. Kirkpatrick. Evaluating Training Programs: The Four Levels, 3rd edn. Berrett-Koehler Publishers, San Francisco, pp 379, 2006.

Author CVs

Christiane Gresse von Wangenheim

Christiane Gresse von Wangenheim is a professor at the Federal University of Santa Catarina (UFSC). Her research interests are software process improvement, including project management. Previously, she worked at the Fraunhofer Institute for Experimental Software Engineering and the UNIVALI. She received a PhD in Production Engineering at the Federal University of Santa Catarina (Brazil) and a PhD in Computer Science at the University of Kaiserslautern (Germany). She's also a PMP - Project Management Professional and Assessor of the Brazilian Process Improvement Model MPS.BR. She's a member of the IEEE Computer Society, the Project Management Institute, and the Working Group ISO/IEC JTC1/SC7/WG24—SE Life-Cycle Profiles for Very Small Enterprises. Contact her at UFSC, Campus Universitário 88049-200 Florianópolis/SC, Brazil; gresse@gmail.com

Jean Carlo R. Hauck

Jean Carlo Rossa Hauck is visitor researcher at Dundalk Institute of Technology (DkIT) and SEPG manager of the CYCLOPS Research Group at the Federal University of Santa Catarina (UFSC). His research interests are in software process improvement and project management. He received his M.Sc. in Computer Science from the UFSC and he is PhD student of Knowledge Engineering and Management at the Federal University of Santa Catarina. Contact him at UFSC - EGC, Campus Universitário 88049-200 Florianópolis/SC, Brazil; jean-hauck@gmail.com

Codified vs. Personalized – A Vertical Approach to the Dilemma of the Knowledge Management Strategies

Karsten Jahn & Peter Axel Nielsen

Abstract

Software process improvement is a huge practical concern in software companies today and it has consequently been addressed in much research. A part of this research has applied a knowledge management perspective. Researchers pointed out that basically two different strategies exist: Personalization (focus on people and their collaboration) and codification (focus on documents and their accessibility). It has been shown that mixtures are difficult to maintain and 80-20 shares between the two strategies are preferred. In our research we address beneficial knowledge management for a software company and by studying its knowledge processes we have identified a number of problems there. To alleviate some of these, we have built a prototype. It supports different knowledge management strategies at different organizational levels (e.g., software managers and software developers). Consisting of a wiki and an enterprise system, we show how each system focuses on one of the strategies and describe the differences for tool support in the strategies, why a combination could be beneficial and how the connection between the two different parts of the prototype works. From this study we then conclude that an equally distributed knowledge management strategy between personalized and codified is applicable and can be supported in a prototype.

Keywords

Knowledge management, personalization strategy, codification strategy

1 Introduction

Software process improvement (SPI) became interesting for many companies searching for a way to improve the operational quality. According to Aaen et al. [1] are there three different approaches for SPI: Evolution, norm and commitment. The norm approach to SPI is a way to adopt the existing norms and the commitment approach stresses the active support with attention and resources of the senior management [1]. The evolutionary approach includes incremental changes based on experiences of previous executions, like the IDEAL model, which organizes the whole process in five phases: Initiating, diagnosing, establishing, acting and learning [2]. Researchers often focused on small and medium-sized companies, as these seem to face the changing environments more often than larger ones. The changing circumstances need to be matched by changed processes in order to stay successful [3]. Kautz studied three small companies according to their process improvement [4; 5]. He points out four critical success factors: a tailored approach, functioning networking, external assistance and external financial support. A different study addresses problems with current software processes through problem diagnosis, i.e., the software developer's own perception of problems and their commitment [6].

Even though SPI differs from case to case, a fundamental part in all of these reports is the sharing of knowledge and/or experience, referred to as knowledge management (KM). Arent and Nørbjerg analyzed the theory of organizational knowledge creation in SPI [7], stating that an organization's software practices are based on the software developers' and managers' knowledge [8]. Mathiassen and Pourkomeylian point out that a KM strategy has to fit to an SPI approach in order to succeed [9].

Generally, the implications on KM can be derived from the six different perspectives of knowledge: Personalized information, understanding, object, process, access and capability [10]. Deriving from these, researchers distinguish between two knowledge management strategies: the personalization strategy and the codification strategy [11]. While personalization focuses on people and provides possibilities to share their knowledge (person to person), the codification strategy focuses on documents and provides possibilities to write down and store information and for others to access it (people to documents to people). KM is a field that has been addressed by researchers from many different angles, but also industry has reacted with a large variety of tools, specialized for different approaches [12]. Widespread examples would include wikis and enterprise systems (ES). Wikis are collaborative knowledge tools that allow the user to create and edit hypertext pages. Being very simple to use with hardly any restrictions for the users, wikis are considered as supportive tools for the personalization strategy [13]. ES on the other hand assist a company operationally by integrating data for use in the whole organization. They can record data, store and access content of many different backgrounds in an equal form and are thus supporting the codification strategy [14].

In this paper we explore the distinction between the codified and personalized KM strategies for SPI and in particular how the strategies can be supported by knowledge tools and systems. We envisage support in different kinds of systems according to the dilemma of the strategies at different organizational levels. We introduce a prototypical system that we have implemented, based on these considerations. Our contribution is a layered strategy to combine oppositional strategies on different organizational levels to a combined KM strategy, which the prototype follows. As the systems are in the evaluation phase at the time of submission, the paper is created without actual test results but with detailed discussions of features and possibilities with target people.

The rest of the paper is organized as follows: In section 2 we describe the two KM strategies in more detail. Section 3 sums up the requirement for KM in our case company, followed by the design of our prototype in section 4. Afterwards we discuss our approach and the impact on the company (section 5) followed by the conclusion in section 6.

2 Knowledge Management Strategies

Hansen et al. analyzed different consulting companies and explained two different strategies for KM that they found. The bottom line is that the KM strategy of a company has to fit to its competitive strategy [11]. However, a knowledge management strategy is not only addressing the company level, even internally departments or services can differ in strategic orientation [9]. In the following we describe the two strategies in detail, based on the two sources mentioned [11; 9]. Additionally we explain cases and give examples for supporting tools. A broader overview for KM and its implications for knowledge management systems (KMS) is described in [10], details about KM in software engineering, including an overview of technological approaches in [12].

2.1 Codification

Especially in process-driven companies, the employees often have to fill out forms and reports about ongoing work or current results. It is part of the process to codify the gained knowledge. By doing so, the company constantly builds up a knowledge base with formalized information to specific tasks or problems. For future executions of processes and tasks, this knowledge base is then accessed, in order to learn from the previously entered data. This approach is called the *codified knowledge management strategy* or codification.

To make use of previously codified knowledge possible, strict formalisms have to be pursued and a strong focus on a precisely defined problem domain is necessary. The principle idea is to provide a rather fine-grained report structure, so that users know where to find the desired information, but also where to insert gained experience to make it accessible and usable in the future. This defines the strategy's goal in providing standardized scaffolds as it allows the organization to solve similar problems in different set ups and thus to specialize in a certain problem domain. According to that a competitive strategy could be the specialization on a certain target (e.g., process, domain). This mostly happens on higher levels of a company. Report-creation is usually responsibility-driven and also used as a communication channel to higher hierarchies. Many software companies make use of codification in project management, as it is a very crucial and complicated task. The processes and reports from previous projects act as guidance. In addition to the knowledge bases created for future projects to learn from, the project manager creates these reports to communicate results to higher hierarchies in the company.

From a KMS perspective it is obvious that all information of the same kind follows the same structure and is thus easily comparable in a software-based system. Spreadsheets or form-based applications with a database are common examples for software-based support. An application that handles data in forms and supports analysis company-wide is called an ES and is typical for the codifying strategy. It is applied in most companies nowadays to support them operationally by integrating data for use in the whole organization [14]. Employees then can insert and access data with different backgrounds in a formalized way. This allows them to search and filter the existing information according to the needs. Many ES have the features for using previously entered data to create an analysis or verification during the data inserts.

An example for a codifying strategy with KMS support is described in [15] where a database system represents the knowledge base. A specified data model allows the users to insert and query data.

2.2 Personalization

An oppositional knowledge management approach focuses directly on people and the concrete communication between them. Especially companies that follow the idea of flat hierarchies force a high degree of internal communication. Employees are encouraged to exchange ideas and experiences among each other. Thus a social network is continuously built and improved to help in experts finding and knowledge locating. This approach is called the *personalized knowledge management strategy* or personalization.

Companies that follow personalization usually embrace the difference of each project or customer to provide a specialized solution. The internal communication helps people to know about their colleagues' expertise (e.g., experience, knowledge, interest, etc.) and supports this flexibility. To sustain this activity the company has to provide a corporate culture that focuses on personal communication, but also provide circumstances to exchange knowledge. These include any kind of direct communication, e.g., in meetings. Examples for companies that follow this strategy are IT consultants providing development and management experts to several different customers of diverse domains. They can then use the corporate network to access first person knowledge about different topics and thus gain experiences themselves, which they share afterwards again.

Support for this strategy from KMSs is rather simple as the focus is not on the use of software-support. Instead the tools usually are those that help users to connect or communicate among each other. Ex-

amples are conversational services like e-mail or instant messaging. Users write messages to each other to explain problems and solutions or fix appointments to meet in person. An example is a mailing list that contains every employee of the company. Users with certain problems can then describe this, and ask anyone for contacting, if he or she has certain experience in that area. Since social software came up, other systems to support communication arose. The most spread one in professional area is probably the type of system referred to as a wiki. A wiki is a web application that allows users to create or edit content in form of articles, i.e., wiki pages. The strong side of many wikis is the search, as it helps with finding the wanted information [13]. Users create pages to discuss or share knowledge about certain topics; others then see that and, as the wiki is a kind of an archive as well, get to know about the expertise of people without knowing them. Additionally wikis support to communicate results to broader circles and also between people that do not otherwise have the chance to meet or get in closer contact.
Researchers analyzed the communication of people and units in organizations and came up with detailed graphs of social networks or knowledge networks. These show the communication lines and illustrate that gaps are difficult to overcome [16; 17]. A KMS like a wiki can help to close these gaps, but also to create new connections.

3 The case

The case company, Logica A/S, provides IT and business solutions with more than 40.000 employees worldwide. In Denmark about 800 IT and software specialists work in 5 different cities. Software is developed in projects of different areas (e.g., banking, governmental agencies, and the educational sector).
The company is process driven and mainly follows a codification strategy. After a joint analysis of the requirements for knowledge management and several discussions we were able to formulate different knowledge management problems, including the following ones:

- There is a large gap between the actual process in a real software project and the process-related knowledge. It is very difficult to create generally applicable processes and related knowledge that is applicable and beneficial for a practical context. Additionally, the contributing people have difficulties to align their knowledge about the various requirements of processes for a beneficial outcome.
- There are isolated islands of information. Different projects use different tools or media for gathering information, which makes it difficult to find information. The way and place of documentation may vary from project to project, as corporate guidelines are difficult to establish. If people from other areas want to access information, they would need to put effort in moving and transforming data, which is the reason, why accessing the information can thus become a very advanced task.

Both problem statements point to a lack of communication and thus typical reasons for a personalization strategy. It is also obvious that these problems occur stronger at lower organizational levels that are close to the actual development including their management than at the top management level, which makes heavy use of ESs and follows a codification strategy. Even though different parts from the company use different ESs, it is obligatory to keep it. The reason for this is very simple; every company has tools that fit their needs and usually employees are working with these for a reasonable time. Switching to new systems comes with high costs; people would have to be trained. Additionally each ES contains a lot of legacy data, which would have to be imported. In most cases a complete data import is very difficult or not feasible at all. So the company actually would lose knowledge and of course this is not acceptable. Thus is it vital to keep existing ESs.
The fact that software development faces problems that the top management level cannot see shows not only the dilemma between the two knowledge management strategies, but also their different needs in different areas in the company. While the top management level follows processes and already has specialized tools to support that, there is at the same time an obvious need for personalization in the development department. Any software process improvement (SPI) approach would obviously contain a mix of the two strategies. It would try to align the contrary needs in order to solve the problems as described above. From a researcher’s perspective, we want to find an SPI related way to improve the efficiency of KM.

4 Design

It is common practice to focus on one of the two knowledge management strategies and use the other strategy to support it. The recommended share is an 80-20 percentage [11]. We agree on that, but see the need for a different strategy at different organizational levels. The top management level has to follow the codification strategy, while it would be beneficial for the development level to follow a personalization strategy. A horizontal strategy split, that allows the two organizational parts to have different knowledge management strategies, would make that possible. However, it is also obvious, that a company must not be divided into separate parts without internal communication and knowledge exchange. To make this possible, we need a middle tier that is responsible for the communication and exchange between the different strategies (Figure 1).

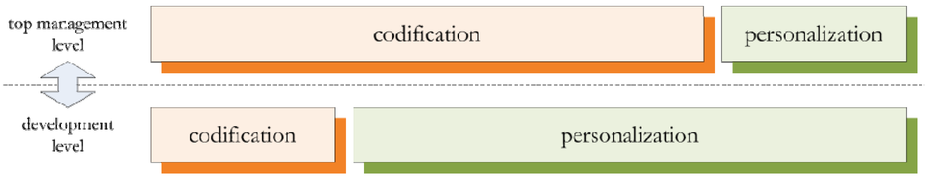


Figure 1: The two tier knowledge management strategy.

Hierarchically organized companies follow simple vertical communication patterns. The most common one is probably the installation of a project manager (PM). Besides all the crucial responsibilities of the PM, including resource planning and financial controlling, one of his tasks is to communicate the progress of the developers to the higher management levels. From this perspective the PM can be seen as the communication interface between those two areas. In its actual implementation, a person that fills this role has to create the plans and appraise the results of the development department according to it. It is among the PM's responsibilities to have an overview of the current situation and the way to reach the target. This contains more details than the report to the top management level, which also consists of information about the situation according to the processes, but not only in an abstract form. The PM collects and compiles data from the developers.
In this description it already becomes visible, that the PM acts as an interface to support the communication between different levels in form of a façade, the road of communication runs through this role. The top management level can work completely different organized way than the development level. Thus one level has not only the freedom to follow the knowledge management strategy that suits it best, but also the possibility to follow it, without influencing the other one directly. On the other hand this approach asks for high responsibilities of the PM, to maintain the communication.

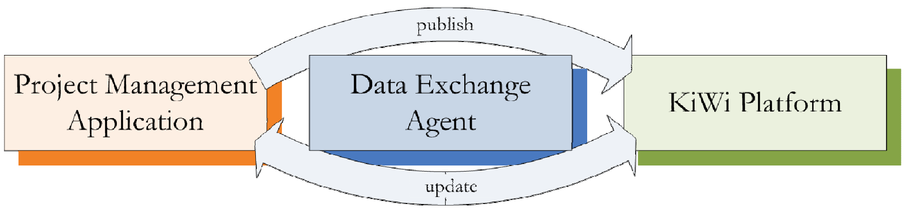


Figure 2: The KiWi Systems and the data exchange between them.

In the system we propose in this paper, this role is advanced by the possibility to interact between the personalization and codification knowledge management strategies. We created a KMS that combines applications for both directions and call it the KiWi[1] Systems (Figure 2). It combines three different applications, an ES, a wiki and a data exchange agent.
The ES, in our case a project management application, is a tool that provides multiple forms to collect precise data (Figure 3). It is aligned with the case company's processes. The PM of each project is responsible for entering the data, but the system is used to create a knowledge base for fulfilling processes and also to communicate the project's result to the top management level. None of the

[1] KiWi – Knowledge in a Wiki, is a larger EU-funded research project. The purpose of the project is to develop wiki-based knowledge management technologies for a wide range of application areas. See: www.kiwi-project.eu

developers has access to this, but instead work on the wiki. This is realized in the KiWi Platform (Figure 4), which combines many different technologies like social media, semantic web, reasoning, information extraction and recommendations [18; 19]. Not only users benefit from this combination of functionalities, but the whole knowledge management strategy does, as the system helps developers and managers in finding experts or information about a desired topic. Users can then collaborate through the KiWi. However, this is not restricted to the development level. Anyone from the company can use it to share information, communicate and cooperate. It is likely though, that the developers are the main users, as they claimed an actual need.

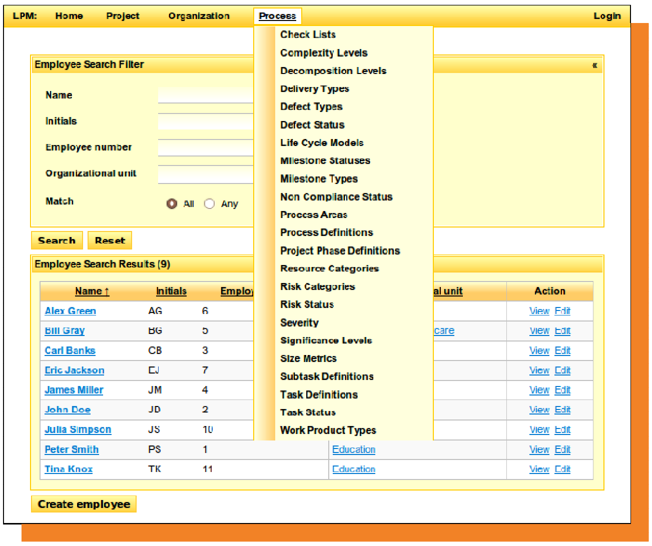


Figure 3: The Project Management Application (screen shot).

The data exchange agent is a rather small, but nonetheless it is an important application. The PM uses it, to publish data from the project management application to the KiWi and later to keep the two systems updated. We describe this in a simple process: A set up of the domain and the target data takes place in the ES, like the company's guidelines require. Then the PM chooses the entities, which might be interesting for the project team and where their collaboration could be beneficial. These entities are then published to the KiWi and thus accessible by the whole team. Whenever the PM finds it necessary, the data is updated in either of the systems. The ES and the KiWi are not synchronized; this provides the opportunity to continue the work on each system without interference. At the time of an update, the PM can choose the direction and the data to be moved.

Figure 4: The KiWi Platform (screen shot).

5 Discussion

The previously presented design allows partitioning of the organization vertically, in order to support the different needs at each level, with the two different knowledge management strategies. During our analysis we learned about two different problems that we will discuss now.

One problem described the difficulties that process managers have to connect to those that actually execute the processes. Even though they try to communicate to the developers regularly, a simple communication line, that is directly related to the process and can be used at the moment of its execution, is missing. But exactly that is the crucial information that would be important: What did the process executor think at the time he or she reads the document. Additionally developers usually do not know whom exactly to contact in case of problems with a process. Anyhow the feedback and the process have to be related manually, commonly through different media. With the introduced prototype a process description can be published in the wiki. Developers and managers would have to visit these pages during the execution of a process to make sure that nothing (e.g., process step, task, or detail) is left out. The current solution provides the process descriptions in documents or on intranet pages. None of these provides a commenting service, like the wiki, where every developer or manager can comment or edit to provide suggestions. The process manager can then discuss the issues in the wiki or respond directly to the reporting person. With this opportunity the communication is spurred and process managers get direct feedback.

The other problem described the difficulties that project managers and developers have to access knowledge that already exist in the company. Through organizational differences from project to project, knowledge and experience is hard to retrieve, and thus hidden or even lost. This is caused by different tools supporting basically the similar tasks, but as explained above a change of these tools is not an option. Our system solves this dilemma by not focusing on one ES but by being able to integrate with many. The system is designed to connect to an ES, either by accessing the database directly or connecting through web services. Also, the only direct connection to the ES is from the data exchange agent. The publishing to the KiWi is decoupled. This allows even multiple different ES to publish data to the wiki and make it generally available and thus directly useable for work in a different project. This strengthens not only the accessibility of data, as there is just one application to search for, but also the reusability. The data can stay untouched, but be directly related through links and so included to the work on a different project.

The system's design addresses directly the requirements pointed out by the case company: We keep the existing ES untouched, and insert an application in the IT infrastructure. This KMS is integrated at the level if data and it supports the communication between the developers and managers. Even more, the three subsystems allow the company to redefine their knowledge management strategy. The presented system supports a division of the company between the top management level and the development department with a PM as the interface between those. Introducing a data exchange agent makes this not only possible, but also realistic. The tasks for the PM change slightly from what they used to be in the traditional approach. However, the difference is actually rather small and the new combined KM strategy is less complex. According to the company's processes, the PM is responsible for the data in the ES. Traditionally, the PM has to collect this data through meetings and discussions with selected people and then enter these to the ES manually. With the system proposed in this paper, the data can be the direct result of collaboration within the project team. Every team member can review the knowledge contents and comment or edit it directly. The PM later chooses what to put to the ES and imports it directly. The data is thus the result of a wider base, with less need for time and resources. The described system also allows the PM to choose which data should be accessible by the developers. This leads to two advantages, first, developers can access data without the possibility to destroy it in the ES and second, the developers can benefit from data that was not readily available before.

Additionally, the KM benefits from this approach even more, as nothing really changes at the top management level, but the development department gets support where concrete need was expressed. Alternative designs have targeted a complete substitution of the legacy ESs by the described KiWi platform. An interface that allows adding and editing data through forms would be possible. But apart from the fact, that the company's ES must stay untouched, this would also soften the border between the top management level and the development department, i.e., between the two knowledge management strategies. The result would be dubious, as both strategies would not find a clean implementation and the system would end up being neither supporting personalization nor supporting codification. This is difficult to maintain, it is described that companies fail with this kind of mixture approaches

[11]. Hansen et al. state that one strategy should be in a leading role, supported by the other one and suggest an 80-20 percentage share. Our proposed system does that but at two organizational levels:

- The top management level follows a codification strategy (supported by personalization). This support is maintained through the wiki, which is also available to the top management level. They can access the wiki and search for information or expertise as needed. Additionally the wiki can be used for internal collaboration without influencing the development department. Support is also ensured through the PM. It is PM's responsibility to organize communication and manage the information flow between the two levels.
- The development level follows a personalization strategy (supported by codification). This support is mainly established through the PM, whose responsibility it is to comply with the company's processes and as such, inserting the correct data in the ES.

The description shows, that the proposed approach is mainly a strategic one. We agree with Kautz and Thaysen that software-based systems can support knowledge management, but not be the main driver [19]. This is what our system represents, an implementation to support a strategy.

6 Conclusion

The main advantage of the presented system compared to the current situation is the fact that nothing is taken away. With the exception of the PMs, every developer and manager can follow the same work steps as before. The development department gains additional support for communication, to approach discussed problems. Even in the case that the wiki does not improve the communication among the developers, the communication between the organizational levels is facilitated. This means that the system, even in the case of denial from the other developers and managers, would reduce some of the knowledge work of the PM, because the system supports the communication between the top management level and the development department. It shows that no bridges are burned, but new ones built. The system allows solving problems that were pointed out and it allows continuing the work as before.

The novelty in this approach is the distinction of different organizational levels in a company and a different knowledge management strategy for each. Hansen et al. reported that it is difficult to maintain a half-and-half approach of codification and personalization. They suggest choosing an 80/20 approach instead, where one strategy is the leading one and the other one supports this [11]. We propose to have a two-level strategy instead, which could be understood as a half-and-half approach, but it is not. Instead it is vertical integration of knowledge and knowledge management. The distinction between personalization in the development department and codification in the top management level is more precisely described as an 80/20 and 20/80 approach.

The gained features provide more flexibility, not only in the way the knowledge management strategies are applied, but also for the company's KM in total. As the distinction between the organizational areas allow specified knowledge management strategies, the overall KM benefits and can change in focus and scope compared to the traditional strategy.

Literature

1. *A Conceptual MAP of Software Process Improvement.* **Aaen, Ivan, et al.** 2001, Scandinavian Journal of Information Systems, Vol. Vol. 13, pp. 81-101.
2. **McFeeley, Bob.** *IDEAL: A User's Guide for Software Process Improvement.* Software Engineering Institute. Pittsburgh, PA : Carnegie Mellon University, 1996. p. 237, Tech. Report CMU/SEI-96-HB-001.
3. *Software Process Improvement in the Small.* **Ward, Robert P., Fayad, Mohamed Ebrahim and Laitinen, Mauri.** 4, New York, NY, USA : ACM, 2001, Communications of the ACM, Vol. 44, pp. 105-107. ISSN:0001-0782.
4. *Making Sense of Measurements for Small Organizations.* **Kautz, Karlheinz.** 2, 1999, IEEE Software, Vol. 16, pp. 14-20.
5. *Software process improvement in very small enterprises: does it pay off?* **Kautz, Karlheinz.** 4, 1998, Software Process: Improvement and Practice, Vol. 4, pp. 209-226.
6. *Situated assessment of problems in software development.* **Iversen, Jakob, Nielsen, Peter Axel and Norbjerg, Jacob.** 2, New York, NY, USA : ACM, 1999, SIGMIS Database, Vol. 30, pp. 66-81. ISSN:0095-0033.
7. *A Dynamic Theory of Organizational Knowledge Creation.* **Nonaka, Ikujiro.** 1, s.l. : INFORMS, 1994, Organization Science, Vol. 5, pp. 14-37.
8. *Software Process Improvement as Organizational Knowledge Creation: A Multiple Case Analysis.* **Arent, Jesper and Nørbjerg, Jacob.** s.l. : Society Press, 2000. Proceedings of the 33rd Hawaii International Conference on System Sciences.
9. *Managing knowledge in a software organization.* **Mathiassen, Lars and Pourkomeylian, Pouya.** No. 2, 2003, Journal of Knowledge Management, Vol. Vol. 7, pp. 63-80. ISSN 1367-3270.
10. *Review: Knowledge Management and Knowledge Management Systems: Conceptual Foundation and Research Issues.* **Alavi, Maryam and Leidner, Dorothy E.** 1, 2001, MIS Quarterly, Vol. 25, pp. 107-136.
11. *What's Your Strategy for Managing Knowledge?* **Hansen, Morten T., Nohria, Nitin and Tierney, Thomas.** Issue: 2, March-April 1999, Harward Business Review, Vol. Volume: 77, pp. 106-116.
12. *Knowledge management in software engineering.* **Rus, Ioana and Lindvall, Mikael.** Issue 3, May-June 2002, IEEE Software, Vol. Volume 19, pp. 26-38.
13. *Wiki: A Technology for conversational Knowledge Management and Group Collaboration.* **Wagner, Christian.** 2004, Communications of the Association for Information Systems, Vol. 13, pp. 265-289.
14. **Davenport, Thomas H.** Putting the Enterprise into the Enterprise System. *Harvard Business Review.* July-August 1998.
15. *Knowledge Management for Building Learning Software Organizations.* **Althoff, Klaus-Dieter, Bomarius, Frank and Tautz, Carsten.** 3-4, 2000, Information System Frontiers, Vol. 2, pp. 349-367. ISSN: 1387-3326.
16. *Knowledge Networks: Explaining Effective Knowledge Sharing in Multiunit Companies.* **Hansen, Morten T.** 3, s.l. : INFORMS, 2002, Organization Science, Vol. 13, pp. 232-248. ISSN:10477039.
17. *Social networks in software process improvement.* **Nielsen, Peter Axel and Tjørnehøj, Gitte.** 2009, Journal of Software Maintenance and Evolution: Research and Practice, Vol. 22, pp. 33-51.
18. *KiWi - A Platform for Semantic Social Software.* **Schaffert, Sebastian, et al.** s.l. : CEUR-WS, 2009. Fourth Workshop on Semantic Wikis. Vol. 464.
19. **Schaffert, Sebastian, et al.** *KiWi Vision (KiWi Project Deliverable D8.5).* s.l. : EU 7FP Project KiWi, 2008. Project Deliverable. Project number: ICT-2007.4.2-211932; Document number: ICT211932/SFRG/D8.5/D/PU/final.
20. *Knowledge, learning and IT support in a small software company.* **Kautz, Karlheiz and Thaysen, Kim.** 4, s.l. : MCB University Press, 2001, Journal of Knowledge Management, Vol. 5, pp. 349-357. ISSN:1367-3270.

Author CVs

Karsten Jahn

Karsten Jahn is PhD Student at the Department of Computer Science of Aalborg University in Denmark. Before that he graduated and worked as an IT-Consultant in Germany. His research interests are principally concerned with knowledge management and software processes. Karsten is part of the EU founded KiWi Project.

Peter Axel Nielsen

Peter Axel Nielsen is Professor Student at the Department of Computer Science of Aalborg University in Denmark. In his research he addresses systems development and how it may be improved. Recent research publications are on: business and software development, agile software development, influence processes in enterprise systems implementation, organisational culture in a software company. Among other research projects is he currently working with KiWi and DISIMIT.

The Diffusion of SPI in Larger Danish Software Companies

Andreas Munk-Madsen[1] & Peter Axel Nielsen[2]

[1]Metodica
metodica@post4.tele.dk

[2]Department of Computer Science, Aalborg University, Denmark
pan@cs.aau.dk

Abstract

In this study we survey medium-sized and large software companies in Denmark to understand the current diffusion of software process improvement (SPI). We investigate policies, knowledge, actions taken, and effects. The level of diffusion is characterized by all these factors. We present the most interesting results, compared to what has already been reported in the research literature. The statistics is primarily descriptive. The main result is that software companies show a large discrepancy between intentions and actions and hence also effects. Many companies want to do SPI but few invest the effort necessary to achieve the desired effect. The population is relatively small and though the sample is a large part of the population we are cautious in generalizing results.

Keywords

Software process improvement, diffusion, measurement.

1 Introduction

The practice of software process improvement (SPI) is as old as the idea of software methods that is more than 40 years. With the advent of the Capability Maturity Model [1-3] and a little later Software Process Improvement and Capability Determination [4] the concept of SPI took a more formal form. Although the formalized concept of SPI has been available for more than 20 years there have been remarkable few studies of the diffusion of SPI.

A review of literature searched in high-level outlets through Web of Knowledge and Google Scholar shows few but important research contributions. The Process Maturity Profile compiled by SEI [5] summarizes data from organizations voluntarily reporting their maturity levels. Reporting an appraised CMMI level does not necessarily imply an ongoing SPI practice, but it is a reasonable assumption that reporting organizations are interested in SPI. Since the report gives no information on the size of the population from which the reporting organizations could be seen as a sample it is not possible to deduce the percentage of software organizations practicing SPI. However, the report does give a minimal absolute number of organizations. Since the report has been produced regularly for seven years, it can describe trends. 3906 organizations have reported their appraisals by July 2009. 92% of these are on level 2-5. Of 3863 organizations reporting size data 85% have more than 26 employees in the area of the organization that was appraised. As the sample increases there is a trend towards more organizations being on level 3 and fewer on 5. Only a couple of Danish organizations have reported appraisals to SEI, and only one of these is in our population.

The research question we are pursuing in this paper is: How many software companies in Denmark are engaged in software process improvement and what is the effect?

The paper is structured as follows. The theories we build on are presented in section 2 together with our model of SPI diffusion. In section 3 we present the research method with a particular focus on determining the population and a reasonable sample. We then analyze the empirical data in section 4 and discuss the findings in section 5. With section 6 we conclude the paper.

2 Models and hypotheses

A study of diffusion of SPI needs to start in the existing literature. Many have studied success in SPI and they have developed conceptual models and instruments.

Wilson et al. [6] develop a quantitative measurement model based on in-depth group interviews in 7 UK software companies. They reach the conclusion that it is important for SPI success that: senior management is committed, SPI is staffed with highly respected people, initial processes are defined, and SPI is explained.

Rainer and Hall [7] survey a sample of 84 SPI managers representing a response rate of 8.4%. In their study they find factors impacting SPI success, e.g., internal leadership, inspections, executive support, and ownership of internal processes. In a follow-up study [8] they supplement the previous study with group interviews in 13 companies and identify 26 factors altogether that impacts SPI.

In another study of 13 UK software companies Baddoo & Hall analyze what motivates software developers and their managers to engage in SPI [9]. It is concluded that practitioners' ownership of processes is important and so is the evidence available on successfulness as well as the resources provided. An almost reverse study of de-motivators was conducted through 49 focus group interviews involving 200 practitioners in 13 UK companies [10]. This study shows that there are de-motivators specific to developers, project managers, and senior managers; e.g., de-motivators for project managers are: lack of measurement, fire-fighting, low process priority, and staff turnover.

Niazi et al. [11] conduct a more comprehensive study of SPI implementation. Seven factors are identified based on interviews with 34 SPI managers. The factors are: high management support, training, awareness, allocation of resources, staff involvement, experienced staff, and a defined implementation strategy.

Dybå [12] develop a measurement instrument to assess which factors are key in influencing or determining success in SPI programs. The instrument is validated in a study in Norway based on 120 respondents. On the same data he reaches the conclusion that small companies are just as effective as large companies and additionally that small companies gain higher performance measures than large companies [13]. A causal model for SPI success is developed alongside a set of hypotheses; these hypotheses are all confirmed with data from the survey of 120 Norwegian software companies [14].

In continuation of the existing literature on SPI success we present a model for SPI to clarify the scope of our study. Our model of a SPI-performing organization is illustrated in figure 1. It is basically an input-process-output model. The process is the SPI-related actions taken in the company. The output of the process is the organizational states that can be seen as effects of the actions. The input to the process is the situation of the organization including both contextual factors and internal factors such as the knowledge of the employees and the decisions and intentions of the organization expressed as policies.

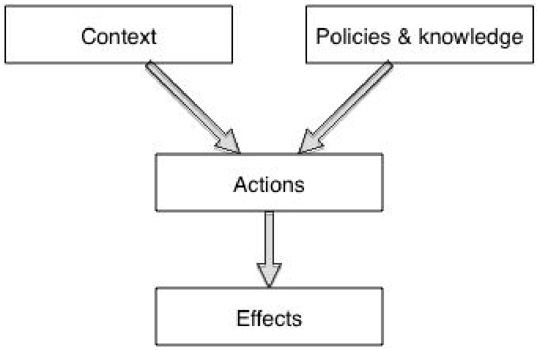


Figure 1: Model of SPI

The context variable is here described by a few factors like the companies' annual revenue, the number of software developers, and their level of formal education.

The policies and knowledge variable is described by the companies’ declared policies and the knowledge about and the attitudes towards SPI.

The actions variable is described by the investment in SPI, the involvement of software developers, the time spent on training and education, and who participated in SPI activities.

The effect variable is described by formal maturity assessments, indicators implied by maturity models like measuring projects, and the use of standards and procedures, as well as measures of reaching planned goals and increasing employee and customer satisfaction.

3 Research Method

To answer the research question we conduct a survey of Danish software companies using a simple questionnaire distributed to SPI managers. The survey method is appropriate because we want to measure the level of diffusion in the entire population and we want to measure the effects of SPI.

The ideal population is all Danish companies that meaningfully could do SPI. Company size matters for deliberate SPI activity. Annual revenue, number of employees, and total IS employees are among the most popular indicators of size in the information system literature [15]. We have used a mix of these indicators for practical reasons.

The primary criterion for including a company into the population was a substantial number of software developers. This limits the survey population and reduces the effort required in this exploratory study. Another criterion is nationality. The developers must be located in Denmark. Thus Danish subsidiaries of international companies may be included and Danish companies may be excluded if the software development is done in another country.

We have no knowledge of existing lists of companies stating the number of their software developers. However, we had available a fairly reliable list of 1400 Danish IT-related companies [16]. The criterion for inclusion in this list is annual revenue above 20 million DKK (1 DKK equals approximately 0.2 USD and 0.13 EUR).

From this list we selected the companies that were involved in software. To further limit the size of the population, we selected the companies that had more than 200 employees. This resulted in 73 companies to which we sent the questionnaire.

After some prompting a total of 35 questionnaires were returned. One of these indicated that the company did not want to be part of the study. Ten of the questionnaires indicated that the company had no or very few (below 20) software developers and one questionnaire indicated that the company had less than 200 employees in total. As these 11 companies do not belong to the population it is reduced to 62 companies. This leaves 23 (of the 62) companies that returned questionnaires useful for our study. The response rate is thus 37% at least.

A weakness of this selection procedure is that in-house software development departments in other trades than IT are excluded. However, we expect that are relatively few of those departments that have a substantial size. Another weakness is that some companies with fewer than 200 employees in total could still have a substantial number of software developers.

The design of the survey questionnaire is guided by three considerations. The expected answers should be represented by indicators for the variable, a person close to the SPI process should be able to answer the questions without much data collecting effort, and it should be possible to answer the questions unambiguously. The indicators for each variable are:

- Context: annual revenue, number of developers and their level of education
- Policies and knowledge: SPI policies, motives, and knowledge on SPI
- Actions: SPI investment, training days, SPI participation
- Effects: maturity measurement, development metrics, standards & procedures, employee and customer satisfaction

The questionnaire was pilot tested before it was distributed. A translation of the final questionnaire is available in [17].

The data analysis is descriptive. We do not intend to generalize to a level where the results are predictive. Partly because the theorizing of diffusion of SPI is still limited. Partly because the response rate and also the absolute number of responses do not warrant further generalization.

4 Analysis

To report our findings we have calculated the answer frequencies for each question. Correlations have been calculated where the model suggests interesting propositions and when a correlation is both interesting and significant it is reported. The reporting follows the structuring in four variables from figure 1.

4.1 Context

Annual revenue range from 50 to more than 5.000 M DKK. Number of developers range from 20 to 2.000. In more than half of the companies the majority of the developers have at least 5 years education, corresponding to a master degree. Only in 2 of 20 companies the majority of the developers do not have a 3-years education, corresponding to a bachelor degree.

4.2 Policies and knowledge

Most of the companies, 19 of 23, agree to a high degree or a very high degree that they have a policy or strategy containing the statement “SPI is important for the efficiency and quality of our company.” Relatively fewer of the companies, 7 of 21, agree to a high degree or a very high degree that they have a policy or strategy containing the statement “Our customers demand that we have SPI.”

Thus, productivity and quality are considered important motivators by 83% of all respondents while only 33% are engaged in SPI because of expectations in their market or direct required by their customers. These two policies are not correlated.

The respondents' knowledge about SPI is reflected in the following table.

Question: To what degree do you as a manager agree to these statements?	Very high degree	High degree	Some degree	Less degree	Not at all	N/A
Know too little about SPI and maturity models to begin improvements			3	7	13	
Decided that SPI is not relevant for them	1	1	2	3	16	
Know enough about SPI and maturity models and would like to do improvement but other tasks do not allow the time for it		4	4	7	8	
Find SPI relevant but lack management support	1		2	9	11	
Find SPI relevant but lack funding		1	4	7	11	
SPI is not a good idea because processes are cumbersome			5	5	13	
SPI is not a good idea because evidence is lacking			2	4	16	1
SPI is not a good idea because of negative experiences			1	5	16	1

The statements are by large capturing de-motivators. Overall, the respondents disagree with all these statements. The only exception is that one third of the respondents agree that “... other tasks do not allow the time for” SPI.

4.3 Actions

Of the 15 respondents who knew the size of the company's investment in SPI, 6 answered it was 2 per cent or less of the revenue from software or systems development. 5 answered the investment was 4 per cent of the revenue or more.

Investment in SPI	1%	2%	3%	4%	5%	>5%
No. respondents	3	3	4	2	2	1

Ten respondents of 21 answered that the number of annual training days per developer was between 5 and 10. 9 respondents answered that the number was below this interval.

Annual training days per developer	0-2 days	3-5 days	5-10 days	10+ days	N/A
No. respondents	3	6	10	1	1

Sixteen respondents of 21 answered that less than 25 per cent of all systems developers were involved in SPI activities. The level of investment correlates well with the number of training days invested per developer (R = 0.51). Investment in SPI cannot be said to cause the training days nor vice versa, and they are simply correlated. Both variables can thus be said to show the level of management commitment to SPI.

Percentage of developers involved in SPI	0-25%	26-50%	51-75%	76-100%	N/A
No. respondents	16		2	3	1

On the question of who are involved in SPI the respondents answer as in the table below.

	Very high degree	High degree	Some degree	Less degree	Not at all	N/A
Developers	2	5	10	5		
Project managers	4	10	1	6		1
Middle managers	1	10	9	2		
Top managers	8	8				5
SPI managers	9	8				5
SPI workers	8	8				5

It seems that SPI people are highly involved, and so are top managers. Middle managers and project managers are involved to a lesser degree; and developers even less.

4.4 Effects

Nine of 23 respondents have within the last 3 years measured their maturity against a maturity model. Eight of these 9 measured against CMMI. Of these were one at level 5, two at level 3, and one at level 2. Most maturity measurements, a total of 6 out of 9 (66%), were performed by third parties.

Only six respondents agree to a high degree that the total SPI initiative has had the planned effect.

Half of the respondents measure employee and customer satisfaction. Only 3-4 have seen increases in satisfaction.

Almost all companies have standards or procedures for producing and using requirement specifications and project plans.

Most of the respondents register time spent on development as well as quality factors (e.g., errors or customer complaints). But only one third registers productivity, publishes metrics, and uses metrics in management.

Very few answers are correlated. The correlation between the degree to which the companies produce statistics on time, quality and productivity and the degree to which they use statistics on project data in project management has a coefficient of 82% with significance level $p < 0.0001$. The correlation between the degree to which they measure time, quality, and productivity and the degree to which they use statistics on these data to manage projects has a coefficient of 49% at a significance level of $p < 0.05$.

Generally, the effect indicators are not correlated.

5 Discussion

From the analysis several issues of SPI diffusion in Danish software companies emerge. In this section we discuss four issues and the limitations of the study.

5.1 Diffusion

The diffusion of software process improvement is a pertinent issue as with all new and maturing technologies [18; 19]. Previous research on measuring SPI activities and success has developed success factors [6; 9; 10; 20], theoretical models and measurements instruments [12-14; 21]. These studies show what leads to success or what causes success in SPI. With similar measurement instruments we are trying in this paper to investigate how widespread SPI is in software companies, i.e., how far software companies have come in diffusing SPI.

We had an initial working hypothesis that there would be a large diffusion of SPI among the large software companies that we studied. Two aspects were expected to be present to a large extent, namely, both the diffusion of maturity assessment and the diffusion of SPI policies among companies. Surprisingly, a little less than half the sample (39%) and hence also a little less than half the software companies in the population have had their software process maturity measured. This should also be seen in the light of how many companies have a SPI policy. Here 83% have a policy that SPI is important for efficiency and quality reasons while only 33% have a policy that SPI is important because customers in their market have demanded it. There is no statistical correlation between those 39% companies that have their software processes maturity assessed and those 83% that are engaged in SPI for efficiency and quality. Nor is there correlation between the companies with assessed maturity and those engaged in SPI because the customers in their market have required it. This leads us to the following proposition.

P1: Despite a large degree of diffusion of SPI in the form of company policies the diffusion of SPI through maturity measurement is clearly lower.

With the high level of diffusion of SPI policies it is surprising how low the diffusion of maturity assessment is.

5.2 Motivation

The presence of SPI policies is the first level of how motivated the companies are to engage in SPI. The presence of SPI policies suggests that the companies are primarily engaged in SPI for their own reasons and not really to please their customers directly (83% on efficiency and quality against 33% on demanded by the customers).

At a deeper level we have analyzed how the companies relate to knowledge and hence also how they relate to motivating reasons. Four companies report that SPI is not relevant for them. Four companies report that they lack time, management support or funding. Very few report that SPI is not a good idea. Taken together there is really no significant pattern in the data that suggests lack of motivation. The only partial explanation would be that SPI as a new and emerging technology is competing for management attention among many other new technologies and also with business issues in software companies. Hence we conjecture that software managers do not perceive SPI as a primary lever for improving the business of software.

P2: Internal motivators are more frequent than external motivators

Dybå [14] showed that business orientation (i.e., the extent to which SPI goals and actions are aligned with business goals) positively influence SPI success. Our study cannot really confirm Dybå's conclusion and rather suggests the opposite that SPI is more a technology for managing software development than a technology to improve business.

5.3 SPI efforts

The low degree of diffusion is explicable when seen in the light of how much effort the companies spend on SPI activity. Fifteen companies of 23 knew how much they invested in SPI. This is a low number considered that those answering the questionnaire are SPI managers or otherwise responsible for the companies' SPI activities. Of these 15 most (66%) spend 1-3% of the revenue for software development and median as well as average are less than 3%. Those 39% companies that assess software processes maturity are spending on average 1% less than those companies that do not assess maturity. A single company spends more than 5% of their software development revenue on SPI; and this company also happens to be involving all software developers in SPI and it offers 5-10 training days per developer.

In addition, most software developers are not involved in SPI activity as 69% of companies only involve less than 25% of their developers. This leads us to proposition P3.

P3: Effort does not match intention

Dybå [14] found that involved leadership causes SPI success. The leadership commitment to invest in SPI and manage effort is quite low. This can thus be a partial explanation why the diffusion of SPI in Danish software companies is also low.

5.4 Effects

Taken together the measured effects of SPI activity are low. Few perceive an increase in employee satisfaction or customer satisfaction hence SPI has not had these effects. There are traces of more standards and procedures for requirements specification and project planning. These are mere means to achieve the ends of better software quality and productivity and contribute little to effects. We thus conjecture P4.

P4: Effects are limited

5.5 Limitations

The limitations are that we have: (1) no respondents from in-house software development; (2) no respondents from small companies; and (3) few respondents from the investigated population.

The limitations of this study are significant, but manageable. We have here intended to report from an initial study in order to form propositions that can be applied in follow-up studies. We find that we have achieved this despite the low number of respondents. The low number of respondents influences significantly the power of the statistics and we have been careful to use descriptive statistics and not claim significant correlations about causality in the underlying measurement model. Further studies are manageable because we now know in much more detail what we are looking for and it will be feasible to begin creating time series of survey data. We now have a database of SPI managers in large Danish software companies that can be activated again to answer a similar yet improved questionnaire.

6 Conclusion

We have in this paper reported from a survey of large Danish software companies. Our declared interest is to know the level of diffusion of SPI in these companies. We have used and extended the existing models and measurement instruments to this end.

Our study is limited in size, but we have identified a few propositions that we believe are questioning several core assumptions in SPI theory. The diffusion of SPI in large Danish software companies is generally low and the diffusion of maturity assessments is particularly low. If SPI theories, models, and activities are good and powerful ideas why is the diffusion then so low? If SPI theory, models and activities are the right means to improve software development why are there then so few effects?

This may be partly explained by the low degree of external motivators found and by the low level of effort invested in SPI. The findings can be summarized in the four propositions:

P1: Despite a large degree of diffusion of SPI in the form of company policies the diffusion of SPI through maturity assessments is clearly lower.

P2: Internal motivators are more frequent than external motivators.

P3: Effort does not match intention.

P4: Effects are limited.

We will use this study to establish a consecutive survey where the same measures can be repeated to achieve time series data.

Literature

1. W. Humphrey, *Managing the Software Process,* Addison-Wesley Publishing Company, Reading, Massachusetts, 1989.

2. M. C. Paulk, B. Curtis, M. B. Chrissis, and C. Weber, “Capability Maturity Model for Software ver. 1.1,” 1993, Software Enginering Institute,

3. B. Curtis, “The global pursuit of process maturity,” *IEEE Software*, 17, 2000, pp. 76-78.

4. K. El Emam, J. N. Drouin, and W. Melo, *SPICE: the theory and practice of software process improvement and capability determination,* IEEE Computer Society Press, 1998.

5. “Process Maturity Profile,” 2009, Software Engineering Institute,

6. D. N. Wilson, T. Hall, and N. Baddoo, “A framework for evaluation and prediction of software process improvement success,” *Journal of Systems and Software*, 59, 2001, pp. 135-142.

7. A. Rainer, and T. Hall, “Key success factors for implementing software process improvement: a maturity-based analysis,” *Journal of Systems and Software*, 62, 2002, pp. 71-84.

8. A. Rainer, and T. Hall, “A quantitative and qualitative analysis of factors affecting software processes,” *The Journal of Systems & Software*, 66, 2003, pp. 7-21.

9. N. Baddoo, and T. Hall, “Motivators of Software Process Improvement: an analysis of practitioners' views,” *The Journal of Systems & Software*, 62, 2002, Elsevier, pp. 85-96.

10. N. Baddoo, and T. Hall, “De-motivators for software process improvement: an analysis of practitioners’ views,” *The Journal of Systems & Software*, 66, 2003, Elsevier, pp. 23–33.

11. M. Niazi, D. Wilson, and B. Zowghi, “Critical success factors for software process improvement implementation: an empirical study,” *Software Process: Improvement and Practice*, 11, 2006,

12. T. Dybå, “An instrument for measuring the key factors of success in software process improvement,” *Empirical Software Engineering*, 5, 2000, Springer, pp. 357-390.

13. T. Dybå, “Factors of software process improvement success in small and large organizations: an empirical study in the Scandinavian context,” *ACM SIGSOFT Software Engineering Notes*, 28, 2003, ACM New York, NY, USA, pp. 148-157.

14. T. Dybå, “An empirical investigation of the key factors for success in software process improvement,” *IEEE Transactions on Software Engineering*, 31, 2005, pp. 410–424.

15. S. Goode, and S. Gregor, “Rethinking organisational size in IS research: meaning, measurement and redevelopment,” *European Journal of Information Systems*, 18, 2009, pp. 4-25.

16. “Brancheguiden,” *Computerworld*, Supplement, 2008, Computerworld,

17. A. Munk-Madsen, and P. A. Nielsen, “Questionnaire for Software Process Improvement Diffusion,” 2009,

18. R. G. Fichman, and C. F. Kemerer, “The Illusory Diffusion of Innovation: An examination of assimilation gaps,” *Information Systems Research*, 10, 1999, pp. 255-275.

19. R. G. Fichman, and C. F. Kemerer, “The assimilation of software process innovations: An organizational learning perspective,” *Management Science*, 43, 1997, INFORMS, pp. 1345–1363.

20. T. Hall, A. Rainer, and N. Baddoo, “Implementing software process improvement: an empirical study,” *Software Process: Improvement and Practice*, 7, 2002, pp. 3-15.

21. T. Dybå, “Enabling Software Process Improvement: An Investigation of the Importance of Organizational Issues,” *Empirical Software Engineering*, 7, 2002, Springer, pp. 387-390.

Author CVs

Andreas Munk-Madsen

Andreas Munk-Madsen received the M.Sc. and the Ph.D. degrees in computer science from Aarhus University, Denmark.

He is a founder of Metodica, a company specialized in consulting and training in system development. He has worked as a developer, researcher, teacher, and consultant in system development. He is the author of *Strategic Project Management* and co-author of *Professional Systems Development* and *Object-Oriented Analysis & Design*. His research interests are focused on project management and system analysis.

Peter Axel Nielsen

Peter Axel Nielsen received the M.Sc. degree in computer science from Århus University, Denmark, and the Ph.D. degree information systems from Lancaster University, UK.

He is currently Professor of information systems at Aalborg University, Denmark. He is also a Visiting Research Professor with Agder University, Norway.His research interests include action research on systems development practice and development of object-oriented methodologies, software engineering and social and organizational aspects of systems development. He is the co-author of Object-Oriented Analysis & Design and Improving Software Organizations: From Principles to Practice. Currently, he is engaged in research on software process improvement.

Enforce Business Responsibility for Software Quality – The Business Case Gap for Software Testing

Dr. Kai-Uwe Gawlik, Tomas Schweigert
SQS Software Quality Systems AG, Stollwerckstr. 11, D-51149 Köln
kai-uwe.gawlik@sqs.de

Abstract

Looking at the business case of software testing a set of costs has to be considered: The costs of error prevention (process improvement), the costs of error finding (testing & QA), the costs of error correction and the costs of error occurring in operation. There are lots of examples of high costs of errors in the technical sector e.g. the Ariane 5 disaster. It is hard to calculate a business case for systematic and automated software testing in practice. When you look at the cost figure, you can give an overall figure for process improvement, you can give a precise figure for software testing, you can estimate the costs of error correction, but very seldom you have any idea of the costs or intangible effects of an error occurring during the operation of the Software in the business field. One reason is that controlling can't match costs of an occurring error to its normal cost structure. The other is that description of an error contains technical data and classification of errors follows these data. The paper shows that much more effort for software testing might be spent if error classification reflects the business impact of an error occurrence. Additionally, the big amount of errors having low severity will become more transparent.

1 A look on some famous software disasters

Technical Desasters

1985-1987 Therac 25 (A radio therapy system)

Death injuries caused by a software Error.

1996 The Ariane 5 Disaster

The first flight of the Ariane 5 failed caused by a software error. Costs of the Ariane 5 program until this failure: app 7 Billion USD. Cost of the lost satellite: 500 Million USD.

1999: Loss of Mars Climate Orbiter

2004: Recall of Siemens mobile phones

Financial Disasters

1990 AT&T Network break down

The AT&T phone network was down for one day. Damage: 1 Billion USD

1992: Sabre

The integration of the SABRE reservation system with other systems failed. Cost: app 165 Million USD.

2004: FISCUS (Tax Authorities)

The German FISCUS project was cancelled after 13 years of development. Cost 900 Million Euro.

Results up to now

Seeing the high risk related to software quality assurance might be expected to be seen as a key success factor for all software related or software dependent industry. Risk justifies investments in highly mature mostly automated testing processes. In reality we often find organizations treating QA and testing as necessary evil. Such organisations often spent only unavoidable costs for these activities performing mostly intuitive ad hoc testing of software.

Why do we find this situation? Most organization do not run rocket constructing projects. They are engaged in commercial day to day business fighting unnecessary costs to stay competitive. But if their competiveness is based on highly available systems with reliable software quality why don't they invest in QA and testing?

2 The structure of cost and effort

To understand the information gap, we should have look into the cost structure of processes and errors.

The typical process cost structure

Trying to get a rough structure of the process cost we find 4 categories:

- Value adding performance (What everybody wants to get out of a process.)
- Support performance (This includes Management, QM, QA, Testing, Software Change and Configuration Management.)
- Idle performance / busy work (Unnecessary activities that lower the output value)
- Value reducing performance (Errors, failures)

As we can see, the support performance efforts are the critical key. If you spend too little, idle and value reducing performance will start to overwhelm and damage the overall performance. But if an organization invests too much in support performance, cost consuming over engineering will take place. The problem is that immature organizations can't calculate or estimate the cost of idle and value reducing performance. But what they can measure is the cost of support performance. As a consequence immature organisations tend to cut down these costs.

The structure of error costs.

When we look at the structure of errors, we see the cost of errors being separated into

- Error prevention costs (e.g. process improvement)

- Error detection costs (e.g. testing)
- Error correction costs (e.g. localization, removal)
- Error occurrence costs (the Adriane V)

While the cost of error prevention and the cost of error detection are always transparent (because they belong to the support performance costs), many organizations have no figures for error correction costs and nearly no organisation has figures for error occurrence costs (because these costs belong to the idle or value reducing performance costs).

Consequence

To justify the costs of better testing by using error and process cost figures, organisations have to improve their measurement. There are 2 potential solutions:

- Improve the accounting system making these costs become transparent
- Improve the error description

To link the issue to the accounting system is – from a pragmatic perspective – for many organisations not a real good idea. Their accounting system is designed to deliver business figures not for storing technical information.

Consequently, the preferred way should be to improve the error description.

3 A look on the typical description of Errors

Today errors are typically collected using tools entering a description and setting a number of attributes. The idea is to get an overview about the current state of the total error situation and to follow and solve the error throughout the error life cycle. Some of the most important attributes are severity and priority giving information about the damage connected with an error and about the order of error fixing, respectively. Since attributes and values are configurable one has to be aware of the correct usage of these attributes. Sometimes priority is having the information belonging to severity.

Severity

Severity is typically described qualitatively. An example could be:

- Showstopper
- High
- Medium
- Low

All people using the database entering or evaluating the error data base should have the same understanding about the semantic of any value. Using different databases for testing and production (incident management) could be based on different values and different meaning. Very often errors get a very subjective severity not having a common understanding. Showstopper for example could mean an error not allowing testing software in a distinct test environment. Thus, using Showstopper as container for the most important errors having to be fixed urgently is another example of wrong usage of an error data base. In this case severity is mixed with the meaning of priority.

Priority

Priority is typically described qualitatively too. Often you will find:

- High
- Medium
- Low

Whilst severity is a constant priority is variable depending on the current error situation. The requirement of having the same understanding and consequences is the same as for severity. Priority allows putting errors into project or maintenance cycles to be fixed. Very often the enterer does not have any understanding about consequences connected with a distinct value of priority.

4 An SQS Research Survey on Error Cost

SQS has plants in Europe, Middle East, South Africa, and Asia with approx. 1500 Test Consultants. We made a survey asking our consultants to give a statement if our customers know the cost of error correction and the business impact of errors occurred during production.

As a result we got a very different picture: In the technical sector (e.g. automotive aerospace, medicine) up to 90% of the companies measure the cost of error correction and the cost of error occurring during usage. But only 10% of our customers in the finance, trade, or logistics area measure the cost of error correction and / or the cost of errors occurring during usage.

5 Consequences of this Situation

When we calculate the business case of test and quality assurance we have to compare the cost of error prevention and error detection on one hand with the cost of error correction and error caused damages on the other hand. If these costs are not measured and no data for proper estimation is available it will be hard to give a correct recommendation how to handle testing and process improvement.

If – for instance – we see the following error statistic (example of errors in production- example is based on the figures of a real SQS customer)

A	Critical	511
B	High	1286
C	Medium	631
D	Low	197

it will not be possible – even if we also know that 60% of all errors occurred during usage in production – to give any advice to improve software testing. If – for instance – the necessary investment to lower the error in production rate from 60% down to 10 % is 3 Million Euros, you can't say this is a sustainable business case or a waste of time and money. Even if the technical description of each error is equal, the difference is made by cost. If a critical error costs 1000 Euros improvement of test is a waste of money if a critical error costs 1 Million Euros it is a valid business case.

6 How the business can solve the problem

As we know, the business suffers from errors occurring during usage. It is the business that can measure or estimate the related costs. So the business has to provide the cost information. To do this we have to change the severity classes in the error management. So we don't rate errors in production any longer due to their technical aspects but to the business impact. One recommendation is the following scheme:

Severity A:	Euro 100.000 and more
Severity B:	Euro 50.000 to 99.999
Severity C:	Euro 20.000 to 49.999
Severity D:	Euro 5.000 to 19.999
Severity E:	Euro 500 to 4.999

By categorizing an error the business has to provide trustworthy data that supports the chosen category. Consequently, the upper scheme is depending on the type of application and the underlying business cases.

Obviously, the return on invest for e.g. improving the development process or the test process by automation can be calculated by cost reduction using the costs of the upper scheme. Improvement decision for companies having mostly A errors based on quantitative costs will be made more easily whilst companies with a high rate of E errors and an insignificant number of A errors will not have reasons to change anything in the software development process. The key for the return on investment calculation is the combination of an error profile and the categorization scheme (using mean values in the lower example):

511	Severity A (100.000 Euro per error)	-	51.100.000	Euro
1268	Severity B (75.000 Euro per error)	-	95.100.000	Euro
631	Severity C (35.000 Euro per error)	-	22.085.000	Euro
197	Severity D (12.000 Euro per error)	-	2.364.000	Euro
Total: 2625 Errors		-	170 Mio	Euro

In total this example gives a potential of about 170 Million Euro of cost savings. The detailed knowledge about error cost based on the business gives arguments for investment.

Additionally, the information of error occurrence should be classified giving information about the error costs per time:

- Costs in every online transaction
- Costs once per day
- Costs once per week
- Costs once per quarter
- Cost once per year

To help to setup the right priority and to get information about time until the bug has to be fixed the date of next occurrence should be given. This information helps to prioritize online and batch transactions the right way and put fixing into the adequate release.

Our hypothesis is that in companies that are not committed to invest in software testing the number of severity A errors will decrease rapidly. Even the same will happen to severity B, C, and D errors.

When we ask why the management refused to spend in systematic and automated testing, we might come to the conclusion that we can't easily transfer technical to business severity. By re estimating the business severity the following figure -based on the example above- might support the management decision:

0	Severity A (100.000 Euro per error)	-	0	Euro
5	Severity B (75.000 Euro per error)	-	375.000	Euro
10	Severity C (35.000 Euro per error)	-	350.000	Euro
20	Severity D (12.000 Euro per error)	-	240.000	Euro
2590	Severity E (1.000 Euro per error)	-	2.590.000	Euro
Total: 2625 Errors		-	3,5 Mio	Euro

By total damages of 3.5 Mio Euro, it is hard to belive that 2 Million investment in systematic and automated testing might create a business case.

One year after the introduction of this new categorization scheme, we think we find several companies with 100% Severity E errors. These companies do not need to invest in software testing. On the other hand if we find a relevant number of companies which now know that they need to invest in software quality.

Additionally the cost effect of errors with low severity is transparent. The fixing of a large number of C errors will have the same effect like one A error. The collection of a large number of errors with low severity is avoided. The effect of using severity as priority is avoided because a distinct semantic is given to any value.

7 Summary

Many organisations are not able to justify the costs for better testing and QA because they have no or insufficient information over the cost of error. To solve this issue, an improvement in the error description by enriching the error description with cost information is necessary. In a short period after implementing this improvement organisations are able to decide if better testing is a business case or a waste of money.

References and further reading

1 Barry Boehm, Tom de Marco, Software Risk Management, http://www.daschmelzer.com/spm/b4.pdf

2 Barry Boehm, Risk Management Principles and Practices, IEEE Software volume 8, Nr. 1, 1991, s. 38-41.

3 Gitzel, Ralf; Krug, Simone; Schader, Martin, Software Error Impact and Cost Analysis in: APSEC 2008 Workshop Proceedings - SPACE 2008 Workshop, 2 December 2008, Beijing, China, pp. 45-51

4 Xuemei Zhang,, Hoang Pham, A software cost model with warranty cost, error removal times and risk costs, IIE Transactions, Vol. 30, Nr. 12, 1998, S. 1135 - 1142.

5 The Econimoc Impacts of Inadequate Infrastructure for Software Testing –Final Report- National Institute of Standards and Technology, 2002

Authors CV

Tomas Schweigert (Ass. Jur.)

Principal Consultant at SQS Software Quality Systems AG, Tomas Schweigert has a long experience with software quality management, software testing and process improvement, staying with SQS from 1991. He started his PI work with the BOOTSTRAP institute, working as a BOOTSTRAP lead assessor in several assessment projects. His special interest is the analysis of projects in crisis situations. Mr. Schweigert is now a SPICE principal assessor (INTACS scheme) his current research topics are the SPI Manager qualification and the test assessment approach spice4test.

Dr. Kai-Uwe Gawlik (Dipl. Phys.)

Head of Application Intelligence (AI) at SQS Software Quality Systems AG, Kai-Uwe Gawlik is experienced in software quality management, software testing and improvement of technical quality (architecture, code, and data), working at SQS since 1997. His special interest is the analysis of software products to enhance efficiency of development and maintenance.

I⁵P – A Framework for Aligning Process Maturity with Knowledge Sharing

Elli Georgiadou[1], Kerstin V. Siakas[2], Bo Balstrup[3]

[1] University of Middlesex, School of Computing Science, Software Forensics Centre, Trent Park Campus, Bramley Road, London N14 4YZ, UK

[2] Alexander Technological Educational Institute of Thessaloniki, Department of Informatics, P.O. Box 141, GR-57400 Thessaloniki, Greece

[3] Center for Software Innovation, Alsion 2, DK-6400 Sonderborg, Denmark

Email: e.georgiadou@mdx.ac.uk, siaka@it.teithe.gr, bb@cfsi.dk

Abstract

In this paper we argue that Knowledge Management (KM) and in particular Knowledge Sharing (KS) is strongly linked to organisational maturity. In this investigation we study the mechanisms that enable this upward movement and depict measurable effects of performance as the organisation climbs from the ad hoc levels to institutionalised high levels of maturity. We propose the I5P visualisation framework which aligns a Knowledge Sharing level to the appropriate maturity level and characterises the process from incidental to innovative. This framework provides the basis, in terms of preparedness and disposition towards knowledge sharing, for estimating and measuring organisational performance.

Keywords

Capability maturity, knowledge sharing, process models, visualisation

1 Introduction

It is widely recognised that people are any organisation's greatest asset. The employees' knowledge and experience is one of the most valuable assets of businesses and an important competitive factor. Hence, an organisation must ensure that the knowledge base of its employees is captured and shared throughout the organisation. Capturing, preserving and sharing knowledge prevents duplication of effort, and ensures that tacit knowledge is not lost when employees leave the organisation. Knowledge Management can be technology oriented i.e. concentrating on the technology infrastructure and the ways in which explicit knowledge can be codified, stored and interrogated, or people-oriented which emphasises the importance of tacit knowledge, the social infrastructure and the business performance. Research concerning the factors affecting knowledge sharing (Nonaka, 1991; Siakas and Georgiadou, 2008) has identified a number of different factors that can be broadly classified as hard (technologies and tools, such as computer mediated communication) and soft (relationships between the individual and the team, department or organisation including motivation, and organisational culture). As the organisation grows the maturity of its processes needs to grow in order for the organisation to be competitive. Knowledge also evolves continuously as the individual and the organisation adapt to changes and influences from the external and the internal environment. KM and KS therefore is moulded by the organisational maturity and also is expected to grow and evolve provided a KM culture and environment exists.

2 Current research and practice

The Capability Maturity Model (CMM) has been used as a benchmark by some practitioners and researchers to model the Knowledge Maturity processes.

For example Elms and Langen (2002), practitioners at Siemens AG/Corporate Technology, proposed the KMMM methodology, which provided a Maturity Model (of five levels reflecting the 5 levels of the CMM), and a 6-step process model which they applied in the case of audits. Their process model emphasises the need for transparency and recommends (in the case of audits) the use of pair auditing (presumably borrowed from the agile concept of pair programming). Although KMMM provides visual representations of Key Areas of Knowledge Management it is unclear how the quantitative results and the resulting maturity profile were '*condensed*'. It is questionable whether the measurements were objective, and the authors themselves recognise that the interventional nature of their investigation may have coloured their conclusions.

Kulkarni and Louis (2003) *'borrowed CMM's framework and applied it at a broad level to define a KM maturity model… for benchmarking knowledge management maturity within an organisation*'. They looked at organisational goals and differentiated between perceptual and infrastructure assessment as shown in table 1. The differentiation between the perceptual characteristics of the process and the availability of technology is useful in that the often mistaken assumption that systems and associated technologies do not guarantee successful KM and Knowledge sharing. They identified among other limitations of their study the fact that the scheme of mapping their survey results into an overall 'crisp' maturity level is not well defined. Also the study suffers from the deficiency of all case study based research i.e. the confidence with which results can be generalised. However, Gallagher and Hazlett (2000) proposed a more refined framework called the Knowledge Management formula (KMf) where the overlaps and synergies of Knowledge infrastructure (Ki), Knowledge culture (Kc) and Knowledge technology (Kt) were identified. Gallagher and Hazlett used their Knowledge Management Maturity Model (KM3) as an evaluation tool.

Table 1: General Maturity Levels (source Kulkarni and Louis (2003))

Maturity Level	**Goals**	
	Perceptual Assessment	**Infrastructure Assessment**
1: Possible	Not discouraged; there is a general willingness to share; some people who understand the value of it, do it	Knowledge Assets are identified
2: Encouraged	Value of knowledge assets is recognised by the organisation; culture encourages all activities with respect to sharing of knowledge assets; sharing is recognised / rewarded	Knowledge assets are stored in some fashion
3: Enabled / Practised	Sharing of knowledge assets is practice; KM-related activities are a required part of normal workflow	Systematic mechanisms exist to enable activities with respect to KM; centralised repositories exist; a taxonomy exists
4: Managed	Employees find it easy to share knowledge assets; employees expect to be successful in locating knowledge assets	Training instruction is available for learning about KM systems usage; change management principles are

	if they exists; tools for supporting KM activities are easy to use	used to introduce KM practices
5: Continuously Improved	Mechanisms and tools to leverage knowledge assets are widely accepted	Intelligent tools exist; tools and mechanisms for sharing are periodically improved / updated; business processes that incorporate sharing of knowledge assets are periodically reviewed

Mohanty and Chand (2004) evolved the 5iKM3 KM Maturity Model for assessing and harnessing the organisational ability to manage knowledge. Number 5 (in 5iKM3) reflects the 5 CMM levels whilst number 3 (in 5iKM3) refers to the three pillars or Key Function Areas (KFAs) of KM namely people, process and technology. This model is used within the Tata Consultancy (referred to also as a framework) 'describes each state of maturity, addresses the objective of each state and its perceived business benefits'. In particular, the purpose of this framework is the size (volume) of improvement as depicted in the inverted conical shape in an effort to emphasise the way in which relatively small changes to the process result in dramatic changes to the impact. Although not justified by actual data at this stage, this proposition focuses the practitioner to consider over time four different possibilities of moving from the stability state to change in the state.

All other studies until now have been based on a conviction and an assumption that the journey from lack of maturity and low KM capability will always proceed from lower to higher levels in a continuous ascending of the 'ladder'.

Paulzen et. al (2002) suggest a Knowledge Process Quality Model (KPQM) where KPQM stage 1 is taken from CMM level 1, KPQM stage 2 is called '*Aware*' instead of '*Managed*', stages 3 and 5 were adopted directly from SPICE and the CMMI term '*Quantitatively managed'* was adopted for stage 4. The stage descriptions are shown in Table 2.

Table 2: Maturity Stages of KPQM (source Paulzen et. al (2002)

1- Initial	The quality of knowledge processes is not planned and changes randomly. This state can be best described as one of chaotic processes.
2 - Aware	Awareness for knowledge processes has been gained. First structures are implemented to ensure a higher process quality.
3 - Established	This stage focuses on the systematic structure and definition of knowledge processes. Processes are tailored to react to special requirements.
4 - Quantitatively Managed	To enhance the systematic process management, measures of performance are used to plan and track processes.
5 - Optimising	The focus of this stage lies on establishing structures for continuous improvement and self optimisation.

A recent survey by Hain and Back (2009) '*has lead to 55 maturity models in the area of collaboration and knowledge management and e-learning. Three categories are identified: scientific, practitioner-oriented (scientific), and practitioner-based. Essentially, many maturity models are derived from the per se standard CMM, but, however, only a few are well enough documented to be further evaluated or applied in practice'.*

All the previous models used CMM as the benchmark basically adhering to the 5 levels which on occasions are slightly renamed. Instead of CMM we adopted the 5 levels of the newer and improved CMMI reference as the benchmark particularly for identifying the scope and focus on institutionalised practices (project-based, process-based, institutionalised). They also refer to Knowledge Management in general without specific emphasis on Knowledge Sharing which in today's globalised world becomes increasingly important in dispersed business environments. Knowledge sharing is the focus of our research.

3 *Maturity and Knowledge Sharing*

3.1 Purpose of Frameworks and Models

All models and frameworks define process models in organisations. Some models are applied to Key Knowledge Areas, in specific organisations; others are theoretical investigations with some propositions and suggestions. The purpose of all models can be distilled into an aid for assessing the KM maturity of an organisation and to provide guidelines for improvements.

In Georgiadou and Siakas (2009) we presented the Knowledge Acquisition and Sharing model (KASL – II) and postulated that *'a learning organisation is able to reflect and capitalise on the achievement of targets which in turn enhance the organisation's competitiveness*'. When all the employees feel empowered and responsible ownership of the process (they are involved with) and when they shed the old way of thinking by replacing the belief in knowledge sharing rather than in knowledge hoarding the organisation will move from '*knowledge is power*' to '*shared knowledge is power'.* KASL – II depicts four knowledge sharing and learning loops which involve individuals, groups, groups of groups (departments/divisions) and the whole organisation. Knowledge is captured, stored and accessed for improved decision making.

3.2 The I^5P Framework

In this paper we propose the I^5P Framework which considers the organisational process as the fundamental expression of the what, who, when and how. The where is considered as either internal or external and the why is understood as the overarching justification for organisational effectiveness and efficiency.

The five CMMI maturity levels are aligned with five knowledge management levels. At each level the nature and characteristics of the process are provided. As the maturity climbs up the 5 levels and at the same time the knowledge sharing practices improve from the minimal level to the institutionalised and optimising level, the organisational performance The architecture and performance of the organisation is depicted (in Fig. 1) as an inverted cone in an attempt here to use the metaphor of the growing volume of the cone as the indicator of the growth of value to the organisation. as the maturity climbs up the 5 levels and at the same time the knowledge sharing practices improve from the minimal level to the institutionalised and measurable.

The nature of the process defined in the I^5P Framework is encapsulated by the main characteristic at each level namely Incidental, Intentional, Implementational, Intelligent and Innovative.

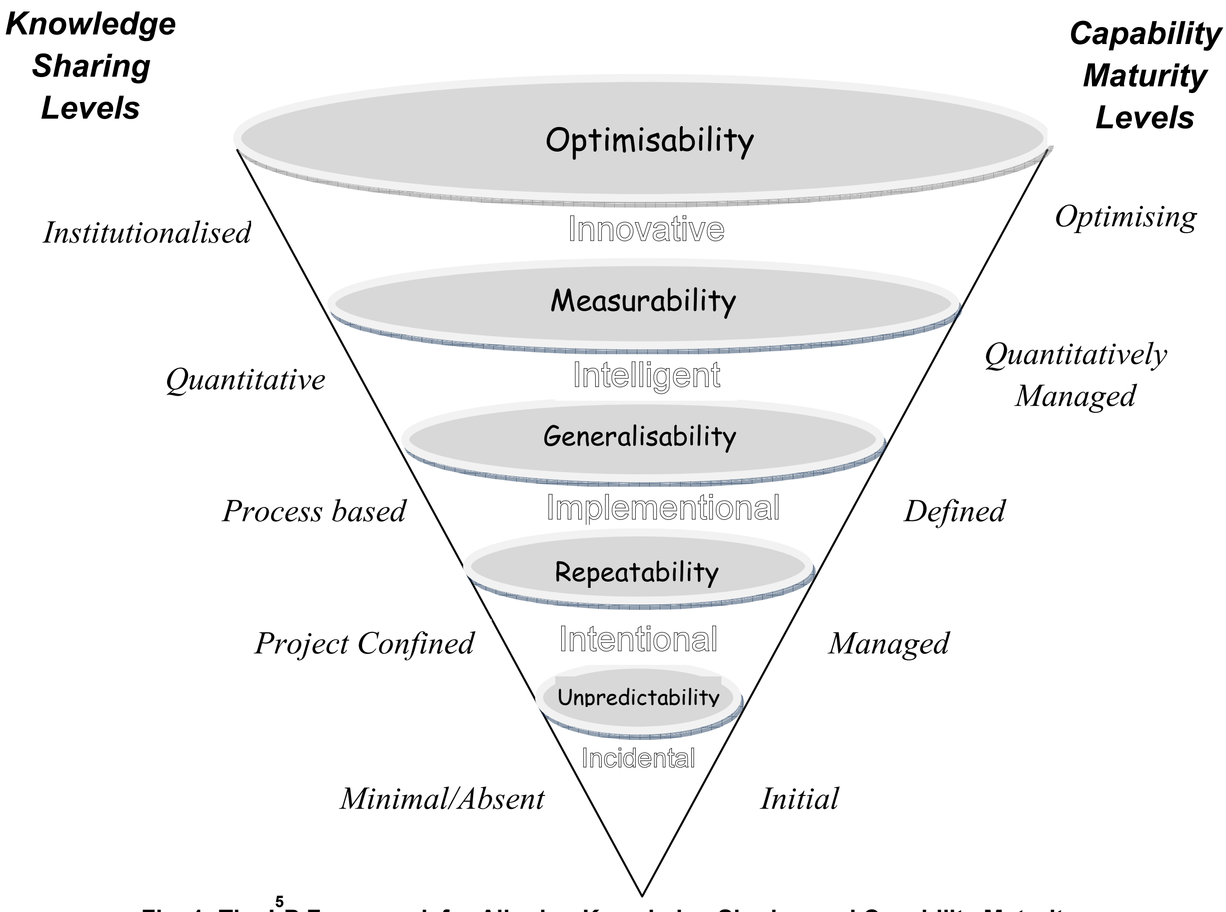


Fig. 1: The I5P Framework for Aligning Knowledge Sharing and Capability Maturity

3.3 The 5 Process Levels and Characteristics of I5P

I(1) - Incidental

This is the level where the processes are simply performed in an ad-hoc manner and are characterised by unpredictability. Knowledge resides with individuals and knowledge sharing is absent or minimal.

I(2) - Intentional

At this level the project management is characterised by repeatability. Best practice has been identified and documented, so that knowledge about successful practices can be shared amongst the project team. Subsequent projects undertaken by the team benefit as the characteristic of repeatability.

I(3) - Implementational

At this level processes are defined and generalisable. Knowledge is shared between projects.

I(4) - Intelligent

At this level processes are performed, repeatable, defined and managed. Measurability is the main characteristic of the processes which in turn provides useful intelligence thereby enhancing performance at an institutional level. Knowledge sharing is institutionalised.

I(5) - Innovative/Improving

This level is characterised by optimisability and continuous improvement. Knowledge sharing is institutionalised and quantitative. Improvements are achieved from continuous feedback, across teams, within and across projects and across the whole organisation. All employees understand, embraced and practice the philosophy of knowledge sharing. Processes are continuously improving and innovative ideas of all employees find fertile ground.

4 Model and Framework Validation

According to Macal (2005) *'Model verification attempts to establish whether a model implements the assumptions correctly'.*

In order to validate the I^5P Framework for Aligning Knowledge Sharing and Capability Maturity specialists in academia and industry were asked to respond to a number of questions based on (Georgiadou and Siakas, 2009; Macal, 2005), but to also give opinions in free format if they so chose. The qualitative validation and verification exercise aimed to collect responses from experts. The opinions, criticisms and suggestions collected were acted upon in one of two ways namely incorporating corrections and ensuring that future work will address issues through the development of specialised models and quantitative estimations and measurements of efficiency, competitiveness and business gains. Below is a summary of the responses by 7 Knowledge Management researchers/experts (mainly from academia but with considerable industrial experience gained through on-going consultatncy on several industrial projects) to the following questions. Where the expert judgements were given on the grading scale of 1 to 4 (where is 1 = Strongly Agree, 2 = Agree, 3 = Disagree, 4 = Strongly Disagree). Critical comments and suggestions given in free format are included.

Question (i): ***Does the model solve an important problem?***

Grading: One respondent thought that the scale was not applicable whilst the others either strongly agreed or agreed (in equal proportion).

Free text responses:

'It seems very relevant and logical'.

'The CMMI general capability maturity model contains largely the same idea but the details would indeed be good to have in special KM application language'.

'In its current form the proposed model is a conceptual framework or working hypothesis. It is compared to CMM, but if you go to original CMM research you will notice that it is based on empirical data from software companies. In order to build a model, you would need to collect similar data as well. The number of steps, as well as the names of steps, are semantic details, the important feature is that the model would described the step order based on empiria'.

Action: The authors clarified (in the paper) that the model is indeed a conceptual framework, and incorporated in the plans for further work the collection of empirical data for the estimation and measurement of performance and value to the organisation.

Question (ii): ***Does the model contain errors, oversights, or bugs?***

Grading: No errors, oversights or bugs were reported by the respondents.

Free text responses:

'*The data will tell this and you can calculate the significance*'.

'*I do not notice anything special, but I am little bit unsure if knowledge management practices are able to improve in same phase so categorical*'.

'The general problem of maturity models is the logical order of levels – which level is 'more' than which other. Thus the names of the levels and the assumptions they give are important. The names of levels 'innovative' and 'intelligent' may be problematic – in some other uses 'innovative' means more creative, soft, less organized processes, while you want to give the picture that this is the highest level of organisation and optimisation. Other names and their order sound good in English but may give problems when translated'.

Action: The hypothesis of stage-wise improvement in a categorical fashion will be tested empirically. The last suggestion was incorporated in the framework (level 5). In future we will carry out evaluations across cultures and languages to iron out any ambiguities of nomenclature.

Question (iii): ***Does the model meet a specified set of requirements?***

Grading: The responses to this question ranged from '*not applicable*' to a very poor '*strongly disagree*'. The average response was a 3.5. The criticisms and suggestions provided insights into the weaknesses of the model description rather than the model structure.

Free text responses:

'I assume that the requirements would be related to question: What are the steps? What type of activities is included at each step? What a company in level X, should do in order to get on level X+1?'.

'*As a model, I think no. The good point with maturity models generally is not that they have levels, but that the company/organisation is measurable against the levels and the measurement – the level the company is on – 'automatically' points to the next improvement direction. The brief description of the model does not include this'.*

'*Well, in this technologically advanced age it might meet the requirements of some companies but I am not sure whether it is meeting the requirements of all the companies in all sectors*'.

Action: The challenges posed by the experts have been acted upon in providing additional descriptions and by incorporating the specified requirements in the plan for future work. In particular, it is intended to use the framework in order to develop customised models for different domains, different types and sizes of organisations and also across different cultures.

Question (iv): ***Do you expect the model to perform as intended?***

Grading: The average response to this question was a 2.5 i.e. half-way between agree and disagree. This is indecisive.

Free text responses:

'Probably'.

'*I hope so, but we cannot be sure* '.

'*Remains to be seen, how the measurement system that I expect will follow will look and function, but generally I would expect the maturity levels to follow the order as presented in the model, and the main division points between them to be what are also shown'.*

'It can vary from industry to industry or sector to sector'.

'I am not able to say on the grounds of this document'.

Action: The free text responses provided clearer suggestions (than the poor grading) which when acted upon will provide strengthened belief in the future performance of the framework and the intended customised models.

Question (v): ***Can we ensure that the model meets its intended requirements in terms of the methods employed and the results obtained?***

Grading: The average grading here was equal to 2.25 – which is again an inconclusive result**.**

Free text responses:

'The level of specifying needs to be more practical and more detailed'.

'*To answer this I would need to know more (at least something) about the methods and results'.*

'*What does a company need to do to get X to level X+1?'.*

'What are the steps? What types of activities are included at each step?'

Action: At this generic, conceptual level it was not possible to detail the recommendations. In future when the empirical data is collected helpful recommendations will be developed.

Question (vi): ***Is the model useful i.e. does the model address the right problem and does it provide accurate information?***

Grading: Several respondents did not provide a numerical grade to this question. The graded answers ranged from 2 = agree to 4 = strongly disagree. This is interpreted as negative because the question was not perceived as clear or relevant. The free text responses gave more insight into the perceptions of the respondents. Due to the low number of responses the result is inconclusive**.**

Free text responses:

> *'It addresses the right problem, but remains to be seen whether the information will be accurate'.*
>
> *'I would say it depends that how and where it is employed'.*

Action: It is evident from the responses that further work is necessary to refine the framework and to develop models detailing the activities, which will ensure that an organisation can move from the lower layers upwards. Additionally measures will be developed for quantifying the business gains at each level.

5 Conclusion and Further Work

This paper presented a study of knowledge management maturity models which culminated in the proposal of the I^5P Framework which is a visual metaphor of the alignment of the process maturity and knowledge sharing within organisations. The volume of the increasingly larger *'cone'* within this metaphor is intended to communicate the business value gained by organisations as they move from the lower to the higher layers. As is the case in all frameworks I^5P is a thinking tool conceptualising a multiplicity of factors through characterising the gradual maturity increases.

There is still the need to enhance the framework by defining measures and activities which will help organisations to move from layer n to layer n+1. Empirical data need to be collected and analysed and further studies will need to customise the framework into implantable models for example for specialised domains, SMEs and global organisations.

In the future it is also our aim to measure the effect of Knowledge Sharing on efficiency, competitiveness and business gains.

References

CMMI for Services V1.2 Carnegie Mellon University (2009).

Elms, K., Langen, M. (2002). Knowledge Management Maturity Model – KMMM® Methodology for assessing and developing maturity in knowledge management, available at http://www.kmmm.org/objects/KMMM_Flyer.pdf (accessed 13.03.2010)

Gallagher, S., Hazlett, S.A. (2000). Using the knowledge management maturity model (KM3) as an evaluation tool, Conference on Knowledge Management Concepts and Controversies, University of Warwick, Coventry, 10-11 February

Georgiadou, E., Siakas, K. (2009). A Dynamic Model for Improving Knowledge Sharing in Virtual Organisations, Industrial Stream of European Software Process Improvement (EuroSPI), Alcala, Madrid, Spain, 2-4 September 2009, pp. 3.37 – 3.47

Hein, S. Back, A. (2009). State-of-the-Art on Maturity Models for Collaboration. White paper (accessed 15.3.2010) available at http://web.iwi.unisg.ch/org/iwi/iwi_pub.nsf/wwwAuthorPublGer/F4A4973DEFFE7034C125765E00502373/$file/State-of-the-Art_01_2009.pdf

Kulkarni, U., Louis R. S. (2003). Organisational Self Assessment of Knowledge Management Maturity, 9th Americas Conference on Information Systems (AMCIS), pp. 2542 - 2551

Macal, C.M. (2005). Model Verification and Validation, Workshop on 'Threat Anticipation: Social Science Methods and Models', The University of Chicago and Argonne National Laboratory, April 7=9, 2005, Chicago, IL (retrieved on 01/03/08 from http://jtac.uchicago.edu/conferences/ 05/resources/VandV_macal_pres.pdf)

Mohanty, S., Chand, M. (2004). 5iKM3 Knowledge Management Maturity Model, White Paper, Tata Consulting Services (http://www.tcs.com/SiteCollectionDocuments/White%20Papers/5iKM3%20Knowledge%20Management%20Maturity%20Model.pdf accessed 13/03/2010)

Nonaka, I. (1991).The knowledge-creating company, Harvard Business Review, Vol. 69 No. 6, pp. 96-104

Paulzen, O., Doumi, M., Perc, P., Cereijo-Roibas, A. (2002). A Maturity Model for Quality Improvement in Knowledge Management, ACIS 2002 Proceedings. Paper 5 available at http://aisel.aisnet.org/acis2002/5 (accessed 15/03/2010)

Siakas, K., Georgiadou, E. (2008). Knowledge Sharing in Virtual and Networked Organisations in Different Organisational and National Cultures, eds. Ettore Bolisani, Building the Knowledge Society on the Internet, Idea Publishing, ISBN: 978-1-59904-816-1, Part 1, Chapter 3, pp. 45-64

Author CVs

Elli Georgiadou

Elli Georgiadou is a Principal Lecturer in Software Engineering at the School of Engineering and Information Sciences, Middlesex University, London. Her teaching and research include Software Quality, Software Process Improvement, Software Measurement (Metrics & Estimation), Methodologies, CASE, Project & Knowledge Management and Cultural Issues and Software Quality Management. She has extensive experience in academia and industry, and has been active in organising/chairing conferences and workshops under the auspices of the British Computer Society and the ACM British Chapter. She is her School's Co-ordinator of European Activities, Projects and International Exchanges. She has a number of academic and industrial project partners in many countries and she has published extensively about her research in journals, book chapters and International refereed conferences.

Kerstin V. Siakas

Kerstin Siakas is an Associate Professor in the department of Informatics at the Alexander Technological Educational Institute of Thessaloniki, Greece. Her teaching includes Management Information Systems, Software Quality Management and Project Management. She is engaged in research in Information Systems Engineering, Multidisciplinary Approaches of Software Engineering, Knowledge Management and Software Quality Management. She has a particular interest in the social, cultural and political approaches and their effect on society, such as globalisation, outsourcing, e-Learning and e-Inclusion, as well as on the subsequent challenges for managers, educators and governments. She has an extensive industrial experience in software development on different levels from many European countries and mainly from multinational organisations. She has a number of academic and industrial project partners in many countries and she has published extensively about her research in journals, book chapters and International refereed conferences.

Bo Balstrup

Bo Balstrup is Managing Director of the Center for Software Innovation in Sonderborg, Denmark. The centre provides research, technology transfer and matchmaking between industries and universities in the area of complex embedded mechatronic systems. He holds a B.Sc.EE from the Danish Technical University and an Executive MBA from Henley Management College, UK. He has formerly been Manager of Software Technology at Danfoss Drives, where he built the global software organisation. He has an extensive 31 years of industrial experience in software development and management from a number of Danish and multinational companies combined with experience in university-industry collaboration projects. He has implemented software quality systems in global organisations and headed an acquisition project as part of company globalisation. His research areas are Software Process and Methodology and Leadership of Virtual Organisations.

The people aspects in modern (S)PI management approaches

Morten Korsaa[1], Jörn Johansen[1], Tomas Schweigert[2], Detlef Vohwinkel[2], Dr. Richard Messnarz[3], Risto Nevalainen[4], Miklos Biro[5]

[1] DELTA Axiom
Venlighedsvej 4, DK2970 Hoersholm, Denmark
www.deltaaxiom.com mko@deltaaxiom.com joj@deltaaxiom.com
[2] SQS Software Quality Systems AG, Stollwerk STR. 11, D-51149 Köln
Tomas.schweigert@sqs.de
[3] ISCN GesmbH, Schießstattgasse 4, A-8010 Graz
rmess@iscn.com
[4] Spinet Oy, Mutkatie 4, 02330 Espoo
risto.nevalainen@spinet.fi
[5] Corvinus University of Budapest
Fovam ter 13-15. H-1093 Budapest, Hungary
miklos.biro@uni-corvinus.hu

Abstract

Since the beginning of the 1990ies process improvement was considered as a formal issue. Focus was on process description and improvement was a somewhat better description. Also process improvement was driven by the customer side. Symptoms were ISO 9001 and CMM®. This situation remained stable even if at the mid of the fist decade of this century ISO/IEC 15504 was published. On the other hand approaches like People CMMI, EU SPRITE Project and the European Qualification Network ECQA were established, but their relevance was not recognized by IT people.

Currently we see a move forward to focus on people because they are recognized as key success factors. The first model that really emphasized the ownership and empowerment of people was the PEMM model of Michael Hammer[1]. In the IT community Ivar Jacobson[2] developed his approach while criticizing the current process description approach. At the same time Jan Pries-Heje did research related to effective SPI approaches[3].

In 2007 a first attempt was made to develop a Training curriculum for SPI Management. From the very beginning it was clear that people are a key factor to be addressed. This process of reorientation of the SPI community reached a milestone, when the SPI Manifesto was published and the Skill Card for the SPI Manager Qualification Scheme was approved by the authorized Job Role committee. Right now the first training is delivered and experience is excellent.

1 The History of People Issues in the SPI Evolution

The concept of innovation teams and human behaviour paradigms has evolved from a classical management approach to a networked learning organisation approach over the last 30 years. SPI is seen as a process innovation, and the innovation studies considered process innovation, including the relationships to product innovation, services innovation and new market creation. European projects have analysed the typical behaviour of people in innovation environments [8], [9], [10], [11] and outlined the following development.

In the original **traditional management model** (till the 80s) the following characteristics were commonly featured: set rules of engagement for every task, division of labour clearly specified, hierarchy of control permeating the firm, job specifications layed out for each task, impersonal nature of the implementation process, financial performance the key imperative, overlooked the social needs of staff.

Conclusion: In such an environment the SPI approach typically is based on a top down command from top management to improve processes while none on the floor understands and the processes are not really understood or implemented. However, all do it just because of a command.

The **Human Behavior Model** (starting in the 90s) recognises the requirement to involve the individuals in an environment of innovation. Here the individual's need for affection & group membership is considered, it takes into account the esteem needs of employees, it encourages individuals to actively seek self-actualization in the performance of their duties and become part of the company vision and goals, and it tries to breed employees who are highly qualified individuals with long-term loyalty to the firm, and aligning their excellence and own advancement with the innovation mission of the company.

Conclusion: In such an environment the SPI approach typically is based on a top management commitment and selected empowered individuals who do the assessment and lead improvement projects. However, only the selected empowered individuals do the SPI and when they are not there the SPI knowledge vanishes.

The **systems based people approach** (end of 90s) considers the networking as a core part for innovation development and break through. A system is a collection of interrelated parts which form some whole, the value of such an approach is that it recognises the criticality connecting individuals in the firm to commonly solve issues/problems and to react with their environment in dynamic and innovative situations.

Conclusion: In such an environment the SPI approach typically is based on a top management commitment, selected empowered individuals who do the assessment and lead improvement projects, and a network of supporters and experts in different divisions who help to really implement the SPI in the projects.

In a **learning organisation concept** (starting from 2003) there exists a shared vision upon which everyone agrees, people disguard their old ways of thinking and problem solving, all organizational processes both internal and external are interrelationships that must be managed cost effectively, communication is honest, without fear of ridicule or punishment, and departmental or personal interests are set aside in favour of the organizational goals and objectives. Individuals are empowered, have long-term perspectives in a borderless interactive environment, life-long learning is a way of life in learning organizations, and they have flat organizational structure that promote ongoing transformational change

Conclusion: In such an environment the SPI approach is based on a top management commitment, selected empowered individuals who do the assessment and lead improvement projects, a network of supporters and experts in different divisions who help to really implement the SPI in the projects, and an organisational strategy empowering a joint SPI vision and financing the team learning across the projects and departments.

The success of SPI largely depends on the human strategy of the organisation and how empowered individuals are supported through a learning organisation environment.

2 People Issues in the ImprovAbility Model

ImprovAbilility (see Figure 1) was made to support process improvement activities. The model suggest a defined approach to analyse 20 parameters that is known to be a source of risk for at process improvement project.

The human aspect has been paramount in the entire model building based on the fact that processes are nothing if not performed by people. Process improvement is for the same reason entirely dependent on the individuals motivation to change the way they are working. One obstructive person can easily compromise the success for a process improvement project, but establishing the exact mechanism that creates the demotivation is impossible. However – analysing the ImprovAbility parameters will generate a valuable insight to the PI professional and enable him to mitigate the largest risks involved.

Initiation category

The “Sensing urgency” parameter is about how the employees interpret the need for a change in the organisation. Employees will be more motivated, if it is obvious to them that the change is needed, either for the sake of the company, or directly for them self.

“Idea Creation” is looking at the amount of ideas that an organisation creates and support. If the environment for the employees are creative and innovative in general, chances are more likely that a given process improvement will succeed, because individuals are used to new ideas, and have learned that supporting them is positive.

“Idea Processing” is the handling of new ideas part. The more ideas an organisation is generating, the more it needs an effective process to prioritize and communicate decisions to the employees. The challenging human aspect here is how to tell the idea-generators, that their idea was not prioritized – without destroying the motivation for creating new ideas.

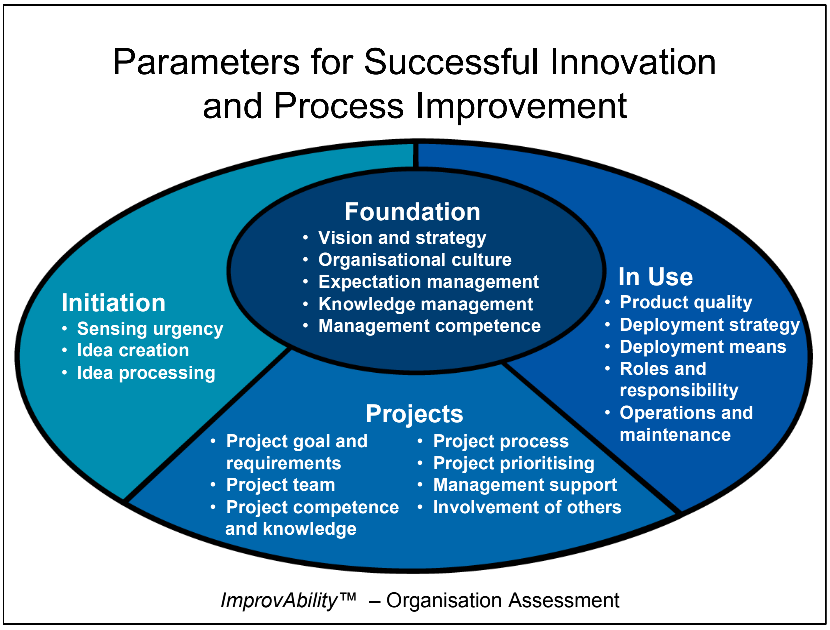


Figure 1 : The Improvability Model

Projects category

“Project goals and requirements”, “project prioritization”, “involvement of others” and “management support” are important, because people are more motivated by having clear goals to work towards and by knowing that their management appreciate their work.
“Project team” is measuring the teams composition with regards to motivation, attitude and personalities.
“Project process” and “Project competence and knowledge” is concerned with the process support for the project.

In Use

“Product Quality”, “Deployment strategy”, “Deployment means” and “Operations and maintenance” are all parameters that deals with how the change/improvement is perceived by the individuals to support their activities, and how the deployment has been planned to meet their personal requirements.
“Roles and responsibilities” are looking at the humans aspects of those who are driving the change.

Enterprise foundation

These parameters are somewhat different than the others because they are looking at the organisational level. The organisational level is however composed as an aggregation of the humans involved, and are as such addressing the human aspects only in general terms. The parameters in enterprise foundation address how well the organisation is supporting the motivation for change of individuals.

Change strategies

In great respect to the culture in organisations, and with the intention to nurture the motivation of individuals, the ImprovAbility method is supporting the choice of a top level change strategy. E.g. Business Process Reengineering (BPR) may be a great change strategy in one organisation, but create great resistance in another! This makes the choice between the ten major change strategies crucial for success, because each of them will create different levels and types of motivation among the individuals.

3 People Issues in the PEMM Model of Michael Hammer

In 2007 Michael Hammer published his Process and Enterprise Maturity Model. His main thesis is that an immature organisation is not able to run capable processes. His Goal is the high performance process and he is fully aware that a high performance process needs adequate people.
1st we should have a look at the maturity model of PEMM:

Organisational Aspects

E-0	Immature Organisation
E-1	Organisation with basic capability of process management
E-2	Organisation with the ability to handle projects with contribution from several units
E-3	Organisation with an established process and project framework
E-4	Organisational ability to integrate processes of customer and supplier

Process Aspects

P-0	Process functions by random
P-1	Process is reliable, predictable and stable
P-2	Process is designed E2E along a value chain
P-3	Process can be integrated with other processes to optimise the performance and output of the organisation
P-4	Process is defined beyond the borders of the organisation integrating customer and supplier

When we look at the model, we find organisational aspects and process aspects. Both are somehow related to people issues.

Let's first have a look at the organisational aspects:

Organisational Aspects			
Leadership • Awareness • Alignment • Behavior • Style	Culture • Teamwork • Customer Focus • Responsibilit y • Attitude towards change	Expertise • People • Methodology	Governance • Process Model • Accountabilit y • Integration

We see, that these organisational aspects might contain personal aspects in the topics Culture and Expertise. Let's look at the leadership aspect and see the alignment topic:

Level	**Description**
E-1	The leadership of the process program lies in the middle management ranks.
E-2	A senior executive has taken leadership of, and responsibility for, the process program.
E-3	There is strong alignment in the senior executive team regarding the process program. There is also a network of people throughout the enterprise helping to promote process efforts.
E-4	People throughout the enterprise exhibit enthusiasm for process management and play leadership roles in process efforts.

We see that the highest level requires the most intensive involvement of people in process efforts.

Let's look at the cultural dimension: Teamwork, Responsibility and Attitude towards Change.

Teamwork:

Level	Description
E-1	Teamwork is project focused, occasional, and atypical.
E-2	The enterprise commonly uses cross-functional project teams for improvement efforts.
E-3	Teamwork is the norm among process performers and is commonplace among managers.
E-4	Teamwork with customers and suppliers is common place

Responsibility:

Level	Description
E-1	Accountability for results rests with managers.

E-2	Frontline personnel begin to take ownership of results.
E-3	Employees feel accountable for enterprise results.
E-4	Employees feel a sense of mission in serving customers and achieving ever-better performance.

Attitude towards Change

Level	Description
E-1	There is growing acceptance in the enterprise about the need to make modest change.
E-2	Employees are prepared for significant change in how work is performed.
E-3	Employees are ready fa major multidimensional change.
E-4	Employees recognize change as inevitable and embrace it as a regular phenomenon.

Lets than have a look at the process aspects:

Process Aspects				
Design • Purpose • Context • Documentation	Performers • Knowledge • Skills • Behaviour	Owner • Identity • Activities • Authority	Infrastructure • Information Systems • Human Resource Systems	Metrics • Definition • Uses

Let’s have a look at the Performers Topic (Knowledge, Skills and Behaviour)

Knowledge:

Level	Description
P-1	Performers can name the process they execute and identify the key metrics of its performance.
P-2	Performers can describe the process's overall flow; how their work affects customers, other employees in the process, and the processes performance; and the required and actual performance levels.
P-3	Performers are familiar both with fundamental business concepts and with the drivers of enterprise performance and can describe how their work affects other processes and the enterprise's performance.
P-4	Performers are familiar with the enterprise's industry and its trends and can describe how their work affects interenterprise performance.

Skills:

Level	Description
P-1	Performers are skilled in problem solving and process improvement techniques.
P-2	Performers are skilled in teamwork and self-management.
P-3	Performers are skilled at business decision making.
P-4	Performers are skilled at change management and change implementation.

Behaviour

Level	Description
P-1	Performers have some allegiance to the process, but owe primary allegiance to their function.
P-2	Performers try to follow the process design, perform it correctly, and work in ways that will enable other people who execute the process to do their work effectively.
P-3	Performers strive to ensure that the process delivers the results needed to achieve the enterprise's goals.
P-4	Performers look for signs that the process should change, and they propose improvements to the process.

As a result of this analysis we can see that people issues are not only addressed but also linked to the highest maturity levels. So SPI in modern times must have a strong emphasis for people issues.

4 People Issues addressed by Ivar Jacobson

In the 1st decade of the 21th Century process performers found that processes are becoming a barrier to change rather than an enabler. The first sheet lightning was the agile manifesto. We see some thesis that is really people related:

- Individuals and interactions over processes and tools
- Working software over comprehensive documentation
- Customer collaboration over contract negotiation
- Responding to change over following a plan

In this tradition of thinking Ivar Jacobsen, one of the spirits behind the Rational unified Process (RuP), set up his mind for process improvement[22]. Even if his reduction of a process to a complex and complete process description seems like bashing a bugbear, it can be found that he redefines process thinking from a performer perspective.

Jacobsen states that on the one hand processes are too big, because they try to be complete but not big enough as they don't provide guidelines for every possible situation.

Performers need freedom and the ability to respond to the unexpected.

Describing this issue, Jacobsen comes to the following result:"This gap between the process as described and the process as applied is what we call the "project-process" gap. This gap causes problems for projects, teams and organizations:

- When a team is successful, it becomes difficult to spread that success to other teams. If another team takes the successful team's process and applies it, the new team won't get the same results because the process description doesn't describe what the successful team has really been doing.
- Process and quality assurance becomes less effective and more expensive. The assurors spend large amounts of time looking at the process rather than at what the people actually do. The gap can even lead teams to waste a lot of time trying to achieve the illusion of process conformance - especially when they realize that they are about to be assessed.
- Teams often find that they have to spend large amounts of time filling the gap by writing large amounts of additional process documentation; for example creating local requirements management plans and configuration management plans. This local process documentation is presented separately from the process, and often contradicts or overrides large amounts of it. The end result is that no-one really knows what process the team is applying.
- It becomes far more difficult to improve or retool the project environment as it is unclear how the team is actually working. Tools are often imposed on the team that support the process description but not what the team does or needs to do .This results in either the tool never being used or a lot of time being spent learning how to configure and change the tool to do something it wasn't designed for. In some organizations, using the tool starts to become more important than doing the job; this is when you know that things have really gone wrong.
- The process should be a description of what the team actually does, rather than a fictional description of what people think the team ought to be doing"[22].

Looking at this statement it becomes clear, that Jacobsen prefers a team centred process instead of constructing processes in an ivory tower filled with experts. In his model he gives the team a pragmatic structure based on a kernel and practice structure.

So he constructs his model consequently on the needs of process performers and the teams they are engaged in.

So we can see similarities between Hammer and Jacobsen.

Both give the process performer an important place in their system as then make the responsible for their process, the performance and the necessary changes.

5 People Issues in the SPI manifesto

The SPI manifesto was created by a large group of SPI experts in 2009. The manifesto is stating what SPI is all about. At the top level there are three principles, the first being:

We truly believe that SPI must involve people actively and affect their daily activities

Again we see a deliberate move toward focus on human aspects. It demonstrates a process improvement trend shift from:

Expert design changes and roll it out to the organisation.

To:

People making full use of their experience, taking responsibility for change on their project and throughout their organisation, and using and improving the processes they have helped to define.

Values

The following “values” are kown to support the principle.

“Know the culture and focus on needs “ tells us, that in order to have success with SPI we must have great respect and insight into the culture of the organisation including the people that constitutes the organisation. All human carried meanings, values and practices are contributing or obstructing process improvements. SPI activities must be aligned with those to have success.

“Motivate all people involved.” Tells us, that if people are not motivated to change, they will most likely try to avoid it.

“Base improvement on experience and measurements” is really an important value. First, it implies that the processes are what people are actually doing. Not what is written on the intranet, if that is not reflected in reality. So process improvement is improving how people are working, not improving process descriptions on the intranet.

First goal is an agreed baseline of processes that all practitioners can accept.

Next goal is to determine improvements based on measurements of process performance. If a process is performed poorly, measure what that means and define what appropriate performance is. Then ask the people who are performing what would be smart to do, and together design the new process based on that input.

“Create a learning organisation” talks about the benefit of a learning organisation, where change is declared permanent, and the organisation is supporting the continuous improvement culture. Again – a strong aspect is the human learning.

6 People Issues in the ECQA

The SPI manifesto and the ECQA certified SPI manager qualification consider these organisational and human concepts in the education. Two thirds of the value and principle statements of the SPI manifesto actually address the organisational and human / people aspects.

For the success of companies in SPI it is therefore of critical importance to understand that if they do not develop these skills and competencies in their firm they will have less success in the SPI implementation.

ECQA (European Certification and Qualification Association) standardised these skills in 18 European countries and offers access to an SPI manager qualification [5]. ECQA is the result of a number of EU supported initiatives in the last ten years where in the European Union Life Long Learning Programme different educational developments decided to follow a joint process for the certification of persons in the industry. Through the ECQA it become possible to attend courses for a specific profession in one country and perform a Europe-wide agreed test at the end of the course.

7 People Issues in the SPI Manager Training

Based on the ECQA Scheme (www.ecqa.org) the SPI Manager Qualification scheme addresses all aspects of modern SPI as stated in the SPI manifesto.

The Skill set of the SPI manager Qualification contains the following units and elements:

PI Involvement and Commitment issues	Improvement Models	Managing PI	PI Implementation
PI Team Communication PI Team Dynamics SPI in Multicultural Environments National Culture Influencing SPI Organisational Culture Influencing SPI Modern Group Motivation Techniques for PI	Process Thinking Process & Life Cycle Models Capability & Maturity Models Process Improvement Models Process Design & Process Description Models	Supporting Top Manager for Organisational Change Management PI Needs & Drivers Analysis Alignment of PI Goals to Business Goals Process Measurement, Data Collection & Analysis PI Leadership	Planning Improvement Deployment of SPI Reporting SPI & Awareness Creation Experience and Good Practice Sharing

The SPI Manager skill set addresses the issues of people as member of a team, part of a culture or member of an organisation as well as their roles of process owners and process performers. Last but not least we find issues of people that want to learn and benefit from experience and also to exchange experience with other people.

As we can see, **the unit "**PI Involvement and Commitment issues " is clearly linked to people issues. Behind the element "Modern Group Motivation Techniques for PI" we find approved best practices like "Real Time Strategic Change" or "Open Space"

Also the unit “PI implementation” deals with people issues as the elements “Deployment of SPI” and “Experience and Good Practice Sharing” show.

We also see that the SPI manager qualification addresses people issues that are in and beyond the range of the SPI Manifesto

8 Summary

Looking back to the last 10 years of SPI we can see a rapid change of focus from models to people. Models like CMMI, SPICE, ITIL or others still will remain useful. But they don’t have the power to create competitive advantage. What we see in all analyzed approaches is that people really count. That competitive advantage has one important source in the commitment and the responsibility of people for their processes in terms of effectiveness, efficiency and results quality. And it has also a source in the peoples attitude to reasonable changes. Even if the approaches of Hammer, Jacobson, the ImprovAbility Model, the SPI Manifesto, the ECQA Scheme and the SPI Manger qualification scheme seem to be different on the first view they all agree on the relevance of people and their needs.

References and further Reading

1. Nathan Baddoo, Tracy Hall, Ciaran O’Keeffe: Using Multi Dimensional Scaling to analyse Software Engineers’ Demotivators for SPI, EUROSPI 2008 Proceedings p. 5.1 – 5.16.
2. Bloom taxonomy, in http://en.wikipedia.org/wiki/Taxonomy_of_Educational_Objectives.
3. Certificates for software testing, in http://www.istqb.org/.
4. Chaos report, in http://www.standishgroup.com/newsroom/chaos_2009.php.
5. ECQA Guide: ECQA European Certification and Qualification Association Guide, in http::/www.ecqa.org/.
6. Jan Pries-Heje, Jørn Johansen: ImproveIT. A book for improving software projects. DELTA, 2007, ISBN 978-87-7398-086-6.
7. Lars Mathiassen, Jan Pries-Heje, Ojelanki Ngwenyama: Improving Software Organizations: From Principles to Practice, Addison-Wesley, 2002, ISBN 0-201-75820-2.
8. Messnarz Richard, Ekert Damjan, Reiner Michael, O'Suilleabhain Gearoid. Human resources based improvement strategies - the learning factor. Software Process: Improvement and Practice 2008:4, pages 297 - 382 (July/August 2008).
9. A Learning Organisation Approach for Process Improvement in the Service Sector , R. Messnarz. C. Stöckler, G. Velasco, G. O'Suilleabhain, A Learning Organisation Approach for Process Improve-ment in the Service Sector, in: Proceedings of the EuroSPI 1999 Conference, 25-27 October 1999, Pori, Finland.
10. Richard Messnarz, Gearoid O'Suilleabhain, Ray Coughlan, From process improvement to learning organisations (p 287-294), in: Wiley Interscience Journal, SPIP Software Process Improvement in Practice Series, Volume 11 Issue 3 , Pages 213 - 335 (May/June 2006), 2006.
11. Messnarz R ., Spork G., Riel A., Tichkiewitch S., Dynamic Learning Organisations Supporting Knowledge Creation for Competitive and Integrated Product Design. In: Proceedings of the Interna-tional CIRP Design Conference, Cranfield, March 2009.
12. Schweigert Tomas, SPI – A Measurement driven Approach – SPICE DAYS Frankfurt 2007.
13. Statz, Joyce, Don Oxley, and Patrick O'Toole: "Identifying and Managing Risks for Software Process Improvement," CrossTalk (April 1997), pp. 13-18. In http:// web: stsc.hill.af.mil/crosstalk/1997/04/identifying.asp.
14. Tim Kasse, Practical insight into CMMI: The look and feel of a successful implementation, Artech House Publishers, 2004, Boston Massachusetts USA. ISBN 1-58053-625-5.
15. Tom Gilb: Project Failure: Some Causes and Cures By Tom Gilb; Web publishing 2004. In http://www.webster.edu/ftleonardwood/COMP5940/Student_Files/Project_Failure/ProjectFailure.pdf.
16. Mark C Paulk, A (Software) Process Bibliography, last update January 2009. In http://www.cs.cmu.edu/~mcp/papers/biblio.pdf.
17. 10 Major Causes of Project Failure. Centreline Solutions Inc. Web publishing, 2004.
18. Jan Pries-Heje, Jörn Johannsen et. al : The SPI Manifesto, web publishing 2010.

19. Michael Hammer: Der große Prozess Check, Harvard Business Manager H. 5/2007, S. 34 ff,
20. Michael Hammer, The Process Audit, Harvard Business Review 04/2007.
21. Michael Hammer, Process and Enterprise Maturity Model www.hammerandco.com/pemm.html.
22. Ivar Jacobsen, Pan Wei Ng and Ian Spence: Enough of Processes - Lets do Practices, JOURNAL OF OBJECT TECHNOLOGY, Vol. 6, No. 6, July-August 2007.
23. Nevalainen Risto, Schweigert, Tomas, Software Process Improvement Manager Training and Certification, SOFTWARE PROCESS IMPROVEMENT AND PRACTICE Softw. Process Improve. Pract. (2009) Published online in Wiley InterScience (www.interscience.wiley.com) DOI: 10.1002/spip.438.

Authors CV

Jørn Johansen

Department Manager, M.Sc.E.E., has more then 28 years experience in IT. He has worked for 15 years in a Danish company with embedded and application software as a developer and project manager. Mr. Johansen has been involved in all aspects of software development: specification, analysis, design, coding, and quality assurance. Furthermore he has been involved in implementation of an ISO 9001 Quality System in the company, and was educated to and functioned as internal auditor. For the last 13 years he has worked as a consultant and registered assessor, performing more then 30 BOOTSTRAP assessments and more then 25 CMMI assessments in Denmark for companies of all sizes. He has been project manager in the Danish Centre for Software Process Improvement project, a more then 25 person-year research project and project manager in the Talent@IT project, a more then 28 person-year Process Improvement reseacrh project. He is now project manager in a 18 person-year research project SourceIT with focus on maturity in relation to outsourcing.

Tomas Schweigert (Ass. Jur.)

Principal Consultant at SQS Software Quality Systems AG, Tomas Schweigert has a long experience with software quality management, software testing and process improvement, staying with SQS from 1991. He started his PI work with the BOOTSTRAP institute, working as a BOOTSTRAP lead assessor in several assessment projects. His special interest is the analysis of projects in crisis situations. Mr. Schweigert is now a SPICE principal assessor (INTACS scheme) his current research topics are the SPI Manager qualification and the test assessment approach test SPICE.

Morten Korsaa

Principal Consultant at DELTA Axiom, a Danish research and consultancy company. Morten has worked as a global SPI manager in a large telecom company and have as a consultant supported companies from 20 people to it departments of 3500. He has performed more than 100 assessments in both vendor and acquire organisations. Mr. Korsaa is SEI certified “Introduction to CMMI Instructor”.

Richard Messnarz

Principal Assessor ISO 15504 and Automotive SPICE, Moderator of SPI task forces in a group of above 20 leading German companies, project leader and expert in a number of European projects dealing with innovation concepts since 1996, Assessments and SPI implementations in leading companies since 1990, such as BOSCH, ZF, Continental Automotive, T-Systems, MAGNA, etc., General Chair EuroSPI, Vice President ECQA.

Miklós Biró

Miklós Biró is associate professor with habilitation at Corvinus University of Budapest with over three decades of software engineering, university teaching (including professorship in the USA), and management experience. He has a Ph.D. in mathematics (operations research) from the Loránd Eötvös University in Budapest, an Executive MBA degree from ESC Rouen, France, and a Master of Science in Management degree from Purdue University, USA. He is fluent in Hungarian, English, and French. He has been initiating and managing the Hungarian participation and consulting activities in numerous European projects and organizations. He is the author, co-author of Hungarian and English language books and numerous international publications in the fields of Software Engineering Management, Decision and Negotiation support Systems, Combinatorial Optimization, etc… He is member of the Editorial board of the journal Software Process Improvement and Practice published by John Wiley & Sons, member of the Hungarian Doctoral Council and of the Public Body of the Hungarian Academy of Science. He is founding president of the professional division for Software Quality Management of the John von Neumann Computer Society, the Hungarian national representative in Technical Committee 2 (TC-2) Software: Theory and practice of the International Federation for Information Processing (IFIP) where he is coordinator of the Manfred Paul Award.

Risto Nevalainen (Lic.Tech.)

Senior Advisor in FiSMA and part-time researcher in Tampere University of Technology Pori Unit. Risto Nevalainen has long experience in software measurement and quality topics. He has been managing director of STTF Oy and then Spinet Oy for last 14 years. His working experience includes position as managing director of Finnish Information Technology Development Center during 1989-1995. Mr. Nevalainen has participated in ISO15504 (SPICE) standard development since beginning. He is Competent SPICE Assessor and ISO9000 Lead Assessor.

Quantitative benefits of a model-based process improvement program at Portugal Telecom Inovação (PTIN): who, what, why, where, how and results

Arthur Maria do Valle, MSc.
Integrated System Diagnostics - ISD
arthur@isdbrasil.com.br

Nuno Alexandre Martins Seixas
Portugal Telecom Inovação
nuno-a-seixas@ptinovacao.pt

Sérgio Oliveira
Centro de Computação Gráfica, Universidade do Minho
sergio.oliveira@ccg.pt

Paula Monteiro
Centro de Computação Gráfica, Universidade do Minho
pmonteiro@dsi.uminho.pt

Abstract

This paper describes how the implementation of a CMMI-DEV ML3 quality improvement program was conducted at PT Inovação - a major Portuguese software development company producing solutions for telecommunications sector. The paper also shows how some indicators already reveal very significant improvements as the CMMI ML3 quality improvement program reaches its completion. For instance 87,5% better On-Time Task Completion, 218,73% better Effort Deviation Rate and 82,67% better Cost Deviation Rate.

Keywords

Process improvement; return on investment; software quality; CMMI

1 Who

Portugal Telecom Inovação (PTIN) is a technological company, part of Portugal Telecom (PT) group. Being the major telecommunication's player in Portugal, PT has in PTIN its major research and development company in terms of telecommunications' technologies. Besides having this internal role of research and development, PTIN is also a world class telecommunications' technology provider, having solutions in several markets. In 2009, PTIN had 50% of its products placed in Europe, 23% in South America, 18% in Africa and 5% in Asia, representing an operational revenue of 110 M€.

PT Inovação has a strong emphasis in R&D, and backs up the development of new products in the results obtained from research initiatives. During 2009, these initiatives represented an investment of 41% of all costs, with 5% of the operational revenue being applied in exploratory research.

The main areas where PTIN places its products are related with telecommunications, being specially aimed at telecommunications providers. Its products range from pure software systems, like Business Support Systems (BSS), Operational Support Systems (OSS) and end-users apllications, to hybrid systems composed of hardware, firmware and software.

A software process improvement initiative was launched by Portugal Telecom Inovação. For this purpose, an internal team was put together with 3 persons working in this project. In order to get help implementing CMMI Model, the company had help from an external consultancy, provided by CCG (Centro de Computação Gráfica) and ISD (Integrated System Diagnostics) in Europe. The final project team was composed of 6 internal persons with experience ranging from 3 to 28 years of experience.

2 What

PTIN´s quality improvement program intended to achieve CMMI-DEV maturity level 3, that was reached in November´2009 by a Class A SCAMPI appraisal. When the project started, PT Inovação Quality System was already certified according to several national and international standards, namely ISO 9001 and ISO 14001. Therefore, an important set of processes was already defined, covering the specific needs of different development departments, each using different software engineering tools and approaches. Related to these differences, one of the main goals was to establish a coherent use of best practices and industry accepted techniques. The objective was that these differences between tools and approaches used were significantly reduced, and therefore alowing for a more coherent set of techniques and tools, and a stronger organizational culture. On the other hand, having in-house knowledge and experience on such diversity of tools and approaches contributed for a more conscious and fundamented choice.

In absolute terms, none of the processes that resulted from CMMI initiative had to be created from scratch. Instead, only the addition of new process components (procedures and activities) and replacement or adaptation of some of the existent process components was required. There was also an important effort in implementing the new processes and adaptations throughout the different development teams, allowing for the necessary institutionalization to be present and real.

3 Where

Software process improvement program was implemented in two departments inside PTIN. One of the departments acts in the Operational Support Systems (OSS) area, while the other works mainly in eLearning solutions.

The department dedicated to the OSS area produces software systems that help the telecommunications providers to maintain, monitor and control their networks, providing and important operational asset to these companies.

The department that works with eLearning solutions delivers this type of solutions to a very broad range of organizations, not just telecommunication providers. This is a different type of product, aimed at the final user and with a very different kind of requirements.

The program integrated in the project portfolio all the projects that are currently being held in those departments, which represents 14 different projects, involving around 100 persons. As it can be seen from the two areas, the projects were very different, both in used technologies, knowledge domain and team size. In terms of technology, the majority of projects develop web based applications with strong database use, some using Java technologies, some using .NET related technologies. In terms of team size, the projects range from small teams of 4 persons to larger teams of 12 persons.

4 Why

By implementing a software process improvement program, and therefore, using a set of recognized best practices adapted to its reality and business needs, PT Inovação aims at increasing its processes maturity throughout all the development lifecycle. This increase in organization maturity is expected to be shown in some specific outcomes: (i) definition, implementation and use of a set of measures that could characterize the organization status; (ii) definition and institutionalization of an estimation method for all the projects; (iii) overall reduction in schedule variation, cost variation and productivity increase. While the two first objectives are to be obtained in a short timeframe, the third one is expected to be monitored and accomplished in a broader timeframe.

5 How

An initial gap analysis, mainly done through revision of the process descriptions' documents, provided a first evaluation of the existent processes. During this process, PTIN´s EPG provided the necessary clarifications to the ISD/CCG consultancy team, allowing the mapping of the descriptions against CMMI. As result of this gap analysis a set of necessary changes to the processes was identified and applied to the existent processes, in order to adapt them to CMMI maturity level 3. Instead of using a cascade approach, the whole improvement process was done in a rather iterative and incremental way. The new and adapted processes were gradually disseminated to the organization, by means of communication, training and support. A motivational survey was also applied and the results were addressed by action itens. Some weeks later, another version of the motivacional survey was applied in order to verify progress, concerning such an important theme in a process improvement program.

Additionally, for the projects in the scope of the quality improvement program, besides regular QA reviews, clarifications were provided by the QA staff during their regular meetings with the projects' teams. In a regular basis, the ISD/CCG consultancy team, assessed and measured the level of institutionalization and, as result of these assessments, corrective actions were implemented, to improve the degree of institutionalization. Upon completion of the definition and institutionalization of all processes, a SCAMPI Class B appraisal was performed. Having resulted in a rather encouraging result, this appraisal provided some minor corrective actions and recommendations, which required an only limited amount of effort to be applied. After that, a readness review and a CMMI L3 SCAMPI A were performed, which demonstred to the organization an official CMMI level.

6 Results and Benefits

As part of the PTIN assets, there is a measurement repository with measures that can also be use to measure progress, and benefits, of the process improvement program.

These measures were already applied to software development projects conducted in 2008 as well as for projects conducted in 2009 using PTIN´s CMMI ML3-based processes and assets. In this sense, some project indicators were selected and were analyzed by the consultancy team:

Absolute Project Completion (%) – GRAabs,

Relative Project Completion (%) – GRArel;

On-Time Task Completion (%) - %RcDP;

Effort Deviation Rate (%) - %DesvioRH;

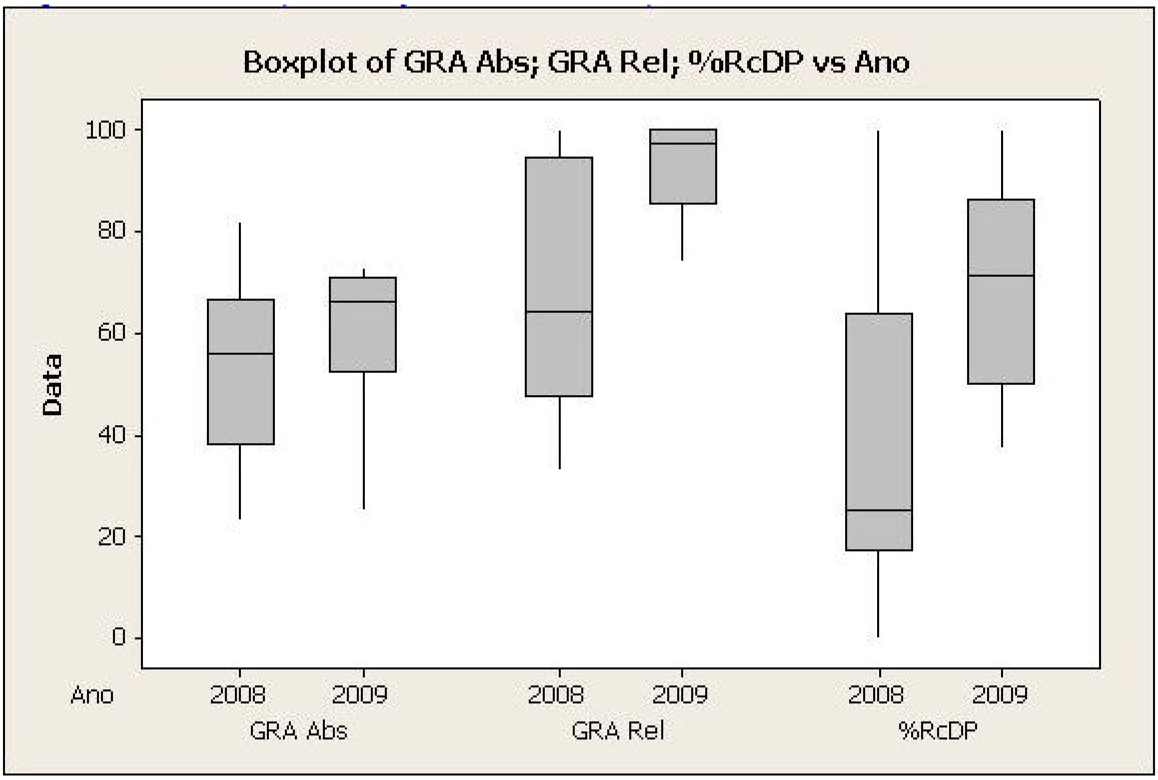
Cost Deviation Rate (%) - %DesvioCustos.

Comparations between third quarter status report of 2008 (14 projects) and 2009 projects (12 projects) were performed. Projects from 2008 did not use CMMI-based processes, but their 2009 versions did.

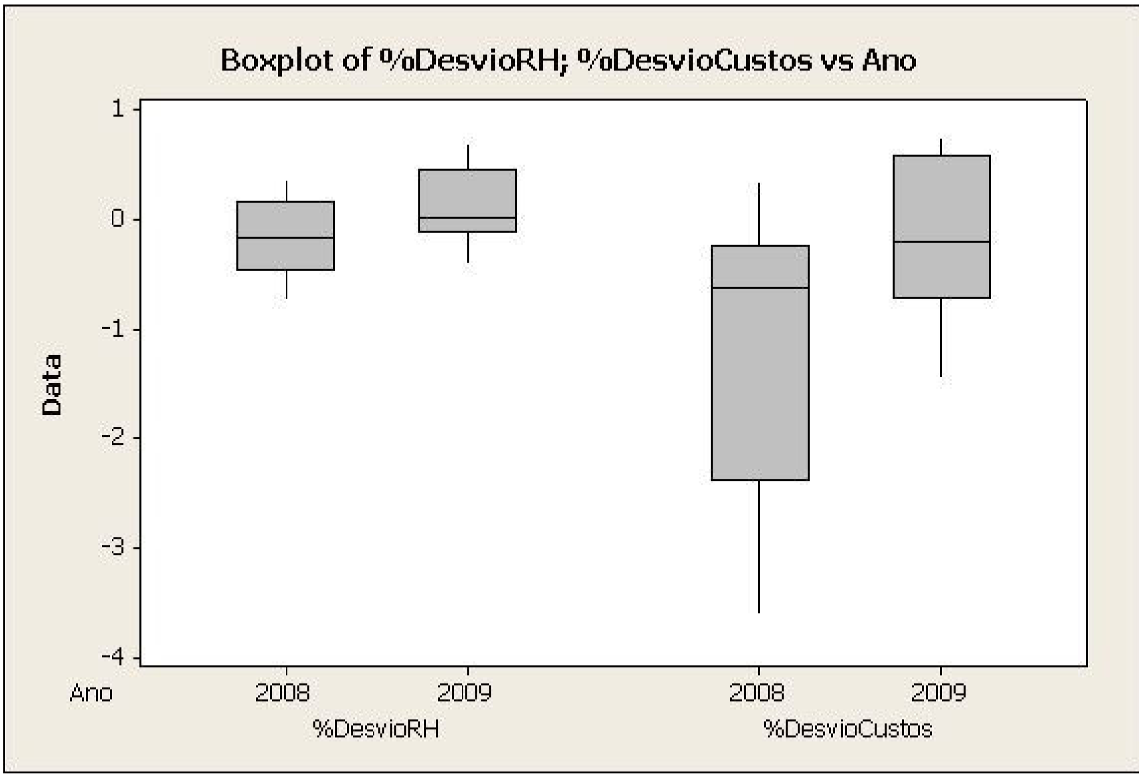
Although some effects are not statistically proven yet, due to small sampling, some interesting benefits related to these measures can already be measured:

- In terms of mean/median, there is improvement for all indicators: Absolute Project Completion (13,2% better), Relative Project Completion (35,3% better), On-Time Task Completion (87,5% better), Effort Deviation Rate (218,73% better) and Cost Deviation Rate (82,67% better).
- Additionally, there is variation reduction for Absolute Project Completion (22,3%), Relative Project Completion (63,08%), On-Time Task Completion (33,25%) and Cost Deviation Rate (41,01%).
- Finally, at least one measure with non-normal distribution had changed to normal distribution (On-Time Completion) that demonstrates gains of stability.

For example, in the graphic below, with some results plotted in a box-plot chart, it is possible to see that varition had decreased in 2009 for GRAabs, GRArel and %RcDP and mean (median, in fact) had improved for these three indicators.



In the other graphic (below), it is possible to see that variation had decreased in 2009 for two indicators (Effort Deviation Rate e Cost Deviation Rate), as well as mean/median had improved. Verify that variation of Cost Deviation Rate has improved from almost 400% of deviation to less than 200%, although has yet much to improve, it is already a first step.



Due to those proven benefits, as well as others factors, PTIN board has decided to disseminate this CMMI experience to all R&D units in 2010.

References

Beth, M., Konrad, M., & Shrum, S. (2007). *CMMI – Guidelines for Process Integration and Product Improvement*. Boston. Pearson Education.

Author CVs

Arthur Maria do Valle

Arthur had graduated in Business Administration at Universidade Federal do Paraná (2000) as well as in Computer Science from Catholic University of Paraná (1997). He has a Masters´s degree in Applied Computer Science at Catholic University of Paraná (2002). Arthur has been working as a process improvement consultant in ISD, where he also act as appraisal team member. Arthur is SEI certified instructor for CMMI-DEV, SVC and ACQ. He has over 12 years of experience in Management and Computer Science / Software Engineering. Arthur has Black Belt Six Sigma formation and is also professor of post-graduation.

Nuno Alexandre Seixas

Nuno Seixas graduated on Master of Software Engineering from Carnegie Mellon University and Universidade de Coimbra on 2008 and graduated on Informatics Engineering from Universidade de Coimbra, on 2004. Joined Portugal Telecom Inovação on 2005, being part of a development team responsible for a telemedicine platform – Medigraf. Participated in several european projects related to medical informatics, like MyHeart and mCare. Became CMMI Authorized Instructor on April 2009. Since January 2009 has been involved in several process improvement projects inside the organization, and is currently the CMMI Project Lead for the organization.

Sérgio Oliveira

Sérgio Oliveira has a Master degree in Information Systems and a graduation in Systems and Computer Science Engineering, both obtained at Universidade do Minho, Portugal, where he is currently a teacher and researcher in Software Engineering, and is also a researcher and consultant at the Centro de Computação Gráfica of the Universidade do Minho. Prior to this current positions he had worked in organizational training and participated in software development projects for 20 years. Part of his most recent activity concerns software process improvement, namely by taking part in the software process improvement project that lead PT Inovação to CMMI maturity level 3, both as a consultant and external appraisal team member.

Paula Monteiro

Paula Monteiro graduated on Masters in IT - Expertise in Distributed Systems, Computer Communications and Computer Architecture by the Universidade do Minho on 2006 and graduated on Mathematics and Computer Science by the Universidade do Minho on 2001. Actually is PhD Student in MAP-I (Doctoral Program in Computer Science – Minho, Aveiro and Porto). Join Universidade do Minho as researcher in 2002 being part of some national and international research projects. Since 2008 has been involved in some process improvement projects.

A Standards-Based Model for the Specification of System Design and Implementation Constraints

Alain Abran, Khalid T. Al-Sarayreh

Software Engineering Department, École de technologie supérieure (ETS)-
University of Quebec, 1100 Notre-Dame west, Montréal, Québec H3W 1T8, Canada
alain.abran@etsmtl.ca, khalid.al-sarayreh.1@ens.etsmtl.ca

Abstract

The European standards series for the aerospace industry (ECSS) include the software design and implementation (D&I) constraints as one of sixteen non-functional requirements for the embedded and real time software. Design and implementation (D&I) constraints are typically described at the system and software levels and within the ECSS standards, there are numbers of concepts and terms used to describe various types of candidate D&I constraints. This paper collects and organizes these concepts into a generic standards-based reference model of the requirements at the software level. The structure of this reference model is based on the generic model of software functional requirements proposed in the COSMIC – ISO 19761, in that way allowing the measurement of the functional size of such requirements implemented through software.

Keywords

Design and implementation constraints (D&I), Non functional requirements – NFR, Functional size, COSMIC – ISO 19761, ECSS International Standards, Software Measurement and SWEBOK Guide (ISO 19759).

1 Introduction

Non-functional requirements (NFR) play a critical role in system development, including as selection criteria for choosing among alternative designs and ultimate implementations. NFR may also have a considerable impact on project effort, and should be taken into account for estimation purposes and when comparing project productivity.

Typically, these non functional requirements are described at the system level, and not at the software level, and there is no consensus yet on how to describe and to measure such system NFR. In practice, NFR may be viewed, defined, interpreted, and evaluated differently by different people, particularly when they are stated briefly and vaguely [1-3]. It is challenging to take the NFR into account in software estimation and software benchmarking: the NFR have received less attention in the software engineering literature and are definitely less well understood than other costs factors [3]. Without measurement, it is challenging to take NFR as quantitative inputs into an estimation process and productivity benchmarking.

In the practice, the requirements are initially addressed typically at the system level [4-7] as either high-level system functional user requirement (system-FUR) or high level system non-functional requirements (system-NFR); such high level requirements must typically next be detailed and allocated to functions which may be implemented in either or both hardware and software, as software FUR (software-FUR) for instance.

For example, a system-FUR will describe what are the required functions needed in a system, while a system-NFR will describe how the required functions must behave in a system [8-10]; in the software requirements engineering step, such system-NFR may next be detailed and specified as software-FUR to allow a software engineer to develop, test and configure the final deliverables to system users.

"Functional" refers to the set of functions the system (including the software) is to offer, while "non-functional" refers to the manner in which such functions are performed. Functional user requirements (FUR) are typically phrased with subject or predicate constructions (i.e. noun/verb) such as: "The system design must include some of the software components to implement various parts/features of the system device". Non-functional requirements (NFR) are typically phrased with adverbs or modifying clauses, such as: "The system design must highly reuse existing software components behaviours that implement various parts/features of the system design".

Within the ECSS European standard for the aerospace industry [11-14] and the SWEBOK Guide [15], a number of concepts are provided to describe various types of candidate design and implementation (D&I) constraints at both the system, software and hardware levels. However, these standards vary in their views, terminology and coverage of D&I requirements.

Currently, there exists no generic model for the identification and specification of software-FUR for implementing system D&I constraints (system-NFR) from the various views documented in international standards and in the literature. Consequently, it is challenging as well to measure these D&I constraints-related software-FUR and to take them into account quantitatively for estimation purposes.

The motivation of this research project is to contribute to better define, describe and measure some of the NFR inputs required for adequate *a priori* cost estimation of software projects. The measurement scope in this paper is to identify separately all functionality allocated to D&I constraints for embedded and real time software, whether software has yet to be built or it has already been delivered.

The focus of this paper is on a single type of NFR that is, system D&I constraints. This paper reports on the work carried out to define an integrated view of software-FUR for system D&I constraints on the basis of international standards, and on the use of the generic COSMIC – ISO 19761 [15] model of software-FUR to measure their functional size.

This paper is organized as follows. Section 2 presents the structured view of software functional user requirements (software-FUR) in ISO 19761. Section 3 identifies the standards describing D&I constraints requirements. Section 4 presents a standard-based definition of a generic model of requirements for software to implement system D&I constraints. Section 5 presents a measurement example. Finally, a discussion is presented in section 6.

2 A generic View of Software-FUR in ISO

In the collection of ISO standards, it is specified in the ISO 14143-1 [16] that a functional size measurement method must measure the software functional user requirements (FUR). In addition, ISO 19761 – COSMIC [15] proposes a generic model of software-FUR that clarifies the boundary between hardware and software. Fig. 1 illustrates the generic flow of data from a functional perspective from hardware to software. From this generic model of software functional requirements in Fig. 1 the followings can be observed:

- Software is bounded by hardware. In the so-called "front-end" direction (i.e. center in Fig. 1), software used by a human user is bounded by I/O hardware such as a mouse, a keyboard, a printer or a display, or by engineered devices such as sensors or relays. In the so-called "back-end" direction (i.e. right-hand side of Fig. 1), software is bounded by persistent storage hardware like a hard disk and RAM and ROM memory.
- The software functionality is embedded within the functional flows of data groups. Such data flows can be characterized by four distinct types of data movements. In the "front end" direction, two types of movements (ENTRIES and EXITS) allow the exchange of data with the users across a 'boundary'. In the "back end" direction, two types of movements (READS and WRITES) allow the exchange of data with the persistent storage hardware.
- Different abstractions are typically used for different measurement purposes. In real-time software, the users are typically the engineered devices that interact directly with the software that is the users are the 'I/O hardware'. For business application software, the abstraction commonly assumes that the users are one or more humans who interact directly with the business application software across the boundary; the 'I/O hardware' is ignored.

As an FSM method, COSMIC is aimed at measuring the size of software based on identifiable FUR.

Once identified, those requirements are allocated to hardware and software from the unifying perspective of a system integrating these two “components”. Since COSMIC is aimed at sizing software, only those requirements allocated to the software are considered in its measurement procedure.

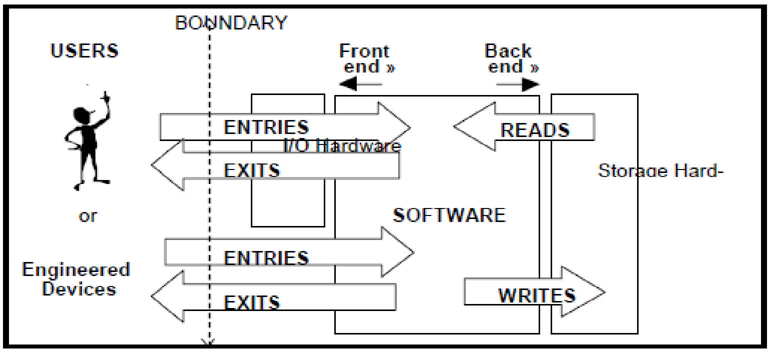


Fig. 1. Generic flow of data groups through software from a functional perspective in COSMIC – ISO 19761

3 *Identification of Standards for Describing D&I Constraint Requirements*

This section presents a survey of the D&I constraints views, concepts and terms in the ECSS standards. This section identifies which standards currently address some aspects of the software-FUR derived from system requirements. The expected outcome is the identification of the various elements that should be included in the design of a standard-based framework for modelling software-FUR for system D&I constraints.

3.1 D&I Requirements in ECSS standards

The elements of D&I constraints are dispersed in various system views throughout different ECSS standards and are expressed as either – see Fig. 2:

- System D&I constraints functional user requirements (system D&I constraints-FUR)
- System D&I constraints non-functional requirements (system D&I constraints-NFR)

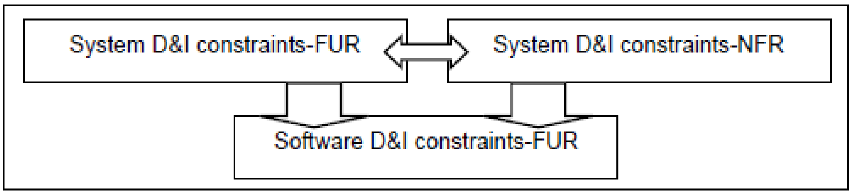


Fig. 2. Mapping system-requirements into software-FUR for D&I constraints

The identification of D&I constraints in the ECSS standards is derived from an analysis of the requirements on the system and its functions. All system requirements are allocated to a set of D&I constraints. Moreover, hardware configuration D&I constraints, software configuration D&I constraints, and human operations D&I constraints shall be subsequently identified from these requirements. The supplier shall transform the requirements for the software D&I constraints into an architecture that describes its top-level structure and identifies the software components, ensuring that all the requirements for the software D&I constraints are allocated to its software components and later refined to facilitate detailed design.

The software architectural design shall describe the D&I constraints within:

- The static architecture (i.e. decomposition into software elements such as packages and classes or modules),
- The dynamic architecture, which involves active objects such as threads, tasks and processes, and
- The mapping between the static and the dynamic architecture, and the software behaviour.

The software D&I constraints requirements shall produce the physical model of the software components described during the software architectural design. For embedded software D&I constraints the following information should be included:

- Type of D&I constraints participating to the real time behaviour, described by stating its logical and physical characteristics with D&I,
- Scheduling types with D&I (e.g. single or multi-threads),
- Scheduling model with D&I (e.g. pre-emptive or not, fixed or dynamic priority based),
- Analytical model with its D&I (e.g. rate monotonic scheduling, deadline monotonic scheduling),
- Tasks identification and D&I priorities,
- Communication and synchronization with D&I,
- Time management through D&I,
- The dependencies of a component should be described by listing the D&I upon its use by other components.

The ECSS-ESA document [17] covers the tailoring of the ECSS-E-40 requirements for the European Space Agency (ESA) software projects. In this document, the software design includes a program design, pseudo-code and flow charts. Software D&I may specify that the processing has to be performed using a particular algorithm and program parameters.

Table 1 presents a list of concepts and vocabulary used in ECSS to describe system related D&I constraints requirements and ECSS mentions that such requirements may be implemented in software.

Table 1. D&I constraints view and vocabulary in ECSS

Key view	Concepts and Vocabulary
Design and implementation (D&I) constraints applicable to various components of the system product	• Software architectural D&I constraints on modules, classes, packages. • Software detailed D&I constraints on tasks and processes. • Physical model of the software D&I constraints described during the software architectural design. • The logical model of the D&I constraints described during the software architectural design.

While conducting the survey of all the D&I constraints concepts and terms described in the ECSS-E-40 and ECSS-Q-series and in ECSS-ESA as the integrated standard for ECSS-E and ECSS-Q, it was observed that:

- These various D&I constraints are described differently, and at different levels of detail within the system design;
- The D&I constraints within the system design are dispersed throughout the various documents: there is, therefore, no integrated view of all types of candidate D&I constraints requirements;
- There is no obvious link for the D&I constraints requirements in ECSS-ESA as the integrated standard and between all other ECSS standards that describe D&I constraints requirements within their system design or within their different ECSS standards contents.
- With regards to European standards, software D&I constraints requirements can be measured within:
  - Static architectural D&I constraints, including modules, classes and packages. and
  - Dynamic architectural D&I constraints including tasks and processes.
- It is also to be noted that ECSS does not propose a way to measure such D&I constraints requirements and, without measurement, it is challenging to take such an NFR as a quantitative input to an estimation process or in productivity benchmarking.

3.2 D&I Constraints Requirements in the SWEBOK Guide

According to the SWEBOK Guide (ISO 19759) [15] ; “Software requirements express the needs and constraints placed on a software product that contribute to the solution of some real-world problem". ISO 19759 mentions explicitly D&I constraints as non-functional requirements in the “Software Requirements’ knowledge area (KA) and implicitly within the context of activities for design in the ‘Software Design’ KA.

Software design is defined in ISO 19759 [15] as both “the process of defining the architecture, components, interfaces, and other characteristics of a system or component" and “the result of [that] process". Furthermore software design in the software engineering life cycle is defined as activities in which software requirements are taken as inputs for analysis in the software design phase.

The architectural design is also described by ISO 19759 as the point at which the requirements process overlaps with software or systems design and illustrate how challenging it is to cleanly decouple the two tasks; software architecture is “a description of the subsystems and components of a software system and the relationships between them”. This means that ISO 19759 is describing the D&I constraints in the Software Requirements KA and these D&I constraints should be reflected on the software design in the Software Design KA.

Moreover, software design consists of two activities that fit between software requirements analysis and software construction:

- Software architectural design (sometimes called top level design): describing software’s top-level structure and organization and identifying the various components.
- Software detailed design: describing each component sufficiently to allow for its construction.

ISO 19759 is decomposing the software D&I constraints into processes, tasks, and threads and deals with related efficiency, atomicity, synchronization, and scheduling issues.

Table 2 presents a list of concepts and vocabulary used in the ISO 19759 to describe system related D&I constraints. The SWEBOK Guide specifies that such requirements be implemented in software design.

Table 2. ISO 19759 views and vocabulary for D&I constraints

Key view	Concepts and Vocabulary
Software requirements express the needs and constraints placed on a software product that contribute to the solution of some real-world problem	• Software architectural D&I constraints on modules, classes, packages or top level structure. • Software detailed D&I constraints on tasks and processes. • Physical model of the software D&I constraints described during the software architectural design. • The logical model of the software D&I constraints described within the software architectural design. • Static and dynamic D&I constraints with system design.

While conducting the survey of all the D&I constraints concepts and terms within system design described in ISO 19759, it was observed that:

- These various D&I constraints are described with system design elements differently, and at different levels of details;
- Measures can be used to assess or to quantitatively estimate various aspects of a software design’s size, structure, or quality.
- Most measures that have been proposed generally depend on the approach used for producing the design.
- These measures are classified into two broad categories:
  - Function-oriented (structured) D&I constraints measures: the D&I constraints structures obtained mostly through functional decomposition; generally represented as a structure chart (sometimes called a hierarchical diagram) from which various measures can be computed.
  - Object-oriented D&I constraints measures: the design’s overall structure is often represented as a class diagram, from which various measures can be computed. Measures on the properties of each class’s internal content can also be computed.

4 A Standard-Based Generic Model of Software-FUR for D&I Constraints Requirements

This section identifies first the terminologies and concepts of D&I constraints dispersed throughout ECSS series and ISO 19759 and assembles next these terminologies into a proposed model of D&I constraints software-FUR, through the use of the generic model of FUR proposed in the COSMIC model. This COSMIC-based generic model can then become a framework for describing the software-FUR from system D&I constraints based on ECSS and ISO 19759.

4.1 D&I Constraints Requirements and Functions to be specified

The types of system D&I constraints can be derived from the physical and logical models; these models include:

- The static design and its D&I constraints,
- The dynamic design and its D&I constraints,
- The mapping between both the static and the dynamic design and its D&I constraints views,
- The behaviour of the system design before and after implementation.

The functions to be specified (and corresponding entities to be measured) are divided into external and internal constraints functions - see table 4. The Internal D&I constraints refer to the expected logical D&I constraints that could appear from the system behaviour, while the External D&I constraints refer to the expected physical D&I constraints.

Table 3. Software D&I functions

Internal D&I constraints	Internal D&I constraints on module(s)	• Module(s) • Process(s) • Channel(s) • Event(s)
External D&I constraints	External D&I constraints on channels	• Module(s) • Process(s) • Channel(s) • Event(s)

4.2 Relationships across Function Types

This section identifies the function types and functional relationships in the software-FUR for system D&I constraints requirements.

D&I constraints functional type 1: Internal D&I constraints on modules.

- Any Process 1 to n can send and receive at least one data group to/from any internal channel or events in the same design module.
- Any internal channel or events can send and receive at least one data group to/from any other process in the same design module.

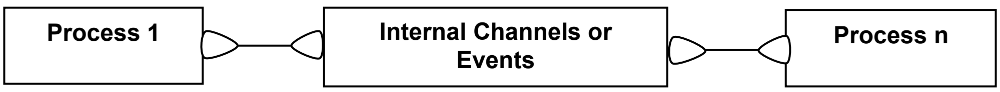


Fig. 3. Internal D&I constraints on modules

D&I constraints functional type 2: External D&I constraints on channels

- Any process in module 1or n can send and receive at least one data group to/from any external channel.
- Any external channel can send and receive at least one data group to/from any other process the different modules.

Fig. 4. External D&I constraints on channels

4.3 Model of Function Types Relationships

Using the COSMIC model for graphical representation, Figure 5 presents an overview of the relationships between the functional types in the D&I constraints software-FUR. More specifically:

- The sub-model of internal D&I constraints functional type 1 can be used to specify (and to measure the functional size of) the internal D&I constraints for the processes and the internal channels or events from the received/send data movements from/to any other processes and internal channels in the same module – See figure 5.

- The sub-model of external D&I constraints on channels functional type 2 can be used to specify (and to measure the functional size of) the external D&I constraints for the external channels from the received/send data movement from/to any other processes in different modules – See figure 5.

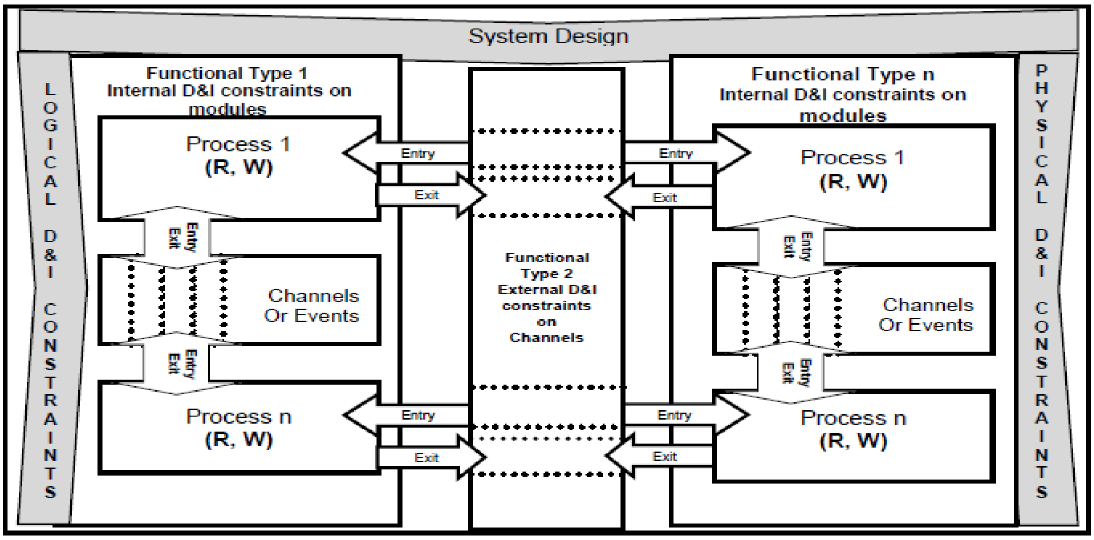


Fig. 5. COSMIC generic model of D&I constraints requirements allocated to software

4.3.1 A Generic D&I constraints requirements services in the Service Oriented Architecture in system-FUR view

This model is referred here as a generic model of software-FUR for system D&I constraints.

- The internal D&I constraints in modules (functional type 1 in Fig. 5): Each module may have many processes, each process may interact using an internal channel or event (for example, through an RPC or remote procedural call) for an internal connection; in this case the processes should be considered as a storage device for such kind of information before data marshalling between the other processes - see also Fig. 6
- The external D&I constraints on channels (functional type 2 in Fig. 5): many modules may interact with each other through their own processes. In this case many processes in different modules may use external channels (for example: through an RMI or a remote method invocation) for external connection - see also Fig. 6
- Process 1.1 starts sending to process 1.n in module 1 (for example process 1.1 represents function and process 1.n represents a sub-function in the same module).
- Process n.1 should start sending to interact process n.n in a module 2 (for example process n.1 represent function n and process n.n represent sub-function in the same module).

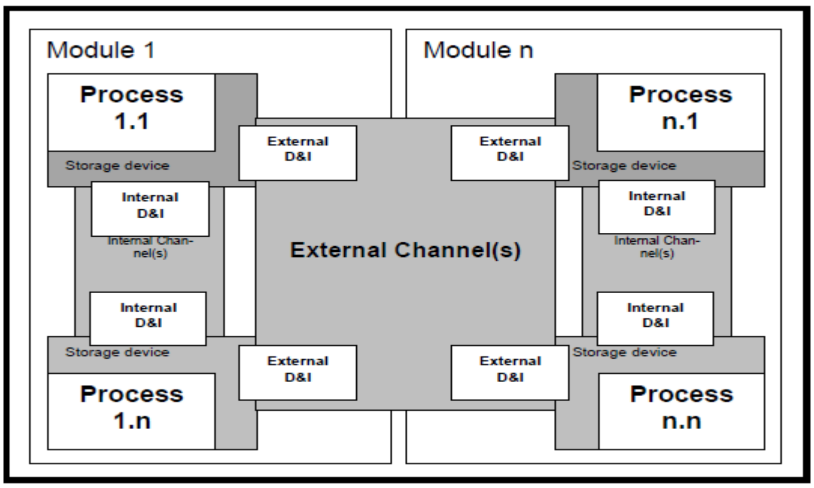


Fig. 6: Generic D&I constraints requirements allocated to software

4.3.2 A Generic D&I constraints requirements data movements of data exchanges between components in Software-FUR view

Fig. 7 and 8 below shows the possible flows of data movements between components, the exchanged of data between components could be direct or indirect exchange movements of data movements to provide the functional user with services.

Fig. 7 shows each process or (component) in the figure could be exchange the data directly to provide services to the functional user, so in this case for the measurements uses, we identify Entry and/or Exit data movements.

Fig. 8 shows indirect exchange of data between a processes which means that a service in one process writes data which is subsequently read by another process. In this situation identify a Write data movement in the next process and a Read data movement by the latter.

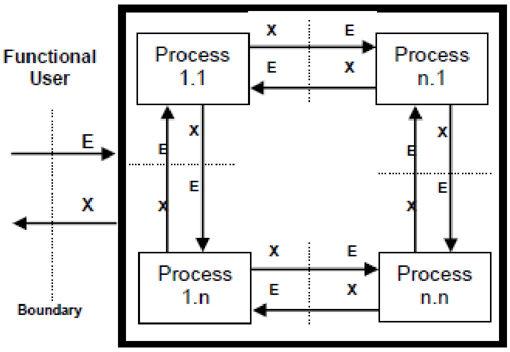


Fig.7: Direct Data Movements

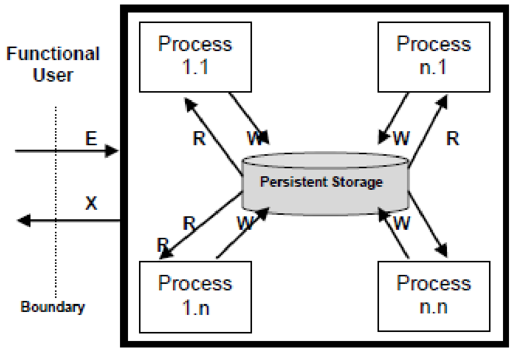


Fig. 8: Indirect Data Movements

5 Discussion

This paper has introduced a procedure for specifying and measuring the software requirements for the internal and external D&I constraints needed to address the system non functional requirements for system D&I constraints.

The main contribution of this paper is the proposed generic Model of software-FUR for system D&I constraints based on ECSS, ISO and IEEE standards. This generic model can be considered as a kind of reference model for the identification of system D&I constraints requirements and their allocation to software functions implementing such requirements. System requirements allocated to hardware have not been addressed in this paper.

Since the structure of the generic model is based on the generic model of software adopted by the COSMIC measurement standard, the necessary information for measuring their functional size is readily available and an example has been presented of a specific instantiation of this reference model.

The model is independent of the software type and the languages in which the software FUR will be implemented. The proposed generic D&I constraints model (i.e. reference model) provides:

- A specification model for each type, or all types, of D&I constraints requirements.
- A specification measurement model for each type, or all types, of D&I constraints requirements.

Future work includes documentation of the traceability of the elements of this generic model to the detailed elements of the ECSS standard as well verification of this generic model to ensure full coverage of design and implementation constraints requirements.

There is no claim that this current version of the generic model of design and implementation constraints requirements covers the full domain of D&I constraints as found in practice and as discussed in the software design literature outside of standards. Discussions with group of experts are necessary to ensure its usefulness across various communities and to develop a consensus on further refinements of such a generic model which could be proposed eventually as a candidate for standardization.

References

1. L. Chung and J. Cesar Prado Leite, "On Non-Functional Requirements in Software Engineering", in "Conceptual Modeling: Foundation and Applications, Essays in Honor of John Mylopoulos", Springer, 2009.
2. L. Chung, B. Nixon, E. Yu, J. Mylopoulos, "Non-Functional Requirements in Software Engineering", Springer, Heidelberg,1999.
3. J. Mylopoulos, L. Chung, B. Nixon, "Representing and Using Nonfunctional Requirements: A Process-Oriented Approach", IEEE Transactions on Software Engineering, vol. 18, pp. 483-497, 1992.
4. M. Shaw, "Larger Scale Systems Require Higher-Level Abstractions: Software Specification and Design", IEEE Computer Society, vol. 14, pp. 143-146, 1989.
5. A. M. Davis, "Software requirements: objects, functions, and states"; Prentice-Hall, Inc., 1993.
6. I. Jacobson, G., Booth, J.,Rumbaugl, "Excerpt from the Unified Software Development Process: The Unified Process", IEEE Software, vol. 16, pp. 96-102, 1999.
7. K. Wiegers, "Software Requirements", 2nd edition, Microsoft Press, 2003.
8. G. Roman, "A Taxonomy of Current Issues in Requirements Engineering", IEEE Computer, pp. 14-21, 1985.
9. B. W. Boehm, "Characteristics of software quality", Amsterdam, New York, North-Holland Pub. Co. , American Elsevier, 1978.
10. A. I. Antón, "Goal identification and refinement in the specification of software-based information systems", PhD Thesis, Georgia Institute of Technology, 1997.
11. ECSS-E-40-Part-1B, "Space Engineering: Software - Part 1 Principles and Requirements", European Cooperation for Space Standardization,The Netherlands, 2003.
12. ECSS-E-40-Part-2B, "Space Engineeing: Software- part 2 Document Requirements Definitions", European Cooperation for Space Standardization, The Netherlands, 2005.
13. ECSS-Q-80B, "Space product assurance: Software product assurance", European Cooperation for Space Standardization, The Netherlands, 2003.
14. ECSS-E-ST-10C, "Space engineering: System engineering general requirements", Requirements & Standards Division Noordwijk, The Netherlands, 2009.
15. ISO-19759, "Software Engineering Body of Knowledge (SWEBOK)", IEEE Computer Society, & ISO, 2004.
16. ISO/IEC-14143-1, "Information Technology - Software Measurement - Functional Size Measurement Part 1: Definition of Concepts", International Organization for Standardization, Geneva (Switzerland), 1998.
17. ECSS-ESA, "Tailoring of ECS: Software Engineering Standards for Ground Segments, Part C: Document Templates", ESA Board of Standardization and Control (BSSC), 2005.

Author CVs

Alain Abran is a Professor and the Director of the Software Engineering Research Laboratory at the École de Technologie Supérieure (ETS) – Université du Québec. Dr. Abran has more than 20 years of industry experience in information systems development and software engineering. Dr. Abran holds a Ph.D. in Electrical and Computer Engineering (1994) from École Polytechnique de Montréal (Canada) and master degrees in Management Sciences (1974) and Electrical Engineering (1975) from University of Ottawa. He was the co-executive editor of the Guide to the Software Engineering Body of Knowledge project – SWEBOK – ISO 19759. He has also been actively involved in software engineering standards as the international secretary for ISO/IEC JTC1 SC7 - Software and System Engineering in 2001-2003; he is also the chairman of the Common Software Measurement International Consortium (COSMIC).

Khalid T. Al-Sarayreh received the B.S. degree in Computer Science from Mu'tah University, Jordan, the MSc in Computer Engineering (Embedded Systems) from Yarmouk University, Jordan, PhD in Computer Information Systems from the Arab Academy for Financial and Banking Sciences, Jordan. PhD. Student in Software Engineering in university of Quebec (ETS) from 2008 Canada. During 2002-2005, he stayed in the KADDB,. From 2005 to 2006, he was an Assistant Professor at the Jordan University. Since 2006-2008, He is as an Assistant Professor at the Faculty of Information Technology, Applied Science University/ His research interests includes: Software Real-Time Embedded Systems, Computer Networks, non-functional requirements for embedded systems and Applied Artificial Intelligence.

A Framework of the Factors that Influence the Software Process Improvement in Small Organizations

Ismael Edrein Espinosa-Curiel, Josefina Rodríguez-Jacobo, and José Alberto Fernández-Zepeda
Department of Computer Science, CICESE
Carretera Ensenada-Tijuana #3918 Ensenada, B.C. 22860, MEXICO
{ecuriel, jacobo, fernan}@cicese.mx

Abstract

Today, many small and medium-sized software enterprises (SMEs) have initiated a software process improvement (SPI) initiative to be more competitive in the software market. The SPI initiatives have not been entirely successful because a large number of social, technical and human factors make them a long, expensive and difficult process. Additionally, there is a lack of information regarding the characteristics and behaviors of these factors. This situation usually limits the SPI managers to design strategies to control, mitigate or solve the factors when they have a negative effect on the SPI initiatives. This work proposes a framework of factors that influence the SPI initiative in SMEs. This framework could be used by SPI leaders to identify potential threats to the SPI initiatives and to propose better SPI strategies.

Keywords

Software process improvement, SPI, factor framework, social factors, personal factors, technical factors, small software organization, SMEs.

1 Introduction

Today, the small and medium-sized software enterprises (SMEs) play an important role in the world economy [1]. The competition in the software market encourages many of such organizations to start a software process improvement (SPI) initiative. The goal of this initiative is to increase the quality and productivity of the software development process of an organization and to reduce its associated times and costs [2]. Recently, a number of processes reference models (PRM) have been proposed to guide and promote the SPI initiatives. Some of the most well-known models are: ISO 9001:2000, CMMI, ISO/IEC 15504:2004, ISO/IEC 12207:2004 and MoProSoft. The SPI initiatives in SMEs have achieved some promising results; however, they have not been entirely successful [3-7].

Besides of the changes required in the processes, tools and software development methodologies, an SPI initiative also requires deep organization changes in the politics, the structure, and the culture [8]. All these changes make it an expensive, difficult and long process [9]. In many cases, this situation and the severe restrictions regarding budget and number of employees, force the SMEs to quit SPI initiatives [10]. Additionally, to increase the probability to successfully complete these changes, it is important to consider a set of knowledge, competences, behaviors, attitudes, perceptions, feelings, and activities at personal, social and organizational levels. In this paper, we refer to all these elements with the word “factors”.

Many researchers have identified a large number of the factors that influence the SPI initiatives in software development organizations [11-14]. However, there is a lack of information about the characteristics and behavior of these factors. This situation limits the success of SPI initiative because the SPI managers usually fail to design strategies to control, mitigate or solve the factors when they have a negative effect on the SPI initiatives [15]. Although some models and methodologies have been proposed to guide the SPI initiatives, they do not clearly define how to make the organizational, social and personal changes, and do not clearly specify the factors to consider in an SPI initiative. This situation motivates this research and raises the following questions:

- *What are the factors that influence SPI initiatives and what are their characteristics?*
- *How can we represent the factors and their characteristics to facilitate the SPI initiative leader the design of SPI strategies?*

To answer these questions we first identified, collected, classified, and analyzed the factors that influence the SPI initiatives. Then, we conducted three case studies in three software development SMEs. We combined qualitative and quantitative techniques to collect and analyze data related to these factors, and identified and evaluated their properties. Later, we develop a framework of factors that influence the SPI initiatives in SMEs. A framework of this type is a set of factors, properties and evaluation criteria that are related and reflects the behavior of the variables that are present in an SPI initiative. An SPI leader can use this framework to foresee potential threats to the SPI initiative and to define better SPI strategies.

This paper is organized as follows. Section 2 reviews the empirical studies that identified factors that influence SPI initiatives. Section 3 briefly describes our methodology. Section 4 describes the way we classified the factors. Section 5 describes the factor properties. In Section 6 we describe the proposed SPI factor framework. Section 7 describes when and how to use the framework and some scenarios where it can be applied. Section 8 presents the conclusions and directions of our future work.

2 Previous work

Researchers have conducted several studies in different countries and in a variety of software development organizations to identify the factors that influence the SPI initiatives. These studies focused on the adoption of different PRMs from the point of view of the software engineering roles; these studies used different data collection and analysis techniques. Next, we discuss some of them. Some researches [1, 5, 6, 13, 16-24] identified a great number of organizational factors that have a strong influence on SPI initiatives. The key organizational factors are: the understanding of the characteristics of the SPI project, the motivation and commitment of the management to achieve the improvement project; the resources assigned to the SPI project; the expected results in short and long term; the efficiency of the organizational mechanisms (*e.g.* the communication, knowledge management, definition of roles and responsibilities, and the promotion of the innovation, creativity and organizational change); and the internal and/or external social, political and financial factors that threaten the existence of the organization.

Additionally, many researches [4, 12, 16, 25, 17, 26] identified SPI project related factors. The key factors are: the efficiency of the planning, implementation, monitoring and evaluation of the SPI project; the duration and cost of the SPI project; the definition and realism of its objectives; the visibility of its benefits; the systematization, use and exploitation of the experience and knowledge; the magnitude of the required changes; the planning of the organizational change; the clarity and accuracy of the specification of the roles and responsibilities; and the identification and control of personal conflicts. Other factors are related to the existence and efficiency of the SPI project mechanisms (*e.g.* communication, motivation and reward, measure performance, resistance to change management, training, etc.); the ease of implementation, comparative advantage, the efficiency, and adequacy of the new processes to the needs and characteristics of the organization.

Other studies [4, 5, 12, 13, 17-20, 16, 25-29] identified that the personal factors have a strong influence on the SPI initiative. The key personal factors are: the attitudes and the behaviors of the stakeholders in the SPI initiative (*e.g.* the motivation, involvement, commitment, resistance to change, etc.); the competences of the stakeholders regarding the SPI and software engineering (*e.g.* knowledge, experience, training, etc.); the collaboration and effort among the employees; the workload of the employees; and the level of conflict between their responsibility regarding the SPI project and their daily activities.

Some researchers have proposed different ways to classify the factors that influence SPI initiatives. Demirörs and Demirörs [10] classified them into three categories: organizational structure, quality models, and the market in which the SMEs operate. Dybå [30] grouped them into six categories: business orientation, leadership involvement, concern for measure, exploitation of existing knowledge, and exploration of new knowledge. Christiansen and Johansen [31] grouped them into four categories: initiation, projects, in use, and organizational foundation. Beechman *et al.* [4] grouped them into three categories: organizational issues, project issues, and software development life cycle issues.

These classifications include a limited number of factors; since one of our objectives is to classify a large number of factors, we expand the classification of Beechman *et al.* [4]. We present our expanded classification in Section 4. Even more, most of the work described above identified the factors that influence the SPI initiatives; however, none of them explicitly describes their characteristics. We used the proposed classification (Section 4) together with the description of the characteristics of each factor (Section 5) to propose an SPI factor framework (Section 6) that can be helpful in the design of SPI initiatives (Section 7).

3 Methodology

We have used a multi-strategy approach for this study. First, we reviewed empirical studies in the SPI field to identify factors that influence the SPI initiatives. We reviewed 24 case studies of SPI and identified 132 factors. Secondly, we conducted three case studies in three Mexican SMEs that have initiated an SPI project. The objective of these case studies was to study the properties and behaviors of factors that affect SPI initiatives in the context of SMEs. The duration of each case study was two years. There were a six-month offset in the starting point of each case study; this offset allowed us to improve the working framework and testing the preliminary results obtained in the previous case study. There were 47 practitioners (developers, project managers, quality managers, and top managers) involved in the three case studies.

In these case studies, we applied action research methodology [32]. Action research describes a way to understand social communities, their activities, the aspects that influence on a particular situation, to take action in close cooperation with those people involved in the situation in order to improve it [33]. The purpose of action research is “influence or change some aspect of whatever is the focus of the research” [34]. The cycle of action research consists of five stages: the diagnosing, the action planning, the action taking, the evaluating, and the specifying learning. Additionally, we used observation, interviews and questionnaires to support action research, to identify the factors, their properties, and to establish a numerical value that is proportional to the “intensity” of each property in each factor. We regularly interviewed 30 people involved in the SPI initiative, including the managers, the SPI project leaders and the software engineers. We interviewed each person approximately twice per month for two years. We used grounded theory [35] to analyze the resulting information.

To increase the precision of the research results and to ensure their validity in the case studies, we used triangulation [36]. Triangulation means to study an object from different angles to provide a broader picture. We used two types of triangulation; first, data source triangulation because we have three data sources (organizations) which provide a great amount of information; secondly, methodological triangulation, because we combined qualitative (field observations and interviews) and quantitative (questionnaires) techniques to collect and to analyze data.

4 Factor classification

In this research, we identified 132 factors that affect SPI initiatives. We grouped them according to their origins. This classification is based on the work of Beecham *et al.* [4] with the following changes: first, we extracted the personal factors from the organizational factors and created a new category; second, we added the SPI project factors category; third, we added the SPI model factors category; fourth, we merged their project issues and software development life cycle issues categories into the software development process category; finally, we added the SPI team factors category. Additionally, we added sub-classifications to some categories to clarify the relationship among factors. Next, we describe our proposed categories.

1. **Organizational factors.** Set of factors related to the characteristics and status of the organization. This set includes the elements that govern the daily activities and that support the SPI initiative. A

subset of this category includes the factors that directly influence the SPI project. Some factors in this subset are the SPI project priority, its budget and allocated times.

2. **SPI Project factors.** Set of factors related to the characteristics of the SPI project. This category includes the efficiency of the SPI project management and the mechanisms necessary to perform the improvement project. A subset of this category consists of the SPI project factors that directly influence the organization. Another subset consists of the SPI project factors that directly influence the people.
3. **Personal factors.** Set of factors related to the competencies, characteristics and emotions that people show when they perform their daily activities. This set includes the skills, knowledge and capabilities that contribute to the success of the organization, and the feelings and expectations that they have regarding the organization (trust, motivation, organizational commitment, etc.). A subset of this category consists of those personal factors that directly influence the SPI project.
4. **SPI team factors**. Set of factors related to the characteristics of the SPI team. This set includes team integration, mechanisms of communication and motivation, and the degree to which the people involved in the SPI team have the right profile, including the SPI project leader.
5. **Factors related to PRM**. Set of factors generated by the characteristics of the PRM that is going to be implemented in the SPI project. This set includes the amount of knowledge, the required organizational changes, the available information of the PRM, the existence of implementation guidelines, and documented successful implementation cases.
6. **Factors related to software development process**. Set of factors related to the status of the software development process and the amount of changes required to satisfy the specifications of the PRM. This set includes the software development phases: requirements, design, coding, testing and maintenance.

5 Factor Properties

Once we classified the factors, we identified and evaluated their properties during the case studies. The properties identified are: stage incidence, approach, impact, complexity, duration, cost, persistence, chain reaction, identification, and communication (see Table 1). We describe these properties and the evaluation procedure:

A. ***Stage incidence***. Refers to the importance of the effect of a factor on a specific stage. First, based on the classification of Espinosa-Curiel [37], we divided the SPI project in three stages: pre-adoption (σ_1), general use (σ_2), and continuous use (σ_3). The goal of the *pre-adoption stage* is to clarify for individuals, groups and organization, the need for change; so they can understand and accept that a change must occur and it is feasible. The *general use stage* comprises the processes of change through which the people involved learn and introduce new behaviors. It includes training, establishment of new working procedures and relationships, determining the vision, objectives, strategies and action plans to be developed. The *continuous use stage* involves the repetition of the behavior until it becomes a new habit. To determine the level of incidence of a factor in each stage, we calculated the frequency of occurrence of the factor in the analysis of transcripts of observations and interviews conducted at that stage. Therefore, the higher the factor frequency in the stage, the higher its presence and the need to control, mitigate or solve it. The value scale is: 1 = very low, 2 = low, 3 = medium, 4 = high, and 5 = very high.

B. ***Approach****.* To control, mitigate or solve the factors, we can use the combination of three different approaches: the *technical approach* (T) which focuses on the knowledge and skills; the *social approach* (S) which focuses on social interactions and relationships of individuals; and the *personal approach* (P) which focuses on feelings, behaviors and perceptions of the individuals. Therefore, depending on the characteristics of each factor and its scope, we establish the order (π_1, π_2, π_3) in which we apply each of these approaches; additionally, we use a digit (α, β, γ) to explicitly indicate the "percentage of the amount of work" for each approach. The value scale is: $1 \leq \alpha, \beta, \gamma \leq 8$ and $\alpha+\beta+\gamma = 10$.

C. ***Impact****.* Refers to the magnitude of the effect of the factor on the completeness of the SPI initiative. The impact is positive when the factor is controlled, mitigated or solved; it is negative otherwise. When the impact is positive, the factor helps the SPI initiative; otherwise, the factor damages the initiative. One factor may affect the following elements: people (ε_1), organization (ε_2), improvement project (ε_3), and development process (ε_4). To determine the impact of a factor in each element, we considered the number and magnitude of changes required to solve, control, or mitigate that factor, and the difficulties generated if it is not solved. We also considered the opinion of stakeholders about how the factor affects the elements. The value scale is 1 = slight, 2 = moderate, and 3 = severe.

D. **Complexity.** Refers to the amount of knowledge and experience required to control, mitigate or solve the factor. To determine the complexity of factors, we considered the required training, the amount of information available, and the opinion of stakeholders about the difficulty to control the factor. The value scale is 1 = easy, 2 = moderate, and 3 = difficult.
E. **Duration.** Refers to the work and/or activities and/or the time required to control, mitigate or solve the factor. To determine the duration of the factor, we estimated the number of man-hours used to solve the factor. The value scale is 1 = short, 2 = moderate, and 3 = long.
F. **Cost.** Refers to the amount of resources needed to control, mitigate or solve the factor. To determine the cost, we considered the costs of training, tools, resources, and the number of man-hours used to control this factor. The value scale is 1 = cheap, 2 = moderate, and 3 = expensive.
G. **Persistence.** Degree of ease to which a factor that was already controlled mitigated or solved, “appears” again. To determine the persistence of a factor, we calculated the number of its recurrences in the analysis of transcripts of interviews and observations. The value scale is: 1 = infrequent, 2 = frequent, and 3 = very frequent.
H. **Chain reaction.** Degree of influence of a factor on other factors in the organization. To determine this property, we observed that when a factor was mitigated or controlled, it was easy to solve some other factors; conversely, to solve a particular factor, it was necessary to first solve other factors. The value scale is 1 = few, 2 = moderate, and 3 = many.
I. **Identification.** Degree of ease for identifying when a factor is present in the organization. To determine this property, we considered the observations and the opinions of stakeholders about the difficulty to view and to be aware of the causes and effects of the factor in the elements of the SPI initiative. The value scale is 1 = easy, 2 = moderate, and 3 = difficult.
J. **Communication.** Degree of ease for communicating the causes and effects of the factor in the organization. To determine this property, we considered the observations and the opinion of stakeholders about the difficulty to explain or understand the causes and effects of the factor in the elements of the SPI initiative. The value scale is 1 = easy, 2 = moderate, and 3 = difficult.

6 Proposed SPI factor framework

The SPI factor framework[1] specifies, in an integrated way, a set of factors, properties and their evaluation criteria that are related and reflects the behavior of the factors that are present in an SPI initiative. The framework includes 132 factors grouped into six categories. For each factor, the framework specifies a concise definition and the properties (see Table 1 and Section 5).
The literature mentions that many of the factors included in this framework also affect large organizations; however, we believe that the values of the properties, behaviors, and impact of the factors may vary depending on the size of the organization. Therefore, it would be useful additional studies to evaluate these values in the context of large organizations.

Table 1. Example of a factor contained in the framework.

SPI FACTOR	PROPERTIES																
	A			B			C				D	E	F	G	H	I	J
	σ_1	σ_2	σ_3	π_1	π_2	π_3	ε_1	ε_2	ε_3	ε_4							
Knowledge management of the organization. Level of systematization, use and exploitation of experience, knowledge and documents in the organization.	1	5	4	T2	S4	P4	3	3	3	2	2	2	2	3	2	1	1

Next, we describe an example of the interpretation of a factor. The factor *knowledge management of organization* has a minimal incidence in the *pre-adoption stage* (σ_1), a very high incidence in the *general use stage* (σ_2), and a high incidence in the *continuous use stage* (σ_3). To solve this factor, follow the order (π_1,π_2,π_3) on column B; in this example we first apply a *technical approach* (T), then a *social approach* (S) and finally a *personal approach* (P). The percentage of work for these approaches is

[1] The framework is online on the following link: http://usuario.cicese.mx/~jacobo/proyecto/spi-factor-framework.pdf

α=20%, β=40% and γ=40%, respectively. This factor has a strong impact on people (ε_1), organization (ε_2), and on the improvement project (ε_3), and a moderate impact on the development project (ε_4). The complexity (D), the duration (E), and the cost (F) of the factor are moderate. This factor is very persistent (G). The chain reaction (H) of this factor is moderate. The identification (I) and communication (J) of this factor are very easy.

7 When and how to use the SPI factor framework

Organizations with new or existing SPI initiatives can use the proposed framework. When the SPI process is in progress, the framework can help to identify, describe, modify or improve the current SPI strategy. This framework can be used by SPI manager, SPI change agents, evaluators, stakeholders, etc. Next, we describe three scenarios in which this framework may be useful.

1. ***Planning the SPI initiative.*** In this scenario, the SPI manager defines the SPI strategy to be used in the organization. The SPI strategy specifies the objectives, the activities to be performed, and the resources and time required by the initiative. Usually, the managers of the SPI initiative are not aware of the great amount of organizational, social and personal factors that affect the SPI initiative; therefore, they focus their efforts on changing methodologies, processes and tools of software development (contemplate only technical factors). This limited plan will affect the SPI initiative.
2. ***Implementing the SPI initiative.*** In this scenario, the SPI manager applies the planned strategy. Usually, when an organization implements an SPI strategy without considering many of the factors that are present in this process, many problems like the following occur: the resources allocated are insufficient, people do not have the training to perform their required activities, the stress increases because people have to make many changes in a very short time, people focus their efforts on troubleshooting the current situation and neglect future situations. Additionally, the stakeholders do not perceive that the SPI initiative is advancing or generating positive results.
3. ***Evaluating the SPI strategy.*** In this scenario, the SPI manager measures the difference between planned results and those obtained with the SPI initiative. Usually, the SPI manager realizes too late that the resulting plan has no realistic objectives and erroneous budgets, resources and time estimations, since he/she is not considering many of the factors that actually occur (especially the personal and social factors). Even more, the plan would also omit many important activities (such as integration, resistance to change management, motivation, commitment, etc.) that are fundamental to successfully implement the SPI initiative. Additionally, the SPI leader has little evidence to evaluate, to justify the resources invested, and to explain the results. For this reason, many organizations discard the SPI initiatives.

The proposed framework can help to partially solve or minimize many of the problems described in these scenarios. Next, we describe the procedure to use the proposed framework:

1. Identify the present SPI initiative stage of the organization (see Section 5).
2. Sort the factors according to their incidence on the present stage. It is convenient to keep track of the origins of each factor.
3. Determine the present control of each factor in the organization. We suggest that transform the description of the factors into statements and integrate them into a questionnaire. For each statement, assign a Likert rating scale of 5 points, where 1 is "strongly disagree", 5 is "strongly agree", and 3 is a neutral perception. Then, apply the questionnaire to the stakeholder of the SPI initiative. Finally, calculate the level of perceived control for each factor.
4. Eliminate from the list those factors that are already controlled (those with values higher than 3 in the previous step).
5. Rank the remaining factors according to the importance of the properties. The properties sorted according to their importance are the following: 1-incidence, 2-impact, 3-complexity, duration and cost, 4-persistence and chain reaction, 5-identification and communication. The organization can adjust this ordering depending on its priorities.
6. Review the list to determine the type of information required to control each factor.
7. Set the necessary control for each factor and the indicators to fulfill this control.
8. Define the activities necessary to control each factor and assign responsibilities.

8 Concluding remarks and future work

The objective of the SPI factor framework is to help organizations to identify the factors that influence their SPI initiatives and to design better SPI strategies. During an SPI initiative, it is necessary that the stakeholders are conscious of the impact and characteristics of the social, human and technical factors that affect or could affect the process improvement. It is important to define these strategies at the right moment to effectively mitigate, control or solve the factors. To define the strategies, the SPI leader should know the main factor characteristics: stage incidence, approach, impact, complexity, duration, cost, persistence, chain reaction, identification, and communication.
For future research, we have three short-term goals: (1) apply and evaluate the framework in some software development SMEs; (2) validate a set of instrument for assessing the factors and to assist the SPI leaders to identify the status of their SPI initiative; (3) develop a SPI initiative guide that specifies the activities required to adopt a PRM and that considers the factors included in this work. A long-term goal of our study is to develop a computer system that integrates the work developed in our short-term goals. This system would help the organization to define and control the SPI strategies.

References

1. Pino, F., García, F., Piattini, M.: Software process improvement in small and medium software enterprises: a systematic review. Software Quality Journal. 16, 237-261 (2008).
2. Paulk, M., Curtis, B., Chrissis, M., Weber, C.: Capability maturity model, version 1.1. Software, IEEE. 10, 18-27 (1993).
3. Batista, J., Figueiredo, A.D.D.: SPI in a very small team: a case with CMM. Software Process: Improvement and Practice. 5, 243-250 (2000).
4. Beecham, S., Hall, T., Rainer, A.: Software Process Improvement Problems in Twelve Software Companies: An Empirical Analysis. Empirical Software Engineering. 8, 7-42 (2003).
5. Fayad, M.E., Laitinen, M., Ward, R.P.: Thinking objectively: software engineering in the small. Commun. ACM. 43, 115-118 (2000).
6. Richardson, I.: SPI Models: What Characteristics Are Required for Small Software Development Companies? Software Quality - ECSQ 2002. págs. 100-113 (2002).
7. Kautz, K.: Software process improvement in very small enterprises: does it pay off? Software Process: Improvement and Practice. 4, 209-226 (1998).
8. Cao, G., Clarke, S., Lehaney, B.: Diversity Management and Organisational Change. Synergy Matters. págs. 61-65 (2002).
9. Mishra, D., Mishra, A.: Software Process Improvement Methodologies for Small and Medium Enterprises. Proceedings of the 9th international conference on Product-Focused Software Process Improvement. págs. 273-288 Springer-Verlag, Monte Porzio Catone, Italy (2008).
10. Demirörs, O., Demirörs, E.: Software Process Improvement in a Small Organization: Difficulties and Suggestions. Proceedings of the 6th European Workshop on Software Process Technology. págs. 1-12 Springer-Verlag (1998).
11. Al-Mashari, M., Zairi, M.: BPR implementation process: an analysis of key success and failure factors. Business Process Management Journal. 5, 87 - 112 (1999).
12. Baddoo, N., Hall, T.: De-motivators for software process improvement: an analysis of practitioners' views. Journal of Systems and Software. 66, 23-33 (2003).
13. Dybå, T.: Factors of software process improvement success in small and large organizations: an empirical study in the scandinavian context. SIGSOFT Softw. Eng. Notes. 28, 148-157 (2003).
14. Rainer, A., Hall, T.: Key success factors for implementing software process improvement: a maturity-based analysis. Journal of Systems and Software. 62, 71-84 (2002).
15. El-Emam, K., Goldenson, D., McCurley, J., Herbsleb, J.: Modelling the Likelihood of Software Process Improvement: An Exploratory Study. Empirical Software Engineering. 6, 207-229 (2001).
16. Kelly, D.P., Culleton, B.: Process Improvement for Small Organizations, (1999).
17. Baddoo, N., Hall, T.: Motivators of Software Process Improvement: an analysis of practitioners' views. Journal of Systems and Software. 62, 85-96 (2002).
18. Horvat, R.V., Rozman, I., Gyorkos, J.: Managing the complexity of SPI in small companies. Software Process: Improvement and Practice. 5, 45-54 (2000).
19. Dangle, K., Larsen, P., Shaw, M., Zelkowitz, M.: Software process improvement in small organizations: a case study. Software, IEEE. 22, 68-75 (2005).
20. Moitra, D.: Managing Organizational Change for Software Process Improvement. Software Process Modeling. págs. 163-185 (2005).
21. Lepasaar, M., Varkoi, T., Jaakkola, H.: Models and Success Factors of Process Change. Product Focused Software Process Improvement. págs. 68-77 (2001).
22. Villalón, J.A.C., Agustín, G.C., Gilabert, T.S.F., Seco, A.D.A., Sánchez, L.G., Cota, M.P.: Experiences in the

Application of Software Process Improvement in SMES. Software Quality Control. 10, 261-273 (2002).
23. Zeineddine, R., Mansour, N.: Software Quality Improvement Model for Small Organizations. Computer and Information Sciences - ISCIS 2003. págs. 1027-1034 (2003).
24. Montoni, M., Rocha, A.: A Methodology for Identifying Critical Success Factors That Influence Software Process Improvement Initiatives: An Application in the Brazilian Software Industry. Software Process Improvement. págs. 175-186 (2007).
25. Mohd Hairul Nizam Md. Nasir, Ahmad, R., Hassan, N.H.: Resistance factors in the implementation of software process improvement project. Information Technology, 2008. ITSim 2008. International Symposium on. págs. 1-10 (2008).
26. Dangle, K., Larsen, P., Shaw, M., Zelkowitz, M.: Software process improvement in small organizations: a case study. Software, IEEE. 22, 68-75 (2005).
27. Valtanen, A., Sihvonen, H.: Employees' Motivation for SPI: Case Study in a Small Finnish Software Company. Software Process Improvement. págs. 152-163 (2008).
28. Biró, M., Messnarz, R.: Process improvement lessons learnt in small and large organizations worldwide. Software Process: Improvement and Practice. 13, 297-299 (2008).
29. Wangenheim, C.G.V., Weber, S., Hauck, J.C.R., Trentin, G.: Experiences on establishing software processes in small companies. Information and Software Technology. 48, 890-900 (2006).
30. Dyba, T.: An Instrument for Measuring the Key Factors of Success in Software Process Improvement. Empirical Software Engineering. 5, 357-390 (2000).
31. Christiansen, M., Johansen, J.: ImprovAbility guidelines for low-maturity organizations. Software Process: Improvement and Practice. 13, 319-325 (2008).
32. Lewin, K.: Frontiers in Group Dynamics: Concept, Method and Reality in Social Science; Social Equilibria and Social Change. Human Relations. 1, 5-41 (1947).
33. Dittrich, Y., Rönkkö, K., Eriksson, J., Hansson, C., Lindeberg, O.: Cooperative method development. Empirical Software Engineering. 13, 231-260 (2008).
34. Robson, C.: Real world research. Wiley-Blackwell (2002).
35. A. Strauss, J. Corbin: Basics of Qualitative Research: Techniques and procedures for developing grounded theory. Sage, CA (1998).
36. Runeson, P., Höst, M.: Guidelines for conducting and reporting case study research in software engineering. Empirical Softw. Engg. 14, 131-164 (2009).
37. Espinosa-Curiel, I.: Modelo de adopción de la norma mexicana del software MoProSoft, en las MIPYMES., Master Thesis, CICESE, Mexico.(2008).

Authors CVs

Ismael Edrein Espinosa-Curiel

He received a BS degree in Computer Science in 2005, a MS in Computer Science in 2008. He is currently a Ph.D. student in Computer Science at CICESE. His research interests focus on software process improvement in small companies, software engineering, knowledge management, and the impact of social and personal factors in software engineering.

Josefina Rodríguez-Jacobo

She received a BS degree in Physics in 1986, a M.S in Computer Science in 1993 and Doctoral degree in Psychology in 2003. She is currently working in the Computer Science Department at CICESE. Her main research interest include software process improvement in small companies, software engineering, knowledge management, and the social and personal factors related to software engineering.

José Alberto Fernández-Zepeda

He received a BS degree in Electronic Engineering in 1991, a M.S in Electronic Engineering in 1994 and a Ph.D. in Computer Engineering in 1999. He is currently an associate professor in the Computer Science Department at CICESE. Some of his research interests include software process improvement in small companies, the impact of social and personal factors in software engineering, design and analysis of algorithms.

Acceptance Test-Driven Development by annotation of existing documentation

David Connolly, Frank Keenan and Fergal Mc Caffery

Abstract

Testing is frequently reported as a crucial stage in the software development process. With traditional approaches acceptance testing is the last stage of the process before release. Acceptance Test Driven Development (ATDD) promotes the role of an expert customer in defining tests and uses tool support to automate and execute these tests. This paper outlines a tool, AnnoTestWeb/Run aimed at expert customers specifying acceptance tests through reuse of existing documentation. Also outlined is a planned evaluation that includes industrial collaboration aimed at considering the impact of this tool on reuse of existing documentation.

Keywords

ATDD, tool evaluation, industrial collaboration, annotations, acceptance tests

1 Introduction

A large part of software development expenditure is attributed to *testing*. Traditionally, customer involvement in plan-driven development acceptance testing, the process of testing functional requirements with “data supplied by the customer" [1], is in the final stages of the development process preformed long after the initial investigation has completed [2]. Many reports, however, highlight that costs can be reduced by detecting errors earlier in development [3]. Also supporting this, in many domains including medical device industry software is developed in a regulatory environment with a tendency for extensive documentation. In contrast, agile approaches require *constant* customer collaboration throughout development, with customer provision of acceptance tests being an important part of this role. Often, it is recommended that tests be identified before implementation commences. In eXtreme Programming (XP) [4], for example, acceptance tests are defined as a part of the User Stories practice and, as such, are written before coding of the story begins. The practice of ATDD permits software development to be “driven by the requirements" [5]. A key advantage of ATDD in its wider context is that it leverages existing agile infrastructure including continuous integration and test-first development. One challenge is that acceptance tests need to be written afresh despite the fact that business rules are often already documented in numerous formats. However, ATDD is currently not well supported with tools that enable reusing existing documents, without rewrites, to create executable tests. A challenge therefore, is to support a suitably informed expert in performing the agile *customer role*, including easily creating tests from existing material. However, successful identification of accurate acceptance tests in this manner is not necessarily straightforward.

2 Related Work

Many approaches to conducting acceptance testing exist. Some concentrate on acting as a “recording device” allowing user actions to be replayed against a system, checking for deviations. However, this approach is mainly limited to Graphical User Interface (GUI) testing of a specific version of a system, using a tool such as the Selenium IDE [7]. Tools for writing acceptance tests in a customer friendly format and appropriate for continuous integration exist. RSpec, for example, is a “Behaviour Driven Development framework for Ruby” [8]. It promotes a workflow that involves writing stories in a somewhat prescriptive natural language style and then manually translating these steps into Ruby. While the authors consider this approach interesting for new stories, it has limitations in dealing with pre-existing documents. Other open source tools aimed at supporting ATDD exist including EasyAccept that supports both tabular and sequential styles [9]. Generally, the Framework for Integrated Tests (FIT) is the most widely accepted tool for managing acceptance tests in agile development and therefore practicing ATDD [10]. In FIT’s simplest workflow a user, places inputs and some expected output into a tabular format, a ColumnFixture [11]. The developer then writes code (fixtures) that executes this data against the systems production code. Other built-in fixture included in FIT include ActionFixtures for testing a “sequence of commands” and RowFixtures for “comparing test data to objects in the system” [11]. FitNesse is a Wiki framework developed to support FIT [12]. It facilitates the editing of FIT tables in a browser allowing non-programming experts to add content. While FIT tables can be written in any tool that can export HTML, such as Microsoft Excel, these generic tools do not have any authoring features directly supporting the task domain. Existing tools that support either FIT or FitNesse includes FitClipse. As FitClipse [13] builds on FitNesse tests are entered using its wiki syntax. Mugridge introduces a process based around a library of fixtures named FitLibrary, which improves FITs “business-level expressiveness” to emphasise a “domain-driven design approach” [14]. It supports a type of fixture, DoFixtures, which is easy to read. Commercial software also supports such a workflow, with GreenPepper [15] supporting “executable specifications” while providing an expressive library of table types. However, none of these tools are focused on reusing existing documentation, so unlike the proposed approach these approaches require re-writes of content. In the requirements authoring process, Melnik and Maurer found that the use of FIT helped students to “learn how express requirements in a precise, unequivocal manner” [16]. In a number of experiments aimed at evaluating the impact of FIT tables on the implementation of change requests Ricca et al. [17], found improvement in the correctness of code produced and a greater impact on experienced students.

3 Tool Description and Workflow

AnnoTestWeb/Run is a browser-based tool built using the Google Web Toolkit and CouchDB. Creation of tests involves using annotations that describe different elements of an acceptance test. A metadata system provides extra detail to annotations for example label, data types. It features a simplified interface and workflow, partially because it is aimed at all members of agile teams, including non-developers. Basic conveniences are also provided, such as automatic saving and synchronization of the document and safe copy and pasting of annotated text between windows of the tool and through email applications through the use of an html-safe micro-format using class attributes.

The annotations are based on elements of good acceptance tests [6]. Three types for elements are defined, each given a colour coding and a greyscale safe symbol:

- Act (From Actors / Actions) are where inputs and outputs are used and expected or events
- Data (From Examples) concerns input data to the system under test
- Results (From Observables Results) concernes expected outputs from the system under test
- N.B. A fourth element “pre-conditions” of [6] is implemented using the order of named tests in a document rather than explicity defined annotations.

Data and Result Annotations have additional meta-data to support model:

- Label (this manditory metadata groups annotations so a named test might contain multiple runs)
- Value (usually the annotated text is the value, optionally it can be overrided)
- Data Types (to provide type conversion when passing the value of the annotation between the tool and the system under test)

The tests created by annotating existing text can be edited in a tabular style equivalent to FIT Tables, from within the tool. The basic user interface of the tool contains a toolbar allowing quick application of annotations, navigation between documents (using URLs) and allows management of named tests created from the document with a tabbed area. The UI with a simple test describing the querying of a timetabling application is displayed in Figure 1 below:

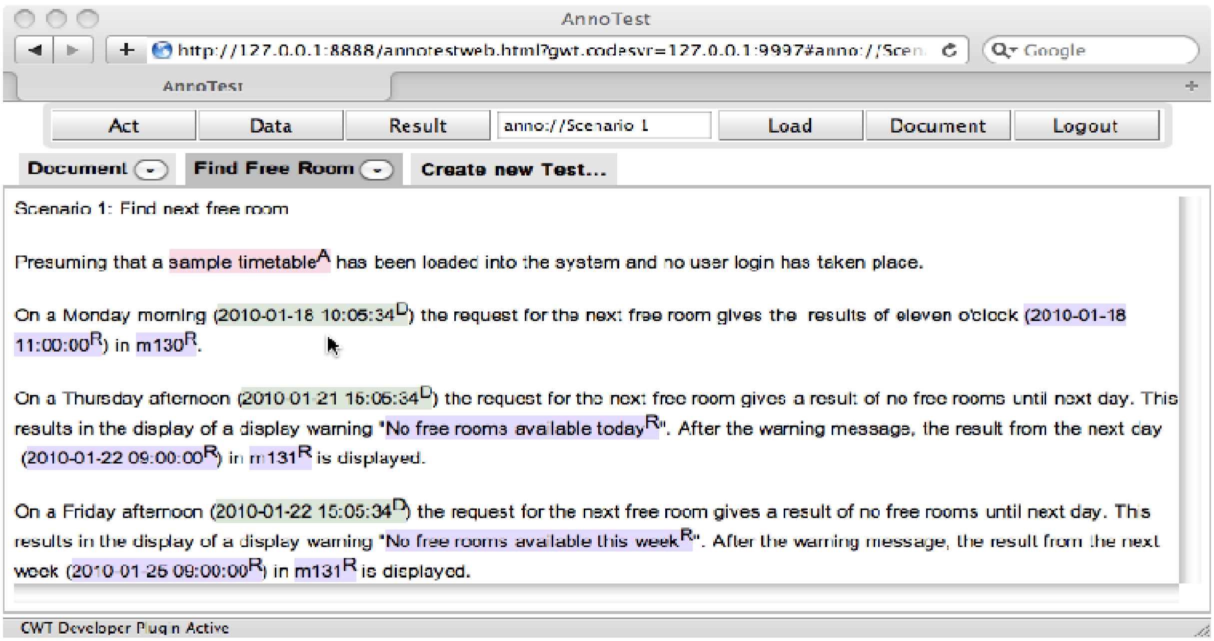


Figure 1: AnnoTestWeb/Run interface with a test ‘Find Free Room’ displayed

The tool is aimed at the person performing an agile customer role or a domain expert, with appropriate support from the software development team either with static review of tests or with direct pair participation. Their workflow for creating a test is simple. First, an appropriate name is chosen and appropriate text is identified then copied and pasted into the tool's rich text editor. Second, a named test is created. Third, highlighting text, selecting the type required and then providing appropriate metadata apply annotations. A fourth optional step is to customise the test with the Tabular editor for example where the test deviates from the natural order of the document. This tabular editor produces simple html tables appropriate for writing FIT fixtures, however a simplified interface for executing tests that integrates results with the annotated document is also included.

For example, sample text is loaded in a document named "Scenario 1" in Figure 1. A named test has been created, "Find Free Room", several annotations have been applied. Here the "Act" annotation is applied to "Sample Table". This is a reference to a class in the system under test, but the annotation type is also appropriate for references to standalone events or actors. The "Data" annotation is applied to inputs to the system; here a date "2010-01-18 10:05:34" has been annotated. The "Result" annotation is applied to the expected outputs of the system; here a free time "2010-01-18 11:00:00" and a free room "m130" have been annotated. As noted previously, metadata can be provided; here a label "freeRoom" has been applied to the later "Result" annotation, as visible in Figure 2 below:

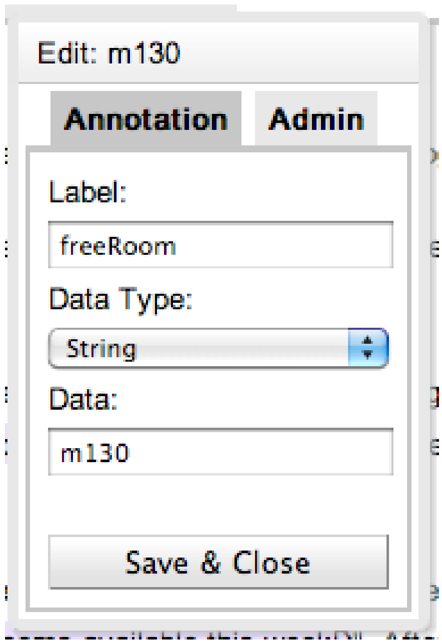


Figure 2: Metadata Editor displaying metadata of an annotation.

Delivering the data present in the annotated document and reporting of test results generated by program execution is designed to occur in a simplified fashion through a JavaScript Object Notation (JSON) API. This means that tests can execute in a wide variety of programming languages and environments. A library to speed up implementation of tests is also provided for Java and C#. This library has also been used to provide for input of manual steps that can occur test execution; this is in consideration of the combined software and hardware environment of our industrial partner in the next section.

The visualisation of the results of tests takes the form of these styles:

- Annotations (Simple pass or failure is visually displayed on Result annotations)
- Tabular (Actual value verses expected values are displayed in a table)
- UML (Actual value verses expected values are displayed in a clabject representation)

4 Industrial Evaulation Plan

The AnnoTestWeb/Run tool has been demonstrated to a domain expert and development team, to positive responses. The tool and the annotations in isolation have been tested experimentally with students, showing some benefits with annotations [18] and a favourable usability experience. Another prior empirical investigation showed annotations helped to reduce Errors in tests. Both these investigations considered similar metrics however the long running nature of the project will provide additional clarity.

The next step in the evaluation is to apply the tool in a long running industrial project. This evaluation will examine the use of the tool in a setting where existing documentation is available, including for example design documents and UML diagrams. At this planning stage the company will not be identified. The project background is of a new product containing both software and hardware components, developed using a plan-driven methodology. The project duration is of one year in total, with implementation at an advanced stage. The goal of the collaboration is to consider the tools impact on reusing existing documentation to generate acceptance tests, rather than in the context of acceptance driven development.

The following metrics will be gathered over the lifetime of the project:

- Errors:

  Gathered by recording mistakes in tests as they are corrected over the lifetime of project.

- Correct Elements / Missing Elements:

  Gathered by recording addition of annotation to tests over lifetime of project.

- Annotation Density:

  Gathered by comparing annotated text of tests to source documents.

The collection of these metrics will help to establish the impact of the tool and associated workflow on the testing process. Further as the acceptance tests created with the tool will not be the only aspect of testing on the project. This will allow detailed comparisons to be drawn between tests written by development team without the tool and tests written by a domain expert with assistance in the tool. This will include test quality metrics determined by the team and comparisons on time taken.

In addition structured interviews and questionnaires will be used to detail the experience of the team adopting the tool. A special focus in these supplementary efforts will be placed on human factors and usability of the tool.

5 Conclusions

This paper has presented a prototype tool and some of the background that led to its development. The work outlined in the previous section will be used to validate the tool in an industrial software development environment containing high quality existing documentation. Following this, improvements will be made to the tool based on the results of the case study.

The evaluation will be a prelude to a wider release of the prototype tool and further evaluation with teams using or adopting acceptance test driven development. Indeed the present industrial partner could be well placed to use the tool earlier in their processes. It is hoped that familiarity with the tool will put them in a position where adopting acceptance-test driven development is a logical conclusion of their document reuse.

Acknowledgments

This research is partially supported by Science Foundation Ireland through the Stokes Lectureship Programme, grant number 07/SK/I1299 and the Principal Investigator Programme (grant no. 08/IN.1/I2030)

Literature

1. Sommerville, I. Software Engineering, 8th edition, Addison-Wesley, pages 80-81, 2007.
2. Pressman, R. S. Software Engineering: A Practitioners Approach, European Adaption, 5th edition. McGraw-Hill, 2000.
3. Tassey, G. The economic impacts of inadequate infrastructure for software testing National Institute of Standards and Technology (NIST), May 2002.
4. Beck, K. and Andres C. Extreme Programming Explained: Embrace Change, 2nd edition, Addison Wesley, Boston, 2005.
5. Park, S.S. and Maurer, F. The benefits and challenges of executable acceptance testing, in APOS 08: Proceedings of the 2008 international workshop on Scrutinizing agile practices or shoot-out at the agile corral, ACM, New York, NY, USA, pages. 19-22, 2008
6. N. Jain, "Acceptance Test Driven Development". http://www.slideshare.net/nashjain/acceptance-test-driven-development-350264.
7. Kasatani, S. Selenium IDE. Website, URL last accessed 1st December 2008: http://seleniumhq.org/
8. RSpec Development Team. Website, URL last accessed 1st December 2008: http://rspec.info
9. Sauvé, J.P., Cirne, W., Osorinho and Coelho, R. EasyAccept Sourceforge Project. Website URL last accessed 3rd Dec 2008: http://easyaccept.sourceforge.net
10. Jain, N. Acceptance Test Driven Development. Presentation URL last accessed 30th Nov 2008. http://www.slideshare.net/nashjain/acceptance-testdriven-development-350264/
11. W. Cunningham, Framework for Integrated Test, September 2002. Website, URL last accessed 16th January 2009: http://fit.c2.com
12. FitNesse.org. Website, URL last accessed 7th February 2008: http://fitnesse.org
13. Deng, C., Wilson, P., and Maurer, F. "FitClipse: A FIT-based Eclipse plug-in for Executable Acceptance Test Driven Development". In XP 2007: Proceedings of the 8th International Conference on Agile Processes in Software Engineering and eXtreme Programming. 2007.
14. Mugridge, R. "Managing agile project requirements with storytest-driven development", IEEE Software, volume 25 (Jan.-Feb. 2008), pages 68-75, 2008.
15. Pyxis Technologies inc., GreenPepper Sofware, Website, URL last accessed 19th January 2009: http://www.greenpeppersoftware.com/confluence
16. Melnik, G. and Maurer, F. "The practice of specifying requirements using executable acceptance tests in computer science courses". In OOPSLA '05: Companion to the 20th annual ACM SIGPLAN conference on OOPSLA, pages 365-370, 2005.
17. F. Ricca, M. D. Penta, M. Torchiano, P. Tonella, M. Ceccato, and C. A. Visaggio, "Are fit tables really talking?: a series of experiments to understand whether fit tables are useful during evolution tasks", in ICSE '08: Proceedings of the 30th international conference on Software engineering, 2008, pages 361-370.
18. Connolly, D., Keenan, F., and McCaffery, F. Developing acceptance tests from existing documentation using annotations: An experiment. In Automation of Software Test, 2009. AST '09. ICSE Workshop on (18-19 2009), pp. 123 –129.

Author CVs

David Connolly

David received his B.Sc. (Hons.) in Computing in Internet Technologies in 2007 from the Dundalk Institute of Technology. He is now undertaking research as part of a Ph.D. in the area of agile acceptance testing, at DkIT's Software Technology Research Centre (SToRC) under the guidance of Dr. Frank Keenan and Dr. Fergal Mc Caffery.

Frank Keenan

Frank lectures in Software Engineering in the Computing department at Dundalk Institute of Technology and is a member of SToRC. His current research interests include requirements engineering, agile development and context analysis. A particular interest is the transfer of knowledge gained to form innovative delivery approaches to subjects taught.

Fergal Mc Caffery

Dr Fergal Mc Caffery is a Lecturer with Dundalk Institute of Technology and a member of SToRC. He has been awarded Science Foundation Ireland funding through the Stokes Lectureship and Principal Investigator Programmes to research the area of software process improvement for the medical device domain.

Approaches Facilitating WS-BPEL Testing

Dessislava Petrova-Antonova, Iva Krasteva, Sylvia Ilieva

Abstract

Business Process Execution Language for Web Services (WS-BPEL) is a language for behavior modeling of composite web services in the context of business processes. The WS-BPEL inherently brings a challenge for testing due to its specific syntax, dynamic binding during execution and the fact that it integrates web services implemented by various providers. The paper outlines the currently known approaches that meet this challenge. Our main research question is to identify the key features of the current approaches for WS-BPEL testing and based on it to give suggestions for their application in a new testing tool. In order to answer it, we have analyzed these approaches and have compared them based on their essential characteristics. The results are used in the design of specific tool named Data Dependency Analysis (DDA) tool. DDA tool is an integrated part of a developed testing framework named Testing as a Service Software Architecture (TASSA). It enables automatic test case generation for path coverage functional testing and provides fault injection mechanisms for negative functional testing.

Keywords

SOA, Testing, WS-BPEL

1 Introduction

Service Oriented Architecture (SOA) provides a new generation of software architectures intended to incorporate loosely-coupled applications. A common implementation solution is based on web services delivered over the network. Often, their functionality is combined into more complex services (composite web services) interacting in the context of business processes described with Business Process Execution Language for Web Services (WS-BPEL). The WS-BPEL presents a challenge for testing due to its specific syntax, dynamic binding during execution and the integration of web services that come from various providers.

Research in verification and validation applied to compositions of web services can be basically classified into two categories: formal verification approaches and testing techniques. The verification approaches are well described. Still, there are not as much research results available on the definition of testing methods for web services [14]. This paper addresses the second category, i.e. testing techniques. It outlines the current approaches for testing web service compositions described with WS-BPEL. Our main research question is to identify the key features of the current approaches for WS-BPEL testing and based on it to give suggestions for their application. In order to answer it, we have analyzed these approaches and have compared them based on their essential characteristics. The results are used in the design of specific tool named Data Dependency Analysis (DDA) tool. The tool searches for the path through all activities of a BPEL process starting from one initial activity. For each control activity, the tool calculates the condition that should be met in order for the process to continue execution further along the discovered path. DDA tool is part of the TASSA framework used to enable automatic test case generation for path coverage functional testing, as well as to provide fault injection mechanisms for negative functional testing [4].

The content of the paper from this point forward is organized as follows. Section 2 introduces the basic concepts used in the current approaches for WS-BPEL testing. Section 3 discusses the advantages and drawbacks of these approaches. A comparison of the discussed testing approaches is presented in a table. Section 4 presents the DDAT and its functions enabling automatic test case generation and fault injection for path coverage functional testing and negative functional testing. Finally, section 5 concludes the paper.

2 BPEL Testing Approaches

The WS-BPEL inherently brings a challenge for testing due to its specific syntax, dynamic binding during execution and the fact that it integrates web services implemented by various providers. This section presents a review of various techniques, methods and tools that meet this challenge. They can be classified based on intermediary model used for generation of executable paths of the BPEL process and test cases namely Control Flow Graph (CFG), Petri net (PN), Model checking techniques, including Stream X-machine (SXM), Web Service Automata (WSA), UML 2.0 activity diagram, PROMELA, etc.

2.1 Control flow graph transformation

In [15] authors propose graph-search based approach to BPEL test case generation, which contains four steps: (i) transformation of the BPEL process into an extension of CFG, called BPEL Flow Graph (BFG); (ii) traversing of the BFG and generation of test paths using a constraint solving method; (iii) filtering of the infeasible test paths, and generation of test data for feasible test paths; (iv) generation of abstract test cases by combining test data and paths. The path searching method on BFG is based on Depth First Search (DFS) algorithm. The resulted test cases are not directly executable and must be transformed to a program written in a specific programming language. The authors have implemented a module to generate executable test cases automatically from abstract ones in Java code and have integrated it into a testing environment.

The generation of the test suite for basis path testing of WS-BPEL and an accompanying tool that can be used by service testers are presented in [7]. Basis path testing starts with creating a CFG from which McCabe's cyclomatic complexity is computed. As a result, the number of basis paths is determined, which corresponds to the number of test cases needed for the flow. The testing tool analyzes variables relevant to each path to generate test data for the basis paths according to the value constraints. The tester specifies range and length of all relevant data values as well as necessary constant values. The generated values and manually-specified constants form the test cases for the BPEL process. For each loop execution the testing tool additionally generates an auxiliary Web service together with its WSDL.

In [8] the authors propose a gray-box testing approach that has three key enablers: test-path exploration, trace analysis, and regression test selection. Test-path exploration and trace analysis work in a complementary way. A group of test cases is designed directly from requirements and represents traditional specification-based test cases. Another group of test cases is designed from BPEL processes and represents test cases derived using the technique proposed by authors. During BPEL test-case generation the BPEL process is transformed into BFG supporting fault-handling logic. Trace analysis takes traces collected in a testing or production environment, identifies the transactions, and maps each transaction to the test paths of the related BPEL processes. A new regression test selection method is also proposed. A minimization algorithm is provided to select a minimal set of test cases that can cover all activities impacted by the process changes.

The approach in [13] uses an Extended CFG (XCFG) to represent a BPEL program, and generates all the sequential test paths from XCFG. Fault and event handling is presented in XCFG together with BPEL basic activities. In order to reduce the sequential test paths the authors use two test coverage criteria – basic path coverage, based on McCabe's cyclomatic complexity, and user directed path coverage. The sequential test paths are combined to form concurrent test paths. The idea of symbolic execution method is adopted in order to extract a set of constraints from the test paths and employ a constraint solver BoNuS to solve the constraints of these test paths and generate feasible test cases.

In [6] a formal model for an abstract-based workflow framework that can be used to capture a composed web service under test is introduced. It is focusing on verifying, based on structural-based testing strategies that a composed web service can function correctly according to its semantic, activities and data dependencies. The semantic-flow graph is used to represent how web services are choreographed to perform a request. The CFG specifies the execution order of activities (based on a specific semantic flow). The data-flow graph "connects" the referenced data to the activities in the control-flow graph, and the web services in the semantic-flow graph. The authors define formal notations that are related to test cases, test suites and applicable structural test adequacy criteria.

2.2 Petri net transformation

In [2] the authors use High-level Petri nets (HPNs) to model BPEL web service composition. The relationship between BPEL conceptions and HPNs is specified in four levels according to inter-service, intra-service, inter-activity, and intra-activity. The testing tool provides performs a reachability analysis for a HPN generated from the specification analysis. The analysis is based on path coverage and branch coverage. The result of reachability analysis is a binary tree that contains all necessary information for test case/suite generation, and provides data support for test case/suite generation. The test suite is generated in a text mode.

The approach proposed in [12] addresses the problem of checking and quantifying how much the actual behavior of a service, as recorded in message logs, conforms to the expected behavior as specified in a process model. The expected behavior is defined using the BPEL. BPEL process definitions are translated into Petri nets (PNs). This transformation is implemented in a tool called BPEL2PNML. For the purpose of conformance checking another tool called WofBPEL is used to simplify the PN produced by BPEL2PNML removing unnecessary silent transitions, and to convert the PN into a so-called WorkFlow net (WF-net). WF-net is the input format required by the ProM Conformance Checker tool that implements the current approach. The mapping of BPEL process to PN includes basic and structural activities, event and fault handling.

2.3 Transformation through model checking techniques

In [14] a model-driven approach toward generating executable test cases for the given business process is presented. The business process model is based on the BPEL specification and UML2.0 activity diagram. With DFS algorithm and typical test coverage criteria applied to the process under test, abstract test cases are generated and transformed into TTCN-3 test case that could be compiled and executed directly.

The approach presented in [16] extends UML2.0 activity diagram to describe the syntax and behaviors of BPEL. The test coverage criteria of UML2.0 activity diagram includes: Action coverage criterion, Transition coverage criterion, Basic path coverage criterion, and Sequence coverage criterion. The proposed test case generation method uses DFS in order to search activity diagram. The branches are decomposed to different sequences, while the circulations are executed only once. Entire test sequences can be obtained after searching of all actions and transitions.

SXM is a formal method, which is capable of modeling both the data and the control of a system. It is applied in [10] to automatically generate test cases for BPEL process. States and transitions are obtained by examining the activity and its sub-activities in BPEL process. BPEL activities are divided into 2 classes: basic and structured. Each activity is translated into an appropriate SXM fragment with starting and ending state. The memory structure is directly derived from BPEL variables. Three kinds of processing functions are specified: operation processing functions, assign processing functions, and predicate processing functions. The test cases are generated with the DSXM testing strategy [5].

In [17] WSA is proposed to model concurrency, fault propagation, and interruption features of BPEL process. A model checking based test case generation framework for BPEL is implemented. The SPIN and NuSMV model checkers are used as the test generation engine, and the conventional structural test coverage criteria are encoded into LTL and CTL temporal logic. State coverage and transition coverage are used for BPEL control flow testing, and all-du-path coverage is used for BPEL data flow testing. Two levels of test cases can be generated to test whether the implementation of web services conforms to the BPEL behavior and WSDL interface models. The generated test cases are executed on the JUnit test execution engine.

In [3] the model checking technique is applied to generate test case specifications for BPEL compositions. Firstly, the PUT is transformed into PROMELA. Then, in order to produce test cases, test requirements are identified using a transition coverage criterion. The transitions are identified in the BPEL specification, whether explicit or implicit, and are mapped onto the model. They are expressed like LTL properties. The counterexample obtained from a SPIN run is thus a sample execution of the BPEL process in which at least the transition included in the LTL is exercised. To obtain a set of test cases that provides transition coverage, the tool is repeatedly executed with each identified transition.

2.4 Other approaches

This subsection presents approaches that cannot be classified according to formalisms that are mentioned above. We describe them to achieve completeness of the reported work.

In [9] an approach to unit testing of WS-BPEL and a tool prototype extending JUnit are presented. The key idea of this approach is to transform process interaction via web service invocations to class collaboration via method calls, and then apply object-oriented test frameworks. The WSDL elements should be mapped to Java language equivalents and the Java data type classes are used to define test data.

In [1] a data dependency based test case generation approach is proposed. The data dependencies are defined using XPath expression by developers. Type definitions in WSDL documents are then leveraged to automatically generate independent data which, together with the specified data dependencies, are then used to generate coherent test data. Finally, test cases are composed using these data. The authors propose a method to capture execution information of the PUT using only standard BPEL functions to help developers determine the adequacy of generated test cases.

In [11] the authors propose a layer-based approach to creating frameworks for repeatable, white-box BPEL unit testing, which is applied to new testing framework. This framework uses a specialized BPEL-level testing language to describe interactions with a BPEL process to be carried out in a test case. The test suite document consists of two sections. The deployment section specifies the simulated partners along with their WSDL files, and contains information on how to deploy and undeploy the process under test. The second section contains an arbitrary number of test cases. Each test case contains one thread for each partner that contains a sequence of interactions, called activities. The framework supports automated test execution.

3 Discussion

This section identifies the key features of BPEL testing approaches and based on them presents a comparative analysis. It may be used to evaluate their advantages and drawbacks and to decide on possible improvements. Also, the main steps in the process of test case generation are summarised and illustrated.

3.1 Key features of BPEL testing approaches

As outlined in Section 2, the most of the authors propose to transform the BPEL process into intermediary model using CFG, HPN, etc. in order to find the executable paths of the process and generate test cases. That is why as the first key feature of BPEL testing approaches we identify the formalism that is used for representation of the intermediary model. The coverage of BPEL activities is an important issue during transformation. The obtained intermediary model could cover only basic BPEL activities, or basic as well as specific activities like those concerning exception and fault handling. To the best of our knowledge, it is significant to know which of the approaches are validated and have practical implementation in terms of automated tool supporting test case generation. Although only three of the approaches provide regression testing, it is useful to present such information especially when searching for testing approach that will be applied to frequently changing BPEL processes. In summary, the key features of BPEL testing approaches can be defined as follows:

- Formalism used for representation of the intermediary model of BPEL process;
- BPEL activities coverage;
- Validation via experimental results and case studies;
- Automation;
- Test case generation;
- Support of regression testing.

The advantages of BPEL transformation can be found in several aspects. The transformation of the BPEL process into CFG unravels the folded structures of BPEL (e.g. while loop, dead path elimination) into unfolded structures that are directly traversable in graph searching [15]. Implicit, disjoint control flows are turned into explicit, connected control flows during the transformation process. Also, the quantity of control structure types is reduced due to the uniformly representation of some control structures (switch and pick) according to their similar semantics.

In [2] the authors point as advantage of the PN transformation the opportunity of using the existing mature tools for BPEL process verification. Furthermore, the related researches on PNs can be employed in the testing. HPNs have the ability to model concurrency of the systems, analyze concurrent behavior, and express the dynamically changing software. In contrast to PNs, the model-checking techniques use domain specific language to describe the model, which burden the testing study process. But using model checking for test case generation also has its own advantages because it is automatic [17]. One of the strengths of using a SXM to specify a process pointed in [10] is that, under certain well defined conditions, it is possible to produce a test suite that is guaranteed to determine the correctness of the implementation under test.

In [16] UML2.0 activity diagram is used as a special form of state machine that can be applied to model computing flow and work flow. In the activity diagram, the state machine and the metamodel of activity are separated. Since UML2.0 activity diagram cannot build the business process testing model sufficiently, the approach presented in [16] uses stereotype and tagged value to promote the imprecision with formal specification, which are UML2.0 extension mechanism.

Of course, neither of the transformations can fit completely to the specific of the BPEL language. CFG is a static representation of a sequential program that represents all alternatives of a control flow. However, for concurrent programs there are no such simple graphs for analysis [13]. Data handling logic cannot easily be represented with PNs. The UML activity diagram tends to be limited for a specific programming language like BPEL in that some language features cannot be expressed in a straightforward way. For example, BPEL flow construction allows multiple-choice style workflow pattern, which cannot be represented by a simple UML activity structure [13]. The state space explosion problem is a well known inherent problem to model checking techniques [17].

The presented approaches have advantages and drawbacks due to their specific implementation. For instance, the generated TTCN-3 test case in [14] still needs some effort to develop the adapter and codec to run. The testing tool proposed in [7] may be considered an add-on to Oracle’s BPEL development environment but the approach can be followed to develop a basis path test suite for other platforms. It does not support all XML schema data types in the generation of test data (only integer, float, boolean, and string are supported). Also, only sequence, condition, and repetition patterns of control are allowed. The tool does not consider infeasible paths that cannot be accessed. In order to improve the preciseness of the generated test paths in [8], IBM BPEL extensions, like Java snippets, need to be handled. The experimental results show that the test-generation time is linear to the number of test paths searched. Thus a more efficient generation algorithm is needed to avoid the performance problem for complex processes.

The approach in [13] is more applicable to programs without complex variable sharing or process interaction patterns. The messages’ maximum enablement is limited to one time during the transformation of BPEL process into XCFG. Also, the exception handling logic does not affect the other running “threads”, which run to completion undisturbedly. An advantage of the approach is that it is modularized so that it can be used together with other testing technologies. It avoids the state space explosion problem and is applicable for programs in which concurrent computation units have only very few or no shared variables or other types of synchronization.

BPELUnit framework presented in [11] does not provide much support in test case creation and the monitoring of the PUT. Developers have to manually prepare large amount of coherent XML data and XPath expression to compose a test case. This is a painstaking task considering the complex structure of involved XML data. An open issue in [17] is to prove the correctness of the model transformation. Automatic test suites extraction and generation, test case organization, exception handling, and security flows are not considered in [6]. The proposed framework is presented on abstract level. BPEL-Unit presented in [9] provides the following advantages: allow developers simulate partner processes easily, simplify test case writing, speed test case execution, and enable automatic regression testing.

Table 1 summarizes the comparison of the presented approaches according to their key features. The choice of formalism for representation of the intermediary model of the BPEL process is a basic step in the design of each of the approaches. It is in close relation with the test path generation and BPEL activities coverage. The formalism (F) that is used for representation of the intermediary model is shown in the second column. The next two columns show which approaches consider event handling (EH) and fault handling (FH) BPEL activities. The fifth column of the Table 1, called Automated (A), shows which of the approaches have practical implementation. Some of them like those in [12], [3], [7] and [2] are implemented in tools. Others like that in [6] suggest testing methods without concrete realization. The next table column, named Experimental results (ER) indicates which of the approaches are proved via case studies, experimental results, etc. Only two of the approaches do not provide test case generation (TCG). This can be seen in the next-to-last column. And finally, the last column shows which of the approaches supports regression testing (RT).

Table 1. Comparison of BPEL testing approaches

Approach	F	EH	FH	A	ER	TCG	RT
Yuan [15]	CFG	-	-	-	-	+	-
Lertphumpanya [7]	CFG	-	-	+	+	+	-
Dong [2]	HPN	-	-	+	+	+	-
Yuan [14]	UML2.0 activity diagram	-	-	+	+	+	-
Zhang [16]	UML2.0 activity diagram	+	-	-	+	+	-
Li [8]	CFG	+	+	+	+	+	+
Yan [13]	CFG	+	+	-	+	+	+
Mayer [11]	-	-	-	+	-	+	-
Choy [1]	-	-	-	+	+	+	-
Ma [10]	SXM	+	-	-	-	+	-
Zheng [17]	WSA	+	+	+	-	+	-
Fanjul [3]	PROMELA, LTL	-	-	+	+	+	-
Aalst [12]	PN, WF-net	+	+	+	+	-	-
Karam [6]	CFG, DFG, SFG	-	-	-	-	-	-
Li [9]	Java language	-	-	+	+	+	+
DDA tool	CFG	+	+	+	-	+	-

3.2 Key steps in the process of test case generation

Fig. 1 illustrates the main steps in the process of test case generation. The transformation into intermediary model is a starting step of the presented approaches. It can be implemented using various formalisms (CFG, PN, SXM, etc.). The verification and path filtering steps are presented with dashed line because they are not realized in all of the approaches. As can be seen from the figure below, existing mature tools like SPIN and NuSMV could be used for verification of the intermediary model. Note that these tools are applicable only if the intermediary model is presented with PN. On the other hand, well-known algorithms like DFS could be applied in the step of path searching.

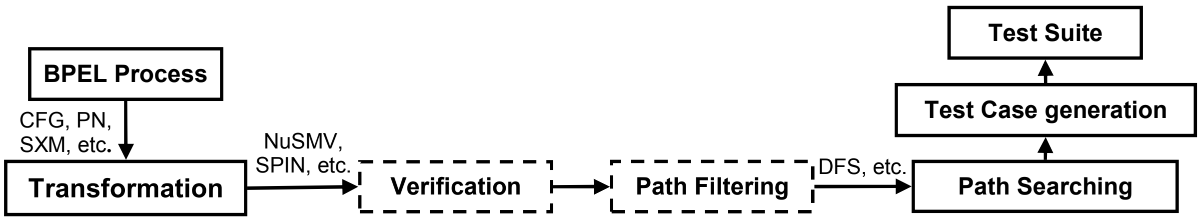


Fig. 1. Key steps of BPEL testing approaches

4 Data Dependency Analysis Tool

As mentioned in the introduction, the results from the research work are used as starting point in the design of DDA tool, which key features are presented in the last row of Table 1.

As can be seen from the Table 1 the CFG transformation of the BPEL process is used in the most of the approaches. This is due to the fact that such transformation allows the conventional graph algorithms for path searching to be applied, while the model-checking techniques use domain specific language to describe the model, which burden the testing study process. In contrast to PNs, CFG allows easy coverage of the exception and fault handling BPEL activities. That is why we have chosen the CFG as intermediate representation of the BPEL process in DDA tool.

In general the Data Dependency Analysis (DDA) tool solves the following problem: given a set of activities {A_1, A_2,... A_n} in the BPEL, find a path from the initial activity I that goes through all activities in the set. Then, find all control activities on this path and for each control activity; calculate what the condition is that must be met in order for the process to continue execution on this path and not on some other branch.

The first task that the DDA tool addresses is to find a path in the BPEL from the initial activity through the activities under question. In order to solve the first task, the DDA tool transforms the BPEL process into an internal structure represented with CFG. The path is then discovered using standard combinatorial algorithms that work on the graph. For the purpose of the second task, the conditions in the process should be analyzed. They are parsed with a suitable parser and are represented as a tree structure. Each leaf in the tree is a rule field of a variable. The leaves are linked with operations that act on the leaves' variables. The conditions are considered dependent on events not on variables. They are internally recognized and handled in a specific way. For example, in the Pick activity there is a split in the flow depending on whether a message arrives or a timeout is reached. Such a condition is recognized and described so as to be understandable for the rest of the TASSA framework's tools. One of the advantages of DDA tool is the coverage of basic as well as structured activities along the transformation to CFG. As is shown on Table 1 exception and fault handling are considered in less than half of the approaches.

The DDA tool is implemented as an open source web service that provides a single method for BPEL analysis. It receives the BPEL and an array of XPath expressions that identify the activities of the BPEL process. The method returns an array of activities along with corresponding conditions for each of them. In case of collective conditions, the BPEL process will go through all relevant activities during execution. The implementation as web service has advantage due to possibility of using DDA tool independently from other tools of TASSA framework. In addition our solution is open source, which makes DDA tool competitive to the presented approaches that are implemented in tools.

A certain limitation for the time being is that the DDA tool only recognizes conditions expressed in XPath language as prescribed by the WS-BPEL standard. However, that limitation is not principal and in the future it is planned it to support also conditions expressed e.g. in Java. Such advanced features are used in IBM's version of BPEL. Due to the fact that TASSA framework is under development, DDA tool is not validated through experimental results. We intend to verify its effectiveness by real case study from industry. Currently the DDA tool does not support regression testing. That is why we recommend it to be used for infrequently changing BPEL processes. In comparison, this feature is implemented only in [8], [13] and [9] and it is object of future work.

5 Conclusion

This paper identifies and systematizes the key features of contemporary approaches for testing web service compositions that are described with WS-BPEL. For this purpose, we have analyzed these approaches and have compared them based on their essential characteristics. First, we have provided a summary of the approaches. After that their essential features characterizing testing process are described and used as a base for comparison analysis. The analysis concerns the formalism that is applied to the BPEL process during test case generation and BPEL activities included in the transformation. It shows that the preferred formalisms rely on CFG and PN as well as the most of the approaches cover only basic BPEL activities. Another point of the comparison analysis determines which approaches are implemented in tools and proved via test cases and experiments. Finally, the analysis examines the support of regression testing and test case generation. Although almost all of the approaches provide test case generation, only three of them support regression testing. Finally, the results from the presented work are used in the design of tool called DDA tool that enables automatic test case generation for path coverage functional testing and provides fault injection mechanisms for negative functional testing.

Acknowledgements

This work has been supported by the National Scientific Fund, Bulgarian Ministry of Education and Science under grant agreement number DO02-182.

Literature

1. Choy Kh. Y., T. Ishio, M. Matsushita, K. Inoue, H. Shinomi, K. Yuura. Data Dependency based Test Case Generation for BPEL Unit Testing. IPSJ SIG Notes, pp. 163-170, (2008).
2. Dong W. L, H. Yu, Y. B. Zhang. Testing BPEL-based web service composition using high-level Petri Nets. Proceedings – IEEE International Enterprise Distributed Object Computing Workshop, (2006).
3. García-Fanjul J., J. Tuya, Cl. de la Riva. Generating Test Cases Specifications for BPEL Compositions of Web Services Using SPIN. International Workshop on Web Services Modelling and Testing, pp. 83-94, (2006).
4. TASSA, Testing as a Service Software Architecture framework, http://tassa.rila.bg.
5. Ipate Fl., M. Holcombe. Testing data processing-oriented systems from stream Xmachine models. Theoretical Computer Science 403, pp. 176-191, (2008).
6. Karam M., H. Safa, H. Artail. An abstract workflow-based framework for testing composed web services. In International Conference on Computer Systems and Applications (AICCSA), pp. 901–908, (2007).
7. Lertphumpanya T., T. Senivongse. Basis path test suite and testing process for WS-BPEL. WSEAS Transactions on Computers, 7(5): 483-96, (2008).
8. Li Z. J., H. F. Tan, H. H. Liu, J. Zhu, N. M. Mitsumori. Business-process-driven gray-box SOA testing. IBM Systems Journal 47, pp. 457-472, (2008).
9. Li Z.J., W. Sun. BPEL-Unit: JUnit for BPEL Processes. Service-Oriented Computing – ICSOC 2006, ISBN 978-3-540-68147-2, pp. 415-426, (2006).
10. Ma Ch., J. Wu, Tao Zh., Y. Zhang, X. Cai. Automatic Test Case Generation for BPEL Using Stream X-Machine. International Journal of u- and e- Service, Science and Technology, ISSN: 2005-4246, pp. 27-36 (2008).
11. Mayer Ph., D. Lübke. Towards a BPEL unit testing framework. In: Proceedings of the workshop on Testing, analysis, and verification of web services and applications, (2006).
12. Van Der Aalst W. M. P., M. Dumas, Ch. Ouyang, A. Rozinat, Eric Verbeek. Conformance Checking of Service Behavior. ACM Transactions on Internet Technology, Vol. 8, No. 3, Article 13, (2008).
13. Yan J., Z. Li, Y. Yuan, W. Sun, J. Zhang. Bpel4ws unit testing: Test case generation using a concurrent path analysis approach. In Proc. of ISSRE. IEEE Computer Society, pp. 75–84, (2006).
14. Yuan Q., J. Wu, Ch. Liu, L. Zhang. A model driven approach toward business process test case generation. 10th International Symposium on Web Site Evolution (WSE), pp. 41-4, (2008).
15. Yuan Y., Z. Li, W. Sun. A graph-search based approach to BPEL4WS test generation. International Conference on Software Engineering Advances, ICSEA'06, (2006).
16. Zhang. G., M. Rong, J. Zhang. A business process of web services testing method based on UML2.0 activity diagram. Proceedings - Workshop on Intelligent Information Technology Application, IITA 2007, pp. 59-60, (2007).
17. Zheng Y., J. Zhou, P. Krause. An Automatic Test Case Generation Framework for Web Services. Journal of Software, Vol.2, No.3, pp. 64-77, (2007).

Author CVs

Dessislava Petrova-Antonova

M.Sc in Computer Systems and Technologies from Technical University of Sofia, Bulgaria, Ph.D. in Computer Science, Technical University of Sofia, Bulgaria. Currently she is assistant professor at the Faculty of Mathematics and Informatics, Sofia University. Main fields of scientific interest – Service Oriented Architectures, Software testing, Web services and Web applications development. She has more than twenty publications in international conferences and journals.

Iva Krasteva

M.Sc. in Informatics from Sofia University "St. Kliment Ohridski", Department of Mathematics and Informatics. In 2004 joined RILA Solutions EAD as Quality Assurance Engineer. From 2007 she is part time Ph.D. student in Computer Science. Main fields of scientific interest – Software Processes, Quality Assurance, Component-Based Software Engineering, Web Services and Service Oriented Architecture. She has more than ten publications in international conferences and journals.

Sylvia Ilieva

M.Sc. in Electronics, Technical University of Sofia, Bulgaria, Ph.D. in Computer Science, Technical University of Sofia, Bulgaria. Currently she is Head of the Software Engineering Department at the Faculty of Mathematics and Informatics, Sofia University, as well as Associate professor at Bulgarian Academy of Sciences. She has a good record of successful participation in a number of European Framework RTD Projects as well as national RTD projects. Main fields of scientific interest – Software process development, Software quality, Service Oriented Architectures. She has more than fifty publications in international conferences and journals.

Automotive and Medical: Can we learn from each other?

Dr. Anne Kramer

Abstract

In this paper we present an analysis of the new standard ISO 26262 for the automotive industry. We compare this standard to existing regulations of another safety-critical domain, i.e. the medical device industry. As can be expected, a large number of activities coincide, since they simply correspond to common best practices that can also be found e.g. in ISO/IEC 15504. However, some interesting differences can be observed both in philosophy and in wording.

This paper is intended to facilitate communication between the two domains that are driven by the same concern: safety of the end user. We will point out the most striking similarities and differences and translate from one domain to the other.

Keywords

ISO 26262, IEC 62304, ISO 14971, ISO 13485

automotive, medical

1 Introduction

The new standard ISO 26262 defines all activities of the so called "safety lifecycle" for electrical or electronic system in road vehicles. ISO 26262 implements the domain-independent standard for functional safety of electrical/electronic/programmable electronic safety-related systems, IEC 61508, for the automotive industry. It is currently been voted and its release can be expected soon.

In this paper we will have a closer look on the requirements resulting from ISO 26262, comparing then to existing regulations of another safety-critical domain, i.e. the medical device industry. An overview on the ISO standards for both domains is given in table 1.

Topic	Automotive	Medical
Quality Management System (general)	ISO 9001	ISO 9001
Quality Management System (domain specific)	ISO/TS 16949	ISO 13485
Life Cycle Management	ISO 26262 ("Safety Lifecycle")	IEC 62304 ("Software Lifecycle")
Safety / Risk Management	ISO 26262	ISO 14971

Table 1: Overview on ISO standards

This paper focuses on ISO 26262. Requirements for general quality management like management responsibilities and resource management are not considered even if they are addressed in ISO 13485 [1].

2 Comparison of ISO 26262 and IEC 62304

2.1 Scope and motivation

ISO 26262 applies to safety-related systems within road vehicles that include at least one electrical or electronic (E/E) system. Strictly speaking, it is only concerned with E/E systems, but it also provides a framework for safety-related systems based on other technologies. The standard provides an automotive safety lifecycle including requirements on verification and confirmation activities to ensure that "a sufficient and acceptable level of safety" is achieved [2].

IEC 62304 provides lifecycle requirements for medical device software. Again, it is intended to provide a development framework to ensure that the resulting software does not represent unjustifiable risks for the users.

Thus, both standards have been developed out of the concern that flaws in the development process of a product might cause damage to people. The authors of both standards also agree that the trend towards increasing software content represents an increasing risk.

2.2 Level of detail

The standards for medical devices listed in table 1 contain rather general requirements. Also, they all reference each other in order to avoid redundancy. No technical details are given. Specific examples for techniques (e.g. FMEA) are given in non-regulatory appendices of ISO 14971 [3].

ISO 26262 is essentially self-contained. The level of detail is higher. Detailed instructions are given, e.g. for conducting the risk analysis and on verification methods. Annex D (informative) of part 6 [4] even gives recommendations for memory checks like parity bits or Cyclic Redundancy Checksums (CRC) to ensure freedom from interference by software partitioning. Also, a clear recommendation is given in favor of requirements management tools.

2.3 Vocabulary

One of the most striking differences between the two domains is the vocabulary. This sometimes makes communication difficult. The following examples illustrate well how misleading the different vocabulary can be:

1. The "safety plan" of ISO 26262 corresponds to the "risk management plan" of ISO 14971.
2. ISO 26262 is about the "safety lifecycle" whereas IEC 62304 is about a "software lifecycle".
3. ISO 26262 requires the determination of "controllability" of a hazard. For medical devices, the "probability of detection" is taken into account.

Other terms are used identically. The highest overlap can be observed for terms like "anomaly", "baseline" or "review" that are also used in other standards (e.g. IEEE).

Note, that the first two examples in the list above also show a difference in approach. ISO 26262 is strictly focused on safety-related systems and not limited to software. The standards for medical devices apply to the entire product, but require more stringent tests and documentation for safety-relevant systems. Standards for hardware development of medical devices exist, but have not been taken into account in this paper.

The highest confusion is created by those terms that have different meanings. For example, "SOP" in automotive stands for "Start Of Production" (instead of "Standard Operating Procedure" in the medical domain). Also, "system" is used differently. ISO 26262 defines a system rather technically as a "set of elements, at least sensor, controller, and actuator, in relation with each other in accordance with a design" [2]. IEC 62304 (in accordance with ISO/IEC 12207) defines a system as a composite of procedures, hardware, software, facilities and personnel providing functionality to perform a given task or achieve a specific purpose [5]. The second definition is broader, giving the term "system test" a different scope.

ISO 26262 mentions the "Hardware Software Interface Specification (HSI)" as work product. This term has no direct equivalence in the domain of medical software, where these interfaces are considered as part of the system architecture.

3 Risk-based approach

Both ISO 26262 and IEC 62304 follow a risk-based approach. There is a common philosophy that the degree of documentation and testing should depend on the risk that an injury may be caused by the device or component. ISO 26262 is strictly limited to these "safety-related" systems, whereas IEC 62304 applies to the entire medical device software.

Both standards define as first activity a hazard analysis and – as a result – the classification of the system regarding to their safety-relevance. This is one of the most striking similarities between both standards (and both domains). However, the classification procedure differs in details.

IEC 62304 distinguishes three classes ranging from class A (no harm or injury possible) to class C (severe injuries or death possible). ISO 26262 classifies the systems according to their "Automotive Safety Integrity Levels" (ASILs), where ASIL A corresponds to the least stringent and ASIL D to the most stringent safety integrity level.

The classification of medical device software is obtained from the hazards analysis. IEC 62304 explicitly refers to the risk management standards ISO 14971 for details on how to determine the risks. ISO 14971 mentions various methods like the Failure Mode and Effect Analysis (FMEA) or the Fault Tree Analysis (FTA) as appropriate methods to determine hazards. Usually, hazards are quantified according to their severity, their probability of occurrence and their probability of detection.

Classification of medical device software	
Previous hazard analysis according to ISO 14971 taking into account severity, probability of occurrence and (optional) probability of detection.	
Class A	no harm or injury to the health of the user, the operator or any other person can be caused (even if the software does not work as specified)
Class B	No major harm can be caused by the medical device. Major harm is defined as injuries or diseases that might directly or indirectly lead to death, result in an irreversible handicap or require medical or chirurgical interventions to repair the damage.
Class C	Major harm or death is possible.
If hardware mechanisms are implemented to reduce the risk caused by software, the software classification can be reduced by one class.	

Table 2: Classification of medical devices

ISO 26262 also mentions FMEA and FTA as methods for a systematical determination of hazards. Other methods mentioned are ETA (Event Tree Analysis), Markov models and reliability block diagrams. For the ASIL determination the standard exactly defines the way the hazards should be classified, using three criteria for the quantification of a hazard. As can be expected, one criterion is the severity of possible harm. The two other criteria differ slightly from those commonly used for medical devices. Instead of rating the probability of occurrence, ISO 26262 deals with the probability of exposure regarding operational situations. There is no real difference between these two criteria. In both cases, the number of vehicles / devices equipped with the system / software and the product lifetime must be taken into account to quantify estimations like "very low probability".

The third criterion for ASIL determination is an estimation of the "controllability", e.g. the probability of bringing a vehicle to complete stop if the breaks fail. The idea behind this criterion is the same as for medical devices. If the "probability of detection" is high, the chances are good that the patient can be saved.

Tables 2 and 3 show the criteria for classification in the two domains. A similarity of both domains is the notion of inheritance of classification. If software is split into modules, these modules inherit the classification of the ensemble unless a rationale can be provided that the module can be classified differently. In that case, the decomposition must be transparent. For automotive systems, sub-systems inherit the ASIL of the entire component. ASIL decomposition is allowed (and even encouraged), but it is subject to specific rules that are defined in part 9 of ISO 26262 [6].

ASIL determination of automotive systems

Previous hazard analysis according to ISO 26262 taking into account

- severity (S1 = "light and moderate injuries" to S3 = "survival uncertain and death"),
- probability of exposure (E1 = "very low" to E4 = "high") and
- controllability (C1 = "simply controllable" to C3 = "difficult to control or uncontrollable).

ASIL A to D, depending on the results of the hazard analysis:

		C1	C2	C3			C1	C2	C3			C1	C2	C3
S1: light and moderate injuries (S1)	E1	-	-	-	**S2:** severe and life-threatening injuries (survival probable)	E1	-	-	-	**S3** survival uncertain, fatal inju-ries	E1	-	-	A
	E2	-	-	-		E2	-	-	A		E2	-	A	B
	E3	-	-	A		E3	-	A	B		E3	A	B	C
	E4	-	A	B		E4	A	B	C		E4	B	C	D

Hazards that are estimated as S0 ("no injury"), E0 ("incredible") and C0 (""controllable in general") do not require ASIL determination.

Table 3: ASIL determination

4 Lifecycle model

IEC 62304 does not impose any specific lifecycle model, even if the V-model is mentioned as example in annex C. It is explicitly stated that the task may be performed iteratively or recursively.

This is an important difference to ISO 26262, where it is explicitly stated in part 2: "The system development process is based on the concept of a V-model (…)" [7]. In fact, the V-model is required at each development level (system, hardware, software) as can be seen in figure 1.

This apperently excludes the application of agile processes like SCRUM which is surprising, especially when you think of the fact that SCRUM is based on the concept of lean production first introduced by Toyota. Also eXtreme Programming (XP) was first introduced at Chrysler by Kent Beck [8]. Agile processes are becoming more popular than ever. The idea of agile processes not being compliant with formal regulatory requirements has definitely been proven wrong for the development of medical devices. On the MedConf 2009, an entire track was dedicated to this subject [9].

It can be predicted that the trend towards more and more software in vehicles will create a conflict between agile methods (with proven efficiency) and the V-model imposed by ISO 26262. ISO 26262 might become more flexible allowing lean approaches, following the motto of the American Food and Drug Administration (FDA) of a "least burdensome approach".

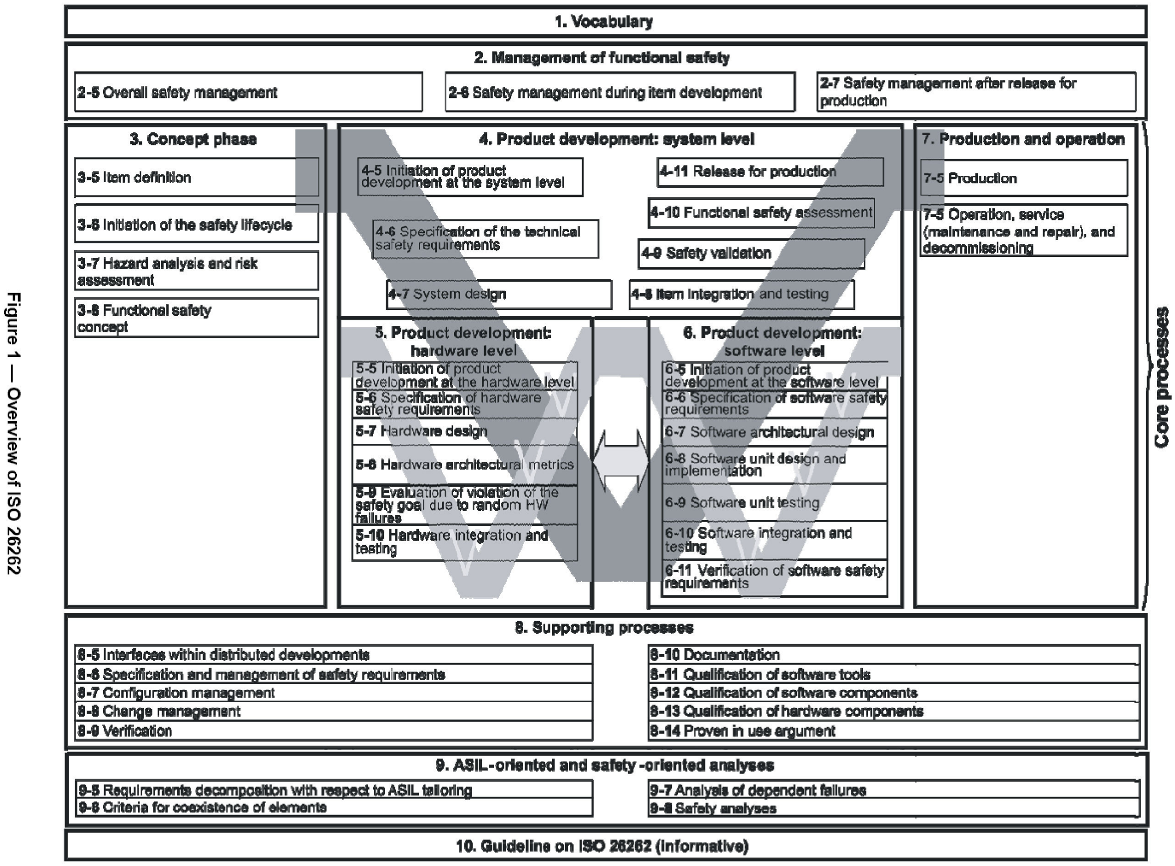


Figure 1: Overview on ISO 26262

5 Verification and Validation

Essentially, the requirements for verification and validation activities are identical. Tests are specified at all levels of integration. All activities shall be planned and documented. Documents reviews are systematically required. Tester qualification must be proven.

Again, ISO 26262 is very detailed on the methods that should be applied to derive test cases and which type of tests (e.g. requirement-based tests or fault injection tests) should be performed. Several tables list requirements and recommendations depending on the system's ASIL. Special attention is paid to configuration and calibration data.

Though model-based development is taken into account, model-based testing is only considered at unit test level. No reference is given to model-based (or model-centric) testing at higher integration level. Model-centric testing represents a systematic and efficient approach to verification and validation [10][11] and should be taken into account as state-of-the-art in any new standard (medical and/or automotive). It differs from the idea of reusing development models for unit testing, since it is based on test design models that are explicitely written to describe the behavior of the system under test. In model-centric testing, the model also contains test management information (e.g. priorities) and represents the central repository for all test-relevant information.

It is interesting that the term "validation" is used, but not defined in ISO 26262. In part 4, validation is required at vehicle level of integration. This includes user test under real-life conditions performed by normal users as testers that are not bound to prior specified test scenarios. Validation shall be based on safety goals, functional safety requirements and the intended use (operational use cases).

Usability aspects that are of major importance for the development of medical devices are not explicitly addressed in ISO 26262. For medical devices, usability is a major issue. It is definitely life-threatening for a patient, if the doctor cannot see a warning or cannot reach an important button in time.

It can be expected that similar usability issues will become more important for automotive safety in the future. With the continuously increasing amount of safety-related functionality in the area of driver assistance and vehicle control, the driver relies more and more on this support. Hidden control lamps (corresponding to hidden warning messages) reduce the controllability and, thus, directly influence the risk analysis.

6 Supporting processes

Supporting processes in IEC 62304 are risk management, configuration management, "software problem resolution" (corresponding to "change management") and software maintenance. Unlike in ISO 26262, verification is not considered as supporting process. Instead, it is part of the different phases of software development (5.1 to 5.8). Other supporting processes are defined in ISO 13485. An overview is given in table 4.

Automotive		**Medical**	
Interfaces with distributed developments	ISO 26262 – 8-5	Purchasing	ISO 13485 – 7.4
Safety management	ISO 26262 – 8-6 ISO 26262 – 2 ISO 26262 – 9	Risk management	IEC 62304 – 4.2 IEC 62304 – 7 ISO 14971
Configuration management	ISO 26262 – 8-7	Configuration management	IEC 62304 – 8
Change management	ISO 26262 – 8-8	Software problem resolution	IEC 62304 – 9
Verification	ISO 26262 – 8-9	*part of software development process*	IEC 62304 – 5
Documentation	ISO 26262 – 8-10	Documentation	ISO 13485 – 4.2
Qualification of software tools	ISO 26262 – 8-11	*part of software development process*	IEC 62304 – 5.1.10
Qualification of software components	ISO 26262 – 8-12	*included in scope*	IEC 62304
Qualification of hardware components	ISO 26262 – 8-13	*beyond scope*	-
Proven in use argument	ISO 26262 – 8-14	*no direct equivalence*	-
Operation, Service	ISO 26262 – 7-5	Software maintenance	IEC 62304 – 6

Table 4: Supporting processes and related chapters

Tool qualification is required for software tools that support the development and production of both medical device software and safety-related systems and those that are delivered together with the product (e.g. virus scan). While IEC 62304 resumes the related requirements in one section, an entire chapter is dedicated to tool qualification in ISO 26262. A detailed analysis shall be conducted to determine the tool's level of confidence. This includes e.g. an analysis of known defects.

According to ISO 26262 a software tool shall be classified according to its impact (Tool Impact: TI) and the probability of error detection (Tool error Detection: TD). From these two factors the Tool Confidence Level (TCL) is determined. TCL1 applies to software tools that have

- no impact on safety related items or
- a high degree of confidence exists, that a malfunction or an erroneous output from the tool will be prevented or detected).

Tools with TCL1 do not require qualification measures. For TCL2 to TCL4, the required qualification activities depend on the maximum ASIL of a safety-related requirement that might be violated by functional misbehavior of the tool. Depending on the ASIL one or several of the following qualification methods are required / recommended:

- Providing evidence for increased confidence from use in previous projects;
- Evaluation of the tool's development process proving compliance with an international standard (e.g. via an assessment according to Automotive SPICE, CMMI or ISO 15504);
- Tool validation;
- Development in compliance with a safety standard, i.e. ISO 26262 (automotive), IEC 61508 (domain-independent) or RTCA DO 178 B (avionic).

The classification instructions given in ISO 26262 could provide guidance for tool validation in the medical device industry where it is often unclear, how much validation is necessary to fulfill the regulatory requirements.

There is no direct equivalence for the "proven in use" argument as qualification methods in the medical standards. The closest match can be found in the Good Automated Manufacturing Practice (GAMP) guide in the context of retrospective validation. Reused or purchased software components that should be qualified according to ISO 26262 are called software of unknown pedigree (SOUP) in IEC 62304. Verification and validation of these components is simply included within the scope of the standard. Qualification of hardware components is without scope of the medical standards considered in this paper.

7 Summary

In this paper we presented a comparison between the new standard for functional safety of road vehicles, ISO 26262 and corresponding standards from the medical device industry (ISO 13485, IEC 62304 and ISO 14971). We analyzed similarities and differences and pointed out, where one domain could profit from having a look at the other domain. Especially the instructions for tool qualification and proven in use arguments from ISO 26262 could provide additional guidance for the medical device industry.

Both domains follow a common philosophy, i.e. a risk-based approach. The degree of documentation and testing required depends on the risk that someone might get severely hurt by using the end product. However, the vocabulary is different. Also, the level of detail is very different. While standards for the development of medical device software are kept rather general, leaving it up to the manufacturer to find and justify appropriate processes, ISO 26262 prescribes the V-model and gives detailed instructions on the way, the requirements shall be implemented.

In our opinion, the ideal way should be in the middle. ISO/IEC 15504 gives an example on how this can be realized, giving general requirements in the normative parts and guidance for implementation in a reference model. This also corresponds to the approach of the American Food and Drug Administration (FDA) that issues regulations for the development and commercialization of medical devices for the American market. The various parts of the Code of Federal Regulations Title 21 (21 CFR), which are a law, do not define any implementation. Recommendations instead are given in additional guidance documents.

Literature

[1] ISO 13485," Medical devices -- Quality management systems -- Requirements for regulatory purposes"

[2] ISO 26262-1, "Road vehicles – Functional Safety – Part 1: Vocabulary"

[3] ISO 14971, "Medical devices – Application of risk management to medical devices"

[4] ISO 26262-6, "Road vehicles – Functional Safety – Part 6: Product development: software level"

[5] IEC 62304, "Medical devices – Quality management systems – Requirements for regulatory purposes"

[6] ISO 26262-9, "Road vehicles – Functional Safety – Part 9: ASIL-oriented and safety-oriented analyses"

[7] ISO 26262-2, "Road vehicles – Functional Safety – Part 2: Management of functional safety"

[8] http://it-republik.de/php/artikel/Agile-Development-mit-Scrum-2166.html

[9] http://www.medconf.de/agenda/1Konferenztag.htm

[10] Florian Prester, "Model-centric testing", testing experience – The Magazine for Professional Testers, March 2010

[11] H. Götz, M. Nickolaus, Th. Roßner, K. Salomon, iX Studie "Modellbasiertes Testen", 01/09, Heise Zeitschriften Verlag GmbH Co KG

All links called on May 19th, 2010.

Author CV

Dr. Anne Kramer

Anne Kramer was born in 1967 in Bremen (Germany). She studied Physics at the University of Hamburg where she obtained her diploma in 1992. In 1995 she received her PhD at the Université Joseph Fourier in Grenoble (France). Immediately afterwards she started working for Schlumberger Systems in Paris - first as software developer for smart card tools, then as project manager for point of sales terminals.

In 2001 she joined sepp.med (near Erlangen, Germany), a service provider specialized in IT solutions with integrated quality assurance in complex, safety-relevant domains. Currently, Anne Kramer is working as project manager and process consultant in the medical domain.

Contextualizing Agile Software Development

Philippe Kruchten, Ph.D., P.Eng.
Kruchten Engineering Services Ltd
Vancouver, BC, Canada

Abstract

This paper presents a contextual model for software-intensive systems development to guide the adoption and adaptation of agile software development practices. This model was found especially useful when the project context departs significantly from the "agile sweet spot", i.e., the ideal conditions in which agile software development practices originated from, and where they are most likely to succeed, "out of the box". This is the case for large systems, distributed development environment, safety-critical systems, system requiring a novel architecture, or systems with an unorthodox business model or governance model.

Keywords

Agile software development, software development process, process adaptation

1 Introduction

1.1 Agility defined

Drawing from earlier definitions from Jim Highsmith [1] or from Steve Adolph and the OODA loop [2], we define *agility* as "the ability of an organization to react to changes in its environment faster than the rate of these changes." This definition uses the ultimate purpose or function of being agile for a business, rather than defining agility by a labeled set of practices (e.g., you're agile when you do XP [3], Lean [4-6], or Scrum [7-9]) or by a set of properties defined in opposition to another set -- the agile manifesto approach [10]. This definition is not too far from that Kieran Conboy arrived at in his survey of agile process development literature [11].

An analogy could be the definition of a road. Would you define a road as something made of crushed rocks and tar, or define it as a surface that is black rather than white, flat rather than undulated, and with painted lines rather than monochrome? Or would you rather define a road as a component of a transportation system, allowing people and goods to be moved on the ground surface from point A to point B? And then let the properties or components of the road be derived from this, allowing some novel approaches in road design.

Note also that it is quite possible to adopt a labeled set of agile practices: Crystal [12], SCRUM, FDD [13], DSDM [14, 15], or a set of practices that perfectly conform to the agile manifesto [10] and not become agile. You would then "do agile", but you are not agile.

1.2 Agile methods successes

Agile software development methods do undoubtedly succeed in contexts that are identical or very similar to the contexts in which they have been initially created. As these contexts—the "agile sweet spot" [16]—are very frequent in software development, representing more than 70% of all software being developed, this may have led sometimes their proponents to a certain complacency: thinking that their method has universal value, that its represents some ultimate recipe, the holy grail of software engineering.

Agile methods may fail in various ways when they are applied "out of the box", i.e., with no or little adaptation, in contexts that are very far, at least on some dimension, from the context in which they have been originally created. Rather than an analysis of the root cause, this usually triggers screams of "you must have not done it right" by its proponents. And this again leads to discussions of "purity", "scrumbuts", etc. [39].

Agile methods can be stretched with variable success outside of the context in which they have been created; for example, scaling them up to larger projects, or across distributed teams. In our experience, the contextual factors that have the greatest risks of derailing agile projects are:

- size
- large systems with a lack of architectural focus
- software development not driven by customer demand
- lack of support from surrounding stakeholders, traditional governance
- novice team
- very high constraint on some quality attribute (safety-critical system, real-time constraints).

As noted by many authors in the last few years [17, 18], we cannot just rely on acts of faith by eloquent process gurus to help us define the adequate process, or set of practices outside of the agile sweet spot. Cold-headed, impartial investigation is required. Such research is generally not very easy to conduct; it is often qualitative, rather than quantitative, it draws more from social sciences than computer science, not easy to publish, not easy to carve down to masters' thesis bite size.

1.3 Overview of this paper

In this paper, drawing from our experience at Rational Software, and subsequently at KESL in consulting engagements with several organizations that have attempted to transition to agile software development methods, we present a contextual framework or model for situating agile practices. We define several factors characterizing software development context, which affect significantly the adoption of agile methods, especially when projects are outside of the "agile sweet spot" in which these methods have been defined and in which they operate at best. We describe four organizations that are developing software outside of this "sweet spot" and have run into difficulties applying agile software development methods "out of the box". We compare our model with other similar proposals. The ultimate purpose of this work, however, is to provide means for organizations to rapidly configure their method, by providing guidance on what agile practice to use in which circumstances.

2 Context Defined

Real estate agents in North America will tell you that only 3 factors do matter in their business: "Location, location, and location." For software process, we claim that only three factors matter: "context, context, and context." In *Voyage in the Agile Memeplex* [19], we had stressed the necessity to put our processes in context. but did not defined what "context" meant. It is very unfortunate that too many advocates of agile development practice are preaching good practices, but completely removed from the context in which they were proven to be successful. It turns some of their followers, several levels of transmission down to become just blind bigots, sometimes rabid bigots (see [19] for this phenomenon and the down side of *decontextualization*).

2.1 Two levels of context

There are 2 sets of factors that make up the context, which can be partitioned roughly in 2 sets: factors that apply at the level of whole organization/company, and factors that apply at the level of the project. In small organizations, with few software development projects, this distinction does not apply, and all factors are on the same level.

The organization-level factors (environment conditions) do influence heavily the project-level factors, which in turn should drive the process and practices that should be used

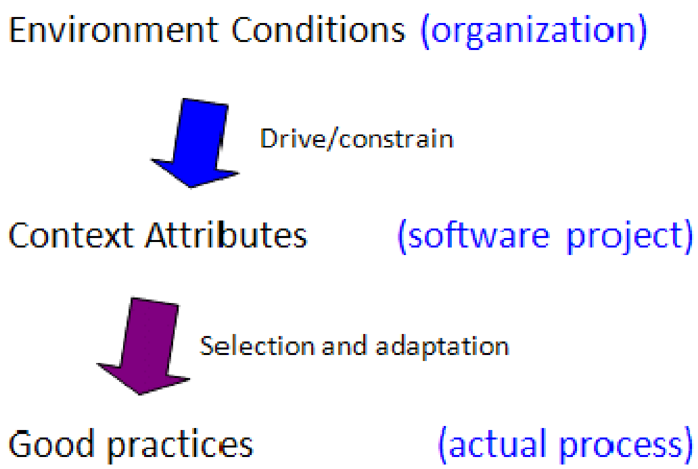


Fig. 1 – Environmental and project level context attributes

2.2 Organizational level: environmental factors

A certain number of factors are attached to the organization developing software, more than to the particular software development project.

1. Business domain

For what domain of activity is this organization developing software? Web-based systems, aerospace embedded systems, small hand-held instrumentation? Is software development the primary business activity of the organization, or at the other end, are we dealing with the IT organization of a business for which software is not at all the primary output. This factor more than any of the other 4 will condition, or constrain many of the project-level factors. For example, aerospace or biomedical instrumentation projects tend to be safety-critical.

2. Number of instances

How many instances of the software system (large or small) will be actually deployed? Are you building one single system, a dozen, a thousand, or millions? One-off systems are often internal to an organization, or developed on demand by a system integrator.

3. Maturity of organization

How long has that organization been developing software? How mature are the processes (and the people) relative to software development? Is the organization a small start-up, an SME (small or mediuym enterprise) or a large multinational firm? A small start-up is more likely to focus on a single, commercial piece of software, or more rarely open-source.

4. Level of innovation

How innovative is the organization? Are you creators or early adopters of new ideas and technologies? Or treading on very traditional grounds? Large IT organizations or government agencies tend to be rather risk-adverse and rarely tread on unchartered territory.

5. Culture

In which culture are the projects immersed? We are speaking here of both national culture and corporate culture? What are the systems of values, beliefs and behaviours that will impact, support or interplay with the software development practices? [20-24]

2.3 Project-level context attributes: the octopus model

At the level of a given software development project, we've identify empirically eight key factors:

1. Size

The overall size of the system under development is by far the greatest factor, as it will drive in turn the size of the team, the number of teams, the needs for communication and coordination between teams, the impact of changes, etc. [25, 26]. Number of person-months, or size of the code,, or development budget are all possible proxies for the size.

2. Stable architecture

Is there an implicit, obvious, de facto architecture already in place at the start of the project? Most projects are not novel enough to require a lot of architectural effort. They follow commonly accepted patterns in their respective domain. Many of the key architectural decisions are done in the first few days, by choice of middleware, operating system, programming languages, etc. [27] Some proponents of agile methods dismiss architecture at Big Up-Front Design, or YAGNI (You Ain't Gonna Need iI) [17, 28, 29] or confine it to a simple explanatory metaphor [3] (which is good, but not always sufficient).

3. Business model

What is the money flow? Are you developing an internal system, a commercial product, a bespoke system on contract for a customer, a component of a large system involving many different parties? Is it free, libre and open-source software (FLOSS)?

4. Team distribution

Linked often to the size of the project, how many teams are involved and are not collocated? This increases the need for more explicit communication and coordination of decisions, as well as more stable interfaces between teams, and between the software components that they are responsible for [30, 31]. Open-source development very often deals with scattered individuals, not teams.

5. Rate of change

Though agile methods are "embracing changes", not all domains and system experience a very rapid pace of change in their environment. How stable is your business environment and how much risks (and unknowns) are you facing? There are still projects with very stable requirement definitions.

6. Age of system

Are we looking at the evolution of a large legacy system, bringing in turn many hidden assumptions regarding the architecture, or the creation of a new system with fewer constraints?

7. Criticality

How many people die or are hurt if the system fails? Documentation needs increase dramatically to satisfy external agencies who will want to make sure the safety of the public is assured [32-35].

8. Governance

How are projects started, terminated? Who decides what happens when things go wrong? How is success or failure defined? Who manage the software project managers? [36-38]

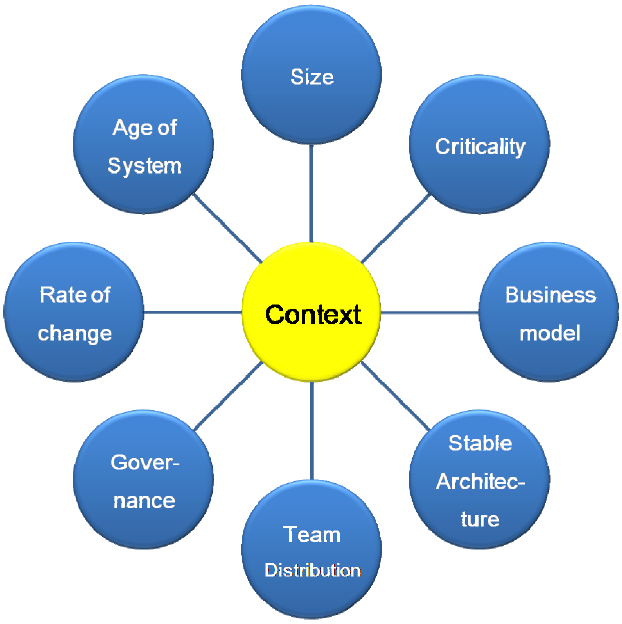


Fig.2 – The octopus: 8 key contextual factors for agile software development

2.4 Relationship between the two sets of factors

As you can expect, the first set of factors (organizational level) impacts and constrains the second set. But there is still a wide range of variation in this second set inside any given organization, especially large software development shops (a.k.a. system integrators) offering "bespoke" software development services.

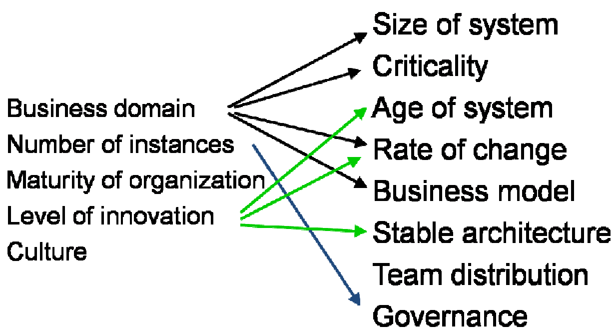


Fig.3 - Relationships between the 2 sets

If your business domain is "aerospace-onboard software", this will pre-condition several project-context factors: criticality, size, business model, etc. This in turn will make some practices suitable or not, will influence amount and type of documentation, the "level of ceremony" required by a given project.

2.5 The agile sweet spot

Figure 4 shows, based on this model, the "agile sweet spot," i.e., the conditions under which most "labelled" agile software development have been developed, and for which success is pretty much assured [16]. This would be the case for example of a web-based e-commerce site, built on dot-net technology by a small team, interacting with their customers.

System Size	• 0 ..12 … 300
Criticality	• Simple, $ losses, … deaths
System Age	• Exploratory, greenfield, legacy maintenance
Rate of change	• Low, medium, high
Business model	• In house, Open source, ….
Stable architecture	• Stable, changed, new
Team distribution	• Collocated, …, …, offshore outsource
Governance	• Simple rules, …, SOX, …

Fig. 4 – A particular context: the agile sweet spot (in blue)

The “agile sweet spot” tends to be for collocated team, of less than 15 people, doing greenfield development for non safety-critical system, in rather volatile environment; the system architecture is defined and stable, and the governance rules straightforward. Again, we are not trying to say that agile practices do not work outside of this sweet spot, but that many are challenged, will need some adaptation, and in some cases not be suitable.

3 Four Agile projects outside of the sweet spot

In our consulting practice we have come across 4 organizations in the last 10 years that have tried to embrace agile software development methods, but have run into difficulties [39]. We found in many cases that this was mostly because they were in some way outside of the “agile sweet spot”.

1. Project FIN

This large legacy software was being re-implemented in Java: 50 developers, collocated for about 18 months. Although progressing very well for the first 6 months using a combination of XP and Scrum, this project “hit a wall” after a while, mostly because of its failure to put in place a solid underlying architecture, driven by a naïve belief that a solid software architecture would gradually emerge as the result of bi-weekly refactorings.

2. Project FAC

A large factory automation project, with again a large amount of legacy software, it felt outside of the sweet spot on several other dimensions: safety-critical, very low rate of change, and no concept of “customer” to interact with, since many aspects are just related to physics. Software cannot be tested in real environment, and there is only one release opportunity per year.

3. Project FLY

Multiple inter-related legacy projects and “greenfield” projects, some deployed in the cockpit of aircraft, having to comply to the highest safety critical standards [32], and therefore requiring large amount of very detailed documentation, in particular traceability of artifacts, which run counter to the agile manifesto [10] and created cultural clashes in the development teams.

4. Project ANA

Small start-up company developing novel algorithms to support security trading, but not driven by customers’ demand, but driving the development internally based on models from physics, hence putting many of the agile practices at odds.

In each of these four projects, the way forward was defined the following pattern:

A) develop a clearer understanding of the contextual factors, using our model for example, allowed the project(s) to focus on what was specific to their project, and

b) specify the areas needing improvement from a software process perspective,

c) and finally select carefully the agile practices that would solve the issues they actually were facing, rather than blind application of a labeled agile methods in its totality, all practices lumped together; for example, all XP practices.

4 Review of other similar contextual models

Other authors have attempted to define the *context* of a software process, to define the main factors that affect the process they use/could use/would use. Let us examine a few such models in the agile adoption arena:

1. Boehm-Turner

In their book *Balancing agility and discipline* [39], Barry Boehm and Richard Turner used 5 factors to contrast software process/methods between what they call “plan-driven methods” and agile methods:

1. Size,
2. Criticality,
3. Personnel (their skill, know-how),
4. Dynamism (rate of change) *and*
5. Culture (of the team: thriving on chaos or on order).

Their 5 factors provided us with starting point, but we found that: a) they were mixing organization-level issues, with project-level issues, and b) they were lumping together too many aspects into the factors Personnel and Culture.

2. Cockburn & Crystal

In the Crystal family of processes [12], Alistair Cockburn defines different processes based on

1. Size,
2. Criticality, *and*
3. Skills.

The first two are fully aligned with ours, but we found the ’Skills’, which matches ‘Personnel’ in Boehm-Turner, difficult to use in practice. It is not quite a linear scale. We do capture some of it under “Maturity of the organization” however.

3. Ambler (IBM) and “Agile@Scale”

Closer to our views are those of Scott Ambler in *Agility at Scale* [40] which is summarized by his table reproduced in fig.5. We seem to cover mostly the main grounds, though with some small differences here and there, as shown in table 1. Scott Ambler seems to focus on scaling agility to larger projects, whereas I am mostly attempting to adjust to the context, regardless of the size or ambition.

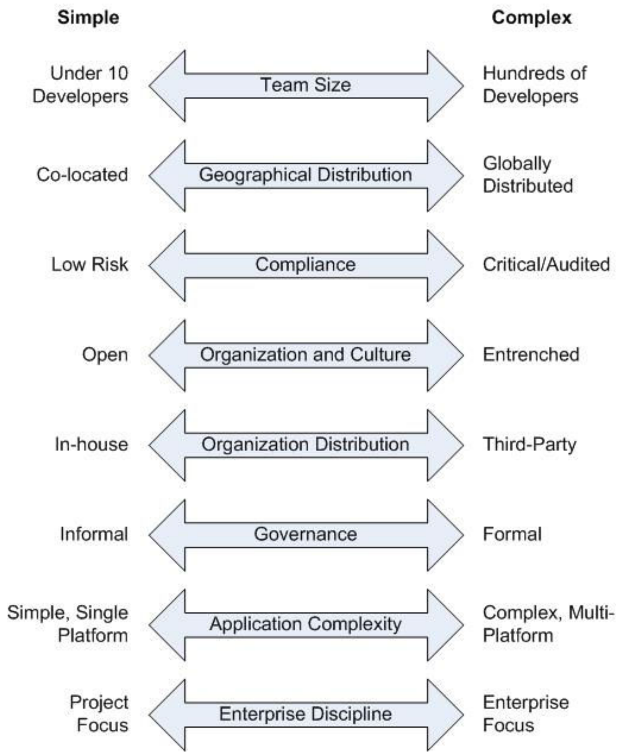


Fig.5- Scott Ambler's scaling factors (from fig. 4 in [40])

Table 1 – Comparing Agile@Scale [40] and our contextual model

Ambler	*Kruchten*	*Comment*
Team Size	Size	Number of people or SLOCs or function points? Either one is a possible indicator of size, as they are strongly correlated.
Geographical Distribution	Team distribution	(Same concept)
Compliance	Criticality	Not quite identical, but linked: critical system have to be compliant to standards and regulations: SOX [42], Cobit, [43], Basel II [43], or Do178B [32] depending on the industry
Organization & Culture	*Culture*	Rarely specific to one project
Organization distribution	Business model	(Same concept)
Application complexity	Stable architecture, Size	Could be also linked to innovation level
Enterprise discipline	*Maturity of the organization*	May vary across projects
Governance	Governance	(Same concept)
	Age of system	Issues with large legacy system
	Rate of change	Drives iteration duration, and the nature of the feedback loop

5 Future work

The authors and several colleagues have used the model in an ad hoc fashion, mostly applying judgement calls, and past (negative) experience to decide which practice could be used in a given combination of project factors. The ideal situation would be to build a kind of recommending system: a tool to which we would provide values for the 5 + 8 factors, that would give an indication of which practices are usable, which are not, which would require adaptation or special consideration, as the starting point for a process configuration. The main hurdle is to objectively populate such a table with reliable data, supported by evidence, and not just guts' feelings, as the number of data points is rather large. A short list of agile practices would be about 40: short iterations, managing a backlog, daily stand-up meetings, test-driven development, continuous integration, etc. With only 3 or 4 values for each of the 8 factors in the "octopus" we have already a thousand data points. Some cases are easier than others: short iterations, or conducting retrospectives have rather wide applicability. Table 2 shows an embryo of what such a table could contain.

Table 2 – Guidance for process configuration

Factor	**Size**			**Criticality**			**Distribution**				**Rate of change**		**Age of system**		**…**
Values	**S**	**M**	**L**	**L**	**M**	**H**	**-**	**M**	**L**	**XL**	**L**	**H**	**G**	**B**	
Iterations															
Daily standup															
Retrospective															
Pair prog.															
Backlog															
Metaphor															
Monthly release to users															
More agile practices…															

Legend:

- Practice is not affected by the factor
- Practice needs caution or adaptation
- Practice not useful, or require considerable adaptation
- Practice to avoid, could be dangerous or counterproductive
- Not enough evidence to conclude

Even in this simple form, the table generates lots of discussions and controversies, and each non-green cell must have an explanation, or rationale, and some caveats. So it should not be used as some kind of prescription, but more as a way to focus the reflection and discussion on the value of practices. Fortunately, as agile methods have crossed the chasm and become mainstream, there is a growing body of evidence to draw from, in particular experience reports, such as the ones collected through agile conference [45, 46], even though many tend to be heavily biased towards the "green", and complemented by studies of researchers, looking at the impact of going outside the sweet spot [47], or just plain old critiques of agile methods [48]. It should be useful to also submit non-agile practices to the same filter.

6 Conclusion

In assisting software organization adopting agile software development processes, we found valuable to first define the context using our model with several factors, then understand in which dimension(s) the project felt outside of an ideal agile sweet spot, and then in turn drive the adoption and possible adaptation of agile practices to this context, rather than a forced-fit, en-bloc adoption of all practices falling under a certain agile label, in the sometimes naïve hope that it will cure all ills. But again, agility should not be defined in terms of practices, but as the ability of an organization to react to changes in its environment faster than the rate of these changes.

References

1. Highsmith, J.A.: Agile software development ecosystems. Addison-Wesley, Boston (2002)
2. Adolph, W.S.: What lessons can the agile community learn from a maverick fighter pilot? In: Melnik, G. (ed.): Agile 2006 conference. IEEE Comp Soc, Minneapolis, MN, USA (2006) 94-99
3. Beck, K.: Extreme Programming Explained: Embrace Change. Addison-Wesley, Boston (2000)
4. Poppendieck, M., Poppendieck, T.: Lean Software Development - An Agile Toolkit. Addison-Wesley, Boston, MA (2003)
5. Poppendieck, M., Poppendieck, T.: Implementing Lean Software Development--From Concept to Cash. Addison Wesley, Upper Saddle River, NJ (2009)
6. Poppendieck, M., Poppendieck, T.: Leading Lean Software Development: Results Are not the Point. Addison-Wesley Professional, Boston, MA (2009)
7. Cohn, M.: Succeeding with Agile-Software development using Scrum. Pearson Education, Boston, MA (2009)
8. Schwaber, K., Beedle, M.: Agile Software Development with SCRUM. Prentice-Hall, Upper Saddle River, NJ (2002)
9. Schwaber, K.: The Enterprise and Scrum. Microsoft Press, Redmond, WA (2007)
10. Agile Alliance: Manifesto for Agile Software Development (2001). At http://www.agilemanifesto.org/
11. Conboy, K.: Agility from First Principles: Reconstructing the Concept of Agility in Information Systems Development. Information Systems Research 20 (2009) 329-354
12. Cockburn, A.: Crystal Clear--A Human-Powered Methodology for Small Teams. Addison-Wesley, Boston, MA (2004)
13. Palmer, S.R., Felsing, J.M.: A Practical Guide to Feature-Driven Development. Prentice-Hall, Upper Saddle River, NJ (2002)
14. Stapleton, J.: DSDM, Dynamic Systems Development Method: The Method in Practice. Addison-Wesley, Reading, MA (1998)
15. Stapleton, J. (ed.): DSDM Business Focused Development. Addison-Wesley, London, UK (2003)
16. Kruchten, P.: Scaling down projects to meet the Agile sweet spot. The Rational Edge (August 2004) At: http://www-106.ibm.com/developerworks/rational/library/content/RationalEdge/aug04/5558.html
17. Abrahamsson, P., Ali Babar, M., Kruchten, P.: Agility and Architecture: Can they Coexist? IEEE Software 27 (2010) 16-22
18. Abrahamsson, P., Salo, O., Ronkainen, J., Warsta, J.: Agile Software Development Methods-Review and Analysis. VTT Technical Research Centre of Finland, Oulu, Finland (2002)
19. Kruchten, P.: Voyage in the Agile Memeplex: Agility, Agilese, Agilitis, Agilology. ACM Queue 5 (2007) 38-44 also at http://queue.acm.org/detail.cfm?id=1281893
20. Borchers, G.: The Software Engineering Impacts of Cultural Factors on Multi-cultural Software Development Teams. 25th International Conference on Software Engineering (ICSE'03). IEEE Computer Society, Portland, OR (2003) 540-545
21. Sharp, H., Woodman, M., Hovenden, F., Robinson, H.: The Role of 'Culture' in Successful Software Process Improvement. 25th Euromicro Conference (EUROMICRO '99), Vol. 2. IEEE-CS, Milan, Italy (1999) 2170-2176
22. Hsieh, Y., MacGregor, E.L., Kruchten, P.: Intercultural Factors in Global Software Development. In: Kasap, S. (ed.): 18th Annual Canadian Conference on Electrical and Computer Engineering (CCECE'05). IEEE, Saskatoon, SK (2007)
23. Kruchten, P.: Analyzing Intercultural Factors Affecting Global Software Development. In: Damian, D., Lanubile, F. (eds.): 3rd International Workshop on Global Software Development (GSD2004), Collocated with ICSE 2004, Edinburgh, Scotland, IEE (2004) 59-62
24. Sharp, H., Robinson, H., Woodman, M.: Software engineering: Community and culture. IEEE Software 17 (2000) 40-47

25. Larman, C., Vodde, B.: Scaling Lean & Agile Development--Thinking and Organizational Tools for Large-Scale Scrum. Addison-Wesley Professional, Boston (2008)

26. Leffingwell, D.: Scaling Software Agility: Best Practices for Large Enterprises. Addison-Wesley Professional, Boston (2007)

27. Kruchten, P.: The Rational Unified Process: An Introduction. Addison-Wesley Professional (2003)

28. Madison, J.: Agile-Architecture Interactions. IEEE Software 27 (2010) 41-47

29. Blair, S., Watt, R., Cull, T.: Responsibility-Driven Architecture. IEEE Software 27 (2010) 26-32

30. Ågerfalk, P.J., Fitzgerald, B.: Flexible and Distributed Software Processes: Old Petunias in New Bowls? Comm. ACM 49 (2006) 27-34

31. Lings, B., Lundell, B., Ågerfalk, P.J., Fitzgerald, B.: Ten strategies for successful distributed development. Proceedings of the IFIP WG 8.6 Conference, Galway, Ireland. Vol. 206. Springer-Verlag (2006) 94-112

32. RTCA/DO-178B Software Considerations in Airborne Systems and Equipment Certification. Radio Technical Commission for Aeronautics, Washington, DC (1992)

33. Leveson, N.G.: Safeware: System Safety and Computers. Addison-Wesley, Reading, MA (1995)

34. Paige, R.F., Chivers, H., McDermid, J.A., Stephenson, Z.R.: High-Integrity Extreme Programming. 2005 ACM Symposium on Applied Computing. Santa Fe, New Mexico, USA, ACM (2005) 1518-1523

35. Poppendieck, M., Morsicato, R.: Using XP for Safety-Critical Software. Cutter IT Journal 15 (2002) 12-16

36. Cockburn, A.: A Governance Model for Incremental, Concurrent, or Agile Projects. CrossTalk: The Journal Of Defense Software 19 (2006) 13-17

37. Dubinski, Y., Yaeli, A., Feldman, Y., Zarpas, E., Nechushtai, G.: Governance of Software Development: The Transition to Agile Scenario. In A. Cater-Steel (Ed.), IT Governance and Service Management Frameworks and Adaptations, Idea Group Publishing, Information Science Publishing, IRM Press, Hershey, PA (2009) 266-284

38. Dubinsky, Y., Kruchten, P.: Software Development Governance (SDG): Report on 2nd ICSE Workshop. ACM SIGSOFT Software Engineering Notes 24 (2009) 46-47

39. Hoda, R., Kruchten, P., Noble J., Marshall, S.: Agility in Context, in: Rinard, M. (Ed.): Proceedings of OOPSLA 2010 (part of SPLASH), Las Vegas, October 17-21, 2010, ACM (2010)

40. Boehm, B.W., Turner, R.: Balancing Agility and Discipline--A guide for the perplexed. Addison-Wesley, Boston, MA (2003)

41. Ambler, S.W.: Agility at Scale: Become as agile as you can be. IBM, Toronto (2009). eBook at http://www.internetevolution.com/ebook/ebookibm7/

42. US Public Law 107-- 204 - Sarbanes-Oxley Act of 2002. US Government Printing Office, Washington (2002).

43. CobiT 4.0--Control Objectives for Information and related Technology--Control Objectives, Management Guidelines, Maturity Models. IT Governance Institute. Rolling Meadows, Il., USA (2005)

44. Basel II--Revised international capital framework, Bank for International Settlements, Basel, CH (2004).

45. Melnyk, G, Poppendieck, M. (Eds.): Proceedings of Agile 2008 Conference, Toronto, Aug. 4-8, IEEE Computer Society, Los Alamitos (2008)

46. Dubinsky, Y., Dybå, T., Adolph, S., Sidiky, A. (Eds.): Proceedings of Agile 2009 Conference, Chicago, Aug 24-28, IEEE Computer Society, Los Alamitos (2009)

47. Šmite, D., Moe, N.B., Ågerfalk, P.J. (Eds.): Agility across Time and Space -- Implementing Agile Methods in Global Software Projects, Springer-Verlag, Berlin (2010)

48. Stephens, M., Rosenberg, D.: Extreme Programming Refactored--The Case Against XP. Apress LP, Berkeley, CA (2003)

Author CV

Philippe Kruchten

Philippe Kruchten is the founder and principal at KESL in Vancouver, Canada. He is also professor of software engineering at the University of British Columbia in Vancouver, which he joined in 2004 after a 32-year career in the software industry, developing systems in telecommunications, defense and aerospace. He holds an NSERC chair in design engineering. His main interests are in software architecture, software project management and software development processes. During his 17 years with Rational Software (now part of IBM) he led the development of the Rational Unified Process® (RUP®), an iterative and incremental software development process. He is the co-founder and secretary of the IFIP working group 2.10 on Software Architecture, and the co-founder and past chair of Agile Vancouver, a special interest group focusing on agile software development approaches.
Dr. Kruchten has a mechanical engineering degree from Ecole Centrale de Lyon (France), and a doctorate degree form Ecole Nationale Supérieure des Télécommunications in Paris. He is a licensed professional software engineer in British Columbia, Canada, an IEEE Certified Software Development Professional (CSDP) and even a Certified Scrum Practitioner (CSP). He is a member of IEEE, ACM, INCOSE and AIS.

Agility meets Systems Engineering: a Catalogue of Success Factors from Industry Practice

Ernst Stelzmann, Christian Kreiner, Gunther Spork, Richard Messnarz, Frank Koenig

Abstract

Agile software development methods are widely accepted and valued in software-dominated industries. In more complex setups like multidisciplinary system development the adoption of an agile development paradigm is much less straightforward. Bigger teams, longer development cycles, process and product standard compliance and products lacking flexibility make an agile behaviour more difficult to achieve. Focusing on the fundamental underlying problem of dealing with ever ongoing change, this paper presents an agile Systems Engineering approach as a potential solution. Therefore a generic Systems Engineering action model was upgraded respecting agile principles and adapted according to practical needs discovered in an empirical study. This study was conducted among the partners of the S2QI agile workgroup made up from experts of automotive, logistics and electronics industries. Additionally to an agile Systems Engineering action model, a list of 15 practical success factors that should be considered when using an agile Systems Engineering approach is one of the main outcomes of this survey. It was also found that an agile behaviour in Systems Engineering could be supported in many different areas within companies. These areas are listed and it is also shown how the agile action model and the agile success factors are related to them.

Keywords

Agility, Systems Engineering, experience

1 Introduction

In IBM's Global CEO Study 2006 65% of all CEOs were expecting substantial change for their companies within the next 3 years. In 2008 this value increased to 83%. During the same period the number of CEOs saying their companies had already dealt with change successfully was only increasing from 57% to 61% [1].

So change is not only a factor that most companies have to deal with, it is also a challenge that a significant number of companies had not handled successfully yet. This paper examines companies that are developing systems with the help of Systems Engineering methods. The challenge this paper deals with is the dynamic market environment for system developers that is influenced by 4 interrelated factors (see Figure 1).

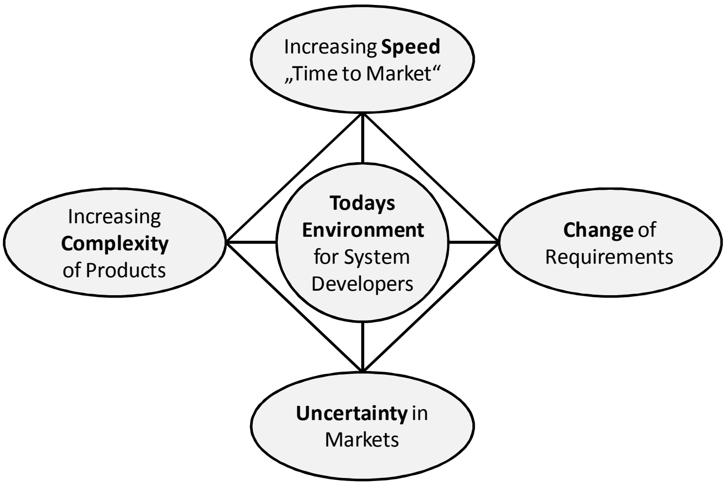


Figure 1: Characterization of Market Environment

For developing software within this environment, some so called "Agile Methods" have been introduced, like Extreme Programming [2], SCRUM [3], Crystal [4] and Feature Driven Development [5]. The specific methods have following common characteristics [6]:

- Learning attitude
- Focus on value for the customer
- Short iterations delivering value
- Neutrality to change
- Continuous integration
- Test-driven
- Lean attitude
- Team ownership

More generic principles for agile SW development are stated by the well known Agile Manifesto (http://www.agilemanifesto.org).

Although Software is often a major component of systems, it is not the only one. Other components need to be treated differently, since they have to be produced physically. Sometimes the production takes a long time and thus it takes also a long time to implement changes (e.g. several months to prepare a casting mould). Development processes are also different for SW and different types of HW (mechanical, electronically, etc.). Hence it is also a necessary Systems Engineering task to synchronise the different development cycles. And usually system development projects are bigger than sheer SW development projects in terms of team size, complexity of product and development time. So it is questionable if these agile development methods for SW can also be applied for developing multidisciplinary systems. It is also questionable if principles, which result in an agile behaviour in SW development, will also support an agile behaviour in systems engineering. Therefore the idea of agility will be analyzed more in detail:

The most comprehensive and generic definition we found for agility is: “Agility is a persistent behaviour or ability of a sensitive entity that exhibits flexibility to accommodate expected or unexpected changes rapidly, follows the shortest time span, uses economical, simple and quality instruments in a dynamic environment and applies updated prior knowledge and experience to learn from the internal and external environment.’’ (Qumer and Henderson-Sellers [7])

An often used tool to visualise the concept of agility is John Boyd’s OODA Loop [8]. It distinguishes 4 phases necessary for responding to change (new information) and calls them Observe, Orient, Decide and Act. It also points out how the later phases influence the earlier ones and shows feedback loops.

In a prior paper [9] we concluded that agility can be supported by enhancing any phase of this concept, but all phases must be considered to generate a comprehensive agile Systems Engineering approach. So it has to be considered:

- The ability to gather new information
- The ability to analyze this information
- The ability to decide for its relevance and how to react
- The ability to respond, which can also be called “flexibility”

And furthermore it has to be taken into account that changes could arise from outside of the development process (like changes of customer needs or laws that have an influence on the product) but could also become apparent after receiving feedback from previous development actions. And this does not have to be feedback on failures. High complexity within systems and short “time to market” often result in the fact, that not all requirements are known, when development starts. Also the fact that behaviour of complex systems can’t be foreseen exactly, but emerges when all components of the system are integrated, makes it necessary to start development, without having all information in advance. So preliminary stated requirements may change or new requirements may show up when more information (feedback) becomes available during development. Since changes are easier (and cheaper) to handle in early development phases [10], the development process has to force early information gathering (feedback) about the behaviour of the system or possible failures within the state of development. And it should also provide the customer/user with all information on the system he needs to be able to give feedback about his real expectations about the system function. So these changes can be caused externally or internally and their moment of appearance is strongly influenced by the Systems Engineering process and interactions with customers and users.

So an agile Systems Engineering approach should be able to deal with both types of changes and should consider all phases of the OODA Loop. And according to the definition of agility it should do it fast.

The agile Systems Engineering approach presented in this paper consists of 3 parts that will be explained in the following 3 chapters:

- An agile Systems Engineering action model
- A list of supportive actions in different areas of companies
- A catalogue of success factors for applying agile methods

2 Agile Systems Engineering Action Model

There is no commonly accepted approach for an "Agile Systems Engineering" existing right now, but several companies try to apply agile SW development methods to develop multidisciplinary systems. So do the companies, which are members of S2QI agile workgroup, an initiative to share experience on selected topics among its members. A survey in these companies (from automotive, logistics and electronics industries) revealed that using agile SW development methods may help in becoming more agile also for developing systems. But this is not a trivial undertaking. A lot of things have to be considered for generating a Systems Engineering approach that should be capable to show agile behaviour but should also comply to general Systems Engineering objectives like effectiveness and efficiency.

The first part of the approach in this paper is an action model for Systems Engineering. As a foundation the Systems Engineering Action Model designed at ETH Zurich [11] was used. It is a generic model that consists of principles and procedures for a good Systems Engineering practice. Other basic principles for the agile action model were taken from the Agile Manifesto and several agile SW development methods. Not all agile principles were applied, only those, which seemed to be appropriate for Systems Engineering. Furthermore ideas from analyzing the basic concept of agility with the help of the OODA Loop were implemented. Figure 2 shows a version of this Agile Systems Engineering Action Model that was already developed further after partners from S2QI agile workgroup had reviewed it. It consists of 4 main principles:

- Esteem Developers: This principle should assure that developers are treated in the way they deserve. Often processes are cared more about then developers. But developers are doing the work, they are the brainpower in each process and they ensure agile behavior. Therefore it is wise to consider all suggestions, stated in the figure, so that the developers can do their job in the best way.
- Incremental Development with close Customer/User Interaction: This principle proposes that it is not necessary to specify all requirements in detail at the beginning, because some (or many) of them will change during development. Therefore a "product vision" should be enough when the project starts. Then development proceeds by developing items that could be any physical part of the system, but also prototypes, simulation models, plans and so on. Items that push information gathering for eliciting requirements or for better understanding of the system should be developed first. A finished item should always be something that could be presented to the customer to receive feedback.
- Iterative Development of Increments: Each development item is developed in iterations of the displayed development cycle. It starts with a planning phase, where it is decided what to develop, who will develop and how long it may take. After that detailed objectives and test criteria are formulated. Then the designated developers start to develop their parts whereas "search for solutions" and "selection of solution" are technical planning steps and "realisation" is the phase in which a system increment is being produced, a SW feature is coded or a document is created. If the search for solution provided different variants, the next iteration can start by selecting another variant, after a negative test result. After a passed test, the developed item is presented to the customer.
- Flexible Design: This should be applied to the product itself, but also to other things like processes, organisation, production and supply chain. The Incremental- and Iterative Development principles are mostly important for agile reactions on changes that come apparent due to new information emerging during development. The Flexible Design principle is imperative for all changes, since there has to be flexibility somewhere (mainly in the product itself) for implementing changes. Some guidelines for making systems more flexible are presented later.

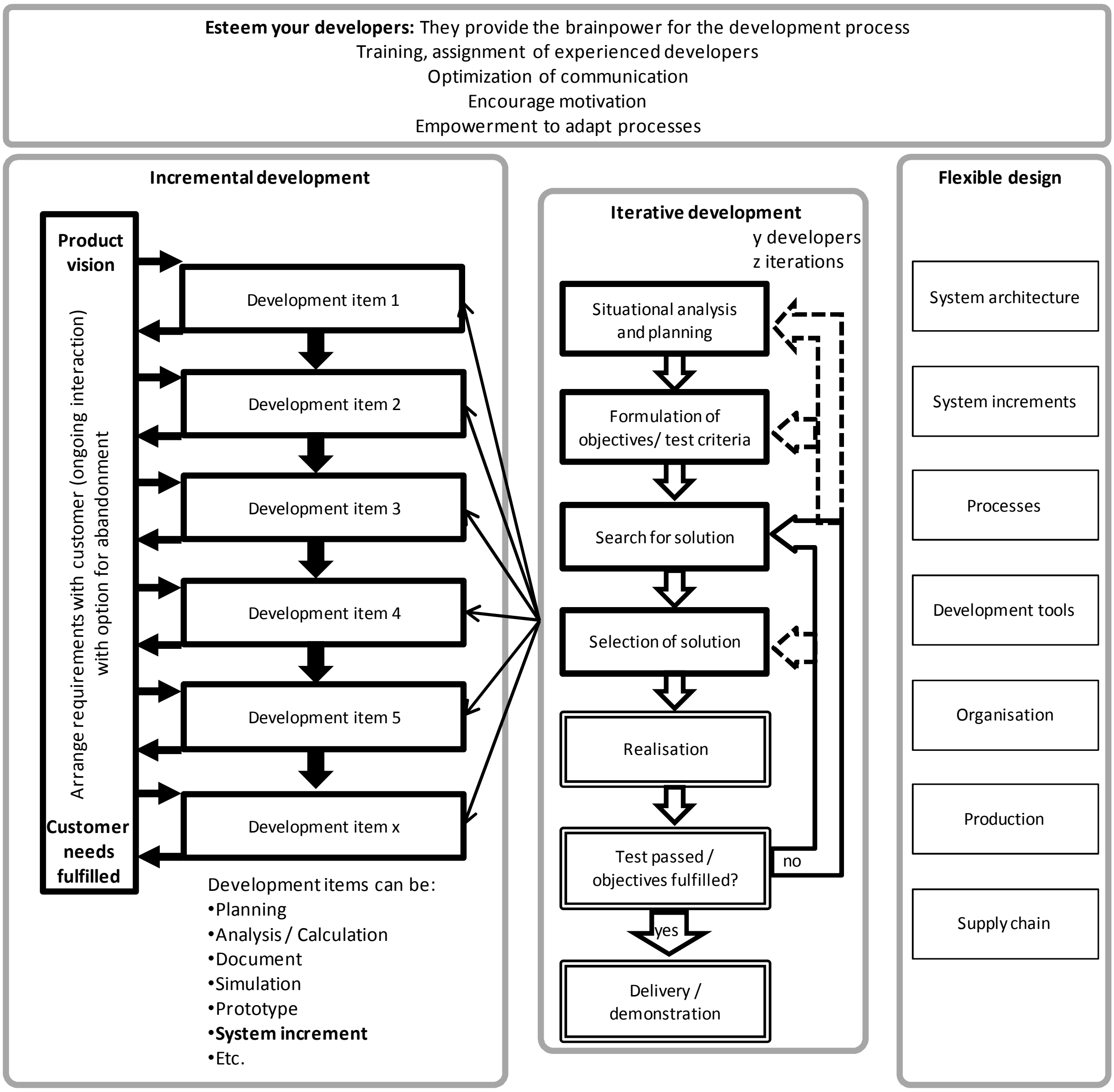


Figure 2: Agile Systems Engineering Action Model

The agile Systems Engineering action model may be used to adapt existing development processes for more agility as well as to create new processes showing agile behaviour without losing necessary characteristics for good Systems Engineering practice. Some parts of it like handling of developers and customers and flexible design are partly process topics but are reaching beyond. Since neither Systems Engineering nor Agility should be limited to process management a comprehensive look on system developing companies was taken to find more starting points for supporting agile behaviour in Systems Engineering.

3 Supportive Actions in Different Areas of Companies

The second part of the agile approach in this paper describes supportive actions for an agile behaviour within Systems Engineering. At first a classification into categories should be found to better handle the large number of possible options. It was chosen to use areas of system developing companies as shown in figure 3.

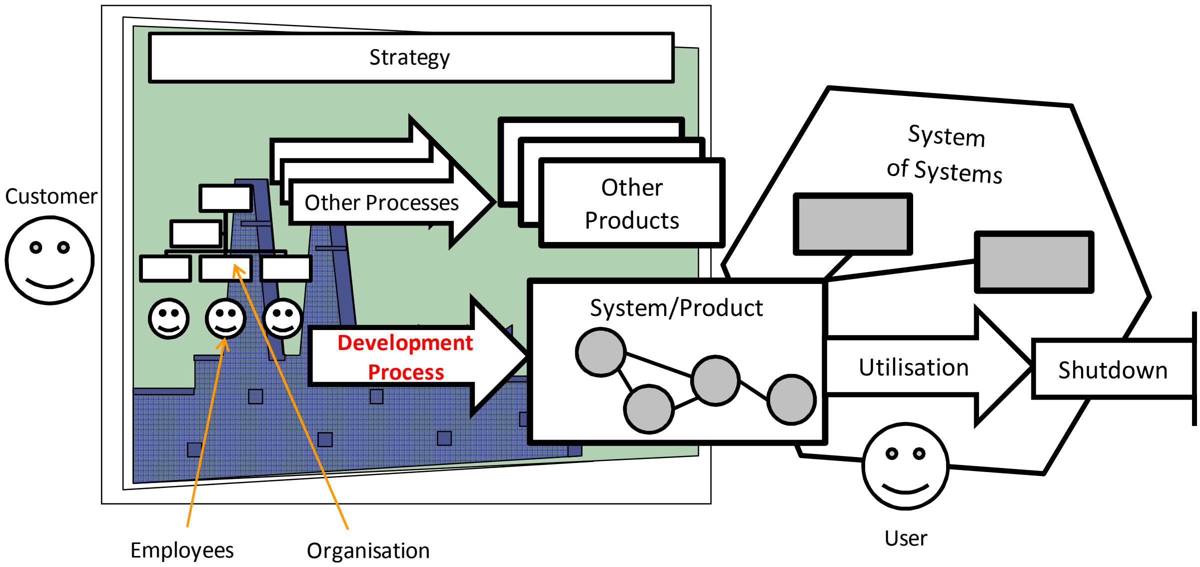


Figure 3: Relevant Areas for Agility within System Developing Companies

- Interaction with customer and user

This is already covered by agile SW development methods [2], [3], [4]. Customer and user should be involved in system development during the whole project. They are required to define their real needs and prioritize them. If their needs change throughout the project or can be stated more precisely, this new information should be considered immediately. This is largely a process topic, but also communication and psychological issues have to be treated, therefore it was chosen to be a category.

- Organisation

The organisational form of the company and of the project is important for communication, which is seen as one of the major points for agile behaviour [2], [3], [4]. Also the composition of teams and the collaboration between different teams is important. Team members who leave the company can also be seen as changes, which have to be dealt with within this area.

- Employees / culture

Employees are the main drivers for agility because they naturally have the ability to gather new information, decide for their relevance and take action to change something. So employees are able to fulfil all phases of the OODA-Loop. Therefore employees should be in centre of all agile considerations. Corporate culture and organization should be optimized to not hinder employees in being agile. It is also considered that high-skilled and experienced employees have more potential for being agile [2], [3], [4].

- Development process

Also the development process should not restrain employees in being agile. It should rather support agility by including tasks for observing new information and information processing. The management of requirements is essential for agility and also a fast providing of helpful prototypes or parts of the system. When changes appear the process should provide routines to adapt the product. An agile process is also adaptable for itself. Agile SW development methods are mainly focusing on process topics [2], [3], [4], [5] and seem to be good starting points for agile Systems Engineering processes.

- Product (System)

The system is the central demand of the customer and it is obvious that the system has to be changed when it doesn't fulfil the requirements. So the system has to provide the flexibility that is needed for reacting on change. To enhance flexibility of product architecture several concepts have been published. Fricke and Schulz [12] list following principles:

Ideality/simplicity (minimize number of interfaces, secondary functions …)

Independence (minimize impact of changing design parameters on other design parameters)

Modularity (minimize coupling among modules and maximize cohesion within modules)

Integrability (compatibility and interoperability applying generic, open or common interfaces)

Autonomy (objects are available to provide basic functionality independent from embedding system)

Scalability (units independent from scale or self-similar/fractals)

Non-hierarchical integration (units linked across system with no respect to type of modularity)

Decentralization (control of information, resources, attributes and properties distributed within system)

- Strategy

This topic addresses super ordinate decisions for the entire company. For example flexibility (limited to a special domain) for the company could be achieved with a platform strategy for similar products. When considering the company as a system itself, principles from 3.5, could be used to improve agile behaviour to form an “Agile Enterprise” [13].

Gathering of outside information could be done more efficiently on company level than in separated systems engineering teams. Changes happening there are not within the area of system developers' daily work. Therefore it makes no sense to make Systems Engineering processes responsive to all changes that might happen. Otherwise they would become very inefficient. An agile behaviour on enterprise (strategy) level might be a solution for this type of changes. Information gathering could be done on enterprise level and also strategies for reacting on change are better applicable by using all resources of an enterprise.

- System utilisation / market

On the one hand systems should be adaptable after launch (e.g. software-updates) and on the other hand it is sometimes possible to use systems for other purposes than originally planned. Therefore even the utilisation phase of a system should be taken into account to provide options for agility.

4 Agile Systems Engineering Success Factors

The third part of the agile Systems Engineering approach is a catalogue of success factors that have to be considered when applying "Agile Systems Engineering". They are not exclusively connected to the agile Systems Engineering action model, but should also be considered when using agile SW development methods for developing systems. The following factors were identified by studying real-world practices in companies working together in the S2QI agile workgroup. A factor was listed, if it was critical for successfully applying agile methods in at least two different companies. There are also several methods, principles and ideas named by S2QI partners presented as possible solutions.

- SF1: Agile project setup (alias project launch meeting)

During project launch it is necessary to set or adapt several organisational and process properties, because standardised values are not always appropriate. Right-sized project setup and management structures balance project management, quality, and development efficiency.

Possible solution: Dependent on the product, albeit starting from the organization's standard, it is possible to scale the approval plan (honouring a minimal set of quality gates), communication/reporting processes, team configuration and number of development cycles (mostly hardware). In an agile mindset, this process could occupy the project's first agile iteration (SCRUM: sprint).

- SF2: Change response strategies

Late changes do occur. Systematically responding resp. preparing to respond to changes can effectively mitigate their – typically negative – impact (extra costs and time). Flexibility exposed in this way is value on its own.

- SF3: Direct customer communication

It is necessary to organise communication between customer and development to be as direct as possible, because direct communication is efficient, has high bandwidth and the least losses.

Possible solution: Form pairs of customers and function owners on several levels to facilitate direct and parallel communication paths and leverage common (sub-) domain knowledge and vocabulary.

- SF4: Customer/market oriented vs. mandatory requirements priorities

Prioritisation of requirements to balance market orientation and technically mandatory requirements (e.g. safety or legal) is needed.

Possible solution: Drive Software development cycles by requirements derived from a prioritised feature/requirements backlog. Mandatory/major feature items are flagged, and cannot be postponed.

- SF5: Software and hardware development coordination and collaboration

Sub-disciplines within product development often have different cycle times, as well as different vocabulary. Coordination/Collaboration is essential to deliver an integrated product successfully.

Possible solution: Synchronise development iterations at several (integration) milestones. Use hardware emulation to enable early and instant test feedback to software developers (software cycles are normally much shorter than hardware cycles, esp. with agile software development)

- SF6&7: Flexible product (line) architecture and systematic reuse

Flexibility of product architecture is inevitable. Systematic reuse was seen as success factor at the beginning of S2QI's research. It is now seen as an option to ensure efficient, albeit flexible product development within a certain domain focus. Therefore SF7 was included in SF6 as possible solution.

Possible solution: Analyse your domain to define your core assets, the scope of your domain and your goals. Develop your components reusable. Take care of their interfaces, parameterisation and internal flexibility. A common architecture is essential. Have a named architect (role). In case of software biased products, this goal oriented, systematic reuse approach is called "Software product line".

- SF8: Effectively linking requirements and tests

Test cases must be updated, in order to be able to continuously check your product's features. Documentation effort should be kept low. Making requirements executable makes them unambiguous (necessary for execution) and reusable as automated test cases. Exercising executable requirements in verification phase and later as test case against the system also lets requirements errors, inconsistencies, etc. show up earlier.

Possible solutions: Make requirements executable so that they can serve as automated test cases. Use Test Driven Development. Red-Red-Green: have the test case executable and failing (red) before unit development. Develop functionality (frequent feedback: still red) until tests are "green". Think of unit tests as "casting moulds" for software units. Automated unit tests and module tests. "Test" the test cases (requirements) in return. Automated acceptance tests (FDD, ATDD). Think of a "function probe".

- SF9: Know your agile method

E.g. when using agile software development methods like SCRUM, it is necessary to clearly distinguish between product backlog (project/product features) and sprint backlog (single iteration features). When adopting agile development methods in conservative environment, multiple time-boxed iterations appear in addition to the global project development cycle. Clearly distinguishing between these is obviously necessary. It is reported however, that this separation is not done quite often.

Possible solution: Introduce all participants to your agile method. Product backlog (mostly) contains the current priority-sorted list of customer features. Sprint backlogs contain features to be realised during the current agile iteration (typically 1 month). Sprint backlogs are planned by the development team by taking features from the product backlog. A systematic, fair method for prioritisation of the product backlog is very helpful.

- SF10: Team work, team thinking, team taking responsibility

Working agile involves a lot of informal communication and tacit knowledge. The better the team spirit, the better their performance.

Possible solution: Do common planning (esp. sprint planning) but also common detailing of product backlog items within team. Develop and maintain the team's common goal orientation. Force "collective" commitment to reach the (self-planned!) sprint goals. Allow collective ownership of the realized substance. Do pair working (on demand).

- SF11: Synchronise sprint cycles with general organisation control cycles

Especially in mixed environments, where agile teams work within classical structures, sprint end delivery/reporting/retrospective events should be synchronised with company standard control structures.

Possible solution: Synchronise (some) sprint cycle ends with non-agile control structure due dates.

- SF12&13: Team and inter-team organisation

Teams only work well, when they are not too big and not too small [14]. Working agile involves a lot of informal communication and tacit knowledge that has to be exchanged across teams' borders. Therefore team size and interfaces between teams are critical. Possible solutions should consider large projects (e.g. SCRUM of SCRUMs), hardware/software teams, geographically separated teams etc.

- SF14: Ensure minimal requirements documentation

While keeping requirements documentation to minimum, sufficient information has to be available for serious sprint planning and task effort estimation.

Possible solution: Have a minimal requirements documentation pattern (e.g. description, additional documents, test cases, acceptance criteria). Possibly it can be supported in an RM tool.

- SF15: Generic requirements

Not really critical but by formulating generic requirements usefully work could become more efficient. E.g. by using parameters for generic requirements they could become reusable.

5 Conclusion

While agile software development is already widely accepted and adopted, there is no commonly accepted approach for agile Systems Engineering. Some companies were already trying to apply agile methods from SW development to the development of complete systems. So did the industry partners of S2QI agile workgroup. Since Systems Engineering is usually a complex endeavour there was no silver bullet found, to provide an agile solution for all issues. Systems show complexity for themselves with interdependencies between components but also Systems Engineering processes have a lot of interdependencies to processes and other things in different areas of system developing companies. Therefore a lot of elements have to be considered when a more agile behaviour within Systems Engineering should be achieved.

To provide a guideline for Systems Engineering with agile behaviour an agile Systems Engineering action model was presented that was combined from knowledge about Systems Engineering and different agile concepts and methods and reviewed by S2QI partners. This action model has an emphasis on process management but also addresses other topics like personnel and flexibility in products. In this sense system developing companies were analysed for possible areas where agility could be supported. Categorized into these areas starting points for agility were presented.

Finally a catalogue of success factors for applying agile Systems Engineering methods was elaborated. This list is the result from a survey conducted among partner companies of S2QI agile workgroup and is not directly connected to the presented agile action model. The catalogue of success factors should also be considered when applying only separate agile principles or an agile SW development method to system development. This catalogue is intended to be a starting point for discussion, extension, abstraction and other strategies to foster deeper understanding as well as widening practical applicability. To provide a comprehensive and agile approach that is capable of solving all issues within Systems Engineering more research is needed.

Literature

[1] S. J. Palmisano, IBM global CEO study 2008 – the enterprise of the future, available: http://www-935.ibm.com/services/us/gbs/bus/html/ceostudy2008.html

[2] K. Beck, Extreme Programming Explained: Embrace Change, Addison-Wesley Professional, October 1999

[3] K. Schwaber, Agile Project Management with SCRUM, Microsoft Press, 2004

[4] A. Cockburn, Crystal Clear: A Human-Powered Methodology for Small Teams, Addison-Wesley Professional, October 2004

[5] S. Palmer and J. Felsing, A practical guide to Feature Driven Development, Prentice Hall PTR, 2002

[6] R. Turner, Towards agile systems engineering processes, 2007, available: http://www.stsc.hill.af.mil/CrossTalk/2007/04/0704Turner.html

[7] A. Qumer and B. Henderson-Sellers, Crystallization of agility back to basics, ICSOFT 2006 (2) pp. 121-126, 2006

[8] Wikipedia: OODA Loop, http://en.wikipedia.org/wiki/OODA_Loop, 26.5.2010

[9] E. Stelzmann, G. Spork, Ch. Kreiner, R. Messnarz, F. Koenig, Practical Use of Agile Methods within Systems Engineering, International Spice Days, Stuttgart, June 21-23, 2010

[10] G. Lock, Project Management, Gower Publishing Limited, Hampshire, England, 2007

[11] R. Haberfellner et al., Systems Engineering: Methodik und Praxis, W.F. Daenzer and F. Huber (ed.), 11th Edition, 2002

[12] E. Fricke and A. P. Schulz, Design for changeability (dfc): Principles to enable changes in systems throughout their entire lifecycle, Systems Engineering Journal, vol. 8, no. 4, 2005

[13] R. Dove, Response Ability: The Language, Structure, and Culture of the Agile Enterprise, Wiley, 2001

[14] F.P. Brooks, The Mythical Man-Month: Essays on Software Engineering, 2nd Edition, Addison-Wesley Professional, 1995

Author CVs

Ernst Stelzmann

Dipl. Ing. Ernst Stelzmann is University Assistant at the Institute of General Management and Organization at Graz, University of Technology. He finished his studies of Industrial Engineering and Management at the same University in 2005. Then he worked for Magna Powertrain as Advanced Product Quality Planner until 2007 when he became University Assistant. The topic of his research and PhD thesis is "Agile Systems Engineering".

Christian Kreiner

Dr. Christian Kreiner is currently senior scientist at the Institute for Technical Informatics, Graz University of Technology (TUGraz) and industry consultant. He holds a Doctoral and a Dipl-Ing. degree in Electrical Engineering from TUGraz. He was co-founder of Salomon Automation GmbH, a logistics systems and software company, where he held several positions including head of R&D and product development. His interests include software/systems development processes, software architecture, models in software, software product lines.

Gunther Spork

Dipl.-Ing. Gunther Spork is working for MAGNA Powertrain. Currently he leads internal and external Automotive SPICE assessments as Competent Assessor and has a scholarship for Production Science and Management at Graz University of Technology. He was responsible for the implementation of ISO 15504 and engineering processes. Before he was responsible for the Quality Management System according ISO TS 16949 and worked as Six Sigma Master Black Belt for MAGNA Powertrain and VA TECH Hydro.

Richard Messnarz

Dr. Richard Messnarz is the Executive Director of ISCN LTD. He studied at Graz, University of Technology and he worked as a researcher and lecturer at this university from 1991 - 1996. In 2 European mobility projects (1993 and 1994) he was involved in the foundation of ISCN and he became the director of ISCN in 1997. He is/has been the technical manager of many European projects and he is ISO 15504 and Automotive SPICE Principal Assessor.

Frank Koenig

Dipl.-Ing.(FH) Dipl.-Inf.-Wiss. Frank Koenig is currently manager for software processes at ZF Friedrichshafen AG.

Improving Safety and Availability of complex Systems by using an integrated Design Approach in Development

Volker Bachmann[1]
Richard Messnarz[2]

[1] *SIBAC, Biberach, Germany*
[2] *ISCN, Graz, Austria*

info@sibac.de
rmess@iscn.com

Abstract

Within the last 5 years the need for a system development and a process that describes this development became more and more obvious. The number of software- and electronic engineers rose even in companies that were traditionally working in the field of pure mechanics. The ISO standard 15504 was consequently increased from the software to the system. Nevertheless even 5 years after this change there is nearly no subsystem mechanics completely described in the same tool as the electronics and software subsystems. This break within the tool chain is the top of an iceberg reaching all the way down to the lived processes in development.

This paper tries to give a solution that was worked out in a group of integrated designers that developed a program for the European Certification & Qualification Association. It shows an example that was tested in an industry project in order to reach a level two on system level in a customer assessment. It shows how to bring together standards that are used in electronics, software and mechanics in order to find an integrated design approach to improve the safety and availability of systems existing out of these parts. Namely this is the IEC 61508 asking for a risk analysis which again finds entrance into the FMEA that is a commonly known tool in the mechanical development. The FMEA again has an interface to the ISO 15504 which is described here as well.

By using the tools mentioned and bringing them together properly an integrated design is the outcome of this process.

Keywords

Integrated Design, FMEA, Safety Standards

1 Introduction

Within the last 15 years there was a great shift of the number of employees in companies working in the production to the ones working in the development. The reason is on the one hand the automation in production reducing the number of employees in this field but also the complexity of the mechatronical systems increasing the number of employees in the development. Whereas it was not so much the number of the mechanical engineers that increased but meanwhile there are often more engineers working in the field of electronics and software than there are mechanical engineers. This leads on the one hand to more complex structures and processes within the development departments asking for a standard like ISO 15504 but on the other hand these complex systems that need to be developed ask for engineers who are capable of using an integrated design approach. This integrated design approach must be supported by tools which best are already known and in use today.

Within the last years there have been a number of standards coming up which 10 years ago stood alone for themselves. Best known example is the previously mentioned ISO 15504 which was originally formulated for software development. After a few years it became apparent that it should spread out on the whole system. But until today there is almost no subsystem description on a mechanical subsystem existent. The two worlds mechanics and electronic/software still live an absolutely parallel live in every large company of the automotive industry as far as documentation is concerned. Since the documentation can be used as a sign for the way to develop, it can also be stated that these two worlds live next to one another in development as well.

Today very many developers believe that the most improvement potential lies within mechatronics. But there is a strong misunderstanding of what mechatronic comprises. Usually the mechanics is not counted as part of the mechatronics but only actuators, sensors and software.

This paper is a try to convince that mechanics is an essential part of mechatronical development and the improvement potential of the future lies in an integrated design approach comprising everything, the software, the hydraulics, the electronics and the mechanics. This is meant as a first try to bring known tools and known processes out of these different worlds together to support this approach.

2 Safety and Availability in Hydraulic and Mechatronic Systems

2.1 The Risk Analysis as a Basis for Safety Evaluation

In the following a brief example is given to lead to an evaluation of a major fault in an ABS braking system. The general knowledge of a brake system is assumed to be known.

In the following this anti-lock braking system (ABS) will be analyzed regarding its possible risk potential. The example describes the ABS without the existence of any safety measures available. Necessary safety measures will be an outcome of this analysis.

At the beginning of the development a system definition is necessary. The following example shows a realistic case in which there is not yet a completely developed product available. At this point at the very start of a development project the risk analysis should be carried out.

Figure [1] shows a typical ABS-Brake-System consisting of 4 wheel-speed sensors a central control unit and actuators on every wheel.

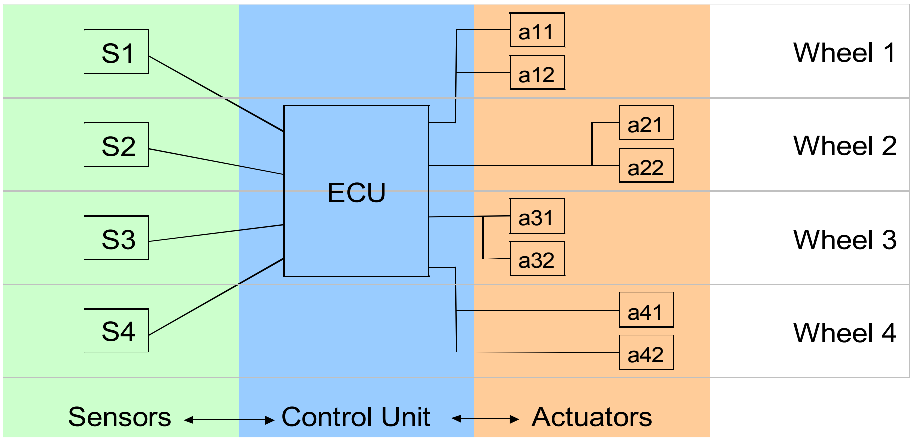
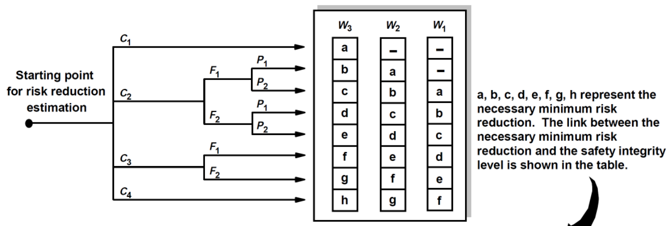


Figure 1. Schematic of an ABS brake System

In this example there is only one central ECU planned for the system that evaluates the signals coming from the 4 wheel-speed-sensors and controlling the valves at every wheel. The original task is to detect wheel slip during a braking situation and to avoid it before it actually occurs.

The challenge is to design the system in such a way as to prevent dangerous failures or to control them when they arise. During hazard identification the regarded system is considered without the existence of any E/E/PE-safety measures. Necessary safety measures will be an outcome of this analysis. A hazard is defined in the standard as a “potential source of harm”. The risks associated with unidentified hazards will remain unreduced. The identified hazards are input for the hazard analysis and risk assessment.

Risk parameter		Classification	Comments
Consequence (*C*)	C_1	Minor injury	1 The classification system has been developed to deal with injury and death to people. Other classification schemes would need to be developed for environmental or material damage.
	C_2	Serious permanent injury to one or more persons; death to one person	2 For the interpretation of C_1, C_2, C_3 and C_4, the consequences of the accident and normal healing shall be taken into account.
	C_3	Death to several people	
	C_4	Very many people killed	
Frequency of, and exposure time in, the hazardous zone (*F*)	F_1	Rare to more often exposure in the hazardous zone	3 See comment 1 above.
	F_2	Frequent to permanent exposure in the hazardous zone	
Possibility of avoiding the hazardous event (*P*)	P_1	Possible under certain conditions	4 This parameter takes into account – operation of a process (supervised (i.e. operated by skilled or unskilled persons) or unsupervised); – rate of development of the hazardous event (for example suddenly, quickly or slowly); – ease of recognition of danger (for example seen immediately, detected by technical measures or detected without technical measures); – avoidance of hazardous event (for example escape routes possible, not possible or possible under certain conditions); – actual safety experience (such experience may exist with an identical EUC or a similar EUC or may not exist).
	P_2	Almost impossible	
Probability of the unwanted occurrence (*W*)	W_1	A very slight probability that the unwanted occurrences will come to pass and only a few unwanted occurrences are likely	5 The purpose of the *W* factor is to estimate the frequency of the unwanted occurrence taking place without the addition of any safety-related systems (E/E/PE or other technology) but including any external risk reduction facilities.
	W_2	A slight probability that the unwanted occurrences will come to pass and few unwanted occurrences are likely	6 If little or no experience exists of the EUC, or the EUC control system, or of a similar EUC and EUC control system, the estimation of the *W* factor may be made by calculation. In such an event a worst case prediction shall be made.
	W_3	A relatively high probability that the unwanted occurrences will come to pass and frequent unwanted occurrences are likely	



C = Consequence risk parameter

F = Frequency and exposure time risk parameter

P = Possibility of avoiding hazard risk parameter

W = Probability of the unwanted occurrence

a, **b**, **c** ... **h** = Estimates of the required risk reduction for the SRSs

Necessary minimum risk reduction	Safety integrity level
–	No safety requirements
a	No special safety requirements
b, c	1
d	2
e, f	3
g	4
h	An E/E/PE SRS is not sufficient

IEC 1 667/98

Figure 2. risk graph method according to IEC61508-5:1998, Annex D.

The only hazard used for this example, which will be looked at in detail, describes the situation that the control unit actuates the valves in a way that building up pressure in the hydraulic brake system becomes impossible. That means although the driver is pressing the brake pedal he can not apply brake force at the wheel.

Hazard analysis is the study of the chains of cause and effect between the various identified hazards and the hazardous events to which they might lead, and of the consequences of the hazardous events. The purpose of this analysis is to derive sufficient information for the assessment of the risks involved. There are two elements of risk, the likelihood of something happening and the potential consequence if it does. Understanding the various causes of a hazardous event allows a calculation or estimation of its likelyhood.

A hazard is categorized in regard to the worst case it can cause. Applied to the shown example this is: ABS avoids build up of hydraulic pressure which means no brake-force at the wheels. Driving fast and having any kind of obstacle in front of the vehicle illustrates the consequences. If a possible obstacle consists of people, the consequences are even worse.

The method used in this example is the risk graph method. In this method a number of parameters are introduced which together describe the nature of the hazardous situation when safety-related systems fail or are not available. One parameter is chosen from each of four sets, and the selected parameters are then combined to decide the safety integrity level (SIL) allocated to the safety-related systems.

These parameters

– allow a meaningful graduation of the risks to be made, and

– contain the key risk assessment factors.

For the chosen example the following is the case:

Consequence: C3

Comment: Without the possibility to slow down a crash is possible. Depending on the speed, passengers in the car as well as traffic participants could be badly injured or killed. Death of several people is possible.

Frequency: F2

Comment: Braking at medium or higher speed is an every-day situation.

Possiblity: P2

Comment: A car without operating braking system is for most drivers not controllable. The average driver and other traffic participants are normally not able to avoid harm in this situation.

Probability: W1

Comment: Today anti-lock braking systems are state of the art. Many years of experience and testing in the field state that this occurrence is quite improbable. Established development processes are available.

The risk parameters C3 / F2 / (P2) / W1 result in a safety integrity level 3 (figure 2)

In Part 4 of the standard, safety integrity is defined as “the likelihood of a safety related system satisfactorily performing the required safety functions under all the stated conditions, within a stated period of time”, and a safety integrity level (SIL) as “a discrete level (one of 4) for specifying the safety integrity requirements of safety functions”.

Every hazard can be verbalized as a safety requirement or a so called safety goal for the safety-related system. Here the safety goal could be: „ABS must ensure a safe build-up of hydraulic pressure.“

Any safety-related system covers all parts of the system that are necessary to carry out the safety function (i.e. from sensor, through control logic and communication systems, to final actuator, including any critical actions of a human operator).

“The major difference between a thing that might go wrong and a thing that cannot possibly go wrong is that when a thing that cannot possibly go wrong goes wrong, it usually turns out to be impossible to get at or repair.”

Douglas Adams, author of The Hitchhiker’s Guide to the Galaxy

2.2 The Advantage of an Integrated Design derived out of Safety and Availability Evaluations

The previous chapter showed that the risk analysis should stand at the very beginning of a system development. Its output is a number of hazardous events that are categorized in Safety Integrity Levels. The hazardous events are used as an input in the System FMEA on the top level (failure effects in figure 3 of the system FMEA). The SIL classification determines the severity in the FMEA. The system should have on the top level functions. To each function there must be at least one failure. Each event out of the risk analysis should be equal to one of these failures. The failures are linked together in a failure tree as shown in figure 3. Most important in this failure tree aspect is that there are as many blocks below the wide orange line as there are subsystems. Instead of System FMEA ECU there should be also a System FMEA calliper, sensors, actuators, wiring harness and so on.

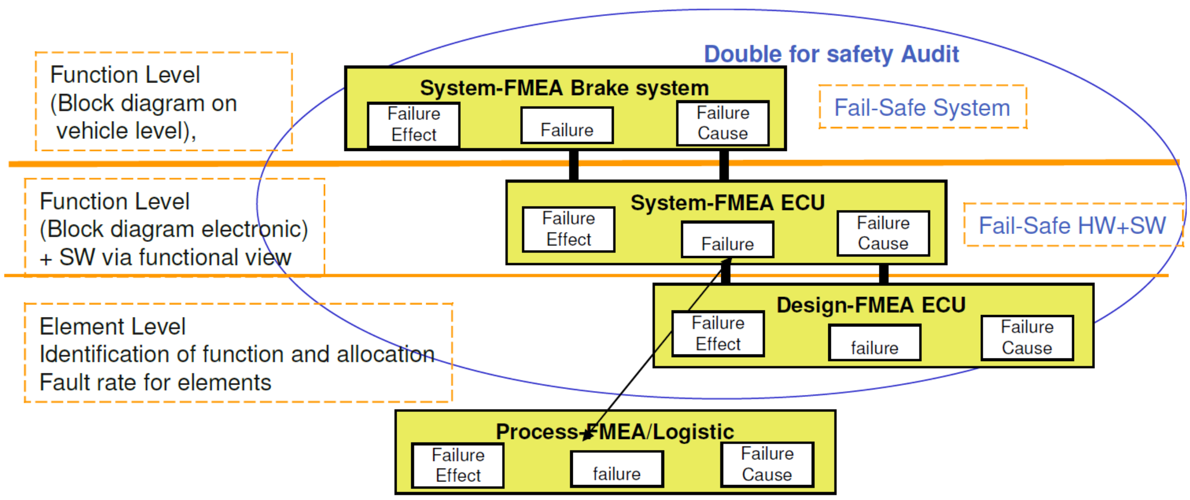


Figure 3. FMEA

Out of this failure tree every potential subsystem failure can be traced to the events found in the risk analysis. This way the contribution of every subsystem to avoid this event becomes visible and helps to get an overview for an integrated system design.

3 *The Role of Logic in Mechanics, Hydraulics, Electronics and Software*

Logical functions can be built in any of the four systems mechanics, hydraulics, electronics and software. Since the opportunities of electronics and software boosted the development in the past years a coordination between the above mentioned systems become more and more important.

3.1 Description of the Current Situation

3.1.1 Fail Safe Systems

Even in present there are still many systems in production that do have an electronic control unit but still rely on mechanical and hydraulical systems as far as safety is concerned. This is in the automotive industry especially the case for example for automatic transmissions and steering systems. Where as the pure steer-by-wire is still prohibited by law in all major industrialized countries because of the high SIL, shifting by wire is allowed and already by some transmission builders in production. What quite frequently occurs is the fact that the development departments do not trust the electronics- / software department enough to lay full responsibility for failure reactions in their hands. Instead there is often a logic within the hydraulics that actuates for example the park pawl in case of a single failure (a chain of two or more failures is not analyzed in SIL 2 Systems). In steering systems only adding force to the steering rack is allowed. The original mechanical steering column still exists and stands for the fall back basis.

3.1.2 Sensors and Diagnostics

Software diagnostics need input signals. These signals can only be provided by sensors. Sensors again cost money. In today’s structures of large automotive companies and their suppliers it is merely impossible to think about the vehicle as a system and to find solutions to requirements outside of a subsystem such as the engine, the steering system or the transmission. Even a very simple problem of a transmission builder stating that a better torque signal coming from the engine via can could help a great deal to improve shift quality can in today’s structures not be solved in a way to analyze which way is best in this case. The idea of a supplier to go to his OEM and analyze together whether it makes the transmission significantly cheaper if he gets a better torque signal can conjure up a smile on every developers face.

Nevertheless in today's systems in which it is possible to get information from any sensor in the system this approach needs to be pushed far more than it is today. Since only 2 or 3 years the software diagnostics is consequently represented in the FMEA. This lead to major irritations in the beginning also because the FMEA tools did not provide this functionality. Meanwhile the FMEA tools are able to handle fault reactions but the schematic how to represent software diagnostics in these tools is still at the very beginning as far as the knowhow of the FMEA moderators is concerned. Out of today's point of view the combination of FMEA and SPICE is the best way to handle this problem properly.

3.1.3 The Integrated Design Approach in ISO 15504

In order to be able to directly compare the results of the FMEA with the requirements formulated for the engineering processes in SPICE the FMEA needs to be worked out thinking in functions and failures. This needs to be done for every subsystem involved. This was until recently not the case. Until short a designer (for mechanics) thought much more in solutions and structures. He for example always knew what part he was working at and what attributes he had to take care of. But it was not always clear what the requirements for the attributes from a functional point of view were. Of course he had to take over many attributes like material, surface roughness, etc. from previous designs in order to keep the efficiency high. Taking over the solutions often helps to answer the question what the solution looks like but not what problem had to be solved.

Similar to that is the behavior in software development. In the software construction it often makes sense to combine modules out of libraries to keep the efficiency high. Most important is to fulfill the requirements of the software architectural design. That does not help to answer the question what system requirement is fulfilled with this module.

In the future the developers of each subsystem have to think in functions and requirements in order to be able to draw links from one level to the next.

3.1.4 System Development Teams allocate Requirements

Figure 4 shows the link between the requirements of the engineering processes and the failures of the FMEA. Also the measures of the FMEA that are linked to the failures are shown in comparison to the test-cases of the engineering processes. The wording is still exclusively for the software, but as it was rolled out to the whole system the v-model is similar in all the other subsystems.

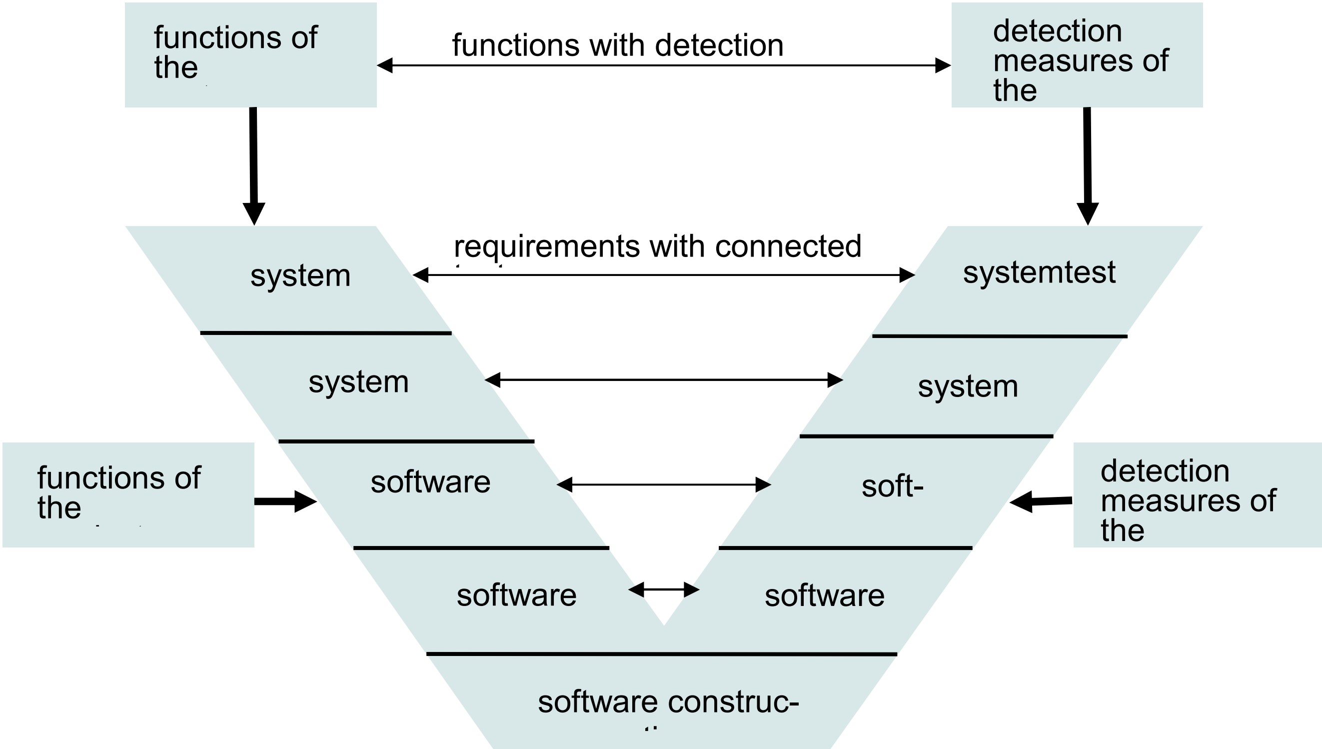


Figure 4. Engineering processes of the V-Model in comparison to functions and detection measures in the FMEA

The demands of the standard do not only help the customers of these systems (e.g. the OEMs) but also the developers of these complex systems, as it enables them to run analysis concerning the project progress in many fields. Such analysis can e.g. be:

How many of my System-requirements are linked to the subsystem-requirements (which answers indirectly the question on how many design-decisions do already exist), or how many of the subsystem-requirements are already positively tested.

Taking this description as a basis it becomes clear that the complete system description of the requirements of a mechatronical product need to be described. The point of view is different than in the FMEA. It is the try to get an overview over complex systems by subdividing them into ever smaller parts. Since the requirements have to be tested down to the lowest subsystem-level the connection to the FMEA is obvious for mechanical systems. As described in the previous chapter in mechatronical systems it is not the mechanics alone that is responsible for safety functions anymore. But especially the diagnostics- software provides the possibility to avoid hazardous situations in case of a mechanical failure or go into a so called limp home mode to raise the availability of a system.

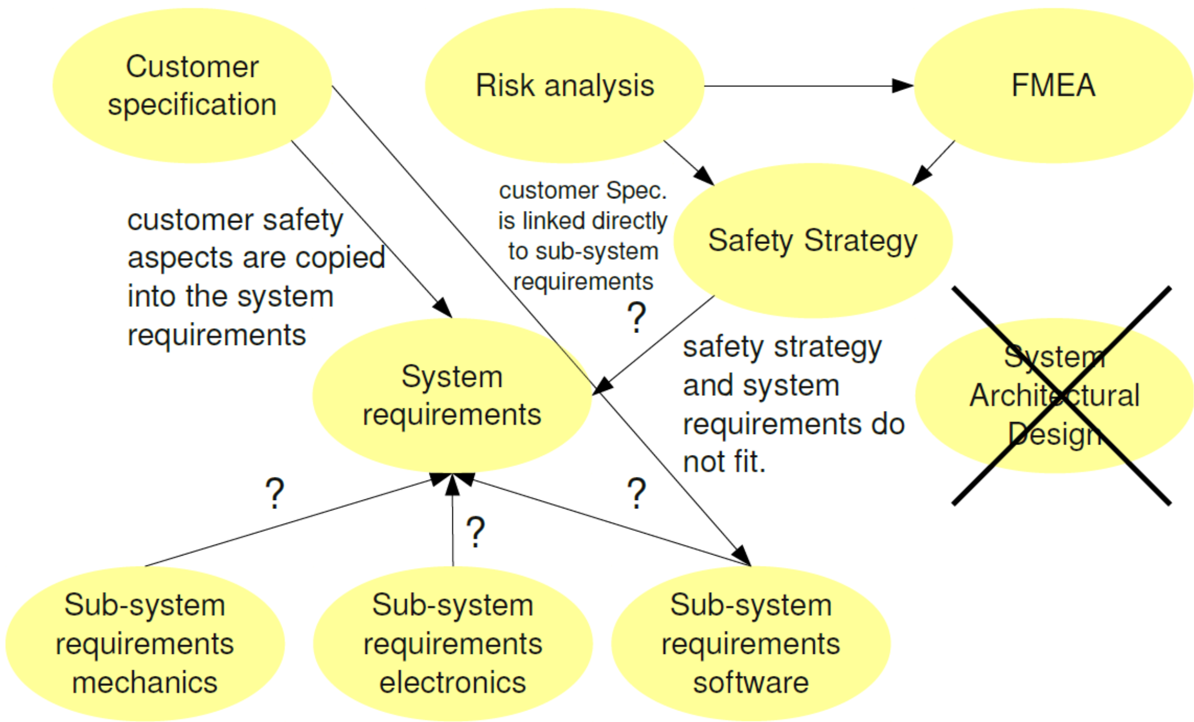


Figure 5. Normal way to try to fulfill the customer requirements and the problems arising out of this way.

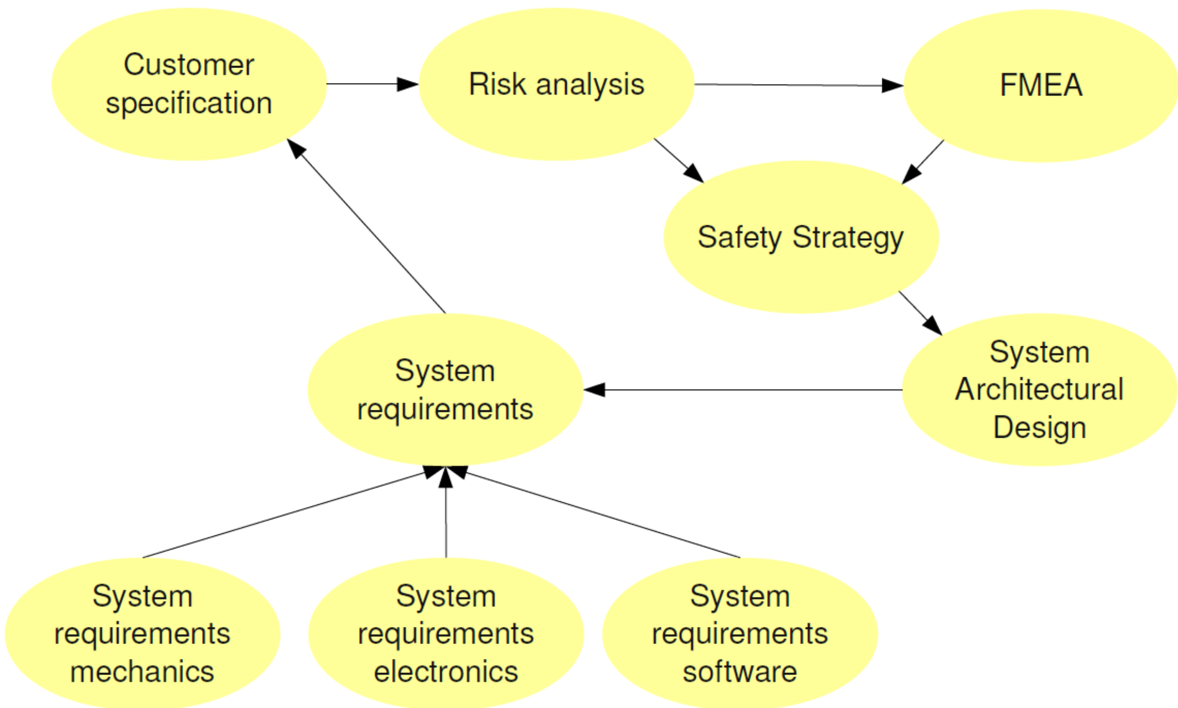


Figure 6. Integrated design approach to fulfill the customer requirements

Going from this example to an overall understanding the following figure 5 explains what most frequently happens.

Usually the customer requirements concerning safety are written in a detail that reaches down into every subsystem. Usually the fastest way to get a good coverage on links to the customer specification is to link directly to the sub-system. This most certainly gives excellent metrics over development time in the beginning of the project. The problems arise later in the project. In order to fulfill the IEC 61508 a risk analysis needs to be carried out as described in the previous chapters. The outcomes are usually taken as an input for the FMEA. The FMEA makes all the dependencies between the subsystems clear and thus leads to a safety strategy for the product. If the safety relevant aspects coming from the customer are directly linked into the system- and sub-system requirements it is not possible to adapt to the safety strategy anymore. Problems in this field usually inhibit to work out a system architectural design because it does not fit to the system requirements at all. In the end there are major problems to link from the subs-systems to the system since some of the links are drawn directly from the customer spec. All these problems lead to a zero in an assessment for the processes eng. 2 and 3 and usually also eng. 9 and 10.

The above figure already somewhat contains the solution that shall be shown in the figure 6.

Using this approach the risk analysis is carried out for the product just as in the previous example with the only difference that the customer requirements may have an influence on the cases which are considered in the risk analysis. That means that every new customer that asks for the the product may enlarge the risk analysis by points or aspects that were not considered until then.

The outcome is taken as an input into the FMEA which again is used to create a fitting system architectural design. This design is the basis on which every decision how to fulfill system requirements in sub-system requirements is based. It is a need to take these design decisions to be able to draw the links from the sub-system to the system requirements. The requirements tree that ends in the system requirements is now used to prove that all the customer specification requirements are met.

In order to be able to do that a work in interdisciplinary teams is essential. In this team there is at least one specialist for every sub system needed just as much as the system designer. It is not possible to come to good design decisions without these interdisciplinary teams. Engineering teams have to get interdisciplinary, and thus demand for a mutual understanding and collaboration between domain expert team members is necessary.

4 Safety and the Influence on Testing on System Level

Considering the functional safety the conflict of goals within the FMEA is solved by changing in case of a failure the severity of the consequence as long as there is a diagnostics software existent that can lead to a different consequence. Using the example of the transmission in which the failure cause for the wrong starting direction can be found in the assembly group level and the consequence is on the level of the vehicle. The severity for this consequence is 10. The consequence can now be changed within the FMEA to a different case taking the exact same failure cause to an eight since the starting in wrong direction was changed by the diagnostic software to a stranded vehicle. This is not safety relevant anymore. Of course there has to be formulated a detection measure that proves the diagnostic software and the according actuators and mechanics to work properly on system level. This can best be proofed in the requirements tree as it is shown in figures 6 and 4.

Working with the SPICE processes in the example of the transmission there should be a requirement on system level that is asking for a detection for the starting off in wrong direction. This would lead to requirements on subsystems such as in the software asking for diagnostics software and into the electronics asking for a sensor for the shift lever position and a sensor measuring the direction of rotation of the output shaft. Each subsystem requirement as well as the system requirement must be linked to a test. The system architectural design is influenced by a great deal and since this is also very much a question of money these decisions need to be based on consequences for the vehicle.

In order to avoid double work in the future there must be found solutions to integrate the tools used today. In a first step there must be a description of the interface for instance of the IQ-FMEA and DOORS or MKS. Today these problems must still be solved by using an indirect way via an html export with an import into the next tool.

Working according to this approach does not only help to get proper design decisions but also leads to a level two in an assessment for eng. 2 and 3, an in case all the tests are existent and linked also eng. 9 and 10. In case this is done the same way for different customers and different products as described in figure 6 even a level 3 is possible.

5 Conclusions

Today most of the engineers agree that the largest potential for improving systems in the near future is to be found in the mechatronical development. A large misunderstanding is still on the way on what belongs to a mechatronical system. For most mechanical engineers this is the combination of actuators, sensors, electronics and software. This paper showed that the mechanics is an essential part of a mechatronical system as well. Since there a clear tool chain break in documentation systems between mechanics and electronics/software the processes do not work hand in hand. Thus this paper showed how an integrated design approach can lead to an improvement of the product system stability and thus to a higher level of safety and availability.

By describing the interfaces of tools that are commonly known and used in the different faculty this paper is meant as a part of a strategy to come to a tool based development. The tools with their interfaces described are the standards ISO 15504, IEC 61508 and the FMEA.

Working according to the approach of an integrated design demands engineering teams that work interdisciplinary. There is a need for a mutual understanding and collaboration between domain expert team members.

References

[1] http://www.pdf-search-engine.com/iec-61508-pdf.html

[2] http://en.wikipedia.org/wiki/Anti-lock_braking_system

[3] IEC 61508:1998

[4] Poth, Alexander: “SPI of the Requirements-Engineering-Process for Embedded Systems Using SPICE”, in: Proceedings of the EuroSPI 2006 Conference

[5] Spork, Gunther: “Establishment of a Performance Driven Improvement Program”, in: Proceedings of the EuroSPI 2007 Conference

[6] Volker, Bachmann, Richard, Messnarz: “Improving the Software-Development for multiple Projects by applying a Platform Strategy for Mechatronic Systems”, in: Proceedings of the EuroSPI 2009 Conference

[7] Riel A., Tichkiewitch S., Messnarz R., Integrated Engineering Skills: Improving your System Competence Level. In: EuroSPI 2008 Industrial Proceedings, Dublin, Delta Series about Process Improvement, ISBN 978-87-7398-150-4pp. 8.9-8.20.

Authors

Dr- Ing. Volker Ovi Bachmann

Dr. Volker Ovi Bachmann (info@sibac.de) is the managing director of SIBAC GmbH. He studied at Darmstadt University from 1988 till 1993 and received a degree in mechanical engineering. From 1993 till 1998 he has worked as a so-called assistant at the automotive department of Darmstadt University as researcher and lecturer and received a degree as Dr.- Ing.
Since 1998 he has been working in the automotive industry for ZF Friedrichshafen AG in several positions the latest being team manager in the quality management. He was in charge for 7 employees and the quality management of 5 large transmission development projects (with more than 60 engineers per project involved).
During his time in industry he received the qualification as a Spice Assessor. He has an experience of 120 assessment hours. In 2008 he become a competent assessor.
In Jan. 2008 he founded SIBAC GmbH. Since then he is working as a process consultant at various companies.

Dr. Richard Messnarz

Dr. Richard Messnarz (rmess@iscn.com) is the Executive Director of ISCN LTD. He studied at the University of Technology Graz and he worked as a researcher and lecturer at this University from 1991 - 1996. In 2 European mobility projects (1993 and 1994) he was involved in the foundation of ISCN, and he became the director of ISCN in 1997. He is/has been the technical director of many European projects.
He is the editor of a book "Better Software Practice for Business Benefit", which has been published by IEEE (www.ieee.org) in 1999 (the leading research publisher in the USA). He is the chairman of the EuroSPI initiative and chair of the programme committee of the EuroSPI conference series. He is author of many publications in Wiley SPIP, Springer LNCS/CCIS, and journals.
He is a lead ISO 15504 assessor. He has worked as a consultant for many automotive firms, such as BOSCH, ZF, Continental, Magna, T-Systems, etc. He is a founding member of the INTACS (International Assessor Certification Scheme) accreditation board, a founding member and vice president of the ECQA (European Certification and Qualification Association), and he is the technical moderator of task forces to which above 20 German companies collaborate, the SOQRATES initiative (www.soqrates.de).

Innovation Management System for the Automotive Supplier Industry to Drive Idea Generation and Product Innovation

Martin Neumann[1]
Andreas Riel[2], *Daniel Brissaud*[2]

[1] *Kolbenschmidt Pierburg AG, Neuss, Germany*
[2] *G-SCOP Laboratory, Grenoble University of Technology, Grenoble, France*

Martin.Neumann@de.kspg.com
Andreas.Riel@grenoble-inp.fr, *Daniel.Brissaud@grenoble-inp.fr*

Abstract

Innovative product development is highly dependent on new product ideas and product information. This especially applies to companies, which obtain their competitive advantages by technological lead like the automotive supplier industry. These companies are inevitably strongly dependent to ascertain which technologies decide on the market success in the future. Therefore the innovation management should not only support the product development with the collection of ideas, but it also has to analyze technologies concerning their further implementation, new arising (substituting) technologies have to be identified in time, and these technologies have to be evaluated concerning their commercialization and their risks. Basic research, which could lead later to new technologies, must be observed, in order to be able to identify technological applications (markets and/or products) in time. This paper gives insight into the industry-sector-specific characteristics of innovation management within the automotive supplier industry, and outlines a basic scheme how to develop an innovation management system with the goal to improve the existing innovation management at the automotive supplier Kolbenschmidt Pierburg AG (KSPG).

Keywords

Innovation Management, Automotive Supplier Industry, Idea Generation, Knowledge Management, Knowledge Mining, Stakeholder Model

1 Introduction

Innovations are the global motor for economic growth and represent at the same time the key factor for increasing competitiveness. Within the R&D- and innovation-driven environment of automotive suppliers the capacity of innovation and the performance of activities have become a major challenge for the success of companies. In 2015, the global automotive suppliers and engineering firms will invest approximately € 65 billions in R&D. This sum is far more than twice as much as the OEMs' budgets [1]. Therefore also the most engineering jobs will be generated by automotive suppliers in future – globally a total of about 250,000 jobs until 2015 [1].

It is all the more important that innovation management in the field of automotive suppliers must cope with the increasingly complex market conditions. Due to its comprehensive and profound interactions with other corporate divisions and the business environment the innovation management has to be open for other scientific disciplines, like for instance knowledge mining [2], to find ways to ensure the generation of new ideas.

This paper highlights the particular role of innovation management to drive idea generation and product development. Section 2 focuses on the general conditions and sector specifics of innovation management in the automotive supplier industry – both theoretically and practically. Section 3 discusses the current situation of innovation management at Kolbenschmidt Pierburg AG (KSPG) and introduces two kinds of approaches to improve the existing system: first of all the enlargement of idea sources by taking into account input from external market stakeholders rather than only internal idea contributors; secondly, the application of knowledge management, in particular knowledge mining techniques. This paper is closed in section 4 with a conclusion and an outlook of further research.

2 Innovation Management in the Automotive Supplier Industry

2.1 General Conditions and Sector Specifics

The automotive industry is one of the most highly innovation-driven industries [3]. This fact applies especially for the German automotive industry. With a total of € 18.9 billions, the research and development expenditure undertaken by the German automotive industry accounted in 2008 for almost a third of the total research and development expenditure of the German economy [4]. In order to enhance its innovative power, the German automotive industry has stepped up its research efforts continuously in the course of the past few years. For example the German automotive industry filed 10 patents per diem in 2007. Given this research intensity, the German automotive industry can claim to be at the forefront of patent statistics [5].

Particularly the automotive supplier industry prepares more and more the way for new automotive technologies worldwide. Because of the automotive suppliers' high involvement and responsibility during the development activities of the OEMs, most of the vehicle parts are engineered and manufactured by the suppliers. The study entitled "Car Innovation 2015" that has been published by the management consultancy OLIVER WYMAN (formerly Mercer Management Consulting) identifies that in 2015 the global automotive suppliers and engineering firms will invest approximately € 65 billions in research and development – far more than twice as much as the OEMs [1]. This outsourcing of innovation activity to automotive suppliers has the consequence that these suppliers apply increasingly independent patents, in order to keep their own innovations exclusive. Thus the large automotive suppliers focus their research on the same areas like the OEMs, particularly to gain new knowledge and the strategically strong patens. Only very innovative suppliers succeed in the development and maintenance of their patent portfolio to strengthen their negotiation position versus the OEMs [6].

To understand innovation management in the automotive industry and based on its findings from the above mentioned study the management consultancy OLIVER WYMAN defined a system called "Innovation Strategy Framework"(ISF) which takes the following success factors of innovation management into account: a clear innovation strategy that is closely connected to the company's overall business model, the right team that has the culture to put the strategy to work, an organization that can effectively and efficiently steer the necessary innovation process and an intelligent business case that enables innovations to be turned into tangible profit [7]. The ISF consists of four elements:

1. Innovation proposition: description of the major benefit and target segment of the innovation and also the primary innovation guideline of the company.

2. Organization and culture: explanations of the innovation process, R&D capacities and facilities structure.

3. Competence focus and collaboration: composition and evaluation of the internal and external competencies and collaborations.

4. Business case: definition of the underlying revenue model for the innovation and the protection of the innovation against exploitation from competitors.

By using the ISF six innovator archetypes have been identified for OEMs and six for suppliers. Each describes a typical ISF profile in which the different elements fit together to form a coherent system. Many companies follow two or more innovation strategies at the same time – suppliers with different product ranges and OEMs with different brands. In addition, innovation archetypes are not static role models, but evolve with time [7]. Table 1 shows conclusively the archetypes of innovation management for automotive supplier.

Table 1: Archetypes of innovation management for automotive supplier [7]

Innovation archetype	Innovation proposition	Focus and collaboration	Business case
Radical innovator	Replaces old systems or establishes new ones	Specialized focus Keeps know-how in-house	Price premium Strong IP protection
Functional enricher	Brings new functions to the market OEM and end customer focus	Functional integration focus Keeps know-how in-house	Price premium Strong IP protection
System connector	Functional process or product optimization End customer focus	Expansion into new systems via coop networks Open interfaces	Price premium or low-cost Fairly weak IP protection
Process champion	Incremental process innovation to serve broader markets Adapts to customers	Process focus Open to coops	Low costs in mature techs Weak IP protection
Niche performer	Product or process innovator serving niche markets End customer focus	Very specialized know-how Selective coops	Price premium Varying IP protection
Module shaper	Focus on module design and processes Defines modules anew	Unique know-how combination Coop with OEM/system connector	Value capture from OEM Cost reduction for modules

The classical approach, to buy parts and components from a variety of suppliers will be increasingly replaced by purchasing more complex, mostly pre-assembled systems (e.g. front-end systems) from only a few suppliers (so called single sourcing). Also against the background of globalization and because of the ever-growing requirements of the OEMs – like e.g. shorter development times and life cycles of the models or the increasing relevance of electronics in vehicles – a further reinforced consolidation process within the supplier industry will take place [8].

The new forms of e-commerce in the B2B sector and the extended EDI standards (Electronic Data Interchange) – made possible by the internet technology – expedite the re-orientation, which proceeds since the mid-90s, of the value-added chain. The use of e-commerce is pushed by the automotive manufacturers in order to obtain savings in time during the product development and implementation of the contract as well as gain further cost reductions. In recent times, so called B2B platforms were established. These B2B platforms offer several companies virtual market places where the relations to suppliers can be organized on-line [8].

By consistent realization both automotive manufacturers and companies of the supplier industry can benefit from system procurement: manufacturers profit by high-quality and innovative products to lower costs, suppliers profit by an increased order quantities, more stable business relationship as well as higher competitiveness. However, through this the dependency of the automotive manufacturers on their suppliers can grow at the same time – and vice versa. Meanwhile, this degree of dependency achieved a historical value with manufacturing depths from only 30 to 35 percent. An efficient system extent and thus purchasing volume can not be generally defined in this situation and depends on both sides: on the procurement strategy and the manufacturing conditions of the automotive manufacturer and on the availability of competent suppliers [8].

In the long term, it has to be assumed that almost the complete vehicle comes from the plants of some few system suppliers and/or mega-suppliers, and the brand dominating OEMs assume only the overall project responsibility and coordination.

Because this complex range of activities cannot be handled by only one supplier, a network of interconnected supplier companies will act under the leadership of one global system integrator. Due to the fact that many suppliers sell their parts, components and systems to other different manufacturers, these relationships result in an interdependent manufacturer-supplier-network [8].

Innovations of the suppliers rank today among the most important characteristics which differentiate an automotive manufacturer from its competition. And these innovations must be generated to a considerable part from the suppliers. Through the generation of innovative ideas, the pre-development of new material, process and product technologies, the technology leadership shifts more and more on the suppliers which, however, requires also extensive personnel and financial resources [9].

2.2 Considerations for Practical Implementation

Product innovations are mainly successful if they are systematically prepared, realized and implemented and they do not happen as a result of pure chance [10, 11]. For that purpose it is necessary to create appropriate basic conditions for the innovation activities and to plan, manage and control individual innovation projects in coordination with other innovation activities [12]. These tasks are summarized under the term innovation management.

Numerous publications discuss from abstract-theoretical perspective the use of supporting instruments

- for the establishment of an adequate business environment,
- for the accomplishment of planning and managing complex and interdependent sub processes,
- to increase efficiency and
- to control and decrease risks

in connection with innovation processes [12]. From an entrepreneurial point of view the methods and instruments offered in the literature can only be used in limited ways due to the missing consideration of (company-)specific characteristics [13]. In this context COOPER and KLEINSCHMIDT [14] stated: "…what the literature prescribes and what most firms do are miles apart when it comes to the new product process."

In view of the all-encompassing definition of innovation management, this discrepancy between theory and practice is understandable. Also the study of OLIVER WYMAN illustrated in section 2.1 verifies that especially in the automotive supplier industry most different and from each other deviating innovation management strategies exist. These different systems are legitimate because of the novelty and the variety of innovations. Thus the innovation management is forced to be defined and adjusted consistently anew. The inevitable question arises to which extent the innovation management can be realized in practice within the sector of automotive supplier industry. However, the organization of innovation management is exceedingly difficult if the product development systems are already well-established within a company. Therefore the innovation management has to concentrate on its central function which has its origin in the process character of innovation and is contained in most definitions of innovation management and finally makes innovation management so unique in relation to other management tasks: the generation of ideas.

The main focus of idea generation is on the systematic development and collection of ideas and problem solutions for new or improved products [15]. In the course of the idea generation the extraction of knowledge is essential. Therefore the innovation management has to deal with the procurement, storage and utilization of new technological knowledge – similar to the discipline of technology management [15]. Thus it is used for the protection of a company's technological competitiveness. Innovation management also has to establish the basic conditions for that the appropriate technologies are provided for concrete development tasks.

3 Innovation Management at Kolbenschmidt Pierburg AG

The following example describes the current situation of innovation management at the automotive supplier named Kolbenschmidt Pierburg AG (KSPG), more specifically of its division Pierburg. Based on this adumbrated as-is analysis a basic structure to improve the existing innovation management will be introduced.

3.1 Description of the Current Situation

3.1.1 Organization

KSPG is the parent company of Rheinmetall's automotive sector. As a global and successful first-tier supplier to the automotive industry, KSPG thanks to its vast capabilities commands foremost positions in the product and component segments air supply, emission control and pumps as well as in the development, manufacture and aftermarket supply of pistons, engine blocks and plain bearings.

Product engineering and development are conducted in close liaison with the leading car assemblers. Low emissions, reduced fuel consumption, upgraded performance, reliability, quality and safety are forces that drive innovation at KSPG.

In line with its strategic focus, the Group has six autonomous divisions:

- KS Kolbenschmidt,
- Pierburg,
- Pierburg Pump Technology,
- KS Aluminium-Technologie,
- KS Gleitlager,
- Motor Service.

With its systems and modules "for every aspect of the engine", KSPG generated sales of around EUR 2,06 billion in 2008. At its production locations in Europe, North and South America plus China, the Group employs a workforce of around 11,700.

Pierburg offers nowadays emission systems, commercial diesel systems, actuators and solenoid valves and was founded in 1909 as a steel trader in Berlin; in 1928, Pierburg began with the production of carburetors and very soon evolved as sole supplier to all German automakers and many international motor vehicle producers and engine manufacturers. In 1986, Pierburg was taken over by the Rheinmetall Group and in 1998 merged with Kolbenschmidt to form Kolbenschmidt Pierburg AG.

KSPG wants to use Pierburg's and its other divisions' experience with innovation management to develop one group-wide innovation management system based on the existing methods within the group. The following sections are dedicated to Pierburg's innovation management.

3.1.2 Existing Innovation Process at Pierburg

Innovation management at Pierburg has been established in 2006 by the advanced development department and describes its overall innovation process by the so called funnel model [16, 17]. This model starts with the monitoring of the market and/or the search field to identify problems or interesting product trends. The monitoring phase of the innovation process attempts the gathering of information to discover market and/or technology trends and to understand the competitive environment. Also there must be an organizational framework which promotes the creation of ideas and the possibility to collect them. As soon as the process of idea generation based on the findings of the monitoring has finished, the collected ideas have to be selected and evaluated according to technical and economical criteria. Successful and promising ideas have to be handed over to the advanced engineering for the elaboration of a concept. The feasibility of this concept has to be proofed before its implementation can be initiated. The final result is a new product. Figure 1 gives a brief outline of the innovation process at Pierburg.

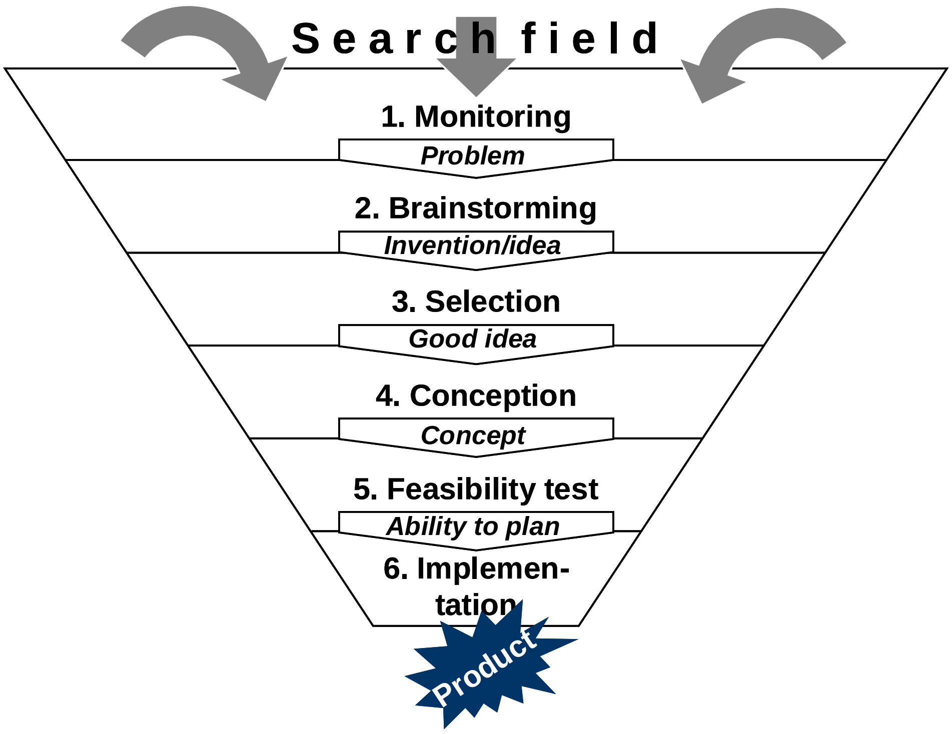


Figure 1. The innovation process at Pierburg

In the process of creating an environment for the promotion of innovations, the next step is the integration of the new innovation management into the existing system. At Pierburg the advanced development process was clearly defined. Before the establishment of innovation management, the generation of ideas was much unstructured. No continuous process was available. The innovation management guarantees an organization of this unstructured pool of ideas and a constant collection of all relevant ideas. It is now the “upstream process stage” to the advanced development process and is responsible for the collection, examination, evaluation and the selection of product ideas – mainly inventions and technical product ideas – to feed the advanced development engineering department with new promising ideas (figure 2).

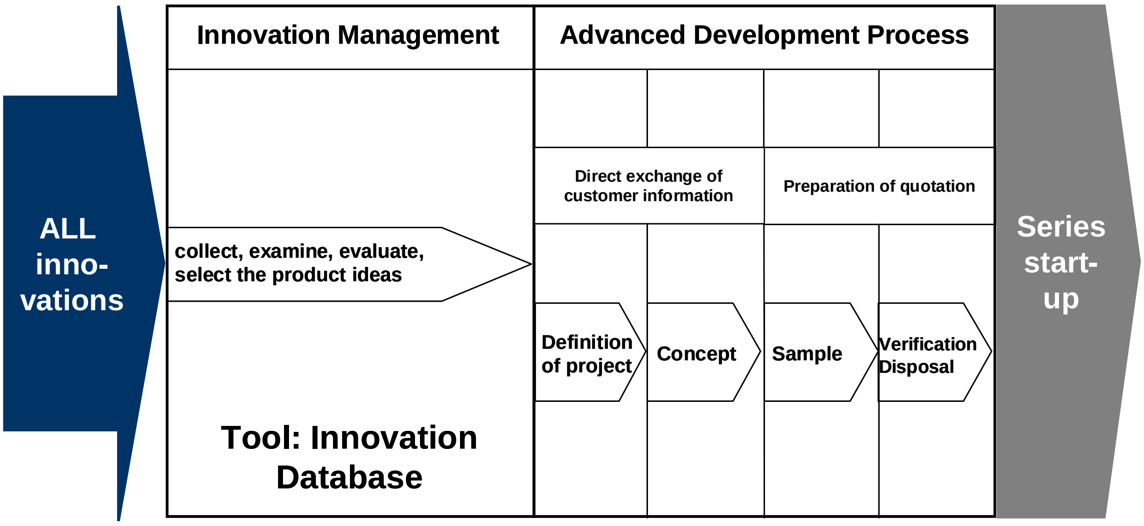


Figure 2. Integration of the Innovation Management into the existing system at Pierburg

3.1.3 Innovation Management Tool

The central tool of innovation management at Pierburg is the Innovation Database. The Innovation Database is available via intranet for all Pierburg employees and supports the innovation management at the collection, evaluation and selection of inventions and technical product ideas. The ideas and inventions collected in the Innovation Database are IT-technically secured and thus protected against the access by external, not company-associated persons. The standardized process cycle of the Innovation Database ensures a simplification and a shortening of the operational workflow.

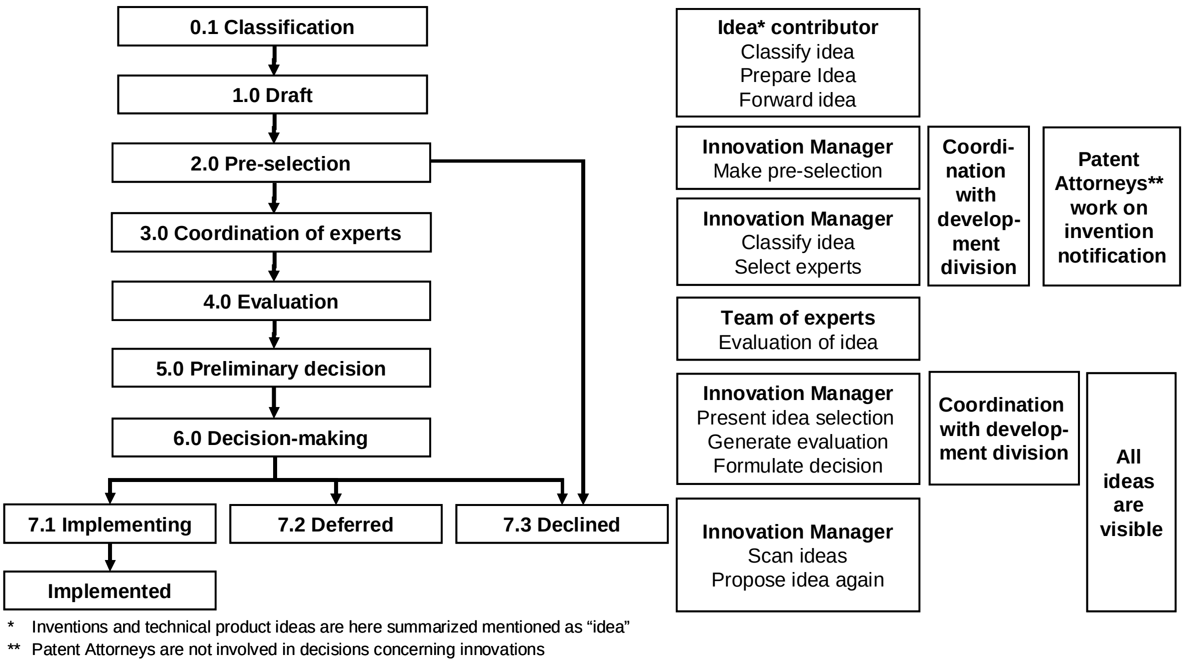


Figure 3. The process cycle of the Innovation Database at Pierburg

The process cycle of the Innovation Database (figure 3) can be described as follows: Idea contributors have now a user friendly possibility to enter their ideas by using the Innovation Database via intranet. Basically the idea contributor has to classify his idea as invention or technical product idea. The technical aspect of the innovation is important and thus represents a first stop criterion. After the idea has been submitted by the contributor, the Innovation Manager makes a pre-selection in coordination with the advanced development department and patent attorneys. In case of inventions a further pre-selection by patent attorneys filters innovations without prospect on success at an early stage. Ideas passing the pre-selection successfully will be assessed by a team of nominated experts. The evaluation criteria of the experts are: technology, patents and strategy, substitution, customer needs and product life cycle, market, sales, invest and budget, start of production, resources. A ranking of the ideas and inventions is done on basis of the experts' evaluations in cooperation with the advanced development department. The next two steps “Preliminary decision” and “Decision-making” terminate the process and describe the transfer of the idea to the different development departments.

3.1.4 Weaknesses of the Existing Innovation Management

The innovation management at Pierburg is mainly dominated by the Innovation Database and the organization of inventions and patent applications. A critical analysis of the existing innovation management system at Pierburg reveals that currently the idea generation lives from a core group of Pierburg employees as idea contributors. By using only ideas and information from well-known sources within the company there could be the threat of stagnation.

A central innovation management system at KSPG will have to improve the following aspects:

1. the generation of new ideas;
2. the enhancement of the analysis of ideas and
3. the implementation of an industrial organization which supports innovation within a standardized advanced development process.

Section 3.2 deals with the redefinition of the existing concept of idea generation at KSPG to improve innovation management and to support the advanced development.

3.2 Improvement Approach

Two kinds of approaches should be verified to implement a group-wide innovation management system and standardized advanced development process throughout all divisions at KSPG:

- The application of knowledge management, in particular knowledge mining techniques [18], as well as
- the enlargement of idea sources by taking into account input from external market stakeholders rather than only internal idea contributors.

3.2.1 Improvement by Knowledge Management

To play an important role in today's competing global market, it is essential to combine satisfaction of the customers, productivity and competitiveness. Companies have “…also to face the growth of technology with a significant increase in the volume of available and accessible information.” [19] This is an important fact for innovation management today and, as this contribution illustrates, knowledge plays a major role, more than ever before, in the idea generation process in the complex field of innovation.

An empirical study by BARACHINI and RANKL [20] lead to the assumption that knowledge management and innovation management seem to be important for the whole automotive supplier industry. They discovered a strong positive correlation between knowledge management and innovation. Therefore these two factors strongly depend on each other. They recommend that knowledge and innovation management should be regarded as key investments in the long run. Only if both techniques are mastered equally well by followers, they will catch up.

Due to the rapidly growing amount of details, the idea evaluation process becomes more and more complicated. In this context, the current usage of the World Wide Web belongs to the most important drivers of the data flood and information overload [21]. An innovation manager will notice numerous negative effects which are in general due to information overload during the analysis; in particular if they concern the work procedures of information gathering and document evaluation.

To cope with the above mentioned phenomenon of information overload the innovation management should realize – during the phase of idea generation – an optimized combination of different analysis techniques from the range of the Knowledge Mining or other disciplines like Competitive Intelligence [22]. This interdisciplinary approach which uses different approved methods of knowledge mining both from internal documents and from the World Wide Web can be the key to the improvement of innovation management in practice and to the successful implementation of innovation management in the automotive supplier industry.

3.2.2 New innovation management implementation planning

By using the model of market stakeholders [23], KSPG can enlarge its sources of product ideas. Not only one group of idea contributors should be responsible for innovations, also other stakeholders of KSPG's corporate environment should be actively involved in the idea generation process. With this identification of additional idea sources it is necessary to analyze methods to extract and format their information and to evaluate it in form of short- and long-term aspects for the company's innovational development. Figure 4 demonstrates this approach.

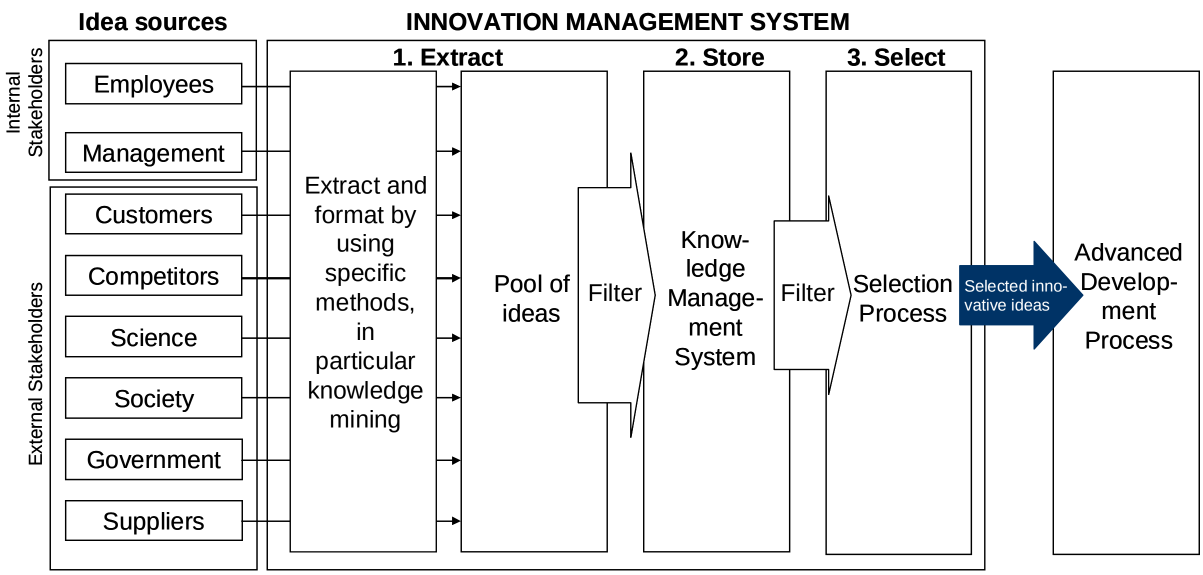


Figure 4. New innovation management system based on stakeholder concept

This new system has to start with an as-is analysis of KSPG's current innovation management with special regard to the idea sources up to now and how ideas are evaluated so far. To capture new ideas from different sources, it is essential to identify potential sources and specific methods to extract and format their data. Also it is has to be analyzed how this information will be collected and in which time frame. Depending on the nature of the idea sources diverse methods and techniques to extract, store and select the ideas have to be chosen individually.

On the one hand ideas can be collected within the company from employees and management. For this purpose the innovation management must provide an organizational framework and IT-infrastructure which makes the collection and evaluation of these ideas possible. This system is partly existent at KSPG thanks to Pierburg's efforts in the past. On the other hand information from external stakeholders must be observed and explored for usable ideas.

KSPG has launched a strategic project to implement an innovation management process according to Figure 4 across several different departments in order to improve its innovation power.

4 Conclusion and Outlook

This paper presents an approach to the improvement of innovation management by IT facilities, with a particular focus on the automotive supplier industry. It does this in the context of a project that aims at the development of an optimized innovation management system at the automotive supplier Kolbenschmidt Pierburg AG (KSPG). The development, realization and coordination of a group-wide process of idea generation embedded in a strategic innovation management are the major goal. This new innovation management system will be associated with a standardized advanced development process at KSPG.

A model based on the stakeholder concept is introduced which provides a solid starting point for further analysis on innovation management in connection with knowledge mining. An additional research should have a close look on how the application of knowledge mining techniques and other methods in the automotive supplier industry – especially at KSPG – can support the innovation management and product development.

The approach of knowledge mining should be discussed specifically as an occasion to combine all these different disciplines in order to increase the understanding of an automotive supplier's environment. Knowledge mining techniques will help discover useful knowledge as patterns and/or models in their own data from internal stakeholders or in data from external stakeholders and information sources.

Knowledge mining will be used to optimize the idea generation process by clustering in-house ideas to get a picture of the current company culture and R&D activities. It will support harvesting knowledge from the World Wide Web for identifying general technology fields or trends to inspire and facilitate product development. This two-way-analysis will be the core of future research in the strategic project. Finally, the research results will increase corporate insights along with improvements in product development quality and value.

Literature

[1] Dannenberg J. (2008). Große Chance für Zulieferer. Interviewed by Tina Rumpelt. In: Automobil Produktion, Sonderausgabe: Innovationen in der Automobilindustrie, May 2008, p. 18-19

[2] Riel A., Tichkiewitch S., Uys W., du Preez N. (2008). Mining Knowledge in the Digital Enterprise. Proceedings of DET2008, 5th International Conference on Digital Enterprise Technology, Nantes, France, 22-24 October 2008

[3] Birchall D. W., Chananon J.-J., Tovstiga G. (2003). Knowledge Sourcing and Assimilation – Innovation in the Automotive Industry. In: Innovation Management in the Knowledge Economy, Series on Technology Management – Vol. 7, edited by Dankbaar B., Imperial College Press, London 2003, p. 167-184

[4] German Association of the Automotive Industry (VDA) (2009). The economic situation of the automotive industry in 2008. Frankfurt/Main 2009

[5] German Association of the Automovie Industry (VDA) (2008). Auto Annual Report 2008. Frankfurt/Main 2008

[6] Gassmann O., Bader M. A. (2007). Patentmanagement – Innovationen erfolgreich nutzen und schützen. 2nd edition, Springer, Berlin Heidelberg 2007

[7] Dannenberg J., Burgard J. (2007). Car Innovation 2015 – A comprehensive study on innovation in the automotive industry. Published by Oliver Wyman Automotive, Munich 2007, http://www.car-innovation.com/pdf/studie_car_innovation_2015.pdf

[8] Mattes B., Meffert H., Landwehr R., Koers M. (2004). Trends in der Automobilindustrie: Paradigmenwechsel in der Zusammenarbeit zwischen Zulieferer, Hersteller und Händler. In: Automotive Management – Strategie und Marketing in der Automobilwirtschaft, edited by Ebel B., Hofer M., Al-Sibai J., Springer, Berlin Heidelberg 2004, p. 13-38

[9] Stockmar J. (2004). Erfolgsfaktoren für Automobilzulieferer – Strategien für 2010. In: Automotive Management – Strategie und Marketing in der Automobilwirtschaft, edited by Ebel B., Hofer M., Al-Sibai J., Springer, Berlin Heidelberg 2004, p. 61-77

[10] Griffin A. (1997). PDMA research on new product development practices. In: Journal of Product Innovation Management (JPIM), 14 (6), p. 429-458

[11] Pleschka F., Sabisch H. (1996). Innovationsmanagement. Schäffer-Poeschel, Stuttgart 1996

[12] Stockmeyer B. (2001). Ansatzpunkte und Methoden zur Effizienzsteigerung im Innovationsmanagement der Ernährungsindustrie. PhD Thesis, Technische Universität München, Germany, 2001

[13] Hambüchen T. E. (1989). Innovation als produktpolitische Maßnahme in der Milchwirtschaft der Bundesrepublik Deutschland. Wisenschaftsverlag Vauk, Kiel 1989

[14] Cooper R. G., Kleinschmidt E. J. (1986). An investigation into the new product process. In: Journal of Product Innovation Management (JPIM), 3 (2), p. 71-85

[15] Brockhoff K. (1999). Forschung und Entwicklung. 5th edition, Oldenbourg, München Wien 1999

[16] Le Corre A., Mischke G. (2005). The Innovation Game. A New Approach to Innovation Management and R&D. Springer, New York 2005

[17] Herrmann C., Bergmann L., Thiede S., Zein A. (2007). Life Cycle Innovations in Extended Supply Chain Networks. In: Advances in Life Cycle Engineering for Sustainable Manufacturing Businesses, Proceedings of the 14th CIRP Conference on Life Cycle Engineering, Waseda University, Tokyo, Japan, June 11th-13th, 2007, edited by Shozo Takata and Yasushi Umeda, Springer, London 2007, p. 429-444

[18] Asakiewicz C. (2008). Knowledge Mining – The Quantitative Synthesis and Visualization of Research Results and Findings. VDM Verlag Dr. Müller, Saarbrücken 2008

[19] Ammar-Khodja S., Bernard A. (2008). An Overview on Knowledge Management. In: Methods and Tools for Effective Knowledge Life-Cycle-Management, edited by Bernard A., Tichkiewitch S., Springer, Berlin Heidelberg 2008, p. 3-21

[20] Barachini F., Rankl S. (2008). The Relevance of Knowledge- and Innovation Management for the European Automotive Supply Industry : A Case Study. In: Knowlegde Management – Competencies and Professionalism, Series on Innovation and Knowledge Management – Vol.7, World Scientific Publishing Co. Pte. Ltd., Singapore 2008, p. 159-169

[21] Finkler W. (2006). Deutsche Börse Systems AG: Textmining im Dienste von Competitive Intelligence. In: Competitive Intelligence – Strategische Wettbewerbsvorteile erzielen durch systematische Konkurrenz-, Markt- und Technologieanalysen, edited by Michaeli R., Springer, Berlin Heidelberg 2006, p. 355-367

[22] Michaeli R. (2006). Competitive Intelligence – Strategische Wettbewerbsvorteile erzielen durch systematische Konkurrenz-, Markt- und Technologieanalysen, Springer, Berlin Heidelberg 2006

[23] Bandulet F. (2005). Finanzierung technologieorientierter Unternehmensgründung – Wirtschaftshistorische und institutionenökonomische Erklärungsansätze von Schumpeter bis Williamson. Deutscher Universität-Verlag/GWV Fachverlage GmbH, Wiesbaden 2005

Author CVs

Martin Neumann

Martin Neumann is Manager Innovation Services within the central research and technology department of Kolbenschmidt Pierburg AG and has been working since 2004 for Kolbenschmidt Pierburg and the Pierburg GmbH in the fields of market research and innovation management. He has professional work experience in marketing, consulting and market research, and he has been certified as European Innovation Manager; Martin.Neumann@de.kspg.com

Andreas Riel

Andreas Riel is contractual expert at Grenoble Institute of Technology and Program Manager of Innovation for the EMIRAcle association. He has ten years of professional work experience in the automotive industry, and three years of experience in the management of the European Network of Excellence VRL-KCiP, which is at the origin of EMIRAcle. Over the recent years he has gained vast experience in innovation management in both product development and manufacturing, and he has been certified as European Innovation Manager and Trainer; Andreas.Riel@grenoble-inp.fr

Daniel Brissaud

Daniel Brissaud has been a professor of engineering design and eco-design at the University of Grenoble (Grenoble-INP) since 1998. He is head of the cluster of research “Management and Organisation of Production Systems and Innovation” for the Rhône-Alpes area, member of the European Manufacturing and Innovation Research Association “EMIRAcle”, CIRP fellow. His scientific interests are: Eco-design, environmental assessment, lifecyle engineering, clean technologies, product-service systems design, integrated design and requirement engineering. He authored more than 60 papers in international journals and books and mentored 16 PhD theses.

Explorative Hazard Analysis Process in Requirement Phase

Masao Ito

Abstract

In the software intensive system, dependability is highly important. However, It cannot be achieved by simply enforcing the V&V activities. We have to consider the additional process in order to avoid the hazardous situation. Recently there are several international standards, for example, IEC 61508 and its derived standard, ISO/DIS 26262. When we conform to these standards, it is hard for software people to identify the hazardous situation because of the characteristics of software: Hardness to modularize software and to measure the reliability of each module. In this paper, I propose the dSPO method and its process. It provides the exploratory method to find out the hazardous situations and the way to consolidate them in requirement phase.

Keywords

Dependability, Hazardous Situation, Functional Safety, SPO, dSPO

1 Introduction

The safety-critical systems, for example airplane or automobile, need the high dependability, but it becomes harder because of increase of the size and complexity of software [1].

Recently several standards about functional safety were issued to assure the system safety. The IEC 61508-3[2] claims the hazard analysis and risk assessment in concept phase. And it suggests using the conventional methods, such as FTA, FMEA and ETA. In software intensive system, however, we meet the difficulties in using them [3]. Because those methods assume that system can be decomposed into the parts and the relationships among them are clearly distinguishable. And the reliability of each part is known through experiments such as rupture test and durability test. But as for software, it is hard to separate modules thoroughly and to know the degree of reliability of each module. Moreover, the complexity and size of system is increasing and we need the good way to analyze the system safety and to describe the safety requirement in early phase.

In this paper, I propose the dSPO for finding the hazard identification in requirement phase. It is based on the Software Process Optimization (SPO), which is the method for finding the solution for software process change. It provides the exploratory and practical way for team thinking. The dSPO also provides the way of finding the hazardous situations and formalizing them. Both methods have the analogous process with explorative and heuristic way of thinking. I will show the usefulness of dSPO in concept phase, but we can use it through the system development life cycle.

First, I show the related work in Section 2. In Section 3 the characteristics of embedded system is simply presented. The SPO is briefly introduced in Section 4. In Section 5 the dSPO is described, and I show the simple example to check the applicability of dSPO in Section 6. Conclusion is drawn in Section 7.

2 Related Work

There are several works that use the conventional techniques in the software-centred system development. For example, software FMEA applies FMEA method to software in the design phase [4]. Software Hazard Analysis and Resolution in Design (SHARD) is the method that uses the Hazard and Operability study (HAZOP) [5, 6]. Those methods are used in system and/or software design phase where hardware/software structure is clear.

As for requirement development, there are several methods. For example, in KAOS [7, 8] approach they use the goal-driven approach and derive the specification from requirements. In this KAOS process, we can also write the non-functional requirement, such as safety relating property. They use the temporal logic based notation in each goal, and the goal model is refined into the object model and operational model. However, it is still tough work for us to find out the hazardous situations before describing them. In my paper, I will show the practical way to find them out in the requirement phase. After finding the hazardous situation, we can use the KAOS in requirement phase or other conventional method in design phase.

3 Structure of Embedded System

First of all, I show the basic structure of embedded system as preparations for explaining the dSPO method.

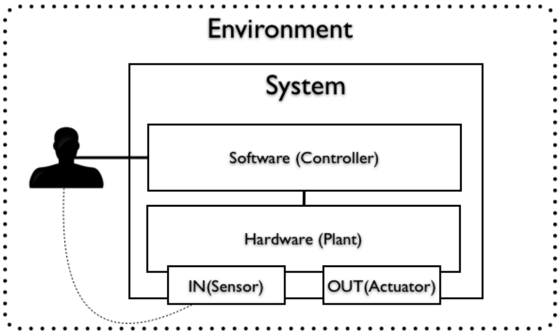


Figure 1. Control-plant-environment structure

System is composed of software and hardware, and software interacts with outer environment through hardware (IN-interface and OUT-interface). The human intervenes between the system and the environment and human-in-the-loop is comprised. Interaction between the human and the system is done through hardware (e.g. button or touch screen), but semantically in system requirement phase we can think it directly interact with software. Because we can postpone our decision about which hardware is suitable for user until design phase.

4 Brief Introduction of Software Process Optimization

The software process optimization (SPO) is the method that does the software process change in order to solve the process problem that the organization has or to adapt their process to the current and future business environment [9]. Figure 2 indicates the overall structure of SPO process.

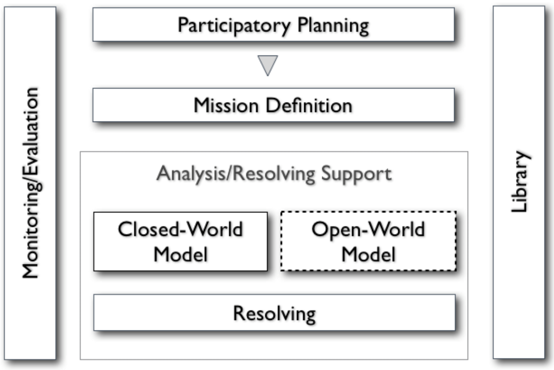


Figure 2. Structure of SPO process

At the first phase, the stakeholder will do participatory planning in order to analyze the current situation. We describe the problem structure, which includes the cause-problem-effect relationship. For example, the cause "no version control" arises the problem "developing done by older version of specification", and then it ensue the undesirable effect "the system is different from the specification". It is the connected tree at a centred problem and including several causes and effects.

In mission definition phase, members carefully choose one cause-problem effect relationship and rewrite it into the output-purpose-goal relationship respectively. We may rewrite the former example like this: output is "doing version control", purpose is "develop is done by right specification" and final goal is "the system is built based on the right specification". Then, the mission that we have to tackle with is defined.

To solve the problem, SPO provides two contrasting techniques: the closed-world modelling and the open-world modelling. These partially come from the idea of Unified Structured Inventive Thinking (USIT) [10]. In close-world modelling the activity that we are interested in are modelled and we look for the solution within the model. We operate on the model by modifying the objects and their relationships, such as changing the number of objects or modifying the attribute of an object. In the open-world model we at the beginning think the ideal solution without the limitation of the reality, then we approach the solution from ideal one. The difference between the closed-world model and the open-world model is that the former starts from the problem itself but the latter starts from the ideal situation. In other words, the former is the depth-centred searching and the later is the breadth-centred searching.

In resolving phase, we chose one idea, apply it into the process and check the result. If it does not work well or we need to increase the effect of process change, we can choose another idea.

We do monitor and evaluate those outcomes of changing process by using SPO Activity Matrix that is a form of ‘process change mission record’, and can keep them in SPO library.

5 dSPO

The dSPO has the same process structure as the SPO has. Here we focus on identifying the problem about dependability, especially, hazardous situation. In the first cycle, we find it and then we solve it in the second cycle. In this paper, I only argue how to find the hazardous situation, solving technique is fundamentally the same as original SPO.

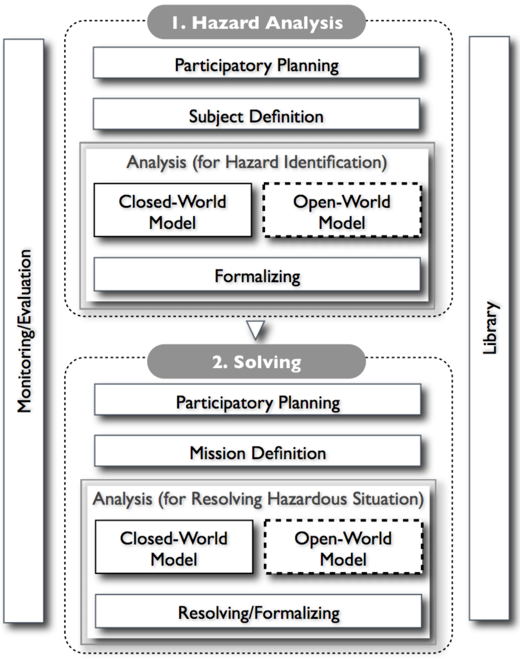


Figure 3. Structure of dSPO process

(1) Participatory Planning

In the meeting, all stakeholders including the engineer and tester who have enough experience about the target domain discuss the target system.

(2) Subject Definition

In this phase, we analyze the functional and non-functional requirements. There are two types of timing to write non-functional requirement. That is, we write it simultaneously with functional requirement, or after describing functional requirement. In the conventional hazard analysis methods, they lead non-functional requirement after functional one is defined. However, the recent requirement engineering methods, such as KAOS [8] analyze and define them all together. We use the KAOS-like approach, and check the hazardous situation out in next phase. We revise the functional and non-functional requirement after this hazard identification.

(3) Analysis for Hazard Identification

We do analysis for hazard identification. As well as SPO, we use two types of technique: the closed-world analysis and the open-world analysis. In the closed-world analysis, we focus on the system boundary, especially the boundary showed in behaviour of the system. On the other hand, in the open-world analysis we broaden this to the environment beyond the target system (cf. Figure 1).

I use the HAZOP method [11] for this purpose in requirement phase (the software HAZOP [12] is also a hazard analysis method, but it is for design phase, not requirement phase).

In close-world analysis, we use the time-relating HAZOP guideword after writing the behaviour of the system. Here we focus on the intended behaviour of system.

Table 1. Time-relating HAZOP guideword

Guidewords	Meanings
EARLY or LATE	Related to the clock time
BEFORE or AFTER	Relating to order or sequence

In the open-world analysis, we mainly use the space-relating HAZOP guideword. In the open-world analysis, the boundary of our thinking area expands and it includes the environment. In this phase, schematic approach is helpful.

Table 2. Space-relating HAZOP guideword

Guidewords	Meanings
NO or NOT	Negation
MORE or LESS	Increases or decreases
AS WELL AS	Qualitative increase
PART OF	Qualitative decrease
REVERSE	Opposite
OTHER THAN	Substitution

(4) Formalizing

In this formalizing phase, we compile the hazardous situation that we have already found in previous phase in tabular form. It has several columns:

- Hazardous situation
- Guideword
- Implication
- Cause
- Protection

"Implication" means the effect that may occur in a hazardous situation. In many cases, the "Cause" and "Protection" are considered in system analysis and design phase, so here those are blank columns.

We can use this information to write the safety case in which we will show the claim / evidence / argument structure [13, 14]. As for claim, we assert that implication will not occur or will be mitigated into the safe level. After design and test, we can prepare the base data for the evidence. Then we can write the argument that evidence shows claim is correct.

If we use the goal decomposition like KAOS, we can also write the safety case in every subgoal.

6 Example of dSPO

In order to show the dSPO process and usage, I show the simple example of the adaptive cruise control (ACC) system. In real world, there are many types of ACC. In addition, we can find several relating documents, for example, SAE's standard [15] or the example requirement for workshop in safety working group [1]. However, the purpose of this chapter is to demonstrate the usage of dSPO, so I will give the simple ACC definition:

ACC can keep the time gap to follow a forward car and it also has the conventional cruise control system functionality, which is control the vehicle speed constant. There is maximum and minimum set speed, and the ACC can only work within this range of speed. If brake or accelerator pedal is pressed, the ACC function is cancelled immediately.

(1) Participatory Planning

The workshop will be opened and every stakeholder, such as hardware/software engineers of brake control and engine control and so on, come to discuss about the object of study and make up a plan for analyzing hazardous situation.

(2) Subject Definition

To share the common image of ACC among participants, first we write the KAOS-like goal model.

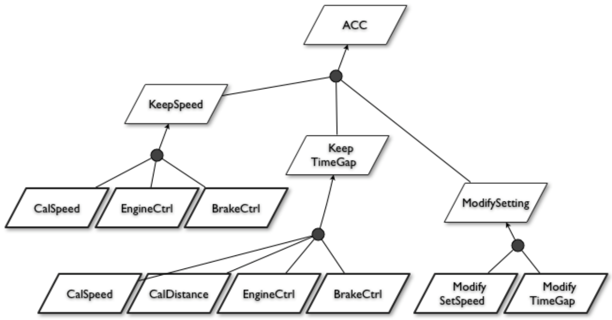


Figure. 4. ACC goal model diagram

In this stage, we just divide the goal into several subgoals. It does not have detailed information that will be added after hazard analysis.

(3) Analysis and Hazard Identification

In the closed-world analysis, we mainly focus on the behaviour of system, especially the behaviour of "controller".

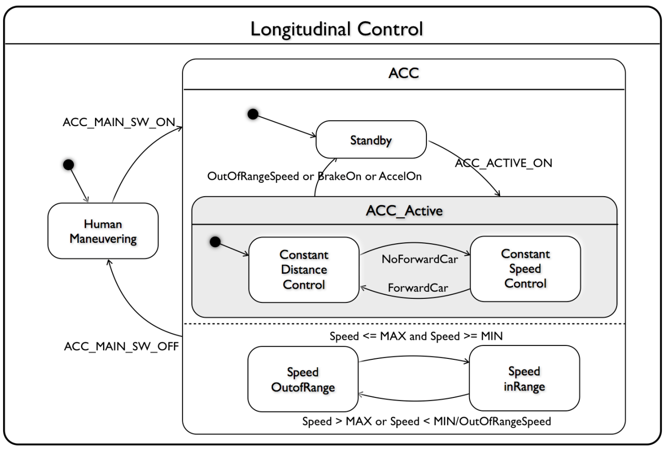


Figure. 5. Simple behaviour of ACC controller

The important point of this behaviour description is that we do not need to describe the detailed states and transition, such as how to show the warning to the driver. Those are design matter and we can postpone describing details until design phase.

With this behaviour model, we use the time-relating HAZOP guideword. In case of the 'LATE' guideword, if ACC is in the constant speed control mode and the 'FowardCar' event is *delayed*, subject car might go close to the forward car. In fact, bad weather or soiled camera/radar might not send the correct information to the ACC. Another example is the guideword 'BEFORE'. When entering the ACC mode and the forward car does not exist, the transition is from default state to 'Constant Distance Control' state, and immediately transit to 'Constant Speed Control'. The intermediate transit to 'Constant Distance Control' is useless in this case, and there may be an unwanted activity in this state. For example, the 'Constant Distance Control' state calculates the time gap but it may have meaningless clearance to the forward car. In this case, the guideword 'BEFORE' remind us this useless transition.

In the open-world analysis, we use the broader boundary (i.e. we also cover the environment) and think about the real object image rather than a model element. It is helpful to stir the image of hazardous situation.

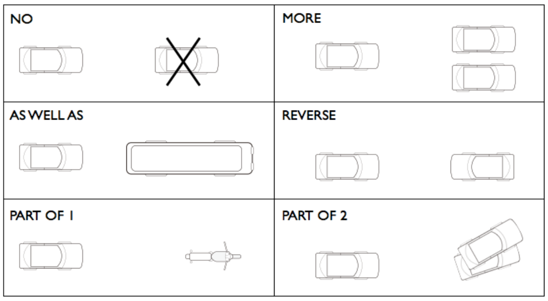


Figure 6. Open-World analysis

Each guideword and figure facilitate answering the 'what if' question. 'NO' means that there is no forward car, but ACC recognizes a car. 'MORE' means that there are multiple forward cars. 'AS WELL AS' means that the forward car is very huge. In 'REVERSE' situation, the forward car runs in the opposite direction. 'PART OF' has two situations. The one is the small transport, such as motorcycle. The other is the forward car changing direction.

Not all those cases directly connect to the hazardous situations. The point is whether we can answer the 'what if' questions or not. This is also useful to depict the safety envelope, because it shows the limitation of system behaviour in a safe manner.

(4) Formalizing

Finally we can write the table of hazardous situation.

Table 3. Sample table of hazardous situation

Hazardous Situation	Guide Word	Implication	Cause	Protection
Delay of 'NO_FORWARD_CAR' event	LATE	Wrong time gap calculation and unwanted control may occur		
The transition from default state to 'Constant Speed Control' state	BEFORE	In 'Constant Distance Control' the unnecessary activity may occur.		

In addition, we can use this table as the base of writing safety case.

7 Conclusion

I may reasonably conclude that the dSPO provides the good process and technique to find out the hazardous situation in requirement phase, and I will show it through the simple example. Exploration in the close-world analysis and heuristic thinking in the open-world analysis are key technique to identify the hazardous situation. Of course, in design and coding phase, there are several methods to achieve high dependability by using formal model or machine-readable code. However, in requirement phase, we do not expect those formal inputs. We carefully and comprehensively think about the problematic situations. The dSPO approach is helpful in this situation.

Literature

1. Leveson, N.: Adaptive Cruise Control System Overview (5th Meeting of the U.S. Software System Safety Working Group), http://sunnyday.mit.edu/safety-club/workshop5/Adaptive_Cruise_Control_Sys_Overview.pdf. (2005)
2. IEC: Functional safety of electrical/electronic/programmable electronic safety-related systems (IEC 61508-3). Part3: Software requirements. IEC (1998)
3. O'Connor, P.D.T., Newton, D., Bromley, R.: Practical reliability engineering. John Wiley, New York (1995)
4. Goddard, P.L.: Software FMEA techniques. Reliability and Maintainability Symposium, 2000. Proceedings. Annual (2000) 118-123
5. McDermid, J.A., Pumfrey, D.J.: A development of hazard analysis to aid software design. Computer Assurance, 1994. COMPASS '94 Safety, Reliability, Fault Tolerance, Concurrency and Real Time, Security. Proceedings of the Ninth Annual Conference on (1994) 17-25
6. McDermid, J.A., Nicholson, M., Pumfrey, D.J., Fenelon, P.: Experience with the application of HAZOP to computer-based systems. Computer Assurance, 1995. COMPASS '95. 'Systems Integrity, Software Safety and Process Security'. Proceedings of the Tenth Annual Conference on (1995) 37-48
7. Lamsweerde, A.v., Fuggetta, A.: ESEC ‘91: 3rd European Software Engineering Conference, ESEC '91, Milan, Italy, October 21-24, 1991: proceedings. Springer-Verlag, Berlin; New York (1991)
8. Lamsweerde, A.v.: Requirements engineering: from system goals to UML models to software specifications. John Wiley & Sons Ltd. (2009)
9. Ito, M.: The Elements of Software Process Optimization: Dealing with the Process Dynamics. Software Process Workshop, 2005 (2005) 449-453
10. Sickafus, E.N.: Unified Structured Inventive Thinking: How to invent NTELLECK (1997)
11. CEI/IEC: Hazard and operability studies (HAZOP studies) - Application guide, CEI/IEC 61882:2001. IEC (2001)
12. Redmill, F., Chudleigh, M., Catmur, J.: System Safety: HAZOP and Software HAZOP. John Wiley & Sons, Inc. (1999)
13. ISO: ISO/DIS 26262, Road vehicles - Functional safety -. ISO (2009)
14. MISRA: Guidelines for Safety Analysis of Vehicle Based Programmable Systems. MISRA (2007)
15. SAE International: Adaptive Cruise Control (ACC) Operating Characteristics and User Interface (J2399). SAE International (2003)

Author CVs

Masao Ito

Mr. Masao Ito is CEO of Nil Software Corp, a Tokyo-based software tool development company. He has long career in IPSE (Integrated Process Support Environment) construction and consulting. He is a board member of SEA (Software Engineers Association of Japan) and a core member of SEA-SPIN. Nowadays, he is also being involved in research activity of Volume CAD application development as a visiting scientist of RIKEN and NIRS (National Institution of Radiological Sciences), and serving as CEO of a venture company VCAD Solutions, Co., LTD.

Initiating Quality Management in a Small Software Enterprise

Markus Suula, Timo Mäkinen, Timo Varkoi
Tampere University of Technology, P.O.Box 300 (Pohjoisranta 11 A), FI-28101 Pori, Finland

Abstract

The quality of a software product is largely determined by the quality of the process that is used to develop and maintain it. The major quality management constraints in small enterprises are the lack of resources and the cost of software process improvement programs. In this paper, we discuss the initiation of quality management in a small software enterprise and tell a story about the first year of a part-time quality manager. We will present an example of how to solve a quality management-resourcing problem using a multirole arrangement. We describe how improvement areas were identified and improvement actions performed. The discussion is based on observations and hands-on experiences in the enterprise during the year 2009.

Keywords

Software process improvement, quality management, problem elicitation, commitment, SPICE

1 Introduction

It is widely accepted that the quality of a software product is largely determined by the quality of the process that is used to develop and maintain it [1]. Process improvement means understanding existing processes and changing these processes to increase product quality and/or reduce costs and development time [2]. The special characteristics of small companies mean that process improvement programs must be applied in a way that is particular to them and visibly different from how this is done in large organizations [3]. A large-scale software process improvement programs are generally not possible in small enterprises because they cannot afford to implement these programs. [4]. Another major characteristic of small enterprises is the lack of resources allocated to quality improvement tasks [5].

In this paper, we will discuss the initiation of quality management in a very small software enterprise (VSE: a company with fewer than 25 employees) and tell a story about the first year of a part-time quality manager. We will present an example of how to solve a quality management resourcing problem in a low-cost and practical way using a multirole arrangement. We describe a problem-based approach to quality management and the actions performed in order to address the issues that are identified. The discussion is based on observations and hands-on experiences in the enterprise during the year 2009.

This paper is structured as follows: Section 2 describes the background of our study. In Section 3 we present how areas for improvement were identified. Section 4 reports the improvement actions and finally Section 5 summarizes the study and provides conclusions.

2 Background

In this section we introduce the enterprise in question and review its process improvement history. We will examine the enterprise's process guide and the status of its process. In addition, we describe how its quality management was set up and the reasons behind it.

2.1 The Context of the Study

The context of our study is a small Finnish software enterprise managed by its owners. The enterprise was founded in Finland in 1994. The wide product repertoire included products for information management systems and industrial solutions. The enterprise became an advanced IBM business partner in 1998 so it was natural that the products were based on IBM technologies. Their customer projects aim to follow the waterfall model. Nowadays, the enterprise has reduced its product portfolio and has concentrated on production monitoring and reporting systems (PMRS) and enterprise content management (ECM). The products are based on IBM technologies such as DB2, DB2 Content Manager, Websphere Portal, and Lotus Domino.

The company's revenue and headcount have grown steadily: in 2006 there was a staff of eight persons and 0.4M€ revenue, in 2007 there were 11 employees and the revenue was 0.6M€, in 2008 the employee count was 13 and the revenue 1.0M€ and by 2009 the revenue was 1.2M€ and the enterprise employed 17 persons. The enterprise has expanded its market area to Scandinavia, the United Kingdom, and the Baltic countries. Software process improvement (SPI) is considered to be crucial in order to gain growth in the international market.

2.2 Process Assessments

The enterprise has a long history of exploiting process assessments to elicit process improvements. The assessments have been carried out in conjunction with SPI-related research projects managed by a local university. The enterprise has actively participated in the projects and, despite its small size, provided consistent research funding. The close co-operation in projects has been rewarded by a detailed understanding of the company and its history.

The first of the SPI initiatives took place in late 1998 when the company started systematically to evaluate its strengths and weaknesses in software development. The first joint effort was to find the priorities for process improvement. The ISO/IEC 15504 (SPICE, Software Process Improvement and Capability dEtermination) processes and process categories were used as the basis for the prioritization and collected the views of management about their importance. The result was that the two highest priority areas were found to be software engineering and management processes, followed by customer-supplier, support, and organizational processes.

The goal of the first assessment in early 1999 was to analyze the situation, establish processes, and improve manageability. The assessment covered requirements for management-related processes, software design, construction and testing processes and also the project management process. The assessment scope was set to capability levels 1-3, Performed, Managed, and Established. Typically the processes were performed but not well managed. The results helped to clarify the process map of the enterprise and to set the goals for the improvement efforts.

The second assessment had two stages. The first stage was carried out late 2000 with the aim to review the improvements made to the processes of the first assessment. The second stage early 2001 concentrated in software testing and integration, and customer support. Some improvement could be noticed but the variation between projects was remarkable. Again, the processes assessed for the first time were mostly on the performed level.

The third assessment took place in late 2002 and covered software requirements and design processes. The result was unclear and alarming in that one process was well managed, but another was not performed properly. Obviously the processes were not established and still relied very much on individuals and their experience. These findings led to the start-up of the efforts presented in this paper. Since then, assessments have been performed in a very informal way, more to check the situation and to get an idea of the capability of the processes. For instance, a light assessment type of process walk-through was performed in late 2006 for all the engineering processes. Nevertheless, process assessment models have provided a sound basis for discussions about the needs for improvement in the enterprise.

In this case the company has changed its software platforms and business domains frequently, which has caused a need to rethink the role of process assessments. At the beginning, the more formal assessments provided the basis to understand the processes. Later on, as the processes become more established, process models can be used systematically to produce ideas for process improvements without formal capability determination, and to model the implemented processes, for example.

2.3 Standard Process Set

In 2004, a study [6] was carried out in the enterprise in order to improve their processes towards ISO/IEC 15504 capability level 2, where a process is performed and managed. The planning of the study consisted of an analysis of the latest process capability assessments conducted in 2001-2002. The main focus was on the management practices so the management process group had the first priority for improvement. The second priority was given to the customer-supplier process group and the third priority to the engineering process group. Based on the assessment report from 2001, the enterprise had recognized and listed ten individual software processes. These ten processes were used to create a standard process for the enterprise called the Standard Process Set (SPS). During the process improvement study the number of processes was increased from ten to nineteen in order to complete the SPS.

Table 1: Processes and Process Categories of the Standard Process Set

Process Categories	CUS - CUSTOMER-SUPPLIER PROCESS CATEGORY	ENG - ENGINEERING PROCESS CATEGORY	SUP - SUPPORT PROCESS CATEGORY	MAN - MANAGEMENT PROCESS CATEGORY	ORG - ORGANIZATION PROCESS CATEGORY
Processes in Category	**SM - Sales and Marketing** **DL - Delivery** CR - Customer Requirement Specification PS - Production Service	FS - Functional Specification TS - Technical Specification SI - Software Implementation **MT - Module Testing** IT - Integration Testing **FT - Functional Testing** ST - System Testing RT - Requirement Testing **PL - Piloting** MS - Maintenance Service	**DN - Documentation** **QS - Quality Assurance** **CM - Configuration Management**	PM - Project Management	**PE - Process Establishment**

The idea of the SPS is that it is customized to the needs of a project. The nineteen processes of the SPS are divided into five categories. Processes and process categories are presented in Table 1. The processes added in 2004 are bolded. The SPS is a folder structure where each of the processes has its own folder containing the process content. Process-related content can be grouped into four groups: 1. Guidance, 2. Checklists, 3. Templates, 4. Misc. The group “Misc” consists of pictures, tables and other miscellaneous content. The SPS content was built to fulfill the requirements of ISO/IEC 15504 capability level 2 [6].

2.4 Characterization of Processes

A case study [7] was carried out in 2008 in which the current process of the enterprise was characterized in depth by examining an ongoing project. In the study it was noticed that the enterprise did not utilize the SPS; instead it moved ahead on an ad hoc basis. The project studied in the case study consisted of two separate subprojects, i.e. the requirement elicitation and the implementation subprojects. The former produced a customer requirement specification which was the main input for the latter, i.e. the implementation project. This project cannot be considered as a success because it took twelve months more than planned and the work hours were exceeded by 300 per cent.

The study lists the following as the key problems in the process:

- Project moved ahead on an ad hoc basis and deviations to the project plan were not acted upon
- Communication problems
- Lack of change and risk management
- Overlooking the importance of documentation
- No project management tool
- Insufficient testing

Many of the problems in the studied project could have been avoided by following the SPS but it had clearly become “shelfware” - documentation that is meagerly used or not at all, with the exception of preparing for audits, etc. The management was still somewhat committed to the SPS and they tried to encourage people to use it even though they knew that the SPS was largely incomplete and needed to be updated and that it did not correlate with the defined set of SPICE processes for the enterprise. The SPICE process architecture consists of the following processes: ENG.1 Requirements Elicitation, ENG.4 Software Requirements Analysis, ENG.5 Software Design, ENG.6 Software Construction, ENG.7 Software Integration, ENG.8 Software Testing, ENG.12 Software and System Maintenance, MAN.3 Project Management, MAN.4 Quality Management, MAN.5 Risk Management, PIM.3 Process Improvement, SUP.1 Quality Assurance, SUP.7 Documentation, and SUP.8 Configuration Management.

In relation to the process from the SPICE point of view [8]: Each of the individual processes has a process definition in the SPS (PA 3.1 Process definition) but the definitions are not utilized (PA 3.2 Process Deployment). The project plan was not executed as planned (PA 2.1 Performance Management) and the documents and plans were not maintained (PA 2.2 Work Product Management). As far as the maturity of the enterprise is concerned, neither the process architecture nor the SPS fulfill the requirements for organizational maturity level 1 [9].

In addition, we can say that the process does not fulfill a single one of Zahran’s process efficiency characteristics [1]. The process is not followed because it is incomplete and is not enforced nor is training given, making it impossible to measure. The process is not owned by anyone so no one is responsible for updating it. And how could the process be updated or improved because there is no feedback? The process guide had become obsolete.

Process improvement has always been declared to be part of the enterprise’s strategy but SPI has not been consistent and the discrete SPI projects have not left a permanent impact. The true commitment of the management to SPI can be questioned for the reason that in spite of the long history of SPI work, the maturity of the enterprise has not evolved, and practices in project management or in software construction have not been established.

2.5 Establishing Quality Management

One of the business strategies of the enterprise was to concentrate more on the markets outside Finland and even establish a branch out of the country. The current maturity of the enterprise was not on such a level that establishing a branch would be possible so there was a need for overall process improvement. The problems identified in the process motivated the management to really invest in SPI activities. The enterprise hired a part-time quality manager (QM).

The QM started at the beginning of 2009. He was a recent graduate who had worked in the enterprise as a software engineer and had also been involved in the 2008 case study. The job description was explicit: the person hired had university roles and enterprise roles. The university roles were *researcher* and *post-graduate student*, and the enterprise roles were *quality manager* and *software engineer*. It was determined that he would work as a software engineer two weeks per month and one week as a QM, with one week allocated for the university roles. The idea was that during the research week the QM would search for the newest and most suitable practices to apply in the enterprise and as a QM would put these into practice. The software engineer role would give the QM a deep insight into how the above-mentioned practices really work. Figure 1 illustrates the arrangement.

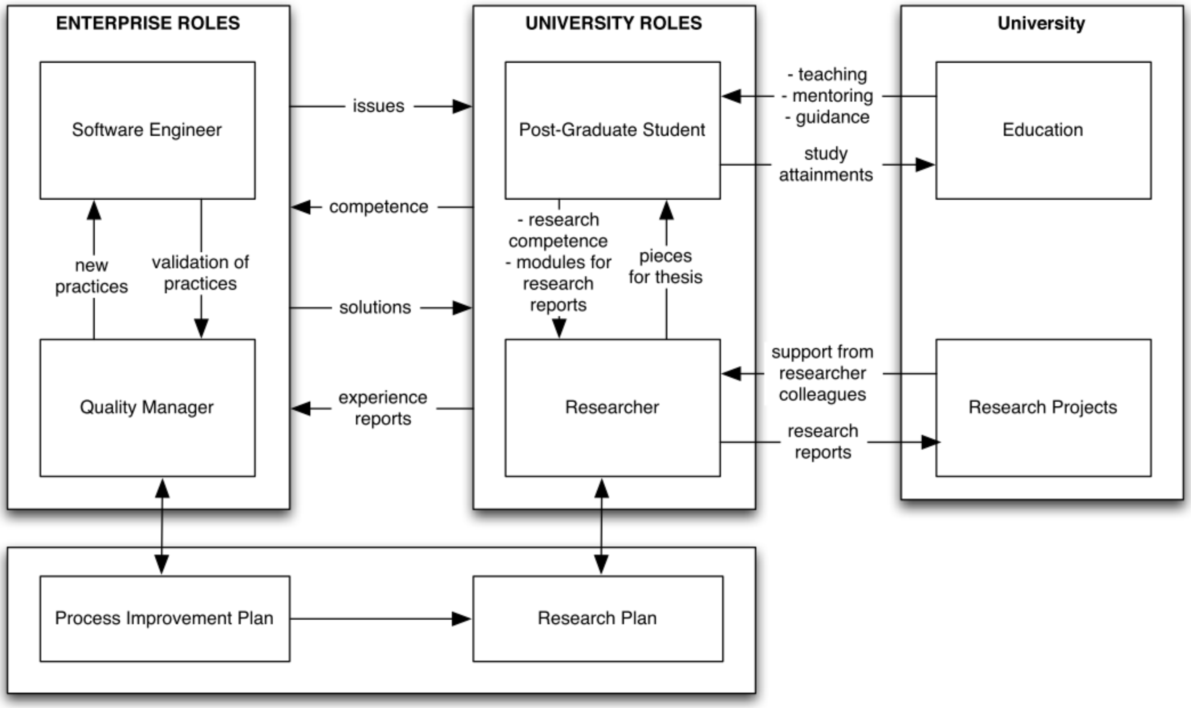


Figure 1: A Multirole Arrangement for Quality Management

There were a couple of reasons for such an arrangement; firstly, the enterprise could not afford to hire a full-time QM so the person had to be able to work as a software engineer. The second reason was that the enterprise was going to participate in an SPI-focused research project which required an enterprise representative. A similar setup was used in the 2008 case study and the experiences encouraged the continuation of this arrangement.

The management of the enterprise set no explicit goals for the QM or for the quality work in general. There had been some SPI projects in the enterprise before which had not made any permanent impact on the process. The QM took a problem-based approach in order to get results and through the results make staff committed to SPI.

3 Elicitation of Software Process Issues

In this section, we present how process issues were elicited and report on these issues.

3.1 Interviews of Key Persons

SPI work started with four interview sessions. The purpose of these sessions was to elicit problems in the current process and to induce a commitment to SPI from key staff members - people are more likely to adopt new practices if they arise from actual needs. The essential parts of the sessions were documented and reviewed by the session participants.

Project Managers
The first session was with the two project managers (PM) in the company who said that effective project management was very challenging and sometimes impossible because of a lack of human resource management practices and the tools to manage and monitor the workload and tasks of the team members. The enterprise had an hour tracking system (HTS) which was used for billing purposes. Staff members entered their work hours into the HTS by filling out a form. In the HTS the

process and the task are selected from a static list which is the same for all projects, i.e. the task is not project-specific. The description field can be used to specify what exactly was done. The HTS cannot be used to track the progress of individual project-specific tasks, so the HTS did not fulfill the requirements of a project management tool. The PMs found it difficult to monitor work hours because people used the HTS in a variety of ways - there was no corporate procedure for using the HTS.

Human resource allocation was the responsibility of the managing director (MD). In the organizational structure of the enterprise, the MD was on top and everyone else was on the same level beneath him. This made it possible to bypass the PMs in the command chain in such a way that it was impossible for the PMs to manage human resources. The PMs were unable to plan their projects because they were not allowed to manage their resources. In most cases the project plan and schedules were made to satisfy the customer, but these plans were not really implemented or even communicated to the project team.

Chief Designers
The second session was with a chief designer (CD) who said that improving internal communication was an important matter and that a project management tool could help with the communication problems. The CD said that the SPS had become shelfware because it was too heavy and bureaucratic. The third session was with another CD who also thought that the enterprise should have a project management tool. The second CD said that the past process improvement work had been too theoretical and that the theories had not materialized. However, the CD was confident that the QM could really make a difference. Both CDs emphasized the importance of enforcement especially at the beginning of process improvement work.

Managing Director
The fourth session was with the MD. When asked about the SPS the MD said that it was designed for large projects but nowadays most projects were short consulting projects and a more agile process guide was needed. The problems that were discussed in previous sessions were presented to the MD who said that the root cause of these problems was the PMs' lack of leadership and discipline. The MD believed that the PMs did not steer and monitor their projects adequately. The MD took the blame for insufficient human resource management and promised improvements. The MD accepted the idea of trying out a project management tool in a project. It was decided that the QM should find a suitable candidate. The MD found the problems identified in the hour tracking system (HTS) trivial. He could not believe that there could be a problem using the HTS, which he did not use himself. The QM managed to convince the MD that the problems were not trivial, and the QM was given a free hand to improve the HTS.

3.2 Hour Tracking System Workshop

The QM scheduled a workshop for improving the HTS. In the workshop the problems of using the HTS were discussed. The process for creating an invoice began with the PM printing the time entries of a project onto a timesheet and filtering the entries that could not be charged. In the case where a customer wanted task-specific hours, the PM had to distribute the working hours of the composite entries to tasks on gut feeling. Managing the timesheet took a lot of the PMs' time but the most irritating and time-consuming problem was that there was no established convention on when and how to use the HTS. Sometimes some of the time entries were created after the invoice had been created so the whole process had to be repeated. There was confusion among the staff about what to write in the HTS.

3.3 Observations as a Software Engineer

As a project team member in the role of software engineer, the QM had noticed that one major problem in projects was ineffective project material distribution. The enterprise utilized a shared folder on a server as a project material repository. This arrangement worked fairly well with project documentation but there were major problems with source code:

- The documentation and source codes of projects were scattered in different places e.g. the developer's workstation or USB sticks, servers - the newest version was usually somewhere on the developer's workstation
- There was no way to determine which version was the latest
- When a document or a source code file was changed there was no way of knowing
- Team members had no history data available about the actions in a project
- There was no way to determine who had changed a file
- Team members could not work with the same file
- Some changes had gone missing because of unintentional actions. These actions could not be reversed

3.4 Job Satisfaction Survey

Despite all the above-mentioned problems, the staff morale was high and they hungered for change. A job satisfaction survey in the enterprise was carried out by a third party, in which members of staff were asked to list and prioritize areas for improvement. The staff gave the highest improvement priority to software process improvement because the staff felt there was no established project culture - the organization worked more or less on a laissez-faire basis. The fifth highest priority was given to improving the hour tracking system and seventh place to improving internal communication. The enterprise's hour tracking system was found to be insufficient and no proper training or guidance in using it had been given.

4 Implementation of Solutions

In this section we present the improvement actions resulting from the problems identified in section 2. We present how the performance and work product management issues were tackled and organization level matters are also discussed.

4.1 Performance management

HTS improvement started with a kick-off meeting where the problems were discussed and improvement ideas brainstormed. It was determined that before any larger improvements could be made, a corporate way of using the HTS had to be created. The QM started to write an HTS guide based on the kick-off meeting and previous guidelines. The HTS guide was reviewed and updated accordingly but never received final approval from the MD and so it could not be released.

The QM suggested that the HTS should be improved by adding a feature that allows the PMs to create project-specific process sets and to edit the tasks under the processes. In this way the HTS could be used as a project management tool. The plan to improve the HTS in such a way was supported by the PMs but rejected by the MD. The MD was afraid that the suggested changes would cause micro-management.

In order to track the progress of and hours spent on a specific project task, the QM suggested that an open source project management tool should be tested. The QM presented a few tools of this type to the PMs and to the MD. The PMs found these tools worth trying but the MD allowed only a project management tool based on IBM technology to be tested. A trial version of the tool was installed and was evaluated by the QM. The tool was found to be worthwhile and, shortly after, the enterprise became a reseller of the tool. Two projects where the tool was piloted started at the end of 2009 and the results were encouraging.

4.2 Work Product Management

The QM suggested an open source SCM (Source Code Management) tool to be used, which was approved by the MD. The deployment had two phases: 1. version control for programmers, 2. version control as a repository for all project material. There were only two coders for phase 1 who were trained for the SCM tool by the QM. The coders participated into two surveys held by the QM in which they had to prioritize the problems and assess how frequent and how disruptive the problems were on a scale of one to four. There was approximately five months between the surveys. The second survey indicated that the version control system helped a lot with the problems. The coders were forced to use the SCM in ongoing projects and after a while the coders also started to put older projects under version control.

In phase 2 all project material was put under version control. The whole project team took part in surveys similar to those in phase 1. After the project we came to the conclusion that even though version control was much a better place for project material than a shared folder on a server, which was the current practice, there should be a better place for project documentation. As a result of these experiments, a new practice was established, putting source codes under version control and project documentation into a commercial collaboration tool.

4.3 Human Resource Allocation

Human resource allocation was a difficult problem to solve. The projects of the enterprise are usually between 150-500 hours and there are multiple projects going on at the same time that normally share resources. This setup makes it really challenging to do any long-term resource allocation planning. The QM created a light human resource management process, which states as follows:

> "*When sales receive a project description from the customer, he/she defines the scope of work (project manager may assist). The sales/project manager asks for the required resources from the resource manager. The resource manager checks from the resource allocation list if/when the required resources will be available and gives a response. According to the response the sales/project manager can decide if and/or when the project will be started. The resource allocation list is checked and updated in a resource allocation meeting which is held every second week. In these meetings project managers report their resource needs and the resource allocation list is updated accordingly and the minutes of the meeting are written up.*"

The resource allocation list was a spreadsheet, which provided a Gantt chart-like view. The allocations were planned on project level and it was up to the PMs which tasks a person was to do. The QM carried out the role of the resource manager. The process helped a lot with the resource allocation problems but it was still just a tool to control the existing chaos. The PMs could still be bypassed in the command chain.

4.4 Organizational Changes

There was a turning point in quality management when the enterprise hired a new MD. The new MD started on 17 August 2009. The QM presented the situation in the enterprise to the new MD who quickly realized that in order to achieve the organizational business goals a true commitment to SPI was needed. The MD approved the HTS guide almost immediately, which was communicated and distributed to the staff. The HTS was supervised by the MD and the QM and the entries started to come in time and the quality of the entries improved significantly.

The MD found the state of the process untenable so a kick-start of the deployment sections of the SPS was launched. The QM proposed a simple set of processes to be followed. These engineering processes were the basic processes of the waterfall model: requirement elicitation, design, implementation, and testing. Project management and documentation projects were also within the scope of the kick-start. These processes were presented to the project teams.

It was soon realized that the problems at organizational level affected projects more than the issues inside a project. The most important issue was human resource allocation. In an enterprise with a rather wide technology repertoire and with a staff with heterogeneous skill sets, resource allocation was difficult. The first step to correct this issue was a reorganization in the enterprise structure carried out by the new MD. In the new structure the staff (sales excluded) was divided into two teams and each team was supervised by a PM. The PMs needed a shared view of the allocation of the other PM’s team members in order to plan projects. The resource allocation list offered this view.

The MD established a project steering group which consisted of the PMs, the QM and the MD himself. The purpose of the steering group was to help the project teams to accomplish their goals. It was decided that the steering group was to have a meeting every Monday. In these meetings process issues and the progress of the projects were monitored and discussed. The PMs presented the status of their projects with a simple slide show. The steering group meeting worked as an excellent communication forum for SPI issues and strengthened team spirit within management.

5 Conclusion

We have presented a way to initiate quality management in a small software enterprise through experiences gathered during the first year (2009) of a part-time quality manager. We have shown an example of a multirole arrangement for quality management. We have described how areas for improvement were identified and what improvement actions were performed.

Based on our experiences, we propose that initiating SPI in a small software enterprise should start with the establishment of the role of a quality manager. In a small enterprise where a full-time quality manager is not an option a multirole arrangement can be recommended. The strength of this arrangement is that it can be applied today; it can start small and expand as experience and needs increase.

We discovered that when a QM also works as part of the project team, e.g. as a software engineer, the QM can get an understanding of how the process really works and where the problems are. It was noticed that when improvement actions are motivated by local problems people are more committed and willing to try something new.

It was found that with the described multirole arrangement the QM also worked as a social and psychological driver at the operational level where he/she leads SPI by example. The project drivers - costs and benefits - are dominant in the project initiation phase but their role weakens due to the inability of the SPI effort to produce quick and meaningful results, even if these are explicitly sought. This phenomenon causes a need for a social and psychological driver [10]. If such a driver is not found, SPI activities are likely to cease to exist, as had happened in the enterprise studied here: the short-term efforts in SPI had left no permanent impact so these efforts could be considered as a waste of time and money.

Lack of commitment is widely recognized as one of the biggest causes of SPI failure [10] and this is borne out by our experiences. If the MD had not been changed, long-term, sustainable changes across the organization could not have been achieved.

Considering the history of SPI activities in this enterprise, a formal and strictly process-focused approach might have caused resistance to change and been considered as a “necessary evil”. We believe that, in a VSE, a more formal SPI strategy should be created only after putting the right organization in place and when a thorough understanding of the improvement areas has been achieved.

The experiences that we gained from the multirole arrangement were encouraging. The overlapping of responsibilities between different roles does not seem to be a problem; on the contrary, the embodiment of different roles in a single person eliminates communication noise; there is no risk of misinterpretation. We found it crucial that the person with multiple roles is required to have a strong ability to show restraint in order to handle the switching between different roles. The greatest challenge in this arrangement appears to be in balancing the workload of the multirole person. Our future work is to continue refining and validating the multirole arrangement for quality management.

Literature

[1] Zahran, S., “Software Process Improvement,” Pearson Education, 1998.

[2] Sommerville, I.; “Software Engineering. Seventh Edition,” Pearson Education Limited, 2004.

[3] Pino, J.F., García, F., Piattini, M.; “Software process improvement in small and medium software enterprises: a systematic review,” Software Quality Journal, Volume 16, Issue 2, Pages: 237 - 261, 2008.

[4] Habra, N., Alexandre, S., Desharnais, J., Laporte, C.Y., Renault, A.;”Initiating software process improvement in very small enterprises: Experience with a light assessment tool,” Information and Software Technology, Volume 50, Issues 7-8, Pages 763-771, June 2008.

[5] Mishra, D., Mihsra, A.;”Software Process Improvement Methodologies for Small and Medium Enterprises,” Proceedings of the 9th international conference on Product-Focused Software Process Improvement, Pages: 273 - 288, 2008.

[6] Nurkkala, R.; “Improving Project Management In a Small Software Organization,” Master’s Thesis, Tampere University of Technology, Department of Information Technology, 73 p, 2004.

[7] Suula, M., Mäkinen, T., Varkoi, T.; “An Approach to Chracterize a Software Process,” Proceedings of the PICMET'09 Conference, Portland, USA, 7 p, 2-6 August 2009.

[8] ISO/IEC 15504-3, “Information Technology - Process Assessment - Part 3: Guidance on Performing as Assessment,” 2003.

[9] ISO/IEC 15504-7, “Information Technology - Process Assessment - Part 7: Assessment of Organizational Maturity,” 2008.

[10] Abrahamsson, P.; The Role of Commitment in Software Process Improvement”, Academic Dissertation, Faculty of Science, Department of Information Processing Science, University of Oulu, 162 p, 2002

Author CVs

Markus Suula

Markus Suula is currently working as a quality manager. He is also a postgraduate student at the Tampere University of Technology (TUT). He has experience in software engineering as a designer and as a programmer. His current research interests include software process improvement, quality and project management, and software process modeling.

Timo Mäkinen

Dr. Timo Mäkinen works at the Tampere University of Technology (TUT) as a senior researcher. He has eight years' practical experience in software engineering as a programmer, systems analyst, and project manager. He has been lecturing in data management and software engineering for over ten years. He has published more than forty articles related to software process improvement (SPI). His current research interests include software process modeling and knowledge management in SPI.

Timo Varkoi

Timo Varkoi (Lic. Tech.) is currently working as a research project manager at the Centre of Software Expertise (CoSE) at the Tampere University of Technology, Pori. His working experience includes both software development and management responsibilities in industrial software organizations. He is a competent SPICE assessor, a board member of the Finnish Software Measurement Association (FiSMA), and an active participant in ISO/IEC Software and Systems Engineering standardization. His current interests include software process assessment and improvement-related research, expertise, and training combined with the development of software-intensive organizations.

A CMMI Based Configuration Management Framework to Manage the Quality of Service Based Applications

Sajid Ibrahim Hashmi, Stephen Lane, Dimka Karastoyanova[1], Ita Richardson

Lero – The Irish Software Engineering Research Centre
University of Limerick, Ireland
{sajid.hashmi, stephen.lane, ita.richardson}@lero.ie
IAAS, University of Stuttgart, Germany[1]
dimka.karastoyanova@iaas.uni-stuttgart.de[1]

Abstract

Service Based Applications (SBAs) have highlighted new challenges related to Configuration Management (CM). This is an important process for the assurance of end to end quality in software systems. As far as the quality of SBAs is concerned, configuration management remains an issue because of the loosely coupled and adaptive nature of the corresponding applications. A smart configuration management approach will allow organizations to make their IT resources more reliable and to utilize them to their maximum. In this paper, we propose a service-based configuration management framework based on SEI CMMI-SVC which contributes to the S-Cube life cycle. Implementing this approach will allow organizations to effectively manage the configurations of their SBAs.

Keywords

Service Oriented Architecture, SBAs (Service Based Applications), CM (Configuration Management), SC (Software Configuration), Quality Assurance, CMMI – SVC (Capability Maturity Model Integration for Services)

1 Introduction

Today's computer world consists of applications which are scattered across different networks and require special effort in terms of integration. For their smooth operation, developers of such applications need to pay special attention to configurations as 60% of service impacts are due to configuration problems [1]. Organizations have business processes in place in order to meet their objectives – for example, sales, administration, and financial departments work together in a "Sales" process. Each of the units involved in an organization needs one or more services (e.g. application software or utilities). These services run on IT infrastructure which includes both hardware and software, therefore it must be managed accordingly to meet organizational objectives [3]. Proper management of IT infrastructure will ensure that the required services by business processes are available.
Configuration Management (CM) is a Software Quality Assurance (SQA) process for managing different configurations of configurable software items (Galin, 2003). In addition, it is part of the IT infrastructure which consists of procedures, policies, and documentation. Many items change during a software product's lifetime and it is important to keep track of these changes. Customers may have different software versions so it is important to know which version each customer is using in order to sup-

port them effectively. This will facilitate customer support as for queries it may be necessary to easily access various version of source code, design documents or support documentation. Issues related to poor Configuration Management (CM) include system related failures, failure of key services, deficiency in performance and reduction in employee productivity, all of which consequently can cause serious business impact. In short, CM is a quality enabling process which provides a logical view of services by identifying, maintaining, and verifying the versions as well as the corresponding configuration items [2].

In service-oriented environments the heterogeneity of resources is dealt with by providing any kind of functionality or resource as a service with a stable interface. However this does not completely remove the need for configuration of the resources, which has to be performed by any service provider of an SBA. Software development is a dynamic process where systems are constantly refined and modified [20]. Consequently, as the system evolves, an efficient CM process becomes increasingly important. Software systems are developed individually, but these systems are integrated to gain the benefits of Service Oriented Architecture. This integration to achieve exchange of information gives rise to different management issues. This is because complexity of data exchange increases as the number of services increase. Additionally, CM process activities will getting increasingly complex as the number of services increase, hence causing the CM process itself to be modified regularly.

Software quality assurance is about identifying the right things to implement and test, and allocating and managing resources in a way that minimizes risks when applications and services are deployed [4]. There are two types of quality assurance activities [21]: constructive and analytic quality assurance. The purpose of the constructive quality assurance is to prevent fault injection when artifacts are being created. Analytic quality assurance deals with cleaning artifacts after they have been constructed. In this research, our aim is to support constructive quality assurance, i.e. to prevent defect injection at design time.

In this paper, we present the development of an initial CM framework that can contribute to the end-to-end quality assurance of SBAs. Effective CM will support the effective management of SBA configuration, and this should help to assure their quality. In terms of end to end quality, a CM process would allow developers more accurately develop and update the correct versions of services. SBAs or other service consumers would also benefit from CM as they would get to know when services get updated, allowing them to update accordingly. The remainder of the paper is organized as follows: the remainder of Section 1 describes background information, and Section 2 describes research methodology, the framework, and the example scenario to validate the applicability of the proposed approach, and finally we sum up our conclusions in section 3 .

1.1 S-Cube

S-Cube project is funded by European Community's 7th Framework Programme [10]. Its objective is to create an integrated European research community in the area of software and service engineering. It is based on an ideology that the engineering and management of SBAs is quite different to traditional software applications as they are built by combining different services which may be provided by third parties with whom there should be a service level agreement.

A reference lifecycle for SBAs has been developed by S-Cube project researchers (see Figure 1). It is composed of two cycles. The evolution cycle depicts classical application design while the adaptation cycle reflects the adaptation of SBAs. SBAs need to accommodate many changes at run time and this two cycle approach provides a balance between the design and runtime operation. The Operation and Management phase, where CM resides belongs to both phases. Therefore, it must be efficient and precise enough to meet the transition needs of the entire life cycle. By further defining the CM process within Operation and Management, the research presented in this paper aims to strengthen the S-Cube life cycle.

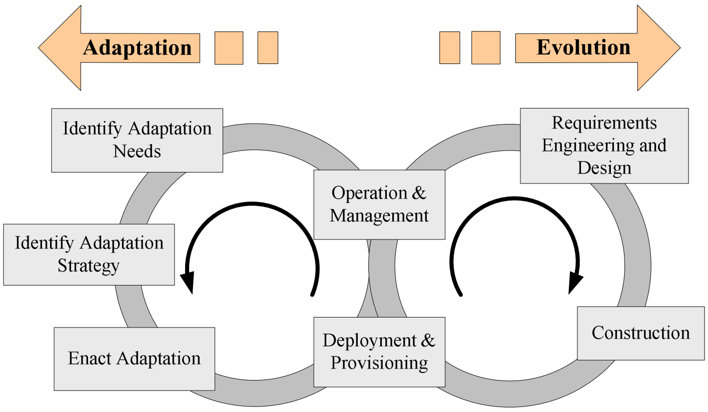


Figure 1: S-Cube Reference Life Cycle [10]

1.2 CMMI-SVC

Capability Maturity Model Integration (CMMI) [5] models are a collection of best practices that help organizations to improve their processes. CMMI - SVC [6] is a CMMI assemblage that covers the activities designed to manage, establish, and deliver services. It has been designed for service industry as a process improvement framework and its goals and practices are relevant to any organization concerned with the delivery of service. CMMI – SVC includes 25 process areas sub-divided into 4 process categories. We used the expert judgment technique [11] to identify process areas and practices which can support CM practices for service based applications.

1.3 Background

In component based development (CBD), software applications can be made up from several standalone components [12]. In SBAs services work as components, and for a good CM, each component included in each application release should be recorded. If the versions of the components are changed then the overall application version should change. This facilitates the quality assurance of the entire software system. In traditional software systems, CM can be achieved successfully if a suitable process guideline or standard is followed. Examples are: IEEE 828:2005, the IEEE standard for software configuration management plans [13] or Leon's guide to software configuration management [14]. Our starting point for the development of the service-based CM framework is the CM process as outlined by Galin [7]. We make use of it because it is comprised of set of configuration activities and their associated action items (see Figure 2). In addition, we map it with the relevant process areas and practices in CMMI-SVC because Galin's model alone did not fullfill the requirements of service based applications. We chose these two models because of their wide use for process management and quality assurance

2 Research Methodology

To develop a service-based CM framework, we make use of a traditional software engineering CM process and supplement it with applicable practices from CMMI-SVC. Galin [7] has defined four levels of CM activities which further contain sets of action items. For each action item, we identified those CMMI-SVC process area(s) and subsequent practice(s) which could support the implementation of that actions item. This one by one mapping of CM supporting action items with relevant CMMI-SVC practices has allowed us to identify an initial service-based CM framework. We then illustrate the implementation of the framework through an example scenarion.

2.1 Initial CM Framework

Galin's four activities are: Software Change Control, Release of Configuration Items (CI) and Software Configuration Versions, Provision of CM Information Services, and Verification of Compliance to SCM Procedures. In our mapping, a process area or practice may be used multiple times to implement different CM activities. Table 1 illustrates the first level of our framework - it displays the mapping between the action items for Software Change Control and appropriate practices from CMMI-SVC and CMMI.

Table 1. Description of CM Framework

Action Items (from Galin)	Relevant CMMI – SVC Process Areas	Corresponding CMMI Practices
Grant approval to carry out changes	• Service System Transition	• Analyze Service System Transition Needs • Prepare Stakeholders for Changes
	• Strategic Service Management	• Gather and Analyse Relevant Data
	• Configuration Management	• Establish change management system
	• Requirements Management	• Manage Requirements Changes • Identify Inconsistencies between Project and Requirements
Control the changes and assure quality of approved changes	• Process and Product Quality Assurance	• Objectively Evaluate Processes • Objectively Evaluate Work Products • Establish Records
Document the approved changes	• Configuration Management	• Establish CM Records • Perform Configuration Audits
Mechanism to prevent simultaneous changes in the same SC item	• Configuration Management	• Track Change Requests • Control Configuration Items

The first CM activity in the framework is Software Change Control. This is an important activity which ensures changes to software systems are carried out with the appropriate levels of governance. This prevents inappropriate or unsafe changes from being made without approval, and becomes particularly important in SOC where changes to services may affect many downstream SBAs. In order to implement the action items, suitable practices were taken from the CM, Requirements Engineering, and Process and Product Quality Assurance process areas of CMMI-SVC and CMMI. The CM and Requirements Engineering process areas provided practices for the steps required to implement software change control, while the Process and Product Quality Assurance process area provided practices for quality assurance during this process.

The second activity in the framework is Release of Software Configuration Items and Software Configuration Versions. When new software versions are released it is important to record version and installation details. This information assists with trouble shooting and diagnosing software errors. With regard to services, the recording of installation sites is not usually an issue as they are usually installed in one location with multiple applications accessing the same services. The release of software configuration items and software versions have differing implications depending on whether services or SBAs are being considered. When new versions of services are released it is important to have access to details of previous versions in the event that they are required. An example would be an incompatibility issue with a service consumer. When SBAs are considered a new application version may be released by adding services or removing services from an existing SBA. Similarly an SBA may require a new version if its component services are updated. In both of these cases configuration details and document version releases should be recorded. Documentation and source code for each release is an important resource for support and quality assurance activities. This activity can be

achieved in SOC using practices from CMMI-SVC activities such as CM, Project Monitoring & Control, and Process and Product Quality Assurance.
The third activity in the framework is Provision of Software Configuration Information Services, it ensures that information about status of changes, versions, and documentation is maintained. Whereas, the fourth activity Verification of Compliance to Software Configuration Procedures deals with verifying compliance to SCM procedures. Our identification of the practices has allowed us to modify the S-Cube life-cycle as shown in Figure 2.

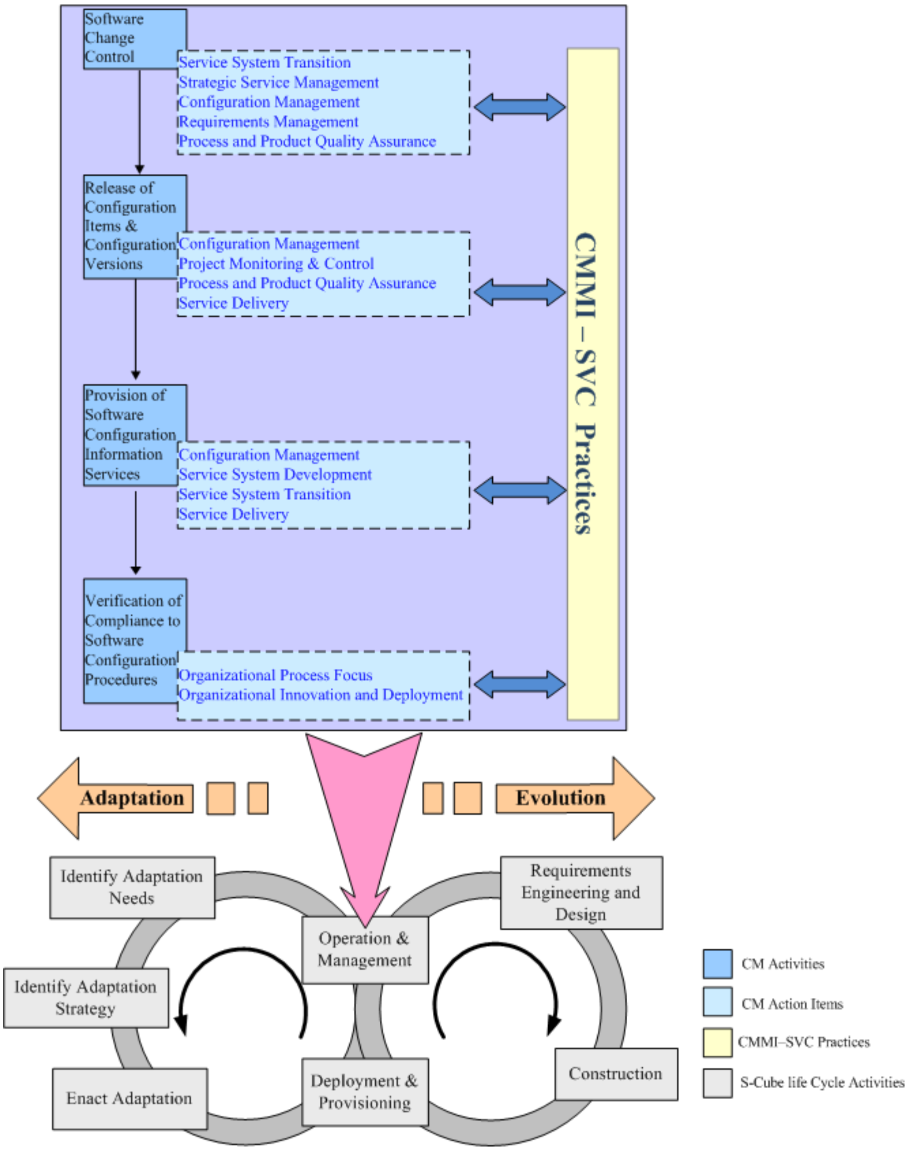


Figure 2: S-Cube Lifecycle with Detailed Operations and Management Phase

2.2 Example Scenario

An example scenario has been designed in the S-Cube project. This is a complex and geographically distributed supply chain in the automotive sector which has been offered by researchers of the companies 360Fresh and IBM [9]. We use this case study to illustrate the possible implementation of our proposed framework. We determined business goals and domain assumptions for the purpose of this illustration. Figure 3 illustrates the global business scope of the service network in the case study. It

highlights the main actors and the interactions, concerning both material and information flow, that occurs between them.
The service network consists of multiple warehouses, scattered across different geographical locations where finished products are stored from the manufacturing factory. If management want to reorganize the current set up (reasons might be due to changing demand patterns or the termination of a lease for a number of existing warehouses), they need to consider their business challenges.

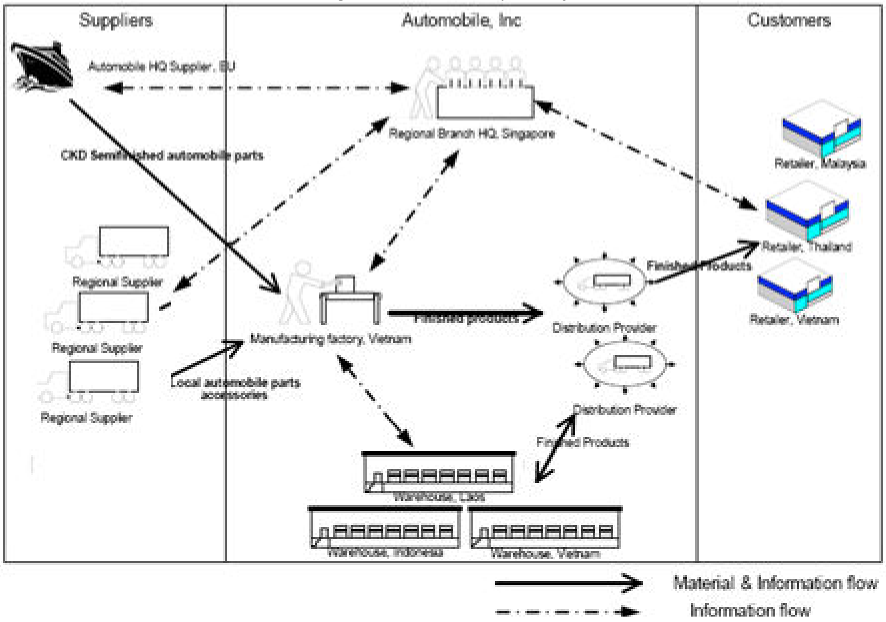


Figure 3: The Business Scope of the Service Network

Reconfiguring a distribution network may require subsequent changes, such as a new flow pattern of goods throughout the network or a change in production levels. They need to consider how to select their new warehouse locations in order to meet changes in demand patterns. The overall purpose of the network is that merchandize is produced and distributed at right quantities, to the right locations, and at the right time, in order to minimize the system wide costs while satisfying service level requirements. For the sake of simplicity we show the warehouses (W_1-W_3) in the network (Figure 4).

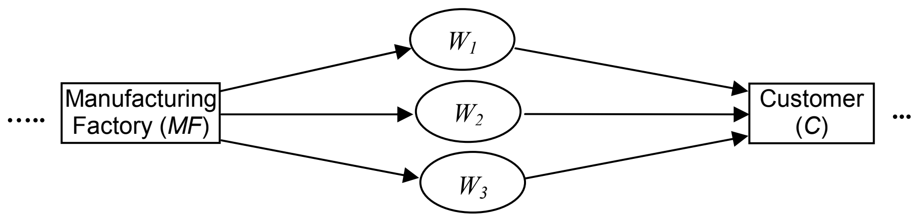


Figure 4: A Simplified View of Service Network

Our objective is to manage the distribution network and reconfigure it based on the optimal strategy when different warehouses are merged into one. The distribution strategy must be able to manage the flow of products from the suppliers through the warehouse to the market areas without interruption. We have three ways to route the finished product to the customer.

$$MF \rightarrow W_1 \rightarrow C$$
$$MF \rightarrow W_2 \rightarrow C$$
$$MF \rightarrow W_3 \rightarrow C$$

Several criteria can be considered to make this decision such as cost, overhead, and distance. However, in a complex logistic network, it is too hard to obtain the optimal path of the network [15] as there are lots of issues involved. For reconfiguration, the supplier evaluation indexes were first presented by

Dickson [16] and Weber [17], and then this index research was expanded [18]. Considering these evaluations and problem indexes allowed us to align them as shown in Figure 5.
The framework we propose in section 2.1 can benefit us in order to address issues associated with fusion of multiple warehouses into a single one. In Table 2, we identified a set of CMMI practices

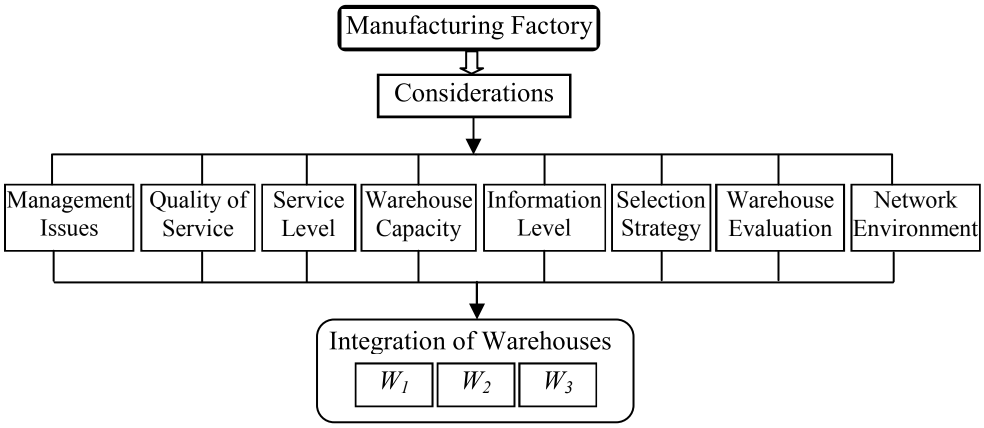


Figure 5: Considerations in Integration of Warehouses

Which can help us to support the CM process, i.e. transition from a multiple into a single warehouse. We may select a set of practices depending on the situation. Thus, our framework has been useful in upporting the business change required.

Table 2: The Service-based CM Framework Supporting Network Modification

Issues	CMMI - SVC Practices	Description
Management Issues	• Analyze Issues • Establish change management system • Prepare stakeholders for Changes	It supports and addresses issues associated with integration
Quality of Service	• Objectively evaluate processes • Objectively Evaluate work products • Conduct Progress Reviews • Select Improvements for Deployment	It supports quality of services by evaluating processes, work products, progress, and improvement deployments
Service Level	• Analyze Service System Transition Needs • Ensure Interface Compatibility • Validate the Service System	Levels of services are addressed by these practices
Warehouse Capacity	• Gather and analyze relevant data • Prepare for service system operations	It addresses assessment of warehousing
Information Level	• Receive and Process Service Requests • Identify configuration Items • Validate the Service System	Information level issues are addresses by related practices
Selection Strategy	• Establish CM Records	Selection strategy analysis is facilitated by CM records
Warehouse Evaluation	• Objectively evaluate processes • Establish Records	Warehouse evaluation is addressed by evaluating current processes and Quality records
Network Environment	• Prepare for Service System Operations • Establish Service Delivery Approach • Deploy Service System Components	Network environment can be readjusted by these practices

3 Conclusions

Services have made the world more connected - allowing producers, consumers, and other human resources to communicate frequently across the globe. The service industry is a significant driver for the growth of worldwide economy. Therefore, guidance on improving service management development can serve as a key contributor to the customer satisfaction, performance, and profitability of the business. In this research, we have proposed a service-based CM framework to manage the configuration of service based applications. The development of the framework is supported by a case which depicts the effectiveness of the approach. A special case with Service Oriented Architecture is that the customer does not see the change of services as long as Service Level Agreements are met. Yet, this is not how it is currently carried out, and therefore remains a future research issue for us. Another issue is that sometimes the providers of services in an SBA do not agree with the SBA provider and this may only be discovered dynamically during execution. We intend to use configuration information for the purpose of audit and for ensuring compliance between them.

4 Acknowledgements

The research leading to these results has received funding from the European Community's Seventh Framework Programme FP7/2007-2013 under grant agreement 215483 (S-Cube). It was supported, in part, by Science Foundation Ireland grant 03/CE2/I303_1 to Lero – the Irish Software Engineering Research Centre (www.lero.ie), and a Higher Education Authority grant PRTLI 4 to the Lero Graduate School in Software Engineering.

Literature

[1] Enterprise Management Associates. "Configuresoft, Inc.'s Enterprise Configuration Manager: A Continuous Compliance Approach to Configuration Management," 2006.
[2] Configuration Management Best-Practice Recommendations, Sun Services White Paper, May 2007
[3] The ITIL Toolkit – The ITIL Guide, ITIL - Office of Government Commerce, The Stationary Office, London, United Kingdom, Typical Page. [http://www.itil-toolkit.com/itil-guide.htm]
[4] Software Quality Management for SOA: Enterprise quality managers take the helm, white paper, Published by Hewlett-Packard
[5] Capability Maturity Model Integration, Technical Report CMU/SEI-2009-TR-001, ESC-TR-2009-001
[6] Capability Maturity Model Integration for Services version 1.2, Technical Report CMU/SEI-2009-TR-001, ESC-TR-2009-001
[7] Software Quality Assurance: From Theory to Implementation, Daniel Galin, Addison Wesley September 21, 2003
[8] Software Services and System Network, www.**s**-cube-network.eu
[9] J. Sairamesh, S. Zeng, B. J. Steele, and M. A. Cohen, "Dynamic service networks: Case study on supply-chain collaboration for early detection and recovery through information services," technical report available from S-Cube on request.
[10]S-Cube deliverables." [Online]. Available: http://www.s-cube-network.eu/results/deliverables
[11] A Guide to the Project Management Body of Knowledge (PMBOK Guide), Project Management Institute; 2000 ed edition
[12] Business component Factory, P. Herzum, O. Sims J. Wiley & Sons Inc., 2000.
[13] IEEE Std 828-2005 - IEEE Standard for Software Configuration Management Plans, IEEE Computer Society, 2005.
[14] A guide to Software Configuration Management, A.Leon, Artech House, Boston, MA, 1999.
[15] Gao Yi; Lin Xinda; Wang Dong;" A computer based strategy design for automobile spare part logistics network optimization", Internet Technology and Secured Transactions, 2009. ICITST 2009. International Conference for
[16] G. W. Dickson, "An Analysis of Vender Selection System and Decisions," Journal of Purchasing, vo1.2, pp.5-17, September 1966.
[17] C. A. Weber, 1. R. Current, and W. C. Benton, "Vendor Selection Criteria and Methods," European Journal of Operational Research, vo1.50, pp.2-18, January 1991.
[18] E. L. Wilson, "The Relative Importance of Supplier Selection Criteria: A Review And Update," International Journal of Purchasing and Materials Management, vo1.25, pp.35-41, May 1994

[19] Wanshan Wang, Liang Tang, Shuang Liang, Junming Hou; “A Study on CBR-Based Supplier Selection under Networked Manufacturing”, Service Operations and Logistics, and Informatics, 2008. IEEE/SOLI 2008. IEEE International Conference on
[20] Nguyen, T.N.; Munson, E.V.; Boyland, J.T.;” An infrastructure for development of object-oriented, multi-level configuration management services”, Software Engineering, 2005. ICSE 2005. Proceedings. 27th International Conference on, Page(s): 215 – 224
[21] Omar Alshathry, Helge Janicke, Hussein Zedan,, Abdullah AlHussein, “QUANTITATIVE QUALITY ASSURANCE APPROACH”, New Trends in Information and Service Science, 2009. NISS '09. International Conference on

Author CVs

Sajid Ibrahim Hashmi

Sajid Ibrahim Hashmi is a doctoral researcher at Lero – the Irish Software Engineering Research Centre, supervised by Dr. Ita Richardson. He holds a M.S in Software Engineering from Korea Advanced Institute of Science and Technology (KAIST). His research interests include software engineering for service-based systems with particular focus on quality assurance and management.

Stephen Lane

Stephen Lane is a doctoral researcher at Lero – the Irish Software Engineering Research Centre, supervised by Dr. Ita Richardson. He holds a M.Tech. in Computer Integrated Manufacturing from the University of Limerick. His research focuses on software engineering processes for service-based systems with a particular interest in adaptability and maintenance.

Dimka Karastoyanova

Dr. Dimka Karastoyanova is an associate professor at the Institute of Architecture of Application Systems at the University of Stuttgart. Her research includes applying the workflow technology in scientific simulation, Web Services (WS) and the related standards and technologies, Business Process Modelling, techniques for enabling adaptability of business processes, in particular BPEL processes, extensibility and parametrization constructs promoting reuse, flexibility and composability of processes, methodologies for creating and executing reusable WS-flows with run time flexibility, toolkit for supporting the automated development of WS-flow definitions, their execution and analysis, transaction processing and workflow management and thier interrelationships, semantic description of WSs and WS-flows and Middleware for Semantic Web Services.

Ita Richardson

Dr. Ita Richardson is the Research Area Leader of Practice, Process and Methods within Lero – the Irish Software Engineering Research Centre and a Senior Lecturer with the Department of Computer Science and Information Systems at the University of Limerick, Ireland. Her research focuses on software quality and software process improvement with an interest in specific environments such as global software engineering, health care and medical devices, small to medium sized enterprises and software services. Much of her research is carried out in industry and in Government-funded services, particularly in Education and Health.

How Green is your Black Belt?

Dick Theisens (1)

Symbol BV
Institutenweg 26, 7521PK Enschede, the Netherlands
www.lssa.eu dick.theisens@symbolbv.nl

Abstract

If your company wants to stay or become more competitive it is important to constantly focus on the main elements Time, Quality and Costs. Applying both Lean and Six Sigma is the right approach for achieving this goal.

Lean Six Sigma is a management philosophy. You find this philosophy explained in many books and articles. The top management decision that Lean Six Sigma is the way to go, is key for success. But after this decision you have to deploy it. Then you will realize implementing Lean Six Sigma is more than a philosophy. It is also about applying the enormous number of tools in the right way. Especially Six Sigma contains many sophisticated analytical and statistical tools. On top of that is the Human Factor that, especially in Lean transformation, is also a key element.

Within Six Sigma employees can be trained at various belt levels. Nowadays these levels are also used to appoint how experienced one is in applying Lean methodology. These levels are called Master Black Belt, Black Belt, Green Belt, Orange Belt and Yellow Belt. Although there are several companies worldwide that train in Lean and Six Sigma, there is no common standard in what elements should be applied within a certain belt. As a consequence the Belt-levels can mean many things. You can train your employees or hire people that call themselves Green Belt or Black Belt, but how do you know this person has the skills you are looking for? How ‘Green’ is your Black Belt?

To meet this problem, the LSSA was established in September 2009. LSSA stands for ‘Lean Six Sigma Academy’. The LSSA main objective is to establish a common European certification standard by developing skill sets, training material and an exam portal. People will be able to apply for a European certificate for the above mentioned levels. Four Skill sets have been derived that exactly describe which of the overall Lean Six Sigma tools are expected to be part at a certain Belt level. The ASQ - Body of knowledge [5], [6] have been taken as a baseline and have been updated according the latest insights.

1 Origins of Lean Manufacturing

The first person to truly integrate an entire production process was Henry Ford by lining up fabrication steps in process sequence, using standardized work and interchangeable parts, which he called Flow production (1913). The problem with Ford's system was its inability to provide variety. The Model T was limited to one color (Black) and it was also limited to one specification so that all Model T chassis were essentially identical up through the end of production in 1926.

Kiichiro Toyoda, Taiichi Ohno, and others at Toyota looked at the Ford's situation in the 1930s, and more intensely just after World War II (1950). While Ford produced 8000 vehicles per day, Toyota had produced 2500 vehicles in 13 years. Toyota wanted to scale up production but faced a lack of financial resources for the huge number of inventory and sub assemblies they saw at the Ford's plant. It occurred to them that a series of simple innovations might make it more possible to provide both continuity in process flow and a wide variety in product offerings. Toyota developed the Toyota Production System (TPS). TPS borrowed Fords ideas but since they couldn't afford the huge inventories Toyota introduced its Just in Time (JIT) philosophy and the 'Pull Concept'.

A detailed description of the Toyota Production System and its 14 principles are described in the book 'The Toyota Way', (2004) [Jeffrey K. Liker, PhD], [1]. The thought process of Lean was thoroughly described in the book 'The Machine That Changed the World '(1990) [2] and in a subsequent volume, 'Lean Thinking' (1996), [James P. Womack and Daniel T. Jones], [3] in which the described the five Lean principles.

1. Customer Value	Specify the value desired by the customer
2. Waste elimination	Identifying and eliminating non-value added activities
3. Continuous flow	Make the product flow continuously
4. Pull instead of Push	Using pull between steps where continuous flow is possible
5. Continuous Improvement	Manage toward perfection

Toyota became in 2008 the world's largest Automaker in terms of overall sales. This continued success has over the past two decades created an enormous demand for greater knowledge about Lean thinking. There are literally hundreds of books and papers and numerous other resources available to this growing audience.

2 Origins of Six Sigma

Six Sigma is a long-term, forward-thinking initiative designed to fundamentally change the way corporations do business. It is first and foremost ‘a business process’ that enables companies to increase profits dramatically by streamlining operations, improving quality, and eliminating defects or mistakes in everything a company does. While traditional quality programs have focused on detecting and correcting defects, Six Sigma encompasses something broader: It provides specific methods to re-create the process so that defects are significantly reduced or even prevented at all [4].

The journey began at Motorola in 1979 when executive Art Sundry's stated in a management meeting, "The real problem at Motorola is that our quality stinks!" Facing stiff competition by Japanese manufacturers, Motorola began its search for ways to eliminate waste in its processes. Two Motorola engineers Bill Smith and Mikel Harry were credited for their pioneering work on defects, incited significant debate within Motorola on the process of finding and fixing defects. (first published in 1985). Their work on process capability, tolerance, critical-to-quality characteristics and design margins laid down much of the foundations of what today is called Six Sigma.

Recognizing a link between fewer defects and lower costs, Motorola set out to incorporate this into their manufacturing processes that was called ‘Six Sigma’. Motorola's Six Sigma quality program was so radical that it forced managers to think about the business differently. Applying these concepts to Motorola's electronics manufacturing delivered more than $2.2 billion in benefits within four years and $16 billion within 15 years. Motorola's CEO Bob Galvin cited the work of Bill Smith and Mikel Harry in achieving these benefits.

One of the companies that embraced the Six Sigma philosophy was General Electric. Jack Welch was told that Six Sigma could have a profound effect on GE quality. Although skeptical at first, the GE Chairman initiated a huge campaign called ‘the GE Way’. He made an official announcement launching the quality initiative at GE's annual gathering of 500 top managers in January 1996. He called the program ‘the biggest opportunity for growth, increased profitability, and individual employee satisfaction in the history of our company’. He has set itself a goal of becoming a Six Sigma quality company producing nearly defect-free products, services, and transactions, by taking quality to a whole new level. Welch intention was to infuse quality in every corner of the company. Layer on he called Six Sigma ‘the most difficult stretch goal’, but also ‘the most important initiative’ GE had ever undertaken. General Electric saved more than $12 billion with Six Sigma in the first five years after implementation.

In the last couple years the Lean and the Six Sigma philosophies are combined to Lean Six Sigma, with a combined set of tools and the common approach of reducing lead time and operational costs, and improving quality.

3 Applying Lean Six Sigma

World Class Performance is about developing and producing products and services that are the best in the world. One can become ‘World Class’ by performing at Operational Excellence and developing new products that exceed customer expectation. Operational Excellence is about continuously supplying products and services at the highest quality at lowest cost and at the right moment. This requires a focus on producing products cost-efficient and without failures.

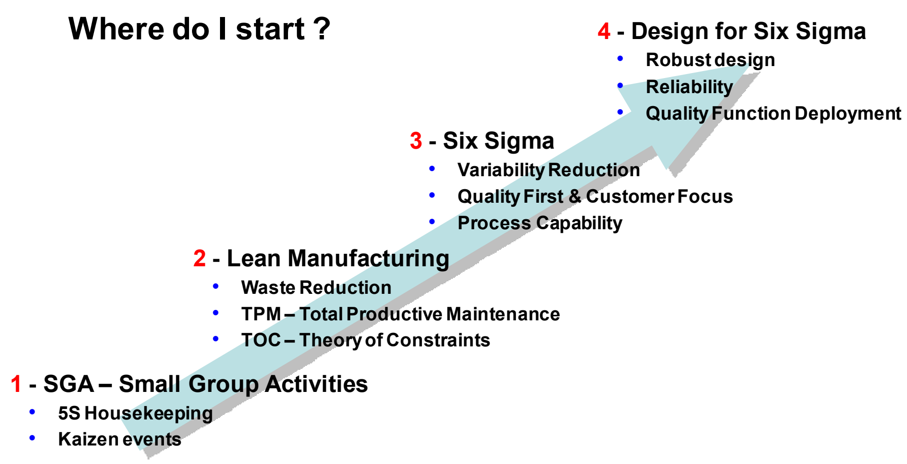
The first level of achieving World Class is organizing the work environment to realize a clear overview of activities and performance. It is also about a professional presentation of the facility. This can be compared by eating in a restaurant. Do you want to eat in an environment which is dirty, disorganized and where the cook has to search for the right ingredients? It is the same for a working environment. Customers expect their products are treated with the outmost care and people are performing the activities in a structured way. Realizing an organized environment can be achieved by applying ‘5S’, which stand for Sort, Straighten, Shine, Standardize and Sustain. An organized environment is the starting point for all improvement activities. You need to get a structured environment first before you employ other methods and tools.

The first level is also about involving operators in quality and problem solving, instead of staff and indirect employees doing this. People at the work floor know very well how problems can be prevented and how operations can be improved. The problem is that very often they are not challenged and supported in these kind of activities. Getting a cooperative organization where the work floor is involved in continuous improvement can be achieved by Kaizen or SGA (Small Group Activities). 5S, Kaizen and SGA focus on the ‘Gemba’ (work floor).

The second level of achieving World Class is implementing Flow and Pull, and stopping operations when problems occur (called Jidoka). At the same time a focus in needed on reducing Waste. The way to realize this is a combination of TPM, TOC and Lean. TPM (Total Productive Maintenance) is a strategy to improve the effectiveness of the environment throughout the company and to reduce unscheduled downtime and quality issues caused by equipment. TOC (Theory of Constraints) is about identifying and eliminating the bottle necks in an organization. Lean Management aims on eliminating waste in every area of production including customer relations, product design, supplier networks and process management. Products and services with excellent quality have to be delivered when the customer wants (Just in time). Examples can be found in assembly operations but also in transactional processes like banking.

The third level of achieving World Class is applying Six Sigma by eliminating variability. Six Sigma is a rigorous and systematic methodology that utilizes information (management by facts) and statistical analysis to measure and improve a company's operational performance by preventing 'defects' and performing breakthrough improvements in solving problems. When procedures, operator training and simple problem solving tools are not adequate to solve a persistent problem, it is time for the Six Sigma approach with sophisticated analytical and statistical tools. For applying Six Sigma it is important to have stable processes. Therefore it is important to focus on an organized work environment and Lean first.

The fourth level of World Class is DfSS (Design for Six Sigma). Both Lean and Six Sigma have a focus on problem solving and prevention in the operations. DfSS has a focus in the development process. The goal of DfSS is to design products that exceed customer expectations; flawless product launch and a predictive reliability. DfSS is a systematic and rigorous methodology using tools, training, and measurements to enable the design of new products and processes that meet customer expectations at Six Sigma quality levels. DfSS has a focus on preventing problems rather than solving them by applying Six Sigma. Examples are found in product development in electronics and automotive industry.

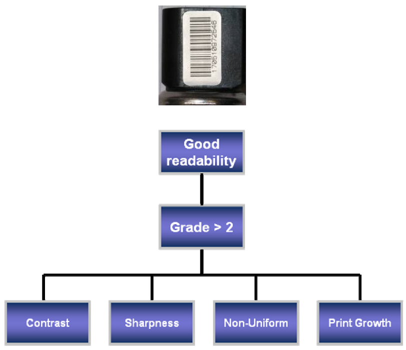


4 *Case description – Laser Coding*

In 2007 an Automotive supplier started a program to change over from labelling their products with a barcode to laser coding. For years their production location faced about 1% of yield loss on this label process which could not be reworked. The problems they faced were labels with bad coding, incorrect position or missing labels. The COPQ (Cost of Poor Quality) was around 300k$ per year.

Laser coding would solve these problems. The issue however was the readability of the laser coding. For almost two years several engineers tried to get a good and stable laser coding quality in place. Since they were not successful for such a long period, management requested the help of a Six Sigma Black Belt to lead this project. The Black Belt started a project that followed the DMAIC roadmap:

D	1 – Project Selection 2 – Project Management
M	3 – Define CTQ & Baseline Performance 4 – Validate Measurement Procedures
A	5 – Diagnose the current process 6 – Identify potential influence factors
I	7 – Define & Implement improvements 8 – Optimize & Verify improvements
C	9 – Improve control system 10 – Close out project

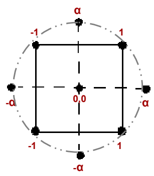


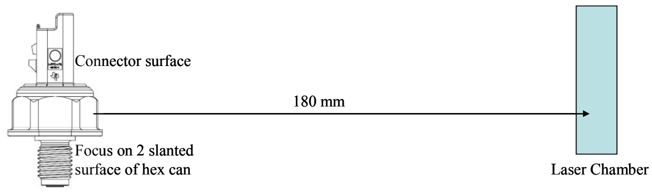
In the 'Measure' phase the first step is to make a CTQ-flowdown that is used to translate the external quality metric from the customer to an internal quality metric that can be measured in production by a label reader. The external CTQ [Good readability] was translated into an internal CTQ [Grade] and detailed out into four internal responses for the label reader [Contrast, Sharpness, Non-Uniformity, Print Growth]. A good performance of each of these four responses will result in a good 'Grade' performance. A good 'Grade' performance will result in a 'Good readability' and a satisfied customer. The internal CTQ 'Grade' and each of the four building blocks had to be measured. The next step then is to perform a Measurement System Analysis on the reader in order to see if the equipment is able to read accurate, consistent and reliable. A Gauge R&R% was conducted on the reader to verify this.

In the 'Analyse' phase a brainstorm session with engineering and the laser supplier was organized to identify potential Factors of Influence that might have an impact on the above mentioned responses. A screening DOE (Design of Experiments) has been performed to select the significant Factors of Influence. The experiment proved that Speed, Current, Frequency and Focal Distance all have an influence. Other potential factors were eliminated.

In the ‘Improve’ phase a second experiment was designed: a Box-Behnken Response Surface Model, which is an optimization experiment. For each of the four Factors of Influences the levels can be found below. Since the Response ‘Grade’ was ordinal, five repetitions were taken for each setting. The Response was calculated as the average for these five repetitions. Minitab software was used to set up the experiment and analyse the results.

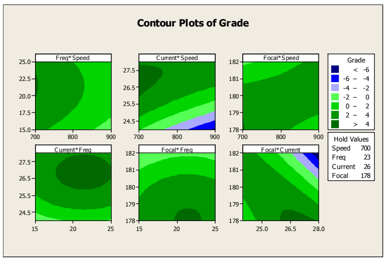
Defined Factors of Influence		**Lo**	**Hi**
• Speed (mm/2)	Time the coding is made	750	850
• Frequency (kHz)	Frequency of the laser	17.5	22.5
• Current (Amp)	Current of the laser	25	27
• Focal (mm)	Distance laser optics and connector	179	181





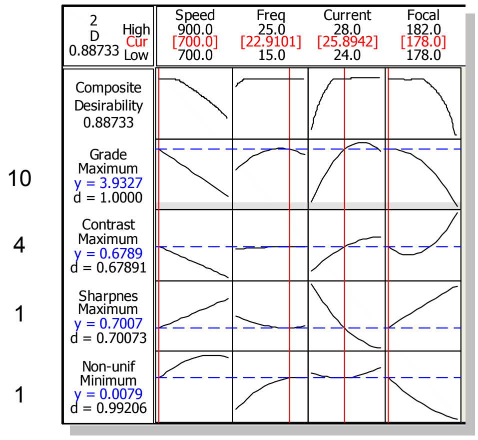
After removing insignificant factors from the model, the Regression Model looked like the below figure (left). Also a contour plot was constructed for the response ‘Grade’ (right). The R-sq (Coefficient of Determination) shows that almost 82% of the total variation can be explained by the model. It means the model is Statistically Significant and conclusions can be derived based on this model.

```
Response Surface Regression: Grade versus Speed, Freq, Current, Focal
The analysis was done using coded units.
Estimated Regression Coefficients for Grade

Term                  Coef  SE Coef       T      P
Constant           2.91206   0.1958  14.873  0.000
Speed             -0.01667   0.1225  -0.136  0.893
Freq               0.28333   0.1225   2.312  0.032
Current           -0.01667   0.1225  -0.136  0.893
Focal             -0.33333   0.1225  -2.720  0.013
Freq*Freq         -0.24362   0.1116  -2.183  0.041
Current*Current   -0.51862   0.1116  -4.646  0.000
Focal*Focal       -0.29362   0.1116  -2.631  0.016
Speed*Current      0.40000   0.1501   2.665  0.015
Speed*Focal        0.40000   0.1501   2.665  0.015
Current*Focal     -0.87500   0.1501  -5.831  0.000

S = 0.600286    PRESS = 27.5484
R-Sq = 81.79%   R-Sq(pred) = 30.38%   R-Sq(adj) = 72.68%
```



The next step in the improve phase was to use the response surface model to determine the optimum settings. Besides determining the optimum settings, there was also a focus to determine robust settings for the response ‘Grade’, in order to assure the readability will be robust over time and won’t be sensitive for small variations from the Factors of Influence.

Using these optimum settings a verification run was performed with 100 devices to determine the capability of the process. For all 100 samples the maximum Grade of ‘4’ was measured. Since the sample run showed no deviation it was not possible to construct a Process Capability Plot, which is normally done in this phase to determine the capability. In this case all samples had the maximum readability performance for ‘Grade’.



Optimize

- When Current is reduced to 26, 'Grade' is getting more robust

Final Optimized Settings

- Speed 700
- Freq 23
- Current 26
- Focal 178

Robust Factors

- Frequency (& Speed)

In the 'Control' phase the process documents were updated and customers were informed on the changed coding. After submitting samples to the customers, they released the new coding and the process. Management appreciated and released the team.

The above mentioned project had actually a very short time line. Within two months the experiments were conducted and the coding process defined. It took almost a year to get customer approval to introduce the new coding. This is an example that changing processes is not only about conducting experiments and statistical analysis, but also about Project Management and Change Management. In some cases applying these Human Dynamics takes more time than applying the experiments itself. A Black Belt is a person that is able to apply both Lean tools, Six Sigma tools and Human dynamics. In this particular project only the Six Sigma tools were applied.

5 LSSA – Lean Six Sigma Academy

Although there are several companies worldwide that train individuals in Lean and Six Sigma, there is no global standard. Some people that followed a four days training and with little experience call themselves Black Belt and some Green Belt trainings do not include statistical analysis. Furthermore theoretical knowledge on methodology and tools is only one aspect. Applying these in real situations is a second aspect which is even more important. As a consequence the Belt-levels can mean many things. You can train your employees or hire people that call themselves Green Belt or Black Belt, but how do you know this person has the skills you are looking for? How 'Green' is the Black Belt that you want to hire?

In the US companies are referring to the standards of the ASQ – 'Body of knowledge' [5], [6]. In Europe there is no such like, although some training agencies refer to the 'Body of knowledge' as well. The shortcoming of the 'Body of knowledge' however is that it focuses on Six Sigma mainly rather than Lean. Most companies are not on the level of Six Sigma yet, so knowing how to apply Lean is important as well.

To meet these problems the LSSA was established in September 2009. LSSA stands for 'Lean Six Sigma Academy' and was raised by Symbol BV (the Netherlands), ROC of Twente (the Netherlands), Institute Polytechnique de Grenoble (France), I.S.C.N. GesmbH (Austria), University of Twente (the Netherlands) and EMIRAcle (Belgium). Detailed information can be found at www.lssa.eu. The LSSA main objective is to establish a common European certification standard by developing skill sets, training material and an exam portal. People will be able to apply for a European certificate for each of the four belt levels.

In Six Sigma one can be trained at a certain level, but generally spoken engineers are trained at a Green Belt level. Team members and work floor are trained at a Yellow Belt or Orange Belt level and Process Improvement project managers and senior engineers are trained at a Black Belt level. A Belt level is called a ‘Job Role’ within the domain ‘Process Improvement’.

Job Role	Vocational Education training	Adult Education / Function
Yellow Belt	Initial VET secondary level	Team member, Operator
Orange Belt	-	Team member, Lean Facilitator, Supervisor
Green Belt	Higher Education	Engineer (Quality / Process / Design), Process owner
Black Belt	-	Senior Engineer, Project Manager, Management, Consultant

For each of the four Job roles Skill sets have been derived that exactly describes which of the overall Lean Six Sigma tools are expected to be part at a certain Belt level. A skill set is a set of ‘Learning Elements’ within eight ‘Units’. The ‘ASQ - Body of knowledge’ for a Green Belt [5] and a Black Belt [6] have been taken as a baseline, and have been updated according the latest insights and Lean methodology. On top of the Green Belt and Black Belt, skill sets for the Orange Belt and the Yellow Belt have been defined. Another difference between the ‘ASQ - Body of knowledge’ and the ‘LSSA Skill sets’ is the structure. Each of the four skill sets within the LSSA has the same structure. The benefit is that you can easily compare what the difference is between a Yellow Belt and a Black Belt in the level of applying a certain set of tools (performance criteria).

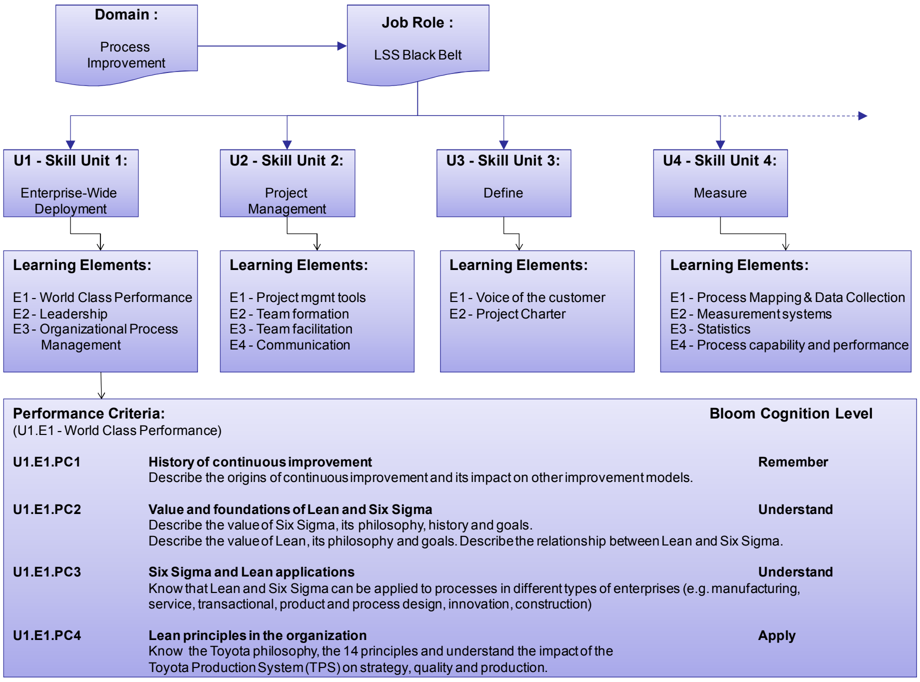
The Skill sets also mention how each Job role refers to the qualification levels within the NVQ (National Vocational Qualification standard) and the EQF (European Qualifications Framework). The NVQ is defined by European legislation and is used for comparability of vocational qualifications from different European countries. The EQF acts as a translation device to make national qualifications more readable across Europe, promoting workers' and learners' mobility between countries and facilitating their lifelong learning.

The structure consists of eight ‘Units’. Each of these units is built up with a number of ‘Learning Elements’ that contains several ‘Performance Criteria’. Each of the ‘Performance Criteria’ has a description and a cognitive level according Bloom [7] at which it should be applied.

Unit	Learning Element
U1 - Enterprise Wide Deployment	E1 - World Class Performance
	E2 - Leadership
	E3 - Organizational Process Management
U2 - Project Management	E1 - Project management tools
	E2 - Team formation
	E3 - Team facilitation
	E4 - Communication
U3 – Define	E1 - Voice of the customer (VOC)
	E2 - Project charter
U4 – Measure	E1 - Process Mapping & Data Collection
	E2 - Statistics
	E3 - Measurement systems
	E4 - Process capability and performance
U5 – Analyze	E1 - Exploratory data analysis
	E2 - Hypothesis testing
	E3 - Analytical methods
U6 – Improve	E1 - Design of Experiments (DOE)
	E2 - Waste elimination
	E3 - Process Improvement Methods
U7 – Control	E1 - Statistical process control (SPC)
	E2 - Quality Assurance
	E3 - Sustain improvements
U8 - Design for Six Sigma (DFSS)	E1 - DFSS methodologies & Roadmap

List of Units and Learning Elements

The number of Performance Criteria, the description and the cognitive level are different for the four Belt levels. The number of Performance Criteria for a Black Belt is 117 and for a Yellow Belt 50. Below an example is given for the first Unit ‘Enterprise Wide Deployment’ and the first Learning Element ‘World Class Performance’.



Sample of the Lean Six Sigma Black Belt skill set.

Experienced Black Belts across Europe have been involved to define and review the Skill sets. Training material has been developed together with Initial VET, Universities and experienced Master Black Belts from Lean Six Sigma training institutes.

An exam portal will be set up in accordance with ECQA (European Certification & Qualification Association). The ECQA has been established through former LLP funded networks and currently certifies several professions in Europe [8]. Since the ability to apply the theory in practice is an important element of Lean Six Sigma, a panel will assess the projects. People that are trained can apply for examination and recognized with a Black Belt, Green Belt, Orange Belt or Yellow Belt certificate. It will also be possible to test people that are already trained and are working on process improvement projects in order to make companies more competitive.

6 Conclusions

More and more companies see the benefits of having employees trained in Lean Six Sigma in order to reduce Lead time, improve Quality of products and services and reduce operational costs. Applying Lean Six Sigma will help companies to stay or become more competitive.

Identifying at what level of expertise one is trained is difficult since people are trained by different agencies and companies. Theoretical training is not a guarantee for successful applying Lean and Six Sigma in real practise.

The LSSA skills sets clearly describe what elements and performance criteria are needed at a certain belt level. Assess employees through the ECQA framework can clarify the level of expertise in applying Lean and Six Sigma.

Abstract and keywords

5S	Sort, Straighten, Shine, Standardize and Sustain
ASQ	America Society for Quality
DfSS	Design for Six Sigma
ECQA	European Certification & Qualification Association
EQF	European Qualifications Framework
JIT	Just In Time
LSS	Lean Six Sigma
LSSA	Lean Six Sigma Academy
SGA	Small Group Activities
TOC	Theory Of Constraints
TPM	Total Productive Maintenance
TPS	Toyota Production System
VET	Vocational Education training
NVQ	National Vocational Qualification standard (England, Wales & N. Ireland)

YB	Yellow Belt
OB	Orange Belt
GB	Green Belt
BB	Black Belt
MBB	Master Black Belt

References and further Reading

1. The Toyota Way (Jeffrey K. Liker, 2004)
2. The machine that changed the world (James P. Womack & Daniel T. Jones, 1990)
3. Lean Thinking (James P. Womack & Daniel T. Jones, 1996)
4. Implementing Six Sigma, Second Edition (Forrest W. Breyfogle, 2003)
5. American Society for Quality, Body of knowledge Black Belt (2005-2008)
6. American Society for Quality, Body of knowledge Green Belt (2005-2008)
7. Bloom's Taxonomy Revised (2001). In *Wikipedia*. Retrieved, April 2010 in

   http://projects.coe.uga.edu/epltt/index.php?title=Bloom%27s_Taxonomy
8. ECQA Guide: ECQA European Certification and Qualification Association Guide, in www.ecqa.org

Authors CV

ir. Dick Theisens,

General Manager and Master Black Belt at Symbol BV, the Netherlands. Symbol is a training institute and consultancy company that is specialized in Lean transformation and Six Sigma deployment. In the past years Dick Theisens successfully applied both Lean and Six Sigma in a number of companies in Malaysia, France, the Netherlands and Mexico. In this role he solved many yield programs and quality complaints using Six Sigma tools. He was also a member of the management team in a Mexican production plant that achieved 22% output improvement in an environment that was already Lean, using Theory of Constraint principles. Between 2007 and 2009 he was a Six Sigma deployment leader coaching management and teaching Green- and Black Belts in executing improvement programs. Nowadays he is a management coach, supporting companies with improving Quality and Lead Time. He is also a teacher Lean Six Sigma at the University of Twente (the Netherlands) and chairman of the LSSA – Lean Six Sigma Academy.

Drifting SPI: Studying Practice

Gitte Tjørnehøj

Aarhus University, Department of Management,

Haslegårdsvej 10, 8210 Århus, Denmark

gittet@asb.dk

Abstract

The SPI field has formed around CMM and is still dominated by the underlying rational ideal of system development. A gap between this ideal and the practice of software firms today is reported, implying a risk of investments in unsuitable SPI. This article discuses SPI practice, based on a longitudinal case study of a small software firm's adoption of SPI. The main result is that SPI practice is characterized by drifting. Plans are made, control is exercised, but SPI drifts in unpredictable directions anyhow. This article characterizes drifting SPI practice, discusses how the dominant SPI theory is challenged and suggests alternative views and practical advice.

Keywords

Software process improvement, practice, CMM, control, drift, improvisation, longitudinal case

1 Introduction

The field of *software process improvement* (SPI) formed around CMM [1]. Even though the models of the field have developed since then the underlying assumptions still prevail and [2] suggest that they may not be appropriate for modern IT-industries. Reviewing the literature of the field revealed a possible gap between how SPI is described in general and the SPI practice of software firms.

If such a gap exists, the field risks missing the target when offering advice to management and even worse firms risk wasting resources investing in unsuitable SPI. To shed light on this problem SPI practice was investigated, through an interpretive longitudinal case study [3,4] of ten years of SPI efforts in a small software firm. Their SPI practice was studied as an ongoing change process, pushing the concrete SPI models in the background.

The underlying results of this study has been reported in detail elsewhere [5,6]. This article sums-up the findings to characterize SPI practice and discuss challenges to the dominating SPI theory. Alternative views and practical advice are offered.

2 The research approach

This longitudinal, interpretive case study [3,4] is based on ten years collaboration [7] with a small firm allowing us to gather a rich [4] collection of data from many different sources. The data was analyzed iteratively. First we described an analytical chronology of SPI events providing an overview and serving as interview guide. Then we identified important encounters and episodes [8] to form the backbone of the case story. Episodes are relatively stable periods of evolution that are punctuated by compact periods of revolutionary events called encounters. We applied actor network theory [9] to explore how interests and actants influenced the change processes.

These two orthogonal analyses served as a basic data analysis for two interpretations of the case digging deeper into different aspect of SPI practice. First the concepts of control and drift by Ciborra [10] explained the case [5] and second the role of organizational improvisation [11] in the change processes was addressed[6].

This article is the result of further analysis returning to both the raw data and the basic analysis looking for characteristic of drifting SPI practice. The chosen characteristic were all very evident in the case and target fundamental aspect of SPI. Since the aim was to enlighten the supposed gap between theories and practice, challenges of the dominating SPI theories were systematically mapped and alternatives and practical advice were suggested.

3 The SPI practice of SmallSoft

This section summarizes the case story of adopting SPI in *SmallSoft*. For more detailed descriptions and profound argumentation see [5,6].

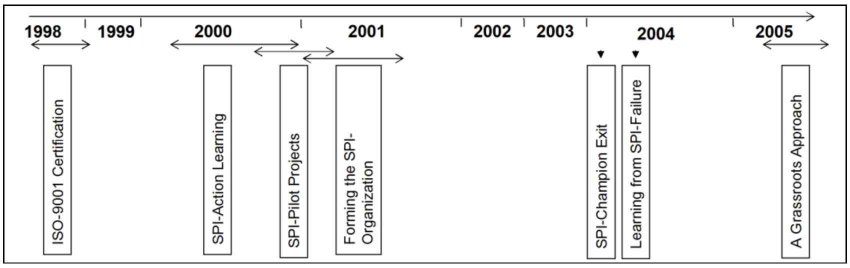
3.1 Ten years of adoption of SPI

SmallSoft has approximately 50 employees in three departments. Their core competence is to combine domain specific engineering knowledge with IT-competencies when developing software solutions. The firm culture displays a low risk attitude and a belief in core engineering skills and long term personal customer relations. The employees are often working alone or in very small teams in close connection with customers.

The ten years involved three main SPI efforts utilizing different SPI technologies. Up to 1999 they designed and implemented a QA-system achieving an ISO9001-certificat. In autumn 2000 they were introduced to CMM and they established of a formal SPI organization in 2001. This centralized organization proved mal functional and in 2004 *SmallSoft* eventually decided on a grassroots approach to SPI.

Some of the encounters we chose as important correspond directly with the launching of these efforts, but also the breakdowns and the learning was noticeable (see figure 1). The encounter “ISO-9001 certification” was well carried out resulting in a certificate and an online quality handbook. But the QA system was soon ignored during the following episode when sales unexpectedly boomed.

The next SPI effort covers the three encounters; “SPI action learning”, “SPI pilot projects” and “Forming the SPI organization”. The local university invited the firm to participate in educational activities on software management. As part of the education an internal assessment (CMM level two) and SPI pilots was conducted. Finally the CMM-like organization was staffed and top management proclaimed the goal “level three in three years” (2001). In the following episode the process group operated orderly for a short period of time pushed by a SPI champion, before it turned inactive in autumn 2001. From that point of time all SPI activity seemed to vanish.



	ISO-9001 Certification	SPI-Action Learning	SPI-Pilot Projects	Forming the SPI-Organization	SPI-Champion Exit	Learning from SPI-Failure	A Grassroots Approach
SPI-activi-ty	Design-ing ISO-9001 QA-system	Action learning education in project planning and SPI	SPI-assessment, -planning and process design. SPI-pilots	Designing and forming new centralized SPI-group	Involvement in an action research project that lead to a short period of hectic SPI-activity before *SmallSoft* has to concentrate all energy on sales activities. This is followed by no or sparse activity	Management realize SPI-breakdown and starts reflecting on present and past improvement practice	A cross department PIT-organization is kicked off and the first PIT's start working
SPI-orga-niza-tion	Ad-hoc repre-senta-tive QA-group of two employ-ees and one mana-ger.	3 key employ-ees appointed to the education. A manager as internal sponsor.	3 SPI-agents. 2 production projects as pilots. Manager as sponsor. Top manage-ment and the supervisor are members of the steering-committee.	A new centralized SPI-group appointed. Two of the key employees and the manager sponsor new members. SPI-initiatives should be carried out in ad-hoc PIT-groups.	For a short period the central SPI-group worked organized as before, but extended with the researchers as consultants, then the SPI-organization in reality collapsed.	SPI-sponsor manager picks up the parts. Suggest a new grassroots organization.	Permanent PIT organization. All employees participate in an improvement team. The SPI-sponsor manager is still sponsor.
SPI-result	Successful ISO-9000 certification & implementation of the new configuration-management – tool, followed by failure of implementation of the QA-system	Successful learning as new knowledge and energy was feed into the firm. No change of practices	Successful CMM-assess-ment, knowledge gained of appropriate change, new processes designed and tested	Successful design and decision on a new SPI organization Members were appointed to the SPI-group and starts working.	Eventually the SPI-champion leaves the firm partly because of the failure of the SPI-initiatives.The episode leading to this showed: No changes of practice from the SPI-initiatives (The SPI-group did not meet). Involvement with researchers gave some results, but was not utilized. The economic situation contributed to a total breakdown in the SPI-effort.	The failure was followed by success as the experience lead to renewed ideas of SPI (grassroots) and renewed collaboration with researchers.	Successful and enthusiastic design and implementation of a new grassroots SPI-organization. It has started working. May or may not bring success and changes og practice around.

Fig. 1. At the top is the historical mapping of the encounters of *SmallSoft's* adoption of SPI. The table below describes the encounters in terms of SPI activity, organization and results.

Early 2003 *SmallSoft* attempted to revitalize the improvements by engaging in an action research project on SPI, but declining markets and downsizing in spring 2004 stopped the efforts and the SPI champion decided to leave the firm ("SPI champion exit"). Reflecting on the breakdown ("learning from SPI failure"), the SPI manager found three reasons for the complete failure of the centralized SPI organization: lack of time in the process group, neglect from everybody else, and the need to maintain monthly economic surplus, limiting investments in innovation. He recognized how the production was more important than SPI and reflected on how several successful improvements had been driven by production needs, especially before the formation of the process group.

Inspired by this, *SmallSoft* applied a grassroots approach hoping build ties across the firm to facilitate learning and knowledge sharing. Most employees were involved in process improvement teams (PITs) in a matrix-like SPI organization. Three month after the launching in 2005, three out of eight planned PITs were active involving eighteen developers and parts of management in an improvement network that seemed stronger than ever during adoption of SPI in *SmallSoft*.

In summary even though plans were made and control was exercised the SPI was drifting[10] in unpredictable but also beneficial directions, often shaped through improvisations following unpredictable and rapid changes in the environment.

4 Characterizing SPI practice

Five characteristic of drifting SPI-practice were identified (see table 1 column 2). The three first outline important conditions, while the two last address acting in drifting SPI practice. Together they outline a profound understanding of SPI practice based on empirical findings. Each row forms an important argument starting from the empirical findings ending with the practical advice (Sections 4.1 - 4.5 presents the arguments).

Findings	SPI practice	SPI theory	Alternatives	Practical advice
• PN[1)] dominated IN[2)] • PN drove successful improvements	**IN is inherently dependent on PN**	Separate SPI organization	Align IN with PN (drive SPI by the strongest network + adapts improve-ments to reality)	• Problem driven SPI • Participative SPI • Integrated SPI
Challenges and possibilities offered from the environment was important	**Sensitive to dynamic environments**	Static SPI strategy	Embracing the dynamic as an advantage	• Short-term changes and evaluations • Utilize dynamic environments continuously
• Micro changes • A row of failures or accumulated learning	**Longitudinal**	Disciplined efforts towards a coherent improvement framework	A web of learning, actions and artifacts laying the foundations for all improvements	• Not only goal driven • Continuous knowledge sharing and learning
• Both control and drift is beneficial • They interact, balance and moderate each other	**Shaped between control and drift**	• SPI theory is strong in control. Drift is at most addressed as failures • Control and drift are irreconcilable	Negotiate adoption of SPI technology between control and drift	• Traditional SPI set direction and push • Be open to *backtalk (drift)* • Cultivate an ability to take advantage of drifting
Improvisation ensures adaptation of improvements and exploitation of possibilities given	**Improvisa-tional**	"If process improvement is not rigorously planned and tracked, it will not happen" (Humphrey 1998)	Take advantage of the improvisational power	• Improvisational culture • Minimal structures and leadership • Procedures are advisory

1) PN "means production network" 2) IN "means improvement network"

Table 1. Summarizes the findings, results and discussion of this study. Column 1 lists the empirical findings as a basis for the characteristic of drifting SPI practice (column 2). The main problems of the dominating SPI theories are listed in column 3 while alternatives are suggested in column 4. Column. 5 provides practice advice.

4.1 Dependent on the production network

The actor network analysis revealed two persistently co-existing networks;

> "…the relatively stable and powerful *production*-network in which managers and software developers across *SmallSoft*'s three departments developed new solutions in response to customer requests." [5, p. 75] and "…the less stable and weaker *improvement*-network through which a small group of different actors over time attempted to improve practices in the production-network through the adoption of new development technologies." [5, p. 75]

The two interacted in a wave like pattern in which the strength of the improvement network fluctuated following the state of the production network. The production network provided powerful feedback, helped focus the improvements on pertinent needs and drove the most successful improvements.

The improvement network thus seems to be inherently dependent on and to benefit from being aligned and integrated with the production network.

In contrast to this, SPI theory advises that SPI should be organized separately from the production. The process group (the SEPG) [1, p. 287] of dedicated change agents initiate, design and drive the improvements and results are measured by an external norm not by the effect on software practice. The detachment constrains potential synergy from shared activities, aligned interests and employees feeling ownership.

Aligning [9] the improvement network with the production network will allow SPI to be fueled by the most powerful network of the organization. This could ease the lack of resources and help ensure that planned improvements fit the firm's reality. It involves acting in the interests of the production network for example as problem driven SPI [12], furthering cross-network activities and knowledge sharing [13] and basing the improvements efforts in extensive user participation [14] or even integrate SPI in the system development practice.

4.2 Sensitive to dynamic environments

The dynamic environments of *SmallSoft* offered both challenges and opportunities that proved important during the adoption. Examples are how the unexpected market fluctuations reduced the ability to invest in SPI both when they had no and too many orders and how the opportunity of action learning eventually changed their SPI strategy. To a large extend *SmallSoft* reacted to and utilized the dynamics of the environment through a flexible, improvisational behavior that allowed for adapted and useful solutions.

Thus drifting SPI practice seems to be characterized by sensitivity to dynamic environments.

Norm-driven approaches to SPI all promote a static strategy [15] of achieving compliance with rather inflexible norms. This constrains the improvement activity and hinders exploitation of a dynamic environment. The norms themselves evolve slowly and the commercial SPI community around them reinforces this. Since environments tend to be increasingly dynamic the misfit is likely to grow, escalating the risk of unfeasible improvements though successful measured by the norms.

Embracing this dynamic as an advantage will allow SPI efforts to be fitted to the actual situation benefitting from possibilities offered from outside. Some resemblance to the agile trend in software development [16] is obvious. Embracing will involve short-term SPI efforts and evaluations according to firm reality. More flexible approaches to SPI is suggested by [17].

4.3 Longitudinal

A web of events, individual and collective learning, new personal practices and tools, communications, discussions, artifacts and management actions over time and across the efforts moved *SmallSoft* forward. One key employee emphasized that, even if the improvement efforts one by one could be evaluated as SPI failures or very short-lived successes, the improvements in the organization were noticeable. In line with the idea of history actively shaping the present and future [4, p. 270] even insignificant or ignored incidents continued to impact future improvements.

Drifting SPI practice is in this way longitudinal through the continued web of leaning, actions and artifacts that leads to the improvements.

One of the key principles of the dominant SPI theory is that improvement is continuous [1, p. 19]. Since people and problems are in a constant flux however, Humphrey suggests disciplined periodic adjustments of task and relations [1, p. 20] in stable periods to achieve an orderly coherent improvement framework. Bits and pieces [1p. 21] do not count.

Longitudinal SPI as suggested above is a rich and realistic way thinking of ongoing SPI practice. Planning and evaluating SPI efforts rationally according to norms introduce the risk of ditching efforts that actually contribute, because of ignorance. To avoid this, organizations needs to focus on the continuous learning [18], knowledge sharing [19] and other social aspects of organizations [10,20].

4.4 Shaped between control and drift

The adoption of SPI in *SmallSoft* was shaped between control and drift [5]. The control elements used, helped management to set the direction, keep the firm vigilant by pushing the organization towards change, and facilitate knowledge sharing. Drifting at the same time exploited human creativity and innovativeness when adapting of the efforts to the firm's realities and when handling lack of resources and knowledge. Everyday coping, bricolage and improvisation by employees were important. Through their interaction control and drift balanced and moderated each other to secure a unique solution for *SmallSoft*. When control elements balance the drift elements, they ensure the effectiveness and efficiency of the practice. When drift elements balance the control elements they ensure adaptation of the models, plans and technologies to the real life of the organization.

Thus drifting SPI practice can beneficially be negotiated between control and drift.

The dominant SPI theories provide profound knowledge on how to utilize control. Drift is only addressed as explanations of failed improvement efforts. Drift theory [10] describes how improvisation, bricolage, hacking, and people's everyday coping in a complex and unpredictable world makes reality drift away from plans.

Management risks that the production network will petrify in an inappropriate practice, if they do not act to push the adoption of SPI through control elements. However, if they insist on a pure control approach without being open to the backtalk from the situation (drift), they will miss the full potential offered by the situation. They need actively to cultivate the organization's drifting abilities.

4.5 Improvisational

SPI practice at all levels of *SmallSoft* was to a large extent improvisational [11]. Improvisation helped employees perform despite scarce resources and allowed the firm to take advantage of opportunities offered from outside. The improvisational culture in some cases led to over-improvisation or even improvisation that was not in the interests of the firm. In all the improvisational culture was a great strength in the dynamic environment when addressing appropriate challenges and when supported by appropriate leadership and coordination. [6].

Appropriate improvisational action and abilities seems to be important for innovation in drifting SPI practice.

In the dominant SPI theories orderly planning is an immensely important principle. The CMM describes how substantial procedural memory is stepwise installed until all software processes are defined and measured, thereby systematically diminishing the conditions for improvisational action; experimental culture, a minimal structure and a low procedural memory [11].

To take advantage of the power of improvisation, an organization needs to consciously cultivate its improvisational competence. They must grow an experimental culture, implement leadership and minimal structures to support and coordinate the improvisational actions and treat procedural memory as advisory [11,21].

5 Conclusion

This article reports from ten years of SPI practice in a small firm. The main result is that SPI practice is drifting [10]. Plans are made, control is exercised, but SPI drifts in unpredictable, but also beneficial directions anyhow. Five characteristics of the practice were identified through this case study of a rather typical small software firm. The further argument forms a critique of the dominant SPI theories for their ignorance of the drifting elements of SPI practice. Support for these empirical based results is to be found in the literature. Authors criticize CMM for being too rational to deal with the complex and massive changes that it imposes on organizations [22], for resting on an old-fashioned managerial assumption platform not suited for modern organizations [2], and for being unable to plan for the unexpected [21].

For SPI practitioners these results means that they need to negotiate control and drift in their SPI efforts, re-interpreting the SPI theories into their special situation as described above. SPI research needs to refocus in order to validate and extend the understanding of SPI as drifting practice and to explore radical new ways of improving software practice “between control and drift”. The goal is situated, flexible and adaptive SPI approaches.

References

1. Humphrey, W.: Managing the Software Process. Addison-Wesley, Reading, MA (1989)
2. Rose, J., I. Aaen, et al.: Managerial and organizational Assumptions in the CMMs. In: P. A. Nielsen and K. Kautz (eds.) Beyond Conventional Software Process Improvement Software Innovation Publisher, Aalborg (2008).
3. Walsham, G.: Interpretive case studies in IS research: nature and method. European Journal of Information Systems 4, (2), 74-81 (1995)
4. Pettigrew, A. M.: Longitudinal Field Research on Change:Theory and Practice. Organization Science 1, (3), 267-292 (1990)
5. Tjornehoj, G. and L. Mathiassen: Between control and drift: negotiating improvement in a small software firm. Information Technology & People 21, (1), 69-90 (2008)
6. Tjornehoj, G. and L. Mathiassen: Improvisation during process-technology adoption: a longitudinal study of a software firm. Journal of Information technology 25, 20-34 (2010)
7. Mathiassen, L.: Collaborative Practice Research. Information, Technology & People 15, (4), 321-345 (2002)
8. Newman, M. and D. Robey: A Social Process Model of User-Analyst Relationship. MIS Quarterly 16, (2), 249-266 (1992)
9. Latour, B.: Science in Action: How to follow Scientists and Engineers through Society. Harvard University Press, Cambridge, MA (1987)
10. Ciborra, C.: The Labyrinths of Information: Challenging the Wisdom of Systems. Oxford University Press, New York (2002)
11. Cunha, M. P., J. V. Cunha, et al.: Organizational improvisation: what, when, how and why. International Journal of Management Reviews 1, (3), 299-341 (1999)
12. Iversen, J., P. A. Nielsen, et al.: Problem Diagnosis in Software Process Improvement. In: International Federation of Information Processing (IFIP).(1998)
13. Nielsen, P. A. and G. Tjørnehøj: Social Networks in Software Process Improvement. Software Process: Improvement and Practice online, (2009)
14. Aaen, I.: Challenging software process improvement by design. In: the 10th European Conference on Information Systems, 379-390.(2002)
15. Arent, J.: Normative Software Process Improvement. Department of Computer Science. Aalborg University. (2000).
16. Beck, K., M. Beedle, et al.: Manifesto for Agile Software Development, http://agilemanifesto.org/. (2001).
17. Aaen, I., A. Börjesson, et al.: SPI Agility: How to Navigate Improvement Projects. Software Process: Improvement and Practice 12, (3), 267-281 (2007)
18. Fichman, R. G. and C. F. Kemere: Assimilation of Software Process Innovations: An Organizational Learning Perspective. Management Science 43, (10), 1345-1363 (1997)
19. Mathiassen, L. and P. Pourkomeylian: Managing Knowledge in a Software Organization. Journal of Knowledge Management 7, (2), 63-80 (2003)
20. Nielsen, P. A. and J. Nørbjerg: Software Process Maturity and Organizational Politics. In: B. Fitzgerald and N. Russo (eds.) Realigning Research and Practice in Information Systems Development: The Social and Organizational Perspective, Proceedings of IFIP WG 8.2 Conference, Boise, Idaho (2001).
21. Aaen, I.: Software Process Improvement: Blueprints versus Recipes. IEEE Software 20, (5), 86-93 (2003)

22. Ngwenyama, O. and P. A. Nielsen: Competing Values in Software Process Improvement: An Assumption Analysis of CMM from an Organizational Culture Perspective. IEEE Transactions on Engineering Management 50, (1), 100-112 (2003)

Process Assets Libraries as knowledge repositories for learning improvement: an experience with agile processes

Antonio Amescua, Leonardo Bermón, Javier García and María-Isabel Sánchez

Abstract

The aim of this work is the definition of a set of guidelines to develop knowledge based PALs to be used in software projects following software engineering best practices. The intention is twofold; in one hand we want to demonstrate that the learning process can be improved using a PAL to transfer the software engineering knowledge, and on the other hand we want to demonstrate that software engineers can be more independent while developing their assigned tasks because the PAL can be accessed always that a piece of knowledge is required. The solution has been implemented and validated in the field of agile methods used in software projects development.

Keywords

Software Engineering, Software Process Technology, Knowledge Management, Web 2.0, Wikis, Agile Development.

1 Introduction

This paper focuses on the use of wikis as Web 2.0 knowledge repositories to help software engineers to find and reuse process assets in projects improving in consequence the learnability and ease of use of software processes.

The software process is a coherent set of policies, organizational structures, technologies, procedures and artefacts that are required to design, develop, install and maintain software products [1][2]. Software process users are not always motivated to do the process if they do not have any supporting technology to create, store and search knowledge in order to develop their activities [3][4]. Therefore, tools are required to provide different support levels for learning and use of processes [1].

One of such tools is a Process Asset Library - PAL. PAL is a repository of information used to keep and make available all process assets that are useful to those who are defining, implementing, and managing processes in the organization [5][6]. However, for an effective use of a PAL, it is necessary to develop domain-specific process assets that can be easily stored and updated [7]. Moreover, it is required to capture the diverse forms of knowledge gained from projects [8]. It is difficult to ensure that new processes are adopted in organizations because are conditioned by the relationships that exist between people and artefacts [9].

The PAL must provide training and learning capabilities to develop collective skills, the motivation to use process and effective communication between teams [10]. Nevertheless, the number of knowledge repositories about agile development process is limited mainly to repositories of documents [11]. In this regard, the challenge is multiple: First, we define the mechanisms for knowledge management that allow a PAL to improve learning and usage of the software process. Then, we apply the theoretical guidance defined on repositories of agile development processes.

To meet these challenges, we have defined regarding the use of a knowledge management-based PAL implemented through a wiki in order to:

- Obtain good products with a greater degree of independence from software engineers to perform processes, that can be achieved by improving the learning process.
- Improve the ease of use of software processes, that can be achieved by storing and reusing the process assets.

To address the above objectives, this study proposes to use Knowledge Management. Knowledge management targets the capture, codification, and dissemination of knowledge across organization to enhance value [12]. Knowledge management in PALs will increase the skills to develop and store knowledge of organizational processes [13] [14]. To support the development of knowledge management systems oriented to software process, Web 2.0 offers a framework of technologies and services with the basic idea of user-contributed content dynamically alters the lifecycle of knowledge itself. Web 2.0 includes the concepts of weblogs, wikis, really simple syndication (RSS) functionality, social tagging, mashups and user defined content. As a vehicle for implementing the solution proposed, the concept of Wiki is used. A PAL can be implemented by a wiki to behave like a lightweight knowledge repository in order to store process assets and to generate shared knowledge [15].

The validation of this work was conducted in a training course to assess knowledge management for learning and executing of software process by a PAL based on a wiki.

The paper is organized as follows. Section 2 presents related work with PAL. Section 3 presents the description of the proposed solution. Next, Section 4 shows the description of the validation performed. Then, Section 5 presents an analysis and discussion of the results. Finally, in Section 6, the authors present their conclusions of the study.

2 Related Work

In this section we review some proposals oriented to achieve the goals listed above.

On the Web, there are some specific repositories that implement corporate PALs. The Spawar System Center Pacific's PAL [16] and Goddard Space Flight Center of NASA's PAL [17] include artefacts such as guidelines, policies, standards, procedures and training material. These PALs have a large amount of information that makes it difficult to understand the processes. They are organized by areas or categories but the assets are dispersed, it is necessary to represent the evolution of assets, and to include easy and flexible editing capabilities.

This work is associated with the works of several authors [18] [19] which recommend knowledge management strategies and the use of wikis to create and maintain process information on collaborative and dynamics ways [20] [21].

In the literature, the wikis are used to support activities as requirements and specifications, architectural models and test cases [22]. The software development teams have begun using wikis in processes such as project management and tracking [23]; communication and collaboration [24] including brainstorming sessions, organization of fragmented knowledge and to facilitate exchange of information [25]; technical documentation of software [26]; requirements management and traceability [27]; architectural design [28]; and management of lessons learned [29].

There are few studies on the use of wikis as PALs. The work of Wongboonsin and Limpiyakorn [30] develops a PAL using a wiki. The process assets stored use CMMI as process improvement model. Our study differs in having a more specific process structure, the types of artefacts used and conducting an experiment during software practices and projects.

In general, the related work lacks of functionalities to address some PAL's aspects as assistants to deliver information about process, to support interpretation and application context [31]; an integrated approach to strategies of codification (to systematize and store information that constitutes the knowledge of the company, and to make this available to the people) and personalization (to support the flow of information in a company by having a centralized store of information about knowledge sources) [32]; support for the activities of knowledge management of process assets (creation, acquisition, adaptation, organization, distribution and application of knowledge) [14]; support to reuse non-structured elements [33], and tools of monitoring and measurement of assets.

The works mentioned above present the main handicap that the PALs implemented are no longer static repositories of information and must become useful repositories of knowledge about agile software process. Systems based on Web 2.0 and wikis offer advantages for developing and implementing lightweight repositories of knowledge.

3 Solution Proposed

This paper seeks to apply knowledge management concepts to software process assets to improve the learning of software processes using Web 2.0 tools such as wikis. The proposal offers an integrated approach of codification and personalization to store artifacts related to the process and allowing users to share knowledge. It also incorporates knowledge management activities in order to acquire, organize, distribute, use, preserve, and measure the assets. The guidelines for building a knowledge management-based PAL include the defining of a structural model, and an incremental processes model. Next, an example of the proposed PAL is applied to agile development processes.

3.1 Structural Model

Assets stored in the PAL must be organized into a structure of elements presented in Figure 1 as a UML (Unified Modelling Language) class model.

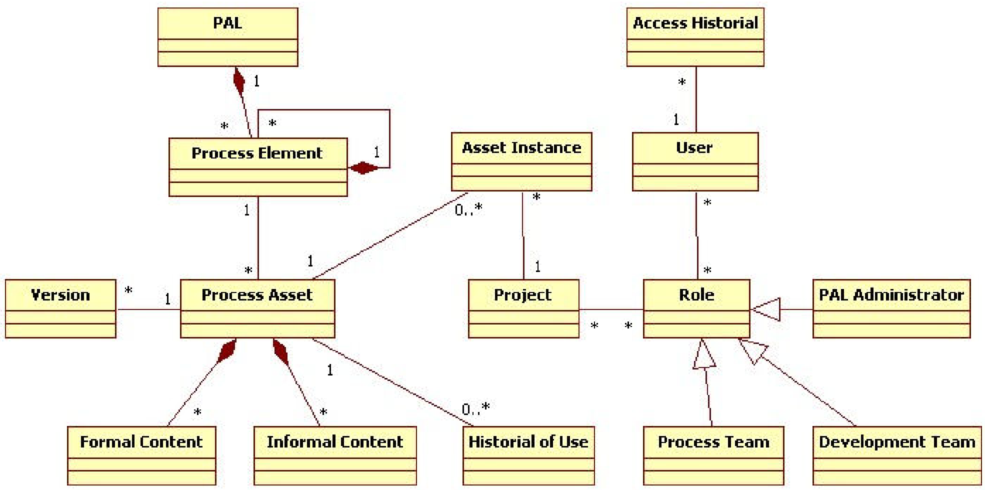


Fig 1. UML structural model for PAL-Wiki.

The *PAL* consists of a set of *Process Elements*. The Process Elements can be standard processes, life cycle models, tailoring guidelines, measurements of processes or work environment standards. The Process Elements are associated with many *Process Assets* that can contain *Formal Content* or *Informal Content*. The Formal Content refers to structured elements as checklists, templates, etc. that implement strategies of codification. The Informal Content refers to unstructured elements such as lessons learned, discussions, etc. that implement strategies of personalization. The Process Assets may have many *Versions* and can be instantiated and reused (*Asset Instance*) in specific *Projects*. During the projects, there are three kind of *Roles*: *Process team* who are responsible to produce, evaluate, edit and enhance knowledge; *Development team* who consume knowledge stored during the projects; and *PAL Administrators* who manage users and contents. Reports about users access (*Access Historial*) and process assets (*Historial of Use*) are obtained from actions performed by *Users* in the PAL.

3.2 Processes Model

The implementation of the PAL-Wiki in an organization is an incremental process composed of three kind of functions. First, a set of core functions must be implemented with the basic requirements on knowledge management to store and use process assets during software development projects. Then, a set of support functions must be implemented to manage users and contents of the PAL-Wiki. Finally, a set of advanced functions adapt to knowledge stored to enhance the usage of the assets. The PAL-Wiki functions are presented in Figure 2 and are presented briefly below.

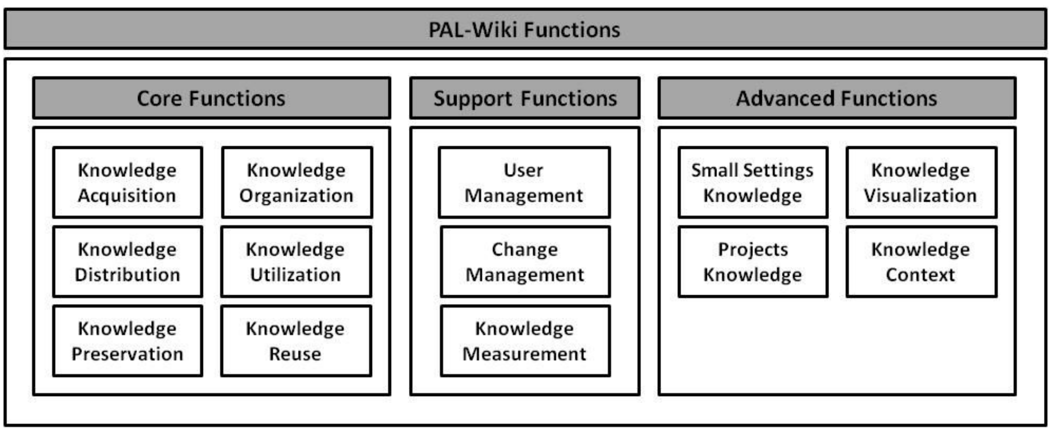


Fig. 2. PAL-Wiki functions.

- Core functions: these functions implements basic processes of knowledge management in the PAL-Wiki. They are related to creation, storage/retrieval, transfer, and application of organizational process assets into the repository.
  - Knowledge acquisition: This is the first function to be used in PAL-Wiki. Process knowledge is obtained with sufficient detail in order to be stored and used in the organization. This knowledge is acquired through forms and templates for different process elements and process assets.
  - Knowledge organization: Next, the knowledge acquired is organized according to some structure following specific formats and categories.
  - Knowledge distribution: Then, the repository should provide facilities for an effective selection of its elements by a structure with multiple types of nodes and associative connections in order to browse, view and search in their contents.
  - Knowledge utilization: The stored assets can be searched, queried, downloaded and used in order to perform activities and tasks in specific projects.
  - Knowledge reuse: The assets used and adapted for specific projects can be stored as examples into the PAL-Wiki according to application context.
  - Knowledge preservation: Finally, the contents of the PAL-Wiki are updated through a collaborative improvement process by editions and feedback from users.
- Support functions: these functions address activities that apply more generally to the processes in the organization.
  - User management: Functions for managing groups and users of the PAL-Wiki.
  - Change management: It incorporates a change control system for the contents of the process assets.
  - Knowledge measurement: Process improvement opportunities are identified through the collection of quantitative measures on user’s access and actions.
- Advanced functions: these functions address processes that are targeted toward the projects and deployment of processes.
  - Small settings knowledge: The repository provides assets with summary and detailed contents for small and medium enterprises.
  - Projects knowledge: The repository provides capabilities to store the work products developed from the usage of assets stored in PAL-Wiki during the projects.
  - Knowledge visualization: The repository includes knowledge maps to coordinate, simplify, highlight and navigate though knowledge stored, providing a framework for capturing and organization around topical areas.
  - Knowledge context: To make reuse more effective of assets, the structuring of knowledge requires first a decontextualization (elimination of application context), and then a subsequent recontextualization when it is applied to a new situation or project.

3.3 An experience with agile processes

This model of PAL based on knowledge management has been instantiated for validation using elements of agile development process. The Process Elements defined were:

- Processes: Agile development processes as user stories management, planning, development, refactoring, tracking, continuous integration, and defect management.
- Activities: Each process is decomposed into a set of activities. The activities have a similar structure that processes.

- Tasks: Set of actions to be undertaken to successfully perform an activity. The tasks are bottom level elements that do not decompose.
- Roles: There are three roles defined: Team coordinator, Software engineer and Customer/User.
- Work items: These are records used to track a activity or task. Work items defined are: Defect, User story, Quality of service requirement, Resource, Acceptance test, Unit test, Class, Task, and Automatic integration element.
- Work products: These are files, documents, specifications, binary files, parts and other tangible items necessary to complete the activities.
- Technical instructions: They present information on how to carry out activities and tasks.

This agile processs-oriented PAL implemented the core and support functions. In total, the PAL-Wiki have 8 processes, 31 activities, 9 work items, 6 work products, 3 roles, 22 sets of slides, 38 videos and 30 examples. Regarding implementation issues, the PAL-Wiki was built by MediaWiki tool, which was installed on an Ubuntu Server with Apache Web Server, PHP programming language and MySQL database. It can be reached at http://wikisel.sel.inf.uc3m.es/dhip/, login: guest, password: guest01.

4 Validation Description

To evaluate the goals defined in the introduction of this paper, we have defined the following hypothesis:

IF a PAL based on knowledge management and implemented through a wiki is developed following the recommendations given in Section 3 THEN

- Good products are produced with a greater degree of independence from software engineers to perform the processes.
- It improves learning of agile development processes.

The participants in this study were two different groups of software engineers during two phases.

4.1 Phase 1: From September 2008 to January 2009

This phase was conducted by group 1 without the PAL-Wiki. During this phase the trainers conducted a set of tutorials to the software engineers. At this phase the following stages were undertaken:

4.1.1 Training stage

This stage lasted 11 weeks. The practices were divided into 8 sessions: a) User stories, b) Acceptance tests, c) Coding standard, d) Simple design, e) Refactoring, f) Unit tests, g) Collective ownership and continuous integration, h) Tracking. The practices were developed by teams of two engineers. The correction degrees of each practice were collected .

4.1.2 Project stage

This stage started on week 12. The project involved the improvement and expansion of a small software application by adapting the concepts of agile development methodologies. The project was a product very similar to a specification of the real world because it contained a set of tasks, teamwork, environment, time pressure, and quite realistic methods and techniques. At the end of the project stage, degrees of correction of work products were collected.

4.2 Phase 2: From September 2009 to January 2010

This phase was conducted by the group 2 using the PAL-Wiki. In this phase instructors solved by email any question made by the software engineers. Statistics on user access and contents accessed and downloaded were collected. At this phase the following activities were undertaken:

4.2.1 Training Stage

The duration of the training stage was 11 weeks. The practices were the same 8 sessions of Phase 1. The teams used the PAL-Wiki to perform the practices. The engineers were familiarized themselves with the PAL exploring its contents. Once completed 8 sessions of practice, it made a first survey on the ease of use and benefits using the PAL-Wiki to develop the practices. The survey consisted of 21 items in the range from 1 to 5 using Likert scale [34] to assess aspects of usability and benefits of the PAL-Wiki, 24 semantic differential items with 5 rankings and 4 open questions items. For this first survey, the items at the level of ease of use were: visual presentation, navigation, structure, search, language, learning, artifacts, help, memorization, access not mandatory, and ease of use. The usefulness level items were: quality of products, saving time, work easier, ease of learning, completeness, usefulness of artifacts, visibility, communication of results, compatibility, and usefulness. The semantic differential items assessed ease of access and understanding of the artifacts, easy to learn the concepts of agile development, and utility of the artifacts.

4.2.2 Project Stage

The project stage started in week 12. The characteristics of the final project were the same as the project in Phase 1. Grades of correction of work products were collected to compare the results obtained in both cases with and without the use of the PAL-wiki. At the end of this stage, a second survey assessed PAL capabilities to provide information during the project. The second survey consisted of 18 items with the range from 1 to 5 using Likert scale, 10 dichotomous questions and 13 semantic differential items with 5 rankings. The items assessed with Likert scale were: Seeking knowledge, presentation of information, GUI, interaction, process control, artifacts, productivity, content, knowledge management, repository of projects, sharing knowledge, and capabilities for future works. The dichotomous questions were related to actions performed for users. The semantic differential items collected data on the quality of processes and artifacts.

5 Analysis of Results and Discussion

The analysis of data obtained in the two validation phases was performed using descriptive statistics and regression and correlation analysis between surveys data and PAL-Wiki access statistics. The results and discussion for each objective are presented below.

5.1 Objective 1: To get good work products with a greater degree of independence from software engineers to perform processes

Software engineers in Phase 2 worked more independently because they learned about the process by searching knowledge and artifacts stored in the PAL-Wiki.

Data on the levels of quality of products during the project stage in both phases were collected in order to determine that an independent work using PAL-Wiki help in learning of processes. Table 1 presents descriptive statistics such as number of products (N), mean and standard deviation (SD) of levels of quality of products collected. The quality of each product developed is evaluated from the completeness of all its constituent parts. A weight is assigned to each part, which represents the relative importance of completing the part. The mean of the quality of each product used a 0-10 scale with one decimal place. The whole information regarding the data gathered can be reached at http://wikisel.sel.inf.uc3m.es/dhip.

Table 1. Levels of product quality.

Product works	Group	N	Mean	SD	Product works	Group	N	Mean	SD
User story	1 2	8 6	10 9.9	0 0.3	Unit test	1 2	8 6	4.1 3.8	1.4 1.5
Acceptance test	1 2	8 6	9.7 8.7	0.7 1.6	Collective ownership	1 2	8 6	8.4 8.9	1.3 1.7
Planning	1 2	8 6	8.4 5.8	1.9 2.5	Continuous integration	1 2	8 6	7.8 8.3	1.1 1.5
Coding stan-dards	1 2	8 6	5 5	5.4 5.5	Tracking	1 2	8 6	8.4 7.1	2.6 2.5
Simple design	1 2	8 6	8 7.5	3.1 1.7	Iteration wrapup meeting	1 2	8 6	7.7 8.1	2.4 2.8
Refactoring	1 2	8 6	9.4 7.2	1.8 2.5					

There were no differences in the quality degree between the products developed by the two groups. This result suggests that the PAL-Wiki had no effect on the improvement of the quality of the products developed, but increased degree of autonomy of software engineers in performing their activities because the number of questions solved by the instructors is lower in the case of using the PAL-Wiki to develop the software products analyzed.

In the PAL-Wiki, 28 user-accounts were created. During Phase 2, 217 PAL-Wiki pages were visited and viewed a total of 9,028 times. The average time on site during the training stage was 8:43 minutes per user and during the project stage was 22:12 minutes per user. The average number of pages for the course was 12.45 pages per visit.

The users used the assets during the training stage (11 weeks) and the project stage (4 weeks). Figure 3(a) shows the accesses historial by week. There is a high access in the first weeks while the users know the structure of the PAL-Wiki. Then, the accesses varies according to practices deadlines. From week 12, the users began to work in the project stage and the final increase is due to last week to deliver the project. Figure 3(b) presents the historial of use (a graphic of quantities of assets downloaded during training). The assets downloaded have a very similar trend to Figure 3(a) except the final week due to they already downloaded in previous practices.

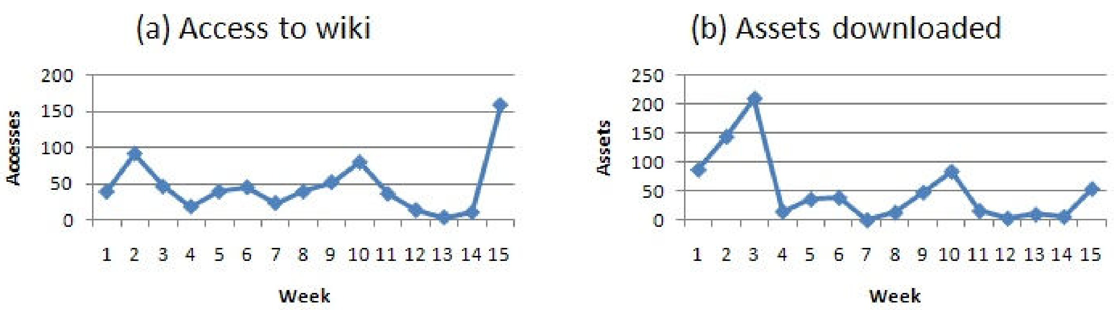


Fig 3. Accesses and assets downloaded in the wiki.

The statistics collected show a continued use of the process with more than one access per user each week. The results show that PAL-Wiki promotes independent work of software engineers and resolves doubts about the description of the software product to be developed.

5.2 Objective 2: Improving the learnability of the development process

For this objective, a survey was conducted at the end stages of training and project in Phase 2 to assess the learnibility of the development process. Learnibility is about how quickly and easily users can be productive with a system that is newto them [35]. Hypothesis tests and regression and correlation analysis were applied in order to determine relationships between the use of the repository and its benefits to perform tasks.

The survey for the training stage obtained a response rate of 53,57% (15 subjects). The survey data indicate a favorable response regarding the benefits and ease of use of the PAL-Wiki. The usefulness of stored artifacts had a very positive assessment for examples, videos and product works because they provide a practical approach on how to perform tasks. Processes such as "Refactoring", "User stories management", "Planning" and "Tracking" which included many artifacts were positively valued. The theoretical material provided had a medium rating because they can not be downloaded and printed for viewing off-line. In general, the ease of use was considered good (3.93 on a scale of 1-5). The training survey indicates that the PAL is easy to learn and use.

Table 2 presents some statistics of access to the processes during the training stage of Phase 2 together with the ease of learning of each process from the survey.

Table 2. Processes access statistics gathered during the training stage.

Processes	Users	Access	Time medium/user in PAL (minutes)	Assets downloaded	Learnability (scale 1-5)
User story	53	1075	11,23	149	3,933
Planning	63	1688	10,68	209	3,933
Development	117	2298	7,43	163,00	3,600
Refactoring	34	725	7,78	33	4,200
Continue integration	48	1281	8,77	175	3,867
Tracking	45	811	5,33	153	3,267

A multiple regression was found (R^2=99.7%) among the variables: users (U), access (A), time (T) and downloaded assets (D) that act as predictors of Learnability (L). The equation is as follows:

$$L = 3{,}58 - 0{,}015\ U + 0{,}0008\ A + 0{,}08\ T - 0{,}005\ D$$

The learnability factor depends on four factors: average time in the PAL, number of accesses, users and assets downloaded. Therefore, the PAL-Wiki is useful for continuous training in organizations.

We also found a high correlation (R^2 = 0.99) among the semantic differential items of the survey of Phase 2 – Training stage related to access and understanding of the artifacts stored and usefulness of these artifacts. The analysis of the survey of Phase 2 - Project stage presented a high correlation (R^2 = 0.842) between the quality of the artifacts and the number of accesses to PAL-Wiki. There is also a high correlation (R^2 = 0.842) between quality of artifacts items from the survey of Phase 2 - Project stage and the items of access and understanding of artifacts from survey of Phase 2 - Training stage. Furthermore, there is a high correlation (R^2 = 0.895) among the items of quality of process information from the survey of Phase 2 - Project stage and the ease of learning from the survey of Phase 2 - Training stage. Therefore, the project stage survey shows a positive assessment on the quality of process and artifacts in learning of software process.

6 Conclusions

This paper provides:

- Guidelines to build a PAL with a structural model and a model of incremental processes based on knowledge management using a wiki system. These models allows building a generic PAL that acts as a repository of knowledge about software processes.
- An example of the proposed PAL applied to agile development processes. The developed example allows assessing the impact of the use of knowledge repositories in the software process learning.

The PAL developed is featured to facilitate an effective learning environment in order to help users learn a process, to work autonomously with process assets, to provide mechanisms for sharing knowledge, and to provide a knowledge repository with useful and accessible artifacts. The results show that software engineers query and reuse the stored artifacts to learn the process. The simple interface of wiki allows rapid reference to main sections. These results are consistent with the evidence that point to wikis as scalable ways for documentation on processes and projects [22][23].

Software engineers reported that the PAL is easy to learn, use and operate; the stored artifacts are useful for solving practices; it is a useful means to publish knowledge about the process, and the assets reuse improves the quality of the products. From the study it is inferred that a wiki system helps learn software processes. The PAL-Wiki is appropriate during the stages of training, and motivates the software engineers to explore autonomously concepts about software processes.

Literature

[1] Fuggetta, A., Sfardini, L.: Software engineering methods and technologies. Software Engineering, Vol. 2: The Supporting Process, Third edition, IEEE Computer Society (2005)

[2] SWEBOK: IEEE Computer Society Professional Practices Committee. Guide to the Software Engineering Body of Knowledge - 2004 Version, IEEE Computer Society, (2004)

[3] Derniame, J.C., Oquendo, F.: Key issues and new challenges in software process technology. SPT, Software Process Technology, Novatica, Spain, 5(5) (2004)

[4] Ruiz-González, F., Canfora, G.: Software process technology: Improving software project management and product software process technology. The European Journal for the Informatics Professional, V(5) (2004)

[5] CMU/SEI-2006-TR-008. CMMI for Development, Version 1.2, Software Engineering Institute (2006)

[6] Turner R. Toward Agile systems engineering processes. Crosstalk, The Journal of Defense Software Engineering, 11--15 (2007)

[7] Nejmeh B.A, Riddle W.E.: The PERFECT Approach to Experience-based Process Evolution. Advances in Computers, M. Zelkowitz (Ed.), Academic Press (2006)

[8] Jalote, P.: Software Project Management in Practice, Addison Wesley (2002)

[9] Humphrey, W.: TSP: Leading a Development Team. Addison-Wesley Publishing Company Inc, Reading (2005)

[10] Bayona S., Calvo, J.A., Cuevas G., San Feliu T., Sánchez, A.: Process deployment in a multi-site CMMI level 3 organization: A case study. Computer and Information Science, 147--156 (2008)

[11] Chau, T., Maurer, F.: A case study of wiki-based experience repository at a medium-sized software company. In: Proceedings of the 3rd international conference on knowledge capture (2005)

[12] Adler, R.: Knowledge engines for critical decision support. Knowledge Management Strategies: A Handbook of Applied Technologies. Chapter V, Information Science Reference, 143--169 (2008)

[13] Edwards, J.S., Shaw, D., Collier, P.M.: Knowledge management systems: finding a way with technology. Journal of Knowledge Management, Emerald Group Publishing Limited, 9(1), 113--125 (2005)

[14] Ward, J., Aurum, A.: Knowledge management in software engineering - Describing the process. In: Proc. of Australian Software Engineering Conference, ASWEC'04, Melbourne, Australia (2004)

[15] Ras, E., Weber, S.: Software organization platform: Integrating organizational and individual learning. Wikis for Software Engineering, Wikis4SE ICSE (2009)

[16] Groarke, B.: Running an SEPG - How SSC San Diego Operates its SEPG. SPEG 2006, Software Engineering Institute (2006)

[17] Raffo, D., Wakeland, W.: Moving Up the CMMI Capability and Maturity Levels Using Simulation. Technical Report CMU/SEI-2008-TR-002, Software Engineering Institute (2008)

[18] Scorta, I.: A knowledge management practice investigation in Romanian software development organizations. WSEAS Transactions on Computers, 8(3), 459--468 (2009)

[19] Aurum, A., Daneshgar, F., Ward, J.: Investigating knowledge management practices in software development organisations - An Australian experience. Information and Software Technology, 50(6), 511--533 (2008)

[20] Majchrzak, A., Wagner, C., Yates, D.: Corporate wiki users: Results of a survey. WikiSym'06, Odense, Denmark (2006)

[21] Radziwill, N.M., Shelton, A.L.: TWiki as a platform for collaborative software development management. Advanced Software, Control, and Communication Systems for Astronomy, Proc. of the SPIE, Volume 5496 Glasgow, UK, 609--617 (2004)

[22] Riechert T., Berger T.: Leveraging semantic data wikis for distributed requirements elicitation. Wikis for Software Engineering, Wikis4SE ICSE (2009)

[23] Louridas, P.: Using wikis in software development. IEEE Software, 23(2), 88--91 (2006)

[24] Mullick, N., Bass, M., Houda, Z., Paulish, P., Cataldo, M.: Siemens global studio project: Experiences adopting an integrated GSD infrastructure. In: Proceedings of ICGSE'06 (2006)

[25] Hasan, H., Meloche, J.A., Pfaff, C.C., Willis, D.: Beyond ubiquity: Co-creating corporate knowledge with a wiki. International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies, UBICOMM '07, 35-40 (2007)

[26] Aguiar, A., David, G.: WikiWiki weaving heterogeneous software artifact. In: Proc. of the 2005 International Symposium on Wikis, San Diego, CA, 67--74 (2005)

[27] Silveira, C., Faria, J.P., Aguiar, A., Vidal, R.: Wiki based requirements documentation of generic software products. AWRE'05, Melbourne, Australia, 42--51 (2005)

[28] Soles C., Ali, N., Babar, M.A.: A spatial hypertext wiki for architectural knowledge management. Wikis for Software Engineering, Wikis4SE ICSE (2009)

[29] Korthaus, A., Aleksy, M., Henke, S.: A distributed knowledge management infrastructure based on a topic map grid. International Journal of High Performance Computing and Networking (IJHPCN), 6(1), Inderscience Publishers, 66-80 (2008)

[30] Wongboonsin, J., Limpiyakorn, Y.: Wikipedia Customization for Organization's Process Asset Management. International Conference on Advanced Computer Theory and Engineering, 467-471 (2008)

[31] Grudin, J.: Enterprise Knowledge Management and Emerging Technologies. In: Proceedings of the 39th Annual Hawaii International Conference on System Sciences, p.57.1 (2006)

[32] Bjornson, F., Dingsoyr T.: Knowledge management in software engineering: A systematic review of studied concepts, findings and research methods used. Information and Software Technology, 50(11), 1055--1068 (2008)

[33] Hyung-Jun, A., Hong-Joo, L., Kyehyun, C., Sung-Joo, P.: Utilizing Knowledge Context in Virtual Collaborative Work. Decision Support Systems, 39(4), 563--582 (2005)

[34] Melnik, G., Maurer, F.: A cross-program investigation of students' perceptions of agile methods. In: Proc. of the 27th international conference on Software engineering (ICSE), ACM Press, 481--488 (2005)

[35] Rafla, T., Robillard, P. N., Desmarais, M. C.: A method to elicit architecturally sensitive usability requirements: its integration into a software development process. Software Quality Journal, 15(2), 117–133 (2007)

Author CVs

Antonio de Amescua

Antonio de Amescua is a full professor in the Computer Science Department at Carlos III University of Madrid. He has been working as software engineer in a public company (Iberia Airlines) and in a private company (Novotec Consultores) as a software engineering consultant. He founded Progresion, a spin-off company, in order to offer advanced Software Process Improvement services. He received his PhD in computer science from the Universidad Politécnica of Madrid. His research interests include Software Process Improvement, Software Reuse, Software Project Management.

Leonardo Bermón

Leonardo Bermon is a PhD student in the SEL-UC3M (Software Engineering Lab) group at Carlos III University. He has worked in different universities in Colombia. He is a professor at National University of Colombia. He has both a bachelor of computer science degree and the grade of Magister in Computer Science from Industrial University of Santander, Colombia. His research interests include Software Process Improvement, Knowledge Management and Software Project Management.

Maria-Isabel Sánchez

Maria-Isabel Sánchez has been a faculty member in the Computer Science Department at Carlos III Technical University of Madrid since 1998. Her research interests include: software engineering in general and software process improvement in particular, interactive systems, and usability in interactive systems. She holds a BSc in Computer Science (1997), an MSc in Software Engineering (1999), and a Ph.D. in Computer Science (2001) from the Universidad Politecnica of Madrid. She is the author of several papers on the improvement of software processes, published in journals such as IEEE Transactions on Software Engineering, Software Practice and Experience, Interacting with Computers, and the Journal of Systems and Software.

Javier García

Javier García is an associate professor in the Computer Science Department at Carlos III University of Madrid. He is a software process improvement consultant in Progresion. He has 10 years experience as a software engineer and consultant in public and private companies. He received his PhD in computer science from the Carlos III University of Madrid. His current research interests are related to SPI in small settings, measurement of software processes, ISO 15504 assessments, software capacity quick audits, and CMMI and management of knowledge related to software engineering.

Proposing a Knowledge Engineering Based Approach for Process Capability/Maturity Models Customization

Jean Carlo R. Hauck[1,2], Christiane Gresse von Wangenheim[1], Fergal Mc Caffery[2], Aldo von Wangenheim[1]
[1]Federal University of Santa Catarina (UFSC)
Florianópolis/SC – Brazil
gresse@gmail.com
[2] DkIT – Dundalk Institute of Technology
Dundalk/Co. Louth, Ireland
jeanhauck@gmail.com

Abstract

Software Process Capability/Maturity Models (SPCMMs) are repositories of best practices for software processes suitable for assessing and/or improving processes in software intensive organizations. Yet, although there is a trend to customize such models to specific domains, little research is done on how such SPCMMs should be developed with quality. In this paper, we, therefore, propose a systematic approach to support the customization of SPCMMs for specific domains. The approach is developed based on standard development processes integrating Knowledge Engineering techniques and experiences on how such models are currently developed in practice. First feedback from piloting the approach in the customization of ISO/IEC 15504, CMMI and MPS.BR for the SaaS scenario indicates that the approach can be useful for creation of SPCMMs.

Keywords

Knowledge Engineering, Process Reference Models, Process Capability/Maturity Models

1 Introduction

Software Process Capability/Maturity Models (SPCMM) are repositories of best practices for software processes, based on good engineering and process management principles, organized with the concept of process capability and/or maturity, suitable for assessing and/or improving processes [1]. Various generic process capability/maturity models have been developed by the software engineering community, such as, CMMI-DEV [2] and ISO/IEC 15504 [3], and their use for software process improvement and assessment is well established in practice. Yet, as these generic models intend to cover a wide range of diverse types of software products and services, processes, technologies, etc., their application in practice often requires a customization to the specific context [4]. Diverse specific software development domains have specific process quality needs that should be covered. Likewise, there are specific standards for software development, especially in the case of regulated sectors, such as health care, that must be observed by the software development process in order to provide the necessary alignment to these domain-specific standards. In order to facilitate such an adaptation, we can observe a current trend to the development of customizations of those generic process models for specific domains. Various initiatives have taken place to specialize and refine generic software process capability/maturity models adapting best-practices for process improvement in specific software development domains/sectors, such as SPICE4SPACE [5], OOSPICE [6], SMCMM [7], etc.

However, most of these initiatives do not adopt a systematic approach for the customization of those generic standards and models [8]. Actually, literature detailing how software-related process capability/maturity models are developed / evolved / adapted is extremely rare [9]. Standardization organizations, like ISO or IEEE, define high-level generic processes for developing and publishing standards. However, they do not describe how to customize existing models or provide detailed technical support for the specific development of SPCMMs.

Alternatively, these SPCMMs may be defined as “best practices” knowledge repositories. Focusing on the extraction and modeling of the knowledge (although, in our case, there is no intention to implement a knowledge-based system), Knowledge Engineering (KE) may provide an important contribution. To date, KE approaches have not been applied to this specific aim.

In this context, we present a proposal for such an approach, based on an analysis on how existing customizations have been performed, integrating standard development procedures and KE techniques. The main contribution of this paper consists on the definition of an innovative methodological process for the customization of the generic SPCMM for specific software development domains. In section 2, the background of SPCMMs is presented. Section 3, presents related software process improvement (SPI) and KE research. In section 4, our approach is proposed, and section 5 presents results from its pilot application. Conclusions and future works are presented in section 6.

2 Related Work

Although, diverse software process capability/maturity model customizations have already taken place [10], research on how to perform such customizations in a systematic way is sparse. One of the few works in this respect is done by Bruin & Rosemann [10], who propose a sequence of steps for the development of Maturity Assessment Models: (i) the definition of the scope of the model, (ii) the design of a new model, (iii) population of the model using domain components as source of specific needs, (iv) test, (v) deployment, and (vi) maintenance of the model. Although, this work considers specific domain needs, it does not address in detail the customization of domain-specific best practices from generic models.

Mettler [11] performs a deeper analysis on the fundamentals of process maturity models, putting the main phases described in [10] under a design science research perspective. In this context, the phases are compared to a model user perspective of the maturity models, indicating a need for more formal methods and studies. Salviano et al. [12] propose the generic framework PRO2PI for the development of process capability/maturity models, based on the authors previous experiences of developing diverse models. The framework consists of seven steps: (i) initial decisions; (ii) sources (of good practices) analysis, including literature, surveys of practitioners, and others; (iii) strategy for development, including how the community of interest will be involved; (iv) model design using ISO/IEC 15504 as the general structure for modeling; (v) draft model development; (vi) draft model validation; and (vii) model consolidation from an analysis of the validation of draft model results. This work represents initial research towards achieving an approach for the customization of software process capability/maturity models. To date, no detailed support is available in relation to this research. Matook and Indulska [9] propose a QFD-based approach for reference modeling incorporating the voice of the reference model users and presenting a compressed measure for the quality of such models. Their approach also provides a means for managing quality reference model development including the following phases: (i) problem definition; (ii) requirements analysis; (iii) information gathering; (iv) setting conventions and rules; (v) documentation; (vi) construction and design and (vii) evaluation. This research works presents the first steps towards the development of more systematic support for the development of reference models. However their principal focus is on the model construction, with no coverage provided of its usage and evolution. Likewise, they do not provide detailed methodological support for the customization of SPCMMs.

Based on a systematic literature review [8] and a survey [13], we also observed that, most publications which propose model customizations (52% of more than 50 models) do not report on how the customization has been done.

From a KE point of view, the customization of such models relates to knowledge acquisition, collecting best practices of a specific domain by customizing generic SPCMMs to domain-specific models. A generic life cycle for KE includes (i) knowledge identification; (ii) knowledge specification and (iii) knowledge refinement [14]. Currently, there exist several methodologies, frameworks and approaches that provide detailed support for the KE development life cycles, such as, e.g., CommonKADS [15]. Yet, again, the usage and evolution of the knowledge models is typically not covered. In addition, KE techniques have so far, not yet been applied for the customization of generic SPCMMs knowledge to specific domains. Therefore, we can observe a lack of methodologies that offer substantial support for the customization of SPCMMs

3 A Knowledge Based Approach for Process Reference Model Customization

In order to facilitate the customization of SPCMMs and to increase the quality of these, as well as increase their adoption rate in practice, we are developing a KE-based approach presented in this section. The approach is based on an analysis of: (i) how currently such customizations are made; (ii) standard development procedures; and (iii) KE techniques.

How it is being developed

With the objective of developing an approach based on scientific procedures we are following the steps presented in figure 1. We started the development with a theoretical approach, covering: KE, SE/SPI and process modeling concepts and approaches. In order to elicit the state of the art with respect to how domain-specific SPCMMs are developed, we performed a systematic literature review [8]. As a result, we identified 52 capability/maturity models, yet, most of them lacking details as to how they were developed. Therefore, we performed a second step, a survey among the authors of the models [13] with the objective to obtain additional information on how these models have been developed.

Based on these results and our experiences in customizing SPCMMs, we developed the first version of a systematic approach for the customization of SPCMMs. The approach itself is being developed under a KE perspective [15] [16], using the customization experiences observed in the literature and descriptions obtained from model authors [8] [13], phases and steps of the ISO International Standard development process [17] and IEEE standard development process [18], PRO2PI [12], and the framework for process maturity models development proposed in [10].

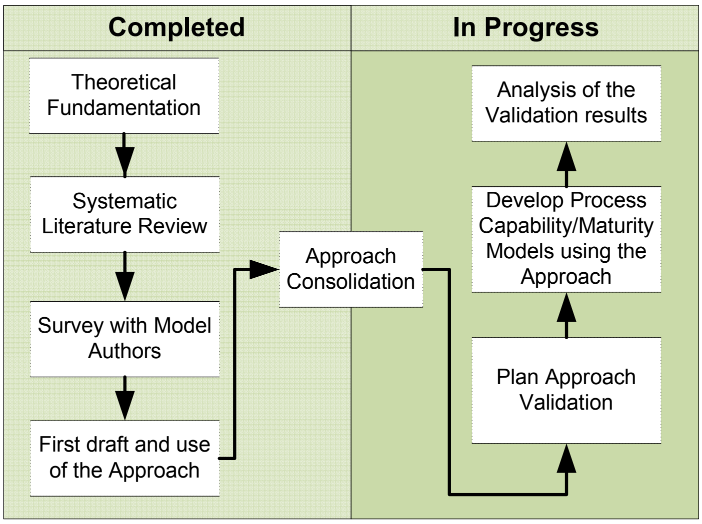


Fig 1. Steps on the development of the Approach

Currently, we are analyzing results obtained from the first usage of a draft version of the approach to consolidate its process and techniques. We will continue the iterative development of the approach while applying it in parallel to customize SPCMMs, until achieving a consolidated state. Then, a validation will be planned, executed and analyzed in order to provide a evaluation of its use.

The proposed Approach

The approach is structured (figure 2) in five phases: (i) Knowledge Identification, (ii) Knowledge Specification, (iii) Knowledge Refinement, (iv) Knowledge Usage and (v) Knowledge Evolution. Each phase is composed by a set of activities that are not necessarily sequentially executed.

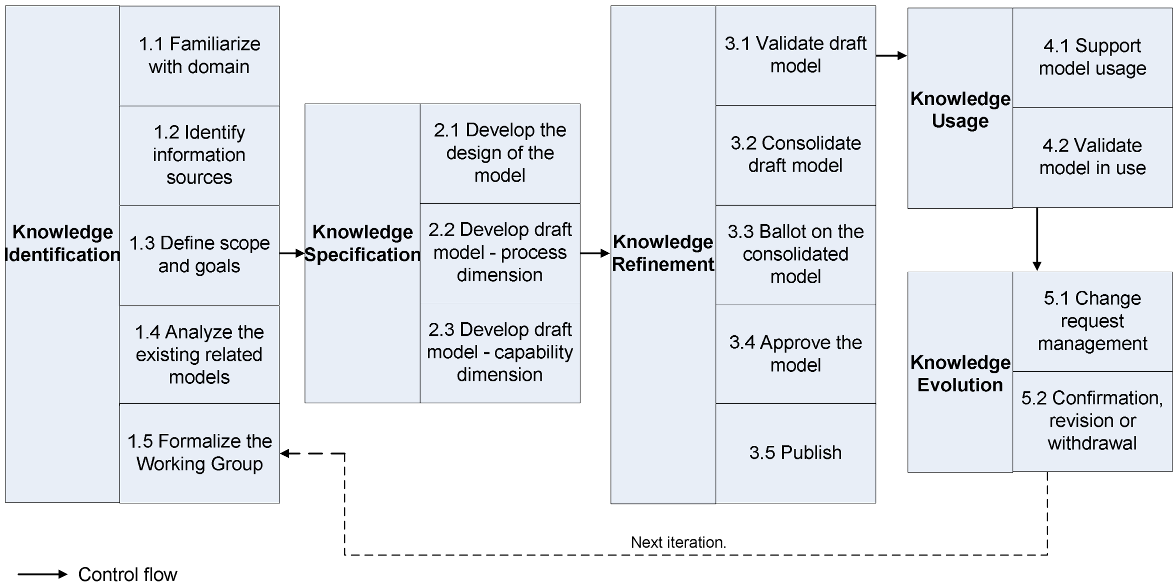


Fig 2. Phases and Activities of the Approach

Phase 1: Knowledge Identification

The main objective of phase 1 is to achieve familiarization with the target domain and a characterization of the context for which the SPCMM will be customized. Related activities are:

Activity 1.1 - Familiarize with domain: Consists in a contextualization in the domain for which the model will be developed. An analysis of domain-related literature provides, in the first place, a deep understanding of exactly what the domain is and its characteristics, providing main concept definitions and terminology, and identifying the underlying general process.

Activity 1.2 - Identify information sources: that will be used as input for the model development. Important information sources consist of: human resources, domain-specific software development standards, generic process capability/maturity models, or reports/papers which identify e.g. important quality / performance aspects. The identification of human sources requires the definition of profile of knowledge agents which in this context means to describe the software development domain experts. It is also necessary to identify which generic SPCMM will be customized for the specific domain. The choice depends on how important each generic SPCMM is for the domain sector in terms of reliability, applicability and market impact.

Activity 1.3 - Define scope and goals: of the model to be developed. The scope of the customized SPCMM must precisely define the limits of the application domain, and define without ambiguity the subject of the model and the aspects covered, thereby indicating the limits of its applicability or particular parts of it. It is important that to identifying the specific goals that must be achieved by the SPCMM to be customized, determining the aims and the interests that may be affected.

Activity 1.4 - Analyze existing related models: once specified the scope of the SPCMM has to be customized, relevant source models are defined and analyzed. This typically involves a mapping of the related models and/or a harmonization effort integrating the existing models into a unified model.

Activity 1.5 - Formalize the working group: for the development of the model. This includes the definition of the allocation of a sponsor/coordinator, working rules and procedures that will be used during the development of the new model. It also includes the invitation of relevant stakeholders to participate and defines who has the rights to vote to approve the model within the working group, who can make change requests and who has the capability to contribute with the model development.

Phase 2: Knowledge Specification

During this central phase, a first version of the customized model is developed, following these proposed activities:

Activity 2.1 - Develop the design/architecture of the model: identifying the main elements of the model. The standard ISO/IEC 15504 establishes a general structure for model design. This structure includes a Process Reference Model and a Process Assessment Model. Typically, within customizations, the structure of one of the principal source models is adopted. Therefore, the structure of those models has to be analyzed and if necessary, modified appropriately.

Activity 2.2 - Develop a draft model – process dimension: in this core activity the process dimension of the SPCMM is developed. Defining a process dimension of the SPCMM implies on identify relevant processes that contain best practices for the specific domain. To identify relevant processes, it is necessary to identify which are important software quality/performance needs within the specific domain. This can be done by extracting this knowledge from the domain knowledge agents (identified in phase 1) using various techniques, such as: interviews, surveys, ontology engineering, focus groups, nominal group technique, etc. either individually or by combining any of those techniques in an iterative and incremental way. Then, in a next step, it is necessary to relate these identified quality attributes to relevant processes. An adapted version of QFD – Quality Function Deployment, involving also SPI experts, can be used to systematically map quality/performance needs with processes and required outcomes/best practices and typical work products. The mapping of related source models produced in activity 1.4 can be used to support the development of the customized SPCMM as a basis by re-using an appropriate process description (as is or by modifying them appropriately) completed by new processes when necessary.

Activity 2.3 - Develop a draft model – capability/maturity dimension: in order to produce a model that can serve as a reference for process assessment, a capability/maturity dimension is developed. Therefore, it is necessary to define attributes and group them into capability levels. This means to define attributes applicable to all processes that describe a facet of the overall capability of achieving process purpose and can be evaluated on a scale of achievement, providing a measure of the capability of the process [3]. Capability levels can be defined as sets of attribute(s) that work together to provide a major enhancement in the capability to perform a process. If it is suitable, processes can also be grouped in levels in order to define a Maturity dimension, following the priority order defined by the quality /performance needs prioritization. Again, the capability/maturity dimension of the underlying source models can be used as a basis, and being adapted when necessary. As result of this phase a draft model is developed.

Phase 3: Knowledge Refinement

In this phase, the draft model is validated, balloted and refined to develop a model approved by a majority of respective community.

Activity 3.1 - Validate draft model: the draft model itself is then validated in order to demonstrate that the draft SPCMM fulfils the general characteristics required of SPCMMs (table 1). In this step, such a validation is typically based on a consensus of relevant stakeholders reviewing the model. Various techniques can be used, including Expert Panel, Delphi etc.

Activity 3.2 - Consolidate draft model: Based on feedback obtained, the draft model is iteratively evolved, until consensus is achieved among the members of the working group. This requires: the discussion, negotiation and resolution of significant technical disagreements in order to prepare a model that will be accepted and widely used.

Activity 3.3 - Ballot on the consolidated model: During this activity, the developed model is distributed, and interested parties vote on the approval or rejection of the model.

Activity 3.4 - Approve the model: Clear criteria for approval must be defined as well as procedures for what happens upon approval or non approval. If necessary, reviews of the model are repeated until the model is approved.

Activity 3.5 – Publish: the resultant model is then made available in an accessible place for the respective domain community.

Phase 4: Knowledge Usage

After its publication, the model is been put in use and results of its usage are collected and analyzed.

Activity 4.1 - Support model usage: it is necessary to define which kind of support will be provided for the model usage, such as, training, user forums, etc. For example, the establishment of a web forum is important to keep the SPCMM development community active.

Activity 4.2 - Validate model in use: in order to validate the model based on its usage in practice, a framework for its validation has to be defined, data collected and analyzed. Such a framework can be developed, using for example, the GQM method. The results will complement the results of the earlier expert validation that was performed and this may be used to develop future new versions of the model.

Phase 5: Knowledge Evolution

Due to various reasons, SPCMMs evolve constantly (maturing of the domain knowledge, technological advances, etc.). Therefore it is necessary to also provide methodological support for the continuous evolution of the model once the model has been implemented in the target domain.

Activity 5.1 - Change request management: it is necessary to define how change requests from different stakeholders are collected in a systematic way and how they are managed.

Activity 5.2 - Confirmation, revision or withdrawal: the process model development group defines which changes will be accepted and how new versions of the model will be published. Each group of changes must follow phase 3 to provide validation of the changes. This process must be supported by a regular configuration management process.

In this way, the proposed approach presents a first proposal for the systematic customization of SPCMMs. In the next section, we show first results and lessons learned we obtained by piloting the proposed approach in the development of a customized SPCMM.

4 First Results and Discussion

The proposed approach for SPCMM customization has been developed in parallel with the customization of a SPCMM for the Software as a Service (SaaS) domain [19].

SaaS is a software solution offered as a service and is developed using SOA. As the SaaS scenario requires specific quality needs, such as, security, availability and service continuation, due to its characteristics of distributed software products as services, a customization of SPCMMs has been done. The SaaS SPCMM has been developed by a group of researchers at the UFSC – Federal University of Santa Catarina/Brazil, involving experts from the SaaS domain and SPI experts. We developed the model through adopting the proposed approach, covering the phases 1 to 3. At present, phases 4 and 5 have not been performed.

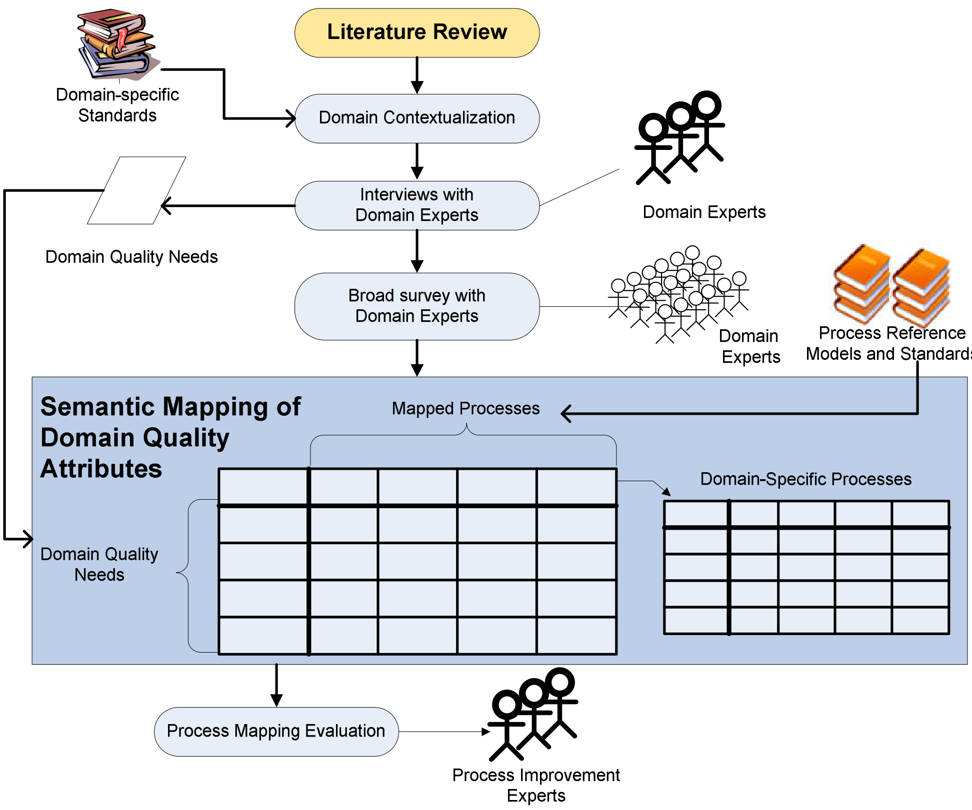
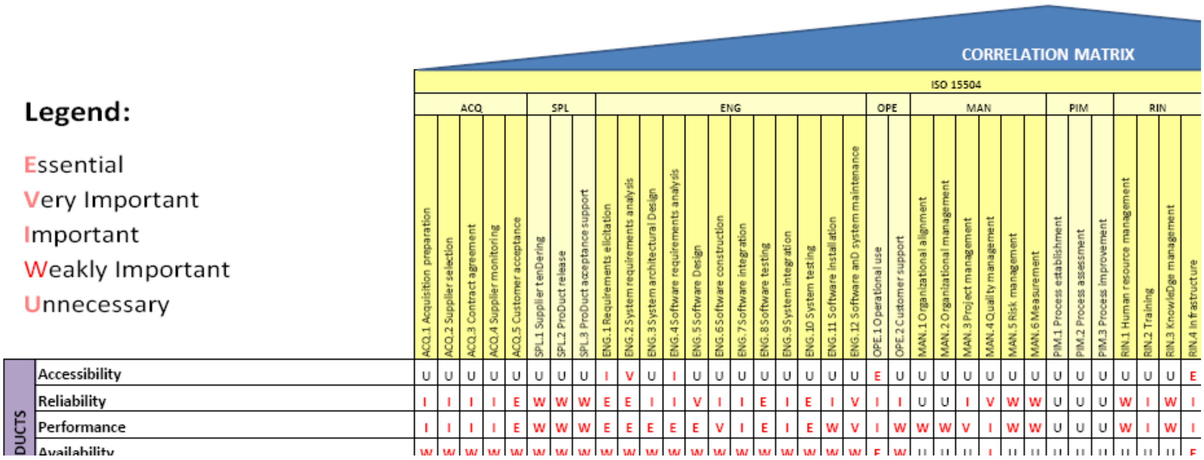


Fig 3. Process adopted for the development of SaaS SPCMM [19].

Following the process illustrated in figure 3, the domain has been contextualized and stakeholders have been identified and characterized. Generic SPCMMs (ISO/IEC15504-5, CMMI-DEV, MPS.BR and CMMI-SVC) have been analyzed and identified as a basis for the customized model. In addition, relevant quality and performance needs in the SaaS domain have been elicitated based on a literature review.

During phase 2, we decided to basically adopt the architecture of ISO/IEC 15504 as the structure of the customized model. Developing the process dimension, in a first step we interviewed 6 SaaS experts in order to complete the elicitated quality and performance needs. The results have then been validated in a second step through a survey, involving 84 SaaS experts, who reviewed and prioritized the identified needs.

Then, a group of 3 SPI experts identified relevant processes and basic practices with respect to the identified quality and performance needs by mapping them using an adapted version of the QFD approach [9]. The result was a draft version of the process model (figure 4). So far, no specific capability/maturity dimension has been developed, adopting simply the capability dimension from ISO/IEC 15504.



ISO 15504	ACQ.1 Acquisition preparation	ACQ.2 Supplier selection	ACQ.3 Contract agreement	ACQ.4 Supplier monitoring	ACQ.5 Customer acceptance	SPL.1 Supplier tenDering	SPL.2 ProDuct release	SPL.3 ProDuct acceptance support	ENG.1 Requirements elicitation	ENG.2 System requirements analysis	ENG.3 System architectural Design	ENG.4 Software requirements analysis	ENG.5 Software Design	ENG.6 Software construction	ENG.7 Software integration	ENG.8 Software testing
Accessibility	U	U	U	U	U	U	U	U	I	V	U	I	U	U	U	U
Reliability	I	I	I	I	E	W	W	W	E	E	I	I	V	I	I	E
Performance	I	I	I	I	E	W	W	W	E	E	E	E	E	V	I	E
Availability	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

ISO 15504	ENG.9 System integration	ENG.10 System testing	ENG.11 Software installation	ENG.12 Software anD system maintenance	OPE.1 Operational use	OPE.2 Customer support	MAN.1 Organizational alignment	MAN.2 Organizational management	MAN.3 Project management	MAN.4 Quality management	MAN.5 Risk management	MAN.6 Measurement	PIM.1 Process establishment	PIM.2 Process assessment	PIM.3 Process improvement	RIN.1 Human resource management	RIN.2 Training	RIN.3 KnowleDge management	RIN.4 Infrastructure
Accessibility	U	U	U	U	E	U	U	U	U	U	U	U	U	U	U	U	U	U	E
Reliability	I	E	I	V	I	I	U	U	I	V	W	W	U	U	U	W	I	W	I
Performance	I	E	W	V	I	W	W	W	V	I	W	W	U	U	U	W	I	W	I
Availability	W	W	W	W	E	W	U	U	U	I	U	U	U	U	U	U	U	U	E

Fig 4. Extract of the mapping of quality and performance needs to relevant processes (The complete version of the matrix is available at http://www.gsigma.ufsc.br/~cancian/msc/mapping.pdf).

During phase 3, the draft model has been reviewed by different SPI experts and the model has been improved based on the obtained feedback.

This experience allowed us to identify strengths and weaknesses of the proposed approach in practice. One of its strengths is the involvement of specialists, although we also identified that in order to stimulate a wide adoption of the model, a much stronger involvement of the community is also required. Another strength is the methodological support which typically, for standard developments, is not available. Using for example, a modified version of the QFD allowed systematic mapping and also allowed explicit derivation of the model. We also observed several improvement opportunities:

- Support for a systematic mapping and harmonization of existing models;
- Better methodological support for consensus building among community representatives throughout the models development and not just elicitation of their knowledge;
- More systematical and formal support for the validation of the models.
- Integration of data-based input to the models if available in the specific domain in order to complete the expert's knowledge.

5 Conclusions

In this paper, we outline an approach for SPCMM customization by integrating a KE perspective, customization experiences from literature and standard development processes. A first application of the proposed approach for the customization of a SaaS SPCMM provides a first indication that the approach can be useful for the customization of such models as well as enabling the identification strengths and weaknesses. Based on the feedback, we are currently evolving and refining the proposed approach as well as continuing its application in parallel for the customization of SPCMMs, such as, for medical devices as well as digital convergence.

Acknowledgements

Authors would like to thank Maiara H. Cancian for sharing experiences on the development of the SaaS Process Reference model.

This work was supported by the CNPq (Conselho Nacional de Desenvolvimento Científico e Tecnológico) and CAPES (Coordenação de Aperfeiçoamento de Pessoal de Nível Superior), entities of the Brazilian government focused on scientific and technological development. This research is also supported in part by Science Foundation Ireland (SFI) through the Stokes Lectureship Programme, grant number 07/SK/I1299, and the SFI Principal Investigator Programme, grant number 08/IN.1/I2030.

Literature

[1] C. F. Salviano and A. M. C. M. Figueiredo, "Unified Basic Concepts for Process Capability Models," in 20th Int Conf on Sw. Eng. and Knowledge Eng. SEKE, San Francisco, USA, 2008, pp. 173-178.

[2] CMMI Product Team, "CMMI for Development, Version 1.2," Carnegie Mellon University/Software Engineering Institute, Pittsburgh, Technical Report CMU/SEI-2006-TR-008, 2006.

[3] International Organization for Standardization (ISO) / International Electrotechnical Commission (IEC) ISO/IEC, "ISO/IEC 15504: Information Technology Process Assessment - Part 1 to 5," International Organization for Standardization (ISO) / International Electrotechnical Commission (IEC), ISO/IEC International Standard 2005.

[4] S. Beecham, T. Hall, and A. Rainer, "Building a Requirements Process Improvement Model," Faculty of Engineering and Information Sciences, University of Hertfordshire, Hertfordshire, Technical Report 378, 2003.

[5] A. Cass and C. Volcker, "SpiCE for SPACE: A method of Process Assessment for Space Projects," in SPICE 2000 Conference, 2000.

[6] J. Torgersson and A. Dorling, "Assessing CBD - What's the Difference?," in 28 th Euromicro Conference, Dortmund, Germany, 2002, p. 332-341.

[7] A. April, A. Abran, R. Dumke, "SMCMM Model to Evaluate and Improve the Quality of the Software Maintenance Process," in 8th Euromicro Working Conference on Software Maintenance and Reengineering 2004, IEEE Computer Society, p. 243.

[8] C. G. von Wangenheim, J. C. R. Hauck, C. F. Salviano and A. von Wangenheim, "Systematic Literature Review of Software Process Capability/Maturity Models," in Spice Conference 2010, Pisa, Italy, 2010.

[9] S. Matook and Indulska, "Improving the quality of process reference models: A quality function deployment-based approach," Decision Support Systems, vol. 47, pp. 60–71, 2009.

[10] T. Bruin and M. Rosemann, "Understanding the main phases of developing a maturity assessment model," in 16th Australasian Conference on Information Systems, Sydney, Australia, 2005.

[11] Tobias Mettler, "A Design Science Research Perspective on Maturity Models in Information Systems," Universität St. Gallen, St. Gallen, Switzerland, Technical Report BE IWI/HNE/03, 2009.

[12] C. F. Salviano, A. Zoucas, J. V. L. Silva, A. M. Alves, C. G. von Wangenheim, and M. Thiry, "A Method Framework for Engineering Process Capability Models," in 16th European Systems and Software Process Improvement and Innovation, Alcala, Spain, 2009, pp. 6.25-6.36.

[13] C. G. W. Wangenheim and J. C. R. Hauck, A. Zoucas, C. F. Salviano, "Software process capability/maturity models survey report," Universidade Federal de Santa Catarina, Florianópolis, Technical Report, under review, 2010.

[14] A. Th. Schreiber and B. J. Wielinga, "Knowledge Model Construction," in 11th Workshop on Knowledge Acquisition, Modeling and Management, Voyager Inn, Banff, Alberta, Canada, 1998.

[15] G. Schreiber, H. Akkermans, Walter V. Shadbolt, and B. Vielinga, "Knowledge Engineering and Management – The CommonKADS Methodology," USA: The MIT Press, 2000.

[16] J. Hua, "Study on Knowledge Acquisition Techniques," in Second International Symposium on Intelligent Information Technology Application, 2008, pp. 181-185.

[17] ISO International Organization for Standardization. (2010, Feb.) How ISO develops standards. [Online]. http://www.iso.org/iso/how_iso_develops_standards

[18] IEEE. (2010, Feb.) IEEE Standards Development Process. [Online]. http://standards.ieee.org/resources/development/index.html

[19] M. Cancian. (2010, Feb.) Process Reference Model for SaaS - Technical Report. UFSC, Florianopolis/Brazil. [Online]. http://www.gsigma.ufsc.br/~cancian/guide/

Author CVs

Jean Carlo R. Hauck

Jean Carlo Rossa Hauck is visitor researcher at Dundalk Institute of Technology (DkIT) and SEPG manager of the CYCLOPS Research Group at the Federal University of Santa Catarina (UFSC). His research interests are in software process improvement and project management. He received his M.Sc. in Computer Science from the UFSC and he is PhD student of Knowledge Engineering and Management at the Federal University of Santa Catarina. Contact him at UFSC - EGC, Campus Universitário 88049-200 Florianópolis/SC, Brazil; jean-hauck@incod.ufsc.br

Christiane Gresse von Wangenheim

Christiane Gresse von Wangenheim is a professor at the Federal University of Santa Catarina (UFSC). Her research interests are software process improvement, including project management. Previously, she worked at the Fraunhofer Institute for Experimental Software Engineering and the UNIVALI. She received a PhD in Production Engineering at the Federal University of Santa Catarina (Brazil) and a PhD in Computer Science at the University of Kaiserslautern (Germany). She's also a PMP - Project Management Professional and Assessor of the Brazilian Process Improvement Model MPS.BR. She's a member of the IEEE Computer Society, the Project Management Institute, and the Working Group ISO/IEC JTC1/SC7/WG24—SE Life-Cycle Profiles for Very Small Enterprises. Contact her at UFSC, Campus Universitário 88049-200 Florianópolis/SC, Brazil; gresse@gmail.com

Fergal McCaffery

Fergal Mc Caffery is a Lecturer with Dundalk Institute of Technology. He is leader of the Regulated Software Research Group in Dundalk Institute of Technology and a member of Lero. He has been awarded Science Foundation Ireland funding through the Stokes Lectureship and Principal Investigator Programmes to research the area of software process improvement for the medical device domain. Additionally, he has received EU FP7 research funding to improve the effectiveness of embedded software development environments for the medical device industry. Contact him at fergal.mccaffery@dkit.ie.

Aldo von Wangenheim

Aldo von Wangenheim is a professor in medical informatics and telemedicine at the Federal University of Santa Catarina's University Hospital. His research interests include medical image analysis for diagnosis support and large-scale telemedicine frameworks for public health and healthcare education. von Wangenheim received his PhD in computer sciences from the University of Kaiserslautern, Germany. He has been nominated to coordinate the newly created Brazilian National Research Institute for Digital Convergence in Health. Contact him at awangenh@inf.ufsc.br.

SPICE Level 3 - Experience with using E-Learning to Coach the Use of Standard System Design Best Practices in Projects

R. Messnarz[1], O. Bachmann[2], D. Ekert1, A. Riel[3]
1I.S.C.N., Austria & Ireland, {rmess, dekert}@iscn.com,
2 I.S.C.N. Group/ SIBAC GmbH, Germany {obachmann}@iscn.com
3Grenoble Institute of Technology, France, ariel@inpg.fr

Abstract

Most improvement initiatives focus on assessments and derived improvement plans. However, the effort to really implement and sustain improvements is much bigger than the assessment effort. Also, it is crucial for SPI success that improvements are not just a collection of formal documentation requirements but show real benefit for the development and help to optimize the development.

Successful internal improvement programs focus on key solutions which can be used by all projects. This way e.g. design practices are exchanged, best practices established, and rolled out and coached to all projects. Thus if you have e.g. 30 customer assessments you do not improve 30 times with project specific budgets, you share and find a proper solution and roll it out to 30 projects. This thinking helps to focus the investments for improvements.

In this paper we describe how such a working group in SOQRATES (a set of shared task forces in German, Austrian and French industry) has set up a team to share best practices in design and how we elaborated this in a learning environment which has started to be rolled out organisation wide.

Keywords

Process Improvement, Learning Organizations, Integrated SPI Learning Strategies, Experiences

1 Introduction & History

In 2003 the SOQRATES initiative has been kicked off supported by the Bavarian software initiative and ISQI. In SOQRATES cross company task forces have been set up to develop key knowledge for industry in the areas of system design, test, requirements management, agile development and functional safety [1], [2], 4], [5]. Since now 7 years the teams work together and deliver annual knowledge releases.

The team about “Systems Design” has been set up in 2003 and the team about “Functional Safety” started in 2006.

What makes the working groups special is the fact that we exchange best practices, base on experiences that really worked out and create concepts which are proven in use and also fit to the overall principle of Automotive SPICE and ISO 15504 [2],[4]. Thus instead of copy and pasting e.g. a standard guideline and book for re-usable design we rather made a research what worked in the leading firms, and then created our own library of best practices. Another difference to the generally known SPI papers and approaches is that we do not restrict ourselves to software development, but we extended the principles to systems and product development concepts.

Till 2008 best practices agreed were shared in a team working portal. In 2009 we developed a new strategy where the task forces create learning components in an e-learning portal so that people from the work place of the participating firms can attend learning sessions. This learning system know how has been provided by the EU project “ELM E-Learning Manager” [10]. This way the system design practices agreed among the members became available in online courses.

In 2010 the SOQRATES [8] partners shared a strategy with the “Integrated Designer Project” [10] which is supported by the EU and runs under the umbrella of the ECQA European Certification and Qualification Association. This allowed us to define the elaborated best practices as modules of a Europe wide qualification. To assure privacy of the working party members only principles were shared but the content which is knowledge of our working group has not been shared. This way the SOQRATES members had a chance for qualification and certification free of cost, unlimited access to the knowledge the group elaborated, and while we shared principles, the concrete success examples remain an ownership and in privacy for the group.

2 Learning Strategy

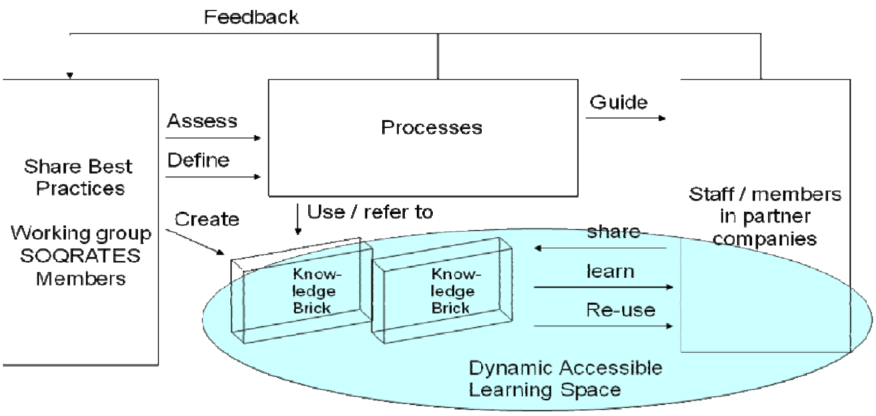


Fig. 1. Strategy - Sharing and Learning Best Practices

Processes describe objectives, roles, activities, results, etc. Knowledge bricks describe real useful and practice oriented work patterns which help to fulfill the technical and process goals at the same time. This synergy is a major driver that “theoretical” processes get acceptance by “practical” staff [2], [4].

3 Implementation Example – System Design Best Practices Transfer

The working parties define goals per year and create a knowledge release per year.

In the system design task force the following main topics had been elaborated until 2008.

- System design and Requirements Traceability
- Re-Usable Design
- Design Metrics
- Using SysML for systems design
- How to measure quality of a design

In 2009 the functional safety task force the following main topics had been elaborated until 2009.

- Additional requirements from IEC 61508
- Additional requirements from ISO 26262 [7]
- Integrating Automotive SPICE with IEC 61508 and ISO 26262 [7], [8]
- Creating an integrated assessment portal for SPICE and safety

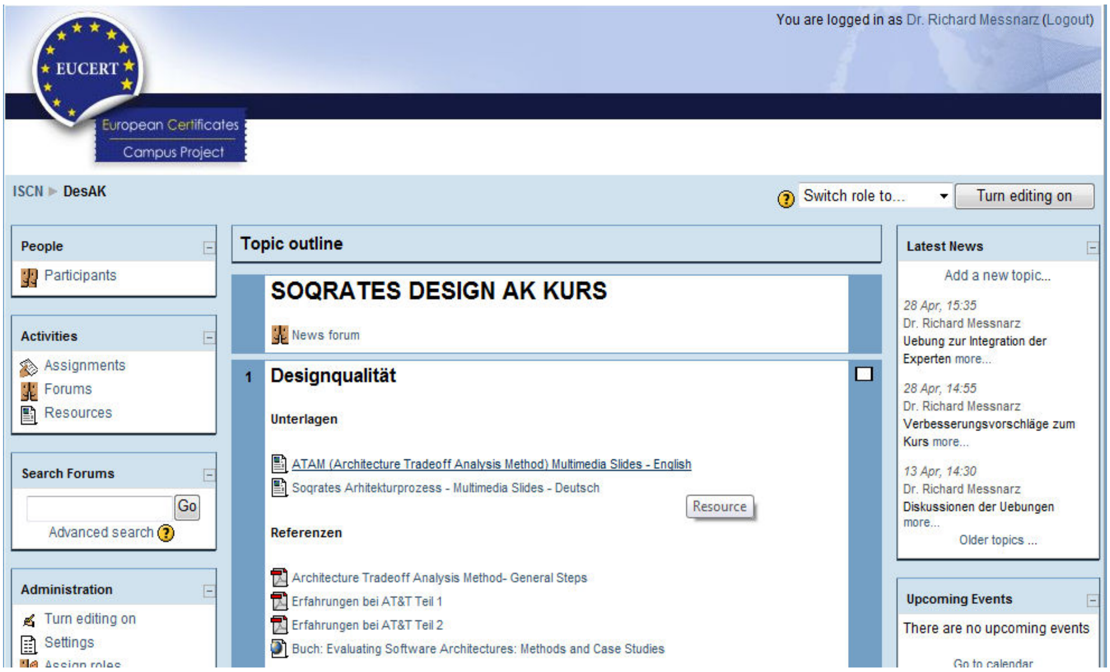


Fig. 2. Learning system for Best Practices

Since 2010 both working parties joined forces and elaborated on two main topics

- Integrating previous knowledge in learning portals and rolling the knowledge out with qualifications / certifications supported by ECQA
- Elaborating on “How to measure complexity on products, systems, and software level”. So far most measures which are published relate to software.

From April 2010 onwards the knowledge releases were available in this online knowledge system.

3.1 Learning Steps

In general each element was taught as follows [10]:

- Step 1

  - Presentations in Graz to those onsite
  - Those in Frankfurt, Erlangen and Nuremberg attended the online multimedia course element
- Step 2
  - Virtual discussion of the materials
  - Discussion of exercise and homework for 2 hours
- Step 3
  - Each company applies the principles on their own system and publishes an example
  - All meet on Skype and discuss the exercise results
  - Trainer concludes the lessons learned

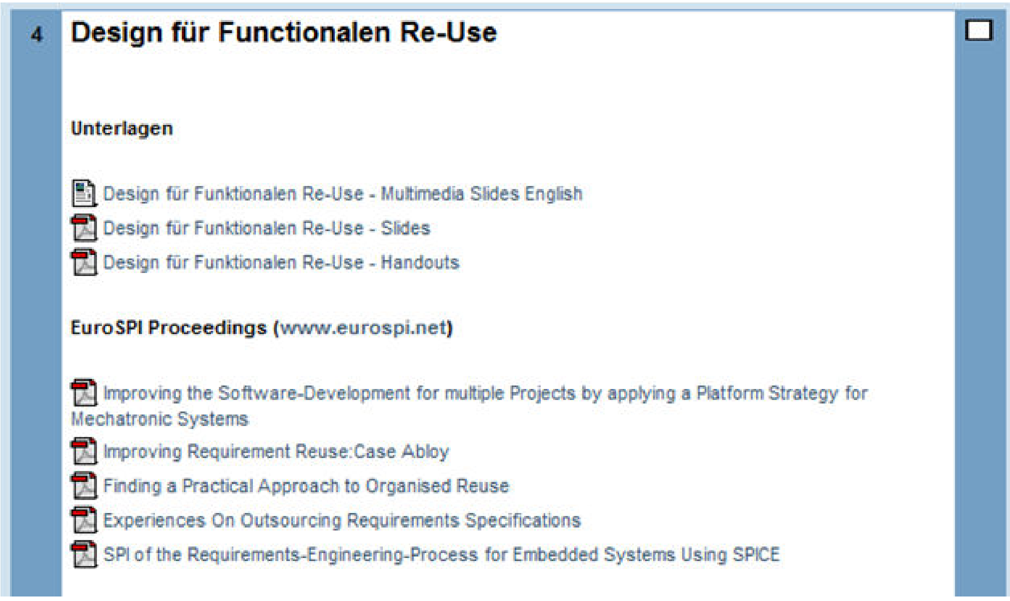


Fig. 3. Course – Functional Re-Use

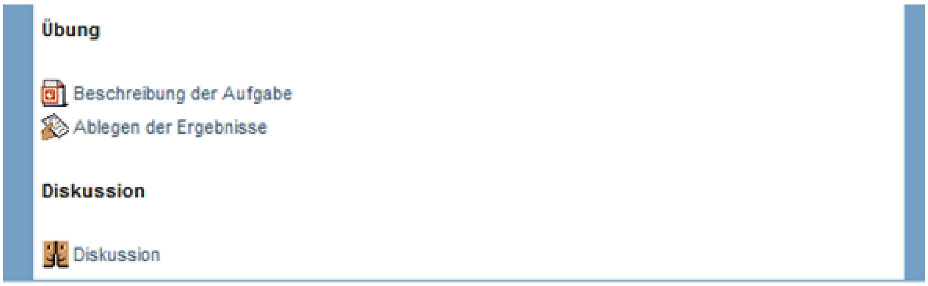


Fig. 4. Exercises – Functional Re-Use

The exercise included the example definition of reusable requirements and functional components for the automotive systems represented by the attendees.

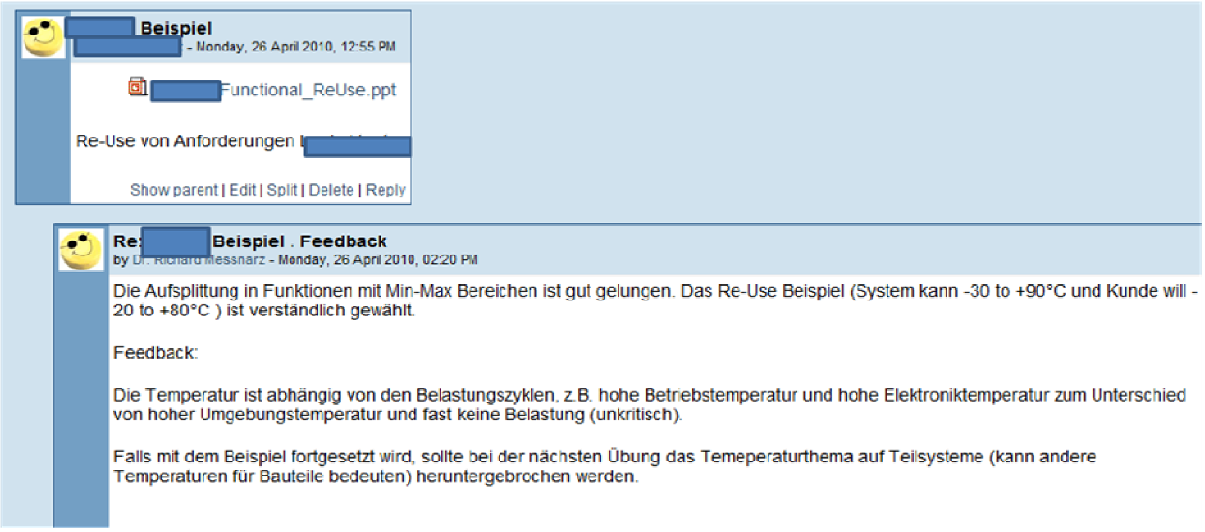


Fig. 5. Feedback – Functional Re-Use

3.2 Knowledge Roll Out

The following learning modules have been created in SOQRATES:

- Design Quality
- Design and Functional Re-Use
- Design and Requirements Coverage
- Functional safety Design

Within 2010 and latest till beginning of 2011 the following learning modules will be added:

- Design Using SysML (Elaboration of knowledge from previous years work)
- Measuring design complexity on product, systems, and software level.
- Combined Safety (IEC 61508 / ISO 26262) and SPICE Assessments plus Tool Training

Two online courses have been performed with leading SOQRATES members like staff from Continental Automotive, ZF Friedrichshafen AG, Magna and a set of middle sized electronic and service companies.

Within 2010 a trail in-house in a world wide active supplier take place. We plan to demonstrate the roll out of SOQRATES task forces knowledge into work places using this new ways of learning.

3.3 Certification for Design AK Skills

In SOQRATES we do not only share and now roll out best practices using new was of learning (where all can access from the work place) but we created the access to certification of staff based on a Europe wide level (recognised in 18 countries).

In the iDesigner project within the ECQA (European Certification and Qualification Association) we defined learning goals for the following elements in systems and product design:

- **Managing Complexity in Systems Design**
  - Design Quality
  - Design and Functional Re-Use
  - Design and Requirements Coverage
  - Functional safety Design
  - Design Integration
  - Design Innovation
- **Understanding Product Life-cycle Engineering**
- **Working in Distributed Engineering Teams**
- **Life-cycle Assessment in Integrated Design: Disassembly and Recycling**
- **Use of VR (Virtual Reality) Technology for Design Support**

The certificate for the iDesigner includes five main areas and elements per area. The SOQRATES modules refer to these elements and currently cover 4 of them.

The iDesigner training is a separate environment while the SOQRATES learning portals (for the elements we cover) contain same principles but more/additional practice and experiences.

This separation protects the interests of SOQRATES members.

However, basing on the same vision allows us to prepare own staff for the European certificate and offer training in-house online free.

The iDesigner project is largely driven by French engineering companies with contributions from a Europe wide manufacturer network including German members.

3.4 Advanced Certification for the Improvement Managers

In the ELM (E-Learning Manager) [10] project within the ECQA (European Certification and Qualification Association) we defined learning goals for improvement managers to use such learning systems to roll out best practices.

All SOQRATES members were offered free access in 2010 to this training and certificate. Here we learned the following topics:

In the management unit they improvement managers learn methods and practices to establish learning based best practice sharing and perform three major exercises where tools and examples are provided:

- **Exercise 1**: Select a recent problem the management wants to solve and where sharing of knowledge is needed. Argue an e-learning solution and analyse ROI scenarios.
- **Exercise 2**: Select your own organisation and try to define the learning process and position the e-learning methodology and technology as part of the strategy. Prepare this as a presentation to convince the manager to also include e-learning functions in the overall strategy of the company. Upload this in the discussion forum and discuss it with the trainer.
- **Exercise 3 (optional)**: Perform an e-learning capability assessment (EMM Model) and

conclude about an improvement plan.

In the technology unit the attendees learn how to implement the best practices in such a learning format.

- **Exercise 1**: Each participant was trained to create their own SCORM compliant package and upload this to the training environment.

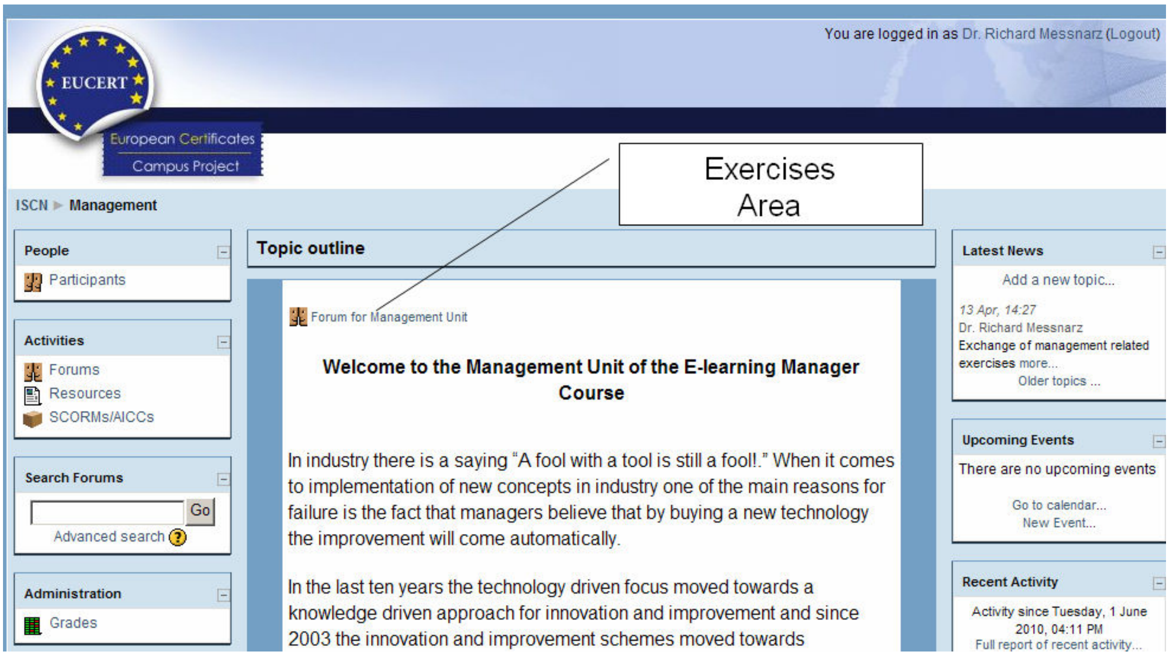


Fig. 6. Learning Manager Skills

3.5 Automotive SPICE Level 3 Strategy

In the SOQRATES [6], [8], [11] partnership the idea is (see Fig. 1 before) that a level 3 is not only a strategy for quality management. ISO 15504 [11] in the section related processes illustrates that level 3 is also related to Knowledge Management, Asset Management and Domain Engineering (see Fig. 7).

These processes are part of the ISO 15504-5 assessment model but have not been selected by Automotive SPICE. However, the SOQRATES group philosophy is that exactly these three processes assure that level 3 is a long term knowledge based approach which brings value to the form and is not just done to satisfy for a specific project a specific customer.

	Process attributes							
Related processes	PA 2.1	PA 2.2	PA 3.1	PA 3.2	PA 4.1	PA 4.2	PA 5.1	PA 5.2
SUP.1 Quality assurance	◆	◆						
SUP.2 Verification		◆						
SUP.4 Joint review	◆	◆						
SUP.5 Audit			◆	◆				
SUP.7 Documentation		◆						
SUP.8 Configuration management		◆						
SUP.9 Problem resolution management	◆	◆						
SUP.10 Change request management		◆				◆		
MAN.1 Organizational alignment						◆	◆	◆
MAN.2 Organization management			◆	◆				
MAN.3 Project management	◆			◆				
MAN.4 Quality management				◆	◆	◆		
MAN.5 Risk management	◆				◆			
MAN.6 Measurement				◆	◆	◆	◆	
PIM.1 Process establishment			◆					
PIM.2 Process assessment				◆	◆	◆	◆	
PIM.3 Process improvement				◆			◆	◆
RIN.1 Human resource management	◆		◆	◆				
RIN.2 Training				◆				
RIN.3 Knowledge management			◆				◆	
RIN.4 Infrastructure			◆	◆		◆		
REU.1 Asset management			◆					◆
REU.3 Domain engineering			◆					◆

Fig. 7. Related Processes

4 *Experiences and Feedback*

Concerning the Design AK modules the general the feedback was very positive and the teams actively collaborated. They also delivered a number of improvement comments.

- We should split between principle slides and case study. The case study slides can be exchanged per application area.
- All presentations should have same/similar multimedia style.
- The exercises and the discussion of exercise results is the highest value, so time for exercise discussions must be prolonged.

In future we plan that in all partner firms the best practices are available online in form of training and coached by practical exercises and examples. People do not travel, its just part of their work environment.

We realised that moving into this way of knowledge sharing requires an upgrade of skills of improvement managers. First they were sceptic but once they realise that such a learning module with nowadays available systems can be established in a few days, they made the exercises and plan now to use it.

Level 3 is more than a defined set of processes. It's a platform of knowledge with shareable best practices that form a standard way of working. Practice shall be the basis of a standard implementation.

Learning nowadays has changed as well. Like we use Google to answer some general questions, we can use such best practice knowledge transport to answer technical and process related issues.

Acknowledgements & SOQRATES membership

We are grateful to ISQI who supports the SOQRATES teams with a team working infrastructure and has helped to kick off the task forces idea in 2003. We are grateful to the experts who have contributed to the Design AK and Safety AK: S. Habel, Leo Ross, R. Dreves (Continental Automotive), F. König, A. Koundoussi (ZF), G. Spork (Magna Powertrain), M. Haimerl (IMBUS), K. Dussa-Zieger (Methodpark), A. Riel (Grenoble INP), J. Unterreitmayer (SQS), and D. Ekert, R. Messnarz (ISCN).

If you want to join the working party please contact Dr Richard Messnarz as moderator of the task forces at rmess@iscn.com. Also please note that the participation requires a commitment to participate in 2 general meetings and additional 4 working meetings per year.

References

[1] H. Höhn, B. Sechser, K. Dussa Zieger, R. Messnarz, B. Hindel, Software Engineering nach Automotive SPICE, Entwicklungsprozesse in der Praxis – Ein Continental Projekt auf dem Weg zu Level 3, dpunkt Verlag, 2009

[2] R. Messnarz, et. al, Assessment Based Learning Centers, in : Proceedings of the EuroSPI 2006 Conference, Joensuu, Finland, Oct 2006, also published in Wiley SPIP Journal (Software Process Improvement in Practice), Volume 12 Issue 6 , Pages 505 – 610, November/December 2007

[3] R. Messnarz, et.al, Better Software Practice for Business Benefit, IEEE Computer Society Press, 1999, Washington, Tokyo, Berlin

[4] R. Messnarz, From process improvement to learning organisations (p 287-294), Wiley SPIP Journal (Software Process Improvement in Practice), Volume 11 Issue 3 , Pages 213 - 335 (May/June 2006)

[5] SOQRATES Safety Team, Richard Messnarz, Hans-Leo Ross, Stephan Habel, Frank König, Abdelhadi Koundoussi, Jürgen Unterrreitmayer, Damjan Ekert, Integrated Automotive SPICE and safety assessments (p 279-288), in Wiley SPIP, Volume 14 Issue 5, September 2009

[6] Automotive SPICE, www.automotivespice.com

[7] ISO 26262, Road vehicles — Functional safety

[8] SOQRATES Initiative, www.sowrates.de

[9] HIS, www.his-automotive.de

[10] ECQA European Certification and Qualification Association, www.ecqa.org

[11] ISO 15504 – 5, Exemplar Assessment Model

Factors that contribute to the effective management of Global Virtual Teams

Javier García, Javier Saldaña, Antonio Amescua, Ana Sanz
Computer Science Department. Carlos III University of Madrid
Avda. de la Universidad, 30, 28911 Leganés (Madrid)
jgarciag@inf.uc3m.es, jsaldana@inf.uc3m.es, amescua@inf.uc3m.es, asanz@inf.uc3m.es

Abstract

The management of globally distributed software teams is complex because of problems of linguistic differences, geographical dispersion, different time zones and the cultural diversity of the team members. These problems are amplified when a single software development team is composed of highly skilled individuals working in dispersed geographical locations.

In this paper, ten of the most important factors that contribute to the correct and effective management of global virtual teams are presented. These factors are obtained from an industrial case study, which lasted 36 months, corresponding to a huge software development project that involved several global virtual teams.

Keywords

Geographically distributed software development, management of IT function, success factors, virtual team efficacy, team processes maturity, software team management

1 Introduction

Nowadays, outsourcing software activities (development, test, maintenance, programming, and incidence management) are indispensible for maintaining the required levels of competitiveness and productivity in large software engineering projects (Herbsleb and Moitra 2001; Sengupta et al. 2006). This situation requires the creation of multidisciplinary teams composed of people working in different locations and on the same software development project.

There are several strategies to configure this type of team depending on the existing interdependence among team members. In some projects, management, requirements specification and architectural design done in one location, and the development or modification of software components by software factories set up in separate geographical locations (Edwards and Sridhar 2003; Smite 2004).

Other projects, however, require the creation of several coordinated multidisciplinary teams, bringing together highly-skilled individuals working in dispersed geographical locations. The work presented in this paper is especially centred on this type of team, called *virtual team* (Cohen & Gibson 2003).

For software development, the specific characteristics that differentiate global virtual teams from traditional ones are:

1) Each team member can work in a different location; members carry out independent tasks across locations (Hyrkkänen et al. 2007).

2) Team members come from several cultures and have different teamwork practices (Powell et al. 2004). There is a great potential for conflict in global virtual teams as members work across cultural, geographical, and time boundaries (Kankanhally et al. 2007).
3) Team members speak several languages. Although English is commonly used, there are different levels and flavours (Gaudes et al. 2007).
4) Knowledge management is different as knowledge is distributed among the different sites (Striukova and Rayna 2008).
5) Communication activities and meetings rely strongly on technologies. So, more time is needed to obtain a common view of the goal and to determine how to achieve it (Petkovic at al. 2006).
6) The time difference makes agenda management more difficult; team work dynamics and the different location work dynamics need to be combined (Lee-Kelley and Sankey 2007).

Many organizations that have implemented Global Software Development have found thatglobal virtual teams are highly complex (Herbsleb and Moitra 2001). This complexity comes from the additional challenges they face: a) Lack of common understanding of goals and requirements assigned to the team that makes team members feel isolated, and they are reluctant to collaborate, share and work together (Striukova and Rayna 2008); b) Difficulties in communication among different team members who are geographically distributed (Fuller et al. 2007); c) Differences between processes, management mechanisms and the associated skills and abilities create problems and bottlenecks in the project execution. (IPRC 2007); and d) Ineffective management of shared knowledge among different team members causes duplication, inconsistencies and lack of knowledge of project assets (Rosen et al. 2007).

To conclude, it can be stated that global virtual and traditional team management are different because of the nature of the teams. Consequently, traditional methods and techniques alone cannot be used to manage global virtual teams since they do not support issues, such as collaboration or communication in geographically dispersed environments. Additional elements need to be considered to manage this type of team, for example:

- Explicit team management practices because they are implicit in traditional teams. However, with virtual teams, these practices should be defined and published for team members (Powell et al. 2004).
- Collaborative work environments have to be available in geographically distributed teams since communication and collaboration rely heavily on technology in these environments (Gaudes et al. 2007).
- Efficient practices for shared knowledge management should be incorporated in distributed environments (Thomas et al. 2007).

A single integration between the previous elements and technical software development activities is necessary to manage the global software team properly (Prikladnicki et al. 2003; Sengupta et al. 2006).

In order to deal with the previously-mentioned problems, it is necessary to identify efficient practices that can help to deal with them, and the key success factors that should be considered to manage effectively global virtual teams for software development (Powell et al. 2004). This work presents a set of key factors identified by the authors of this paper through their experience and participation in four software projects developed by global virtual teams.

2 Description of the case study considered for this work

The goal of this case study is to identify the key success factors for managing global software development teams in which team members are also geographically distributed. For this case study, the authors followed the recommendations defined by (Kitchenham and Pickard, 1998).

This case study is based on the global software development activities considered in the C@R project

(C@R 2005). C@R was a huge project, with a budget of €12 million, which lasted 36 months and involved people from 30 different countries from 2006 to 2009. C@R's objective was to develop software tools (SCTs) to solve problems in rural environments. These problems were related to collaborative logistics, procurement, selling, geographic information systems applied to business incubation, process improvement, rural tourism and territorial planning. Fig. 1 shows the structure of C@R's project teams. From an organizational point of view, this team structure is representative of software development projects (Wallace 2007), but in the C@R project, it was conditioned by the geographically dispersed team members.

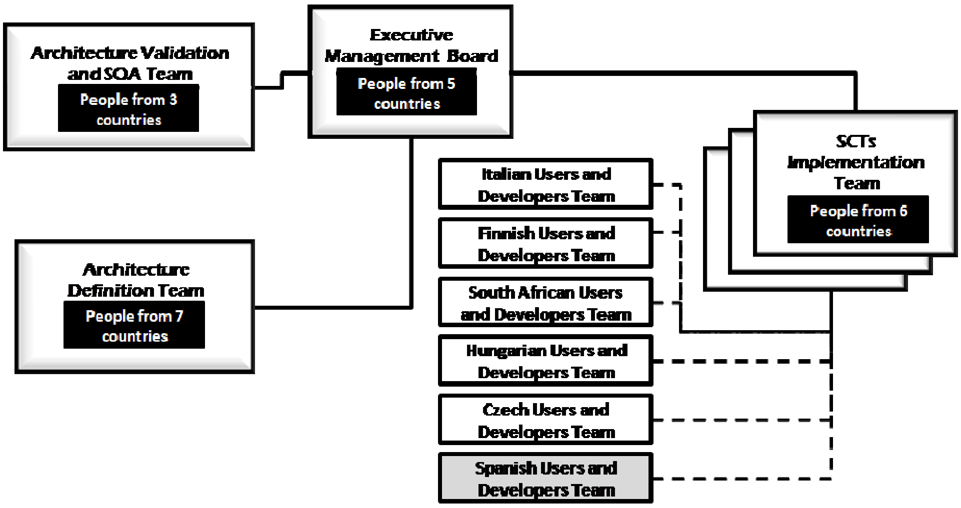


Fig. 1: Team organization in the pilot project

Each team was composed of groups of people from different countries and the members could participate in different teams. The SCTs implementation team was a different case. This team was made up of other six sub-teams working independently to develop different SCTs, but under the same architecture. For this work, only the Spanish Users and Developers Teams were selected because the authors of this paper could not participate actively in the others. Although the architecture validation and the Software Quality Assurance (SQA) teams were different, they were considered a single team for validation purposes because the SQA team was made up of members of the architecture validation team. Some characteristics of the different teams that affected the case study were: types of goal(s) (technical, scientific or administrative), size of team, time needed to perform the tasks, experience and previous training in this type of distributed environment, how critical the success of the work performed was, and cultural and linguistic diversity of team members.

Some authors of this paper took part in the pilot project, so the required resources to organize and run the case study were available.

Three methods were used to collect data. The first, *participant observation*, as defined in (Taylor and Bogdan 1984), refers to "research that involves social interaction between the researcher and informants in the milieu of the latter, during which data are systematically and unobtrusively collected." This method was used to observe how different teams work in the distributed context, the problems they face and the actions taken to overcome them.

The second method was *gathering objective evidences* that demonstrate the execution of each global virtual team management activity - identification of the results obtained, e-mails, minutes of the meetings and logs of the supporting tools used to manage the global virtual team.

The third method was *interviewing,* which was used to collect information and the opinions of team members about their feelings, satisfaction degree, problems and other issues related to working in a global virtual team. Interviews were also used to corroborate if the impressions obtained by the observers were right. Annual interviews were carried out to confirm that the evidences collected were objective, to ensure that the evidence was representative to determine the degree of execution of an activity, and that the complete information was consistent.

3 Reference model to classify the Factors that contribute to the effective management of Global Virtual Teams

Hertel's Five Phases Model (Hertel et al. 2005) was selected to structure the information already available in the literature because it provides a more structured view of the lifecycle of a global software team. This model (Fig. 2) distinguishes five main phases with specific management tasks that have to be addressed in the course of virtual teamwork.

Fig. 2: the Hertel's Five Phases Model

The main goal of each phase is defined below.

a) Preparation: The objectives of this phase are to define the team structure and goals, recruit team members based on the skills identified and the technological support needed to work in a geographically distributed environment.

b) Launch: The objectives of this phase are to develop general rules for teamwork, facilitate communication among team members, define tasks based on a high-level definition of software architecture, assign roles and responsibilities to team members and define the strategy for monitoring and tracking.

c) Performance Management: This phase comprises the period from start to finish of the technical work to be developed. The main goal in this phase is to work efficiently while maintaining the atmosphere of a constructive team.

d) Team development: The objectives of this phase are to execute team assessment and training activities.

e) Disbanding: The goals of this phase are to disband the global virtual team and decide how to reintegrate the members into other teams.

4 Key Success Factors

In the following subsections, the factors identified are presented. However, they are not ordered by degree of importance or relevance.

A) Factors related to global virtual team preparation

The factors considered in this section are related to the determination of the facilities to support the team activity, the composition of the team and the main goals.

FACTOR 1	
Statement	**Collaborative identification of the global virtual team mission**
Problem to solve	Team and individual objectives and goals have to be described. These objectives and goals form the basis to establish the strategy and plan.
Solution proposed / applied	It is recommended to create and manage a distributed requirements list using discussion and vote tools, both synchronous and asynchronous, to estimate and prioritize each item. Moreover, the application of collaboration patterns such as the multimedia meeting room, which provides team members with videoconference and application sharing, and the distributed vote tool are also useful to determine team mission.
FACTOR 2	
Statement	**Identification of the appropriate team members for a specific role and/or responsibility**
Problem to solve	Several dimensions have to be considered. On the one hand, professional and technical competencies such as skills, abilities or knowledge; and on the other hand, general attributes related to three dimensions: general cognitive abilities, taskwork-related attributes (conscientiousness, integrity) and teamwork-related attributes (emotional stability, agreeableness). This problem is

	more difficult in global virtual teams because the skills and competencies to the members are not known and there is no personal contact to obtain tacit information on inter-personal information and collaboration skills.
Solution proposed / applied	In order to solve this problem, the authors found it necessary to take advantage of two approaches. First, it is suggested to develop a competence tree. A competence tree is a tool that visualizes operational and core competencies available in an organization in a hierarchical connection with the corporate vision and the success factors in a competitive domain (Comi and Eppler 2009). It also allows to combine allocation of individual responsibilities with the skills and abilities available in the team (operational and core competences). Second, it is also recommended to use shared blackboard mechanisms in which the team leader defines the skills and abilities each potential team member requires. In this way, the personnel are selected according to these required responsibilities and competencies and the objective evidences of their curriculum vitae that show they have the right skills or abilities.
FACTOR 3	
Statement	**Selection and adaptation of the appropriate technology to enable interaction between the global virtual team members.**
Problem to solve	Nowadays, there are several types of tools that support collaborative working environments. In fact, each organization uses the different software providers' solutions. In this sense, the first challenge for a virtual team is to select a set of tools that are interoperable and can be integrated into the technological infrastructure hosting each of the members that compose a single global virtual team. Moreover, the second challenge is to provide an environment of trust and dynamic communication, and mechanisms that support knowledge management.
Solution proposed / applied	Conventional software engineering and programming tools needed to develop specific technical work have to be also selected because they strengthen collaboration in distributed development (Sengupta et al. 2006). To become useful and increase productivity, the candidate tools must quickly reach a critical mass in the project organization. When collaboration tools are used among organizations, different work cultures sometimes lead to communication problems. The experience gathered during the case studies the authors analyzed indicates that electronic meetings using audio and video communication are common nowadays. The essential prerequisite is, however, a high quality audio and video link. The users also found that multipoint meetings are less tiring than point-to-point meetings. The growing use of presence tools, in combination with face-to-face meetings, often has a positive effect on team building. Any problem concerning the availability of the tools should be solved before the launch phase to prevent lack of synergies and communication in the early phases of team building.

B) Factors related to global virtual team launch

The factors considered in this section are related to the definition and agreement of an action plan that allows global virtual teams to achieve the goals defined in the preparation phase.

FACTOR 4	
Statement	**Multidisciplinary and collaborative definition of the technical approach to be implemented.**
Problem to solve	The first activity is to agree on the definition of a high-level architecture of the software to be implemented. In software engineering, this task is usually performed by means of face-to-face meetings in which design diagrams are prepared among team members. In the case of software teams where team members are also geographically dispersed, it is necessary to define effective ways to share design information dynamically.
Solution proposed / applied	During this phase, team members usually experience difficulties in preparing and agreeing on a common solution, using asynchronous (knowledge sharing repositories and off-line message solutions) or synchronous communication (mainly text or audio chats). Several conflicts may arise. In order to ensure that agreement among team members is based on a correctly-shared vision and to prevent future problems, face-to-face meetings enabled by video-conferencing solutions are recommended. In this type of meeting, the level of productivity required is only achieved through the use of/by using pre-established protocols to create and modify diagrams collaboratively during the virtual meetings. For software development global virtual teams, seamless integration of synchronous and asynchronous communication tools with integrated development environments and management tools is also suggested. In order to prevent loss and duplication of knowledge, these tools have to provide warnings when an asset or artifact has been modified by other team members and show what has been modified. It also recommended to create a constructive working atmosphere and to coordinate the work among members of a geographically distributed team.
FACTOR 5	
Statement	**Distributed and multidisciplinary management of the development strategy**
Problem to	After agreeing on the technological approach to be implemented in order to achieve the global

solve	mission, it is necessary to determine the strategy to achieve the general goal of the team. At this point, the type of contribution expected from each participant has to be stated and the types and the expected synergies of the interactions between the different members have to be defined. Finally, the approach to integrate all contributions has to be clearly planned and communicated to all the team members.
Solution proposed / applied	The definition of the development strategy in global virtual teams consists of determining the best technical approach to implementing the previously defined architecture. In these software engineering environments, this activity should be completed through meetings that normally cannot be done face-to-face, so the efficient use of a multimedia meeting room is important. The application of structured meeting techniques is also important to assure the productivity of these meetings. The Nominal Groups technique could be useful for this purpose. Moreover, it is recommended to create and manage a distributed task list which includes the task purpose, the person in charge, the estimated total effort, the current state and an estimation of the remaining effort to complete the task. In order to manage this artifact efficiently in a geographically distributed context, it is suggested to use the specific practices that allow shared editing of the items included for registering the current status of each task.

C) Factors related to global virtual team performance management

The factors considered in this section are related to developing the planned tasks and tracking progress against the plan to ensure that the quality levels, schedule and budget are being complied with.

FACTOR 6	
Statement	**Effective leadership of the global virtual team**
Problem to solve	One of the main challenges to be addressed in global virtual teams is leadership. In a global virtual team context, leadership refers to decision-making responsibility as regards project and team management; these decisions are based on the current state of both project and team. Moreover the leader is responsible for creating a common vision among virtual team members, motivating and supporting them to achieve the common goal.
Solution proposed / applied	It is suggested to gather intensively and disseminate relevant information on current performance and synergies. In relation to synergies, real and updated information on social processes established among team members must be obtained. The more fluent the personal relationships among team members, the better they communicate, resulting in greater success. There are many tools that allow registration of this information: the challenge is to integrate them correctly with the team processes and the technological environments implemented in organizations. Strategies to increase motivation among team members by means of rewarding systems and recognition of individual achievements make that team members are determined to achieve the common goal and help the leader to ensure that team members have a common understanding of the objectives to be fulfilled.
FACTOR 7	
Statement	**Effective sharing of the knowledge generated by the global virtual team**
Problem to solve	In order to prevent mismanagement of the knowledge generated in global virtual teams, the management practices to be implemented should enhance equal information distribution and systematic processing of unshared information.
Solution proposed / applied	For efficient implementation of knowledge sharing management, it is appropriate to use tools for shared knowledge management (enabling mechanisms for assets linkage and shared access and use of project assets), and collaborative team and individual progress tracking. In addition, a multimedia meeting room should be available for virtual meetings to enable decision-making processes at task level. Therefore, it should be noted that periodic (virtual and/or face-to-face) meetings among team members strengthen commitment. It is also recommended that each team member take charge of several team management responsibilities, thereby increasing trust among team members and motivating them. These strategies to get an effective sharing of the knowledge generated by the global virtual team are also important to reduce problems caused by differences in culture. It is due to periodic meetings and collaborative tools facilitate the interaction among people reducing the sense of isolation caused by the different languages, customs and ways of work.

D) Factors related to global virtual team development

The factor considered in this section is defining and carrying out a training program to enable team members to work with the collaboration technologies selected.

FACTOR 8	
Statement	**Provision of continuous training on the required capabilities to work in a global virtual team**
Problem to solve	As team members usually have different levels of skills and experience in the collaborative working environment selected, the training program should include communication elements of electronic media, clarification of goals, roles and working in culturally diverse environments.
Solution proposed / applied	This training program usually consists of a first stage training and a second stage coaching, but it can be adapted to the specific characteristics of the project. On completion of the initial training, the team leader must carry out periodic assessments to detect problems that team members experience in working in global virtual teams. If members run into difficulties, the team leader has to establish back-up training activities to reduce these deficiencies. First, an expert coaches team members on the application of one specific technique or tool. Second, guides with back-up training material are hosted in a wiki so that they can be consulted periodically, facilitating the inclusion of tips and actions not to be repeated. Specific skills and capabilities considered for working in global virtual teams are: meetings management, identification of skills, interpretation of curriculum vitae, synchronous and asynchronous communication management in distributed environments, establishment of rules to work with shared information, interpersonal conflict resolution adapted to distributed environments, identification, estimation and prioritization of requirements, requirement management and organization and planning, use advanced techniques of shared collaboration, collection, interpretation and analysis of statistical information and analysis of synergies using information collected from the social networks.

E) Factors related to global virtual team disbanding

The factor considered in this section is deciding on disbanding the geographically distributed software team and how to re-integrate the members into other teams

FACTOR 9	
Statement	**Provision of mechanisms for organizational and individual learning and improvement on completion of the software project**
Problem to solve	In this activity experiences and best practices extracted from the project have to be documented. They can be used in future to build and manage virtual teams. Moreover, training activities have to be assessed to determine if they can be improved. Other project management tasks have to be done, for example revise work to check if it has been finished and register data as organizational assets.
Solution proposed / applied	Team disbanding has to be done quickly, carefully and constructively to maintain a high level of motivation and satisfaction among employees and to facilitate their rapid integration into other work teams. In a global software virtual team, these activities should be performed, but under several conditions, for example registered knowledge must be available to comply with the rules on intellectual and industrial property of all the organizations whose employees participate in the team. The necessary security measures to avoid fraudulent or inappropriate use of the assets must be implemented as well. 'Lessons learned' mechanisms must also be available from the beginning of the project to facilitate continuous improvement of the team. In addition to these tools, wikis and team diary management tools, which can help to identify practices to be used in other projects and those that must be rejected, should be available to team members.

5 Conclusions

This paper, which is based on authors' experience in projects involving global teams, has presented the general problems and challenges that have to be addressed in order to manage effectively a global virtual team for software development. One of the relevant cases is described in this paper. This case study has permitted us to identify ten key success factors to manage effectively global virtual teams and to present solutions to these problems. These solutions are based on the authors' practical experience on the case study presented and other similar projects.

The key success factors that have been presented integrate the technical activities of software development with others, enabling effective management of geographically distributed teams. These practices are oriented towards:

- Managing effective communication among team members, preventing problems related to language, culture and time differences.
- Facilitating the integration of the skills and abilities needed to manage the team during the project.
- Providing effective mechanisms to introduce technology support, essential to the success of global software development teams.
- Incorporating effective mechanisms for shared knowledge management in distributed environments.

The research that the authors have already started is centred on defining a model to assess and determine improvement action lines for global software teams launched and managed in a specific organization.

Literature

Braun, A. (2007). A Framework to Enable Offshore Outsourcing. In Proceedings of the 2nd International Conference on Global Software Engineering (pp. 125-129). IEEE Computer Society Press.

C@R Consortium (2005). C@R: A Collaboration Platform for working and living in rural areas. Retrieved from EU project with reference FP6-2005-IST-5-034921, http://www.c-rural.eu. Accesed 25 May 2008.

Cohen, S. G., and Gibson, C. B. (2003). Putting the Team Back in Virtual Teams. In the 18th Annual Conference of the Society for Industrial/Organizational Psychology, Orlando.

Comi, A. and Eppler, M.J. (2009). Visualizing Organizational Competences: Problems, Practices, Perspectives. In Proceedings of I-KNOW '09 and I-SEMANTICS '09 (pp. 115-127). Semantic Web Company.

Durnell, C., Webber, S. (2005). Relationships among geographic dispersion, team processes, and effectiveness in software development work teams. *Journal of Business Research*, 58(6), 758-765.

Edwards, H.K., Sridhar, V. (2003). Analysis of the effectiveness of global virtual teams in software engineering projects. In Proceedings of the 36th Hawaii International Conference on System Sciences. IEEE Computer Society Press.

Fuller, M., Hardin, A., Davison, R. (2007). Efficacy in Technology-Mediated Distributed Teams. *Journal of Management Information Systems*, 23(3), 209-235.

Gaudes, A., Hamilton-Bogart, B., Marsh, S., Robinson, H. (2007). A Framework for Constructing Effective Virtual Teams. *The Journal of E-working*, 1(2), December, 83-97.

Herbsleb, J., Moitra, D. 2001. Global Software Development. *IEEE software* 18(2), 16-20.

Hertel, G., Geister, S., Konradt, U. (2005). Managing Virtual Teams: A review of current empirical research. *Human Resource Management Review*, 15(1), 69-95.

Hyrkkänen, U., Putkonen, A., Vartiainen, M. (2007). Complexity and workload factors in virtual work environments of mobile work. *Ergonomics and Health Aspects*, 4566, 85-94.

IPRC (2007). *A Process Research Framework*. Pittsburgh :Software Engineering Institute.

Kankanhalli, A., Tan, B., Wei, W. (2007). Conflict and Performance in Global Virtual Teams. *Journal of Management Information Systems*, 23(3), 237-274.

Kitchenham, B.A. and Pickard, L.M. (1998). Evaluating software engineering methods and tools: part 9: quantitative case study methodology. *ACM SIGSOFT Software Engineering Notes*, 23(1), 24-26.

Lee-Kelley, L., Sankey, T. (2007). Global virtual teams for value creation and project success: A case study. *International Journal of Project Management*, 26(1), 51-62.

Petkovic, D., Todtenhoefer, R., Thompson, G. 2006. Teaching Practical Software Engineering and Global Software Engineering: Case Study and Recommendations. In Proceedings of the 36th ASEE/IEEE Frontiers in Education Conference (pp. 19-24). IEEE Computer Society Press.

Powell, A., Piccoli, G., Ives, B. (2004). Virtual teams: a review of current literature and directions for future research. *ACM SIGMIS*, 35(1), 6-36.

Prikladnicki, P., Nicolas, J.L., Evaristo, R. 2003. Global Software Development in Practice. *Software Process Improvement and Practice*, 8(4), 267-281.

Ramasubbu, N., Krishnan, M.S. (2005). A Process Maturity Framework for Managing Distributed Development. *IEEE Software*, May/June 2005, 80-86.

Rosen, B., Furst, S., Blackburn, R. (2007). Overcoming Barriers to Knowledge Sharing in Virtual Teams. *Organizational Dynamics*, 36(3), 259–273.

Sengupta, B., Chandra, S., and Sinha, V. (2006). A Research Agenda for Distributed Software Development. In the Proceedings of the 28th international conference on Software engineering (pp. 731-740). ACM Press.

Smite, D. (2006). Global Software Development Projects in One of the Biggest Companies in Latvia: Is Geographical Distribution a Problem?. *Software Process Improvement and Practice*, 11(1), 61-76.

Striukova, L., Rayna, T. (2008). The role of social capital in virtual teams and organisations: corporate value creation. *International Journal of Networking and Virtual Organisations*, 5(1), 103-119.

Taylor, S.J., and Bogdan, R. (1984) Introduction to Qualitative Research Methods. New York: John Wiley and Sons.

Thomas, D.M., Bostrom, R.P., Gouge, M. (2007). Making Knowledge Works in Virtual Teams. *Communications of the ACM*, 50(11), 85-90.

Wallace, S. (2007). The ePMbook. http://www.epmbook.com/ Accessed 17 September 2007.

Yin, R.K. (1994). *Case Study Research: Design and Methods*. Sage Publications: USA.

Author CVs

Javier García Guzmán

Javier García Guzmán is an associate professor in the Computer Science Department at Carlos III University of Madrid. He has 14 years' experience as a researcher at Carlos III University and is a consultant in public and private companies. He received his PhD in computer science from the Carlos III University of Madrid. His current research areas are business and software process management, effective integration of business strategy with software process improvement and software measurement, user and open driven innovation, Global Software Development: Methods, techniques and tools and Organizational Management of Software Test Factories. He has publised several research articles, book chapters and conference papers in these areas.

Javier Saldaña Ramos

Javier Saldaña Ramos holds an Engineering degree in Computer Science at Carlos III University of Madrid, and an MSc in Computer Science and Technology from the same university. He is a PhD student and associate professor in the Computer Science Department at the Carlos III University of Madrid. His research on software engineering is focused on software process improvement through teamwork process improvement. It involves team management improvement, virtual teams management improvement and the use of TSP (Team Software Process) in teamwork. He has several years of experience as a software engineer.

Antonio Amescua Seco

Antonio de Amescua Seco is a full professor in the Computer Science Department at Carlos III University of Madrid. He has worked as a software engineer in a public company (Iberia Airlines) and in a private company (Novotec Consultores) as a software engineering consultant. He founded Progresion, a spin-off company, in order to offer advanced Software Process Improvement Services. He received his PhD in computer science from the Universidad Politécnica of Madrid. His research interests include Software Process Improvement, Software Reuse, Software Project Management.

Ana Sanz Esteban

Ana Sanz Esteban holds an Engineering degree in Computer Science at Carlos III University of Madrid, and Msc in Computer Science and Technology by the same university. She is a PhD student and associate professor in the Computer Science Department at the Carlos III University of Madrid. Her research in software engineering is focused on software process improvement, especially test process improvement. It involves different maturity models related to organizations in general, for example CMMI, and with test process such as TMM and TPI.

Lessons in Global Software Development – Local to Global Transition within a Regulated Environment

Oisín Cawley, Ita Richardson

Abstract

In this paper we use an established framework as a focusing tool to provide a summary of the first author's experiences of GSD in an "Information System" (IS) team of a global supply-chain-management company operating within a regulated environment. While the transition to a GSD configuration was transformational, and introduced an exciting multi-cultural working environment, we discuss several problematic areas and the processes which were employed to successfully resolve them. Some focus is also given to the economic recession, how that has shaped the GSD configuration within the company, and again the processes implemented to meet the changing team dynamics.

Keywords

Global Software Development, Global Transition, Regulated Environment

1 Introduction

In this report we carry out a reflective analysis of one of the author's industrial experiences in Global Software Development (GSD) over an 11 year timeframe in company MyOrg. There are a multitude of characteristics of GSD but for the purpose of this report we are using the broad definition "*any software development lifecycle activity*" [1]. This covers all activities such as writing a requirements document, developing a piece of software, through to code deployment in a live environment and problem resolution in a production setting. We would also categorise the interaction with the business teams as being a prominent activity in this.

Table 1: An overview of the framework of issues in DD [1]

Process	*Dimension*		
	Temporal Distance	**Geographical Distance**	**Socio-Cultural Distance**
Communication	Reduced opportunities for synchronous communication, introducing delayed feedback. Improved record of communications.	Potential for closer proximity to market, and utilisation of remote skilled workforces. Increased cost and logistics of holding face to face meetings	Potential for stimulating innovation and sharing best practice, but also for misunderstandings.
Coordination	With appropriate division of work, coordination needs can be minimised. However, coordination costs typically increase with distance.	Increase in size and skills of labour pool can offer more flexible coordination planning. Reduced informal contact can lead to reduced trust and a lack of critical task awareness.	Potential for learning and access to richer skill set. Inconsistency in work practices can impinge on effective coordination, as can reduced cooperation through misunderstandings.
Control	Time zone effectiveness can be utilised for gaining efficient 24x7 working. Management of project artefacts may be subject to delays.	Difficult to convey vision and strategy. Communication channels often leave an audit trail, but can be threatened at key times.	Perceived threat from training low-cost 'rivals'. Different perceptions of authority/hierarchy can undermine morale. Managers must adapt to local regulations.

While this report is inspired by the work of Lane & Ågerfalk [2], there are some significant differences which, in our opinion, are worthwhile reporting:

1. ***Context*** - The focus here is on projects involving internal Information Systems (IS) teams whereas Lane & Ågerfalk are specifically dealing with "Packaged" teams. In fact they suggest that "*it is likely that application of the GSD framework to IS teams may reveal further insights*".
2. ***Regulation*** - Because MyOrg is a publicly trading US company (NASDAQ listing), it is required to comply with the Sarbanes-Oxley Act of 2002 (SOX) which governs the processes for financial reporting, and therefore the systems and applications which contain and could affect the financial data. This as we report had implications on the software development processes as many of its centres world-wide[1] had to undergo annual audits.
3. ***Transition*** - Our span of review covers a slightly longer time span (11 years) and similar to [3] includes the progression from local site specific IT groups to:
   a. co-located and distributed global development teams operating in parallel
   b. a shared services IT model
4. ***External Contractors*** - The use of onsite and remote contractors at MyOrg depending on the project.
5. ***Recession*** - Experiences as a consequence of a global economic downturn and how that affected the GSD activities.

[1] Only those sites whose turnover exceeded a certain limit were deemed to be in scope for SOX audit.

GSD, has been examined from many viewpoints [1],[4],[5],[6],[7],[8], and cognisant of the critic by [9] of the usefulness of experience reports (albeit for Agile GSD), we believe that this report offers some unique perspectives into the complexities of GSD and implementation and modification of processes specifically for GSD.

2 The Company

MyOrg is a US multinational and is a leader in global supply chain business process management with a focus on the high technology and communications industries. Originally organised as a collection of around 30 international sites in places such as Mexico, United States, Holland, Ireland, China, Japan among others, each performed similar activities but with very different operating procedures, processes and supporting systems. From an Information Technology (IT) point of view, each site originally had its own self sustaining IT department complete with software engineers, technical and application support personnel. This resulted in inefficiencies including duplication of effort, to non-use of best practices and implementation of systems and applications which do not easily support inter-site business processes. Such IT organisation is an anathema to building consistent and reproducible business processes across a global organisation.
Through a series of transformations, over an 11 year period, a true shared services IT model was introduced which included a globally distributed software development team with members located in the United States, Europe and Asia managed from an Irish office. This necessitated a shift in mind-set for the business teams who had to adjust and learn to interact with the IT organisation from the perspective of a true customer as opposed to seeing IT as a readily available extension of their own resources. A form of constructive engagement [4] was utilised in order to instil a new culture necessary to make the globalised and standardised IS function effective.

One of the authors of this paper has a unique perspective on the history of this transformation, having joined the organisation at the start, being an integral part of all stages and finally as the Global Manager of the software development and support groups.

2.1 PACKAGED versus IS Teams

Depending on the context within which a development team operates, they can be referred to as Packaged or IS teams. Packaged teams normally produce an end product. This product is packaged up and sold commercially. IS teams are generally considered to be working internally to support corporate objectives. Carmel & Sawyer [10] state the differences between IS and Packaged software teams include cost pressures versus time-to-market pressures, and bureaucratic versus entrepreneurial cultural milieus. While in general we would agree with these distinctions, software development within MyOrg could be classified as residing somewhere between the two definitions. They believe that *"… packaged software firms function in an environment of intense time-to-market pressure relative to IS development efforts*". However, MyOrg was expected to be operational within the timelines governed by the customer who in turn often operates to their own specific market-driven product release schedules or seasonal consumer activities. From a globalisation perspective, in order to achieve integration, some level of standardisation is required, but, according to [4], the effort for standardisation of packaged teams pales in comparison to the scale of obstacles that a global IS function has to deal with.

The cost and resource pressures which [10] note as being typical of an IS support function were often not relevant in MyOrg. When developing software as part of a new piece of business (directly related to a customer's product), the cost of that development is factored into the outsource contract (either directly or indirectly). The IS function then has the flexibility to look externally for resources and are consequently relatively free of the cost burden. This results in the team becoming more akin to packaged teams developing COTS applications [11].

2.2 Regulation

In the United Sates any publicly trading US company must adhere to relatively new financial accounting and reporting standards as specified in the Sarbanes-Oxley Act of 2002 (SOX)[2]. While the focus of SOX is primarily financial accounting and reporting practices, section 404 of the act stipulates that each company must appoint an internal auditor and perform an annual assessment of the company's controls. A critical element of those controls refers to the ITGCs (Information Technology General Controls) which are intended to ensure Financial data is stored securely, that only the relevant people have access to certain systems and functionality and also that any software/modifications developed which could affect the financial data are developed within a robust and documented software development process. Therefore it was imperative that, within a GSD environment management were confident that each developer, regardless of location, adhered to the internal processes which are aligned with the expectations of SOX.

2.3 Transition: From Self-Sufficient to Shared Services

A key learning from this time period has been the effect on the different business units of moving to a shared-services model of IS development, and this had an impact on how the IS team worked within the global environment. Because MyOrg relied on winning outsource contracts from other companies, what evolved were sometimes similar business processes which were necessarily treated separately in order to satisfy customers' security and reporting requirements. This resulted in small internal business groups each with demands on the same IT resources but each with a customer in the background pushing for deliverables. When this occurred in a global shared-services model with distributed software developers, some issues arose. It became much more difficult to triage the development projects because the demands for resources were coming from multiple sites in different geographical locations with different and often unconnected management structures. From studying the literature on GSD we believe that this aspect has not been fully investigated and warrants further investigation possibly in conjunction with the business research community.

2.3.1 External Contract Staff

MyOrg employed external contractors. However, this resulted in contractors, sometimes in different time-zones, having the required technical ability, but no knowledge of the business processes, user-base or bigger architectural picture. Therefore, they had little hope of giving accurate project estimates. They were also unfamiliar with internal procedures regarding documentation, source control and testing requirements. One particular case ended up costing double the original estimate due to the slow pace of the deliver-test-feedback cycle. Combining this with an unconscious lack of trust due to their remoteness led to a huge overhead in management time and investment shadowing their work. The best method we found to protect against such overrunning development costs was to agree a fixed price up front. We need to remember that contractors are business people who rely on their ability and specialised knowledge to earn a living. It is precisely for this knowledge that we hire them and it would be naive to think that they are not aware of and utilise this information asymmetry to their advantage [12].

3 Some Lessons

Taking the differences discussed into account, we have analysed the experiences within MyOrg using the framework presented in Table 1. Below we describe some of those challenges in more detail and the processes implemented to address them.

[2] http://www.sec.gov/about/laws.shtml#sox2002

3.1 Communication (but more than just with each other!)

Flexible and ad-hoc communication [13] was something that was important within the co-located configuration. When team members had coffee break, unplanned very useful exchanges occurred. This type of chance exchange rarely happened between people physically separated. While information exchanged can be quite trivial it can translate to savings. We noted an example: when discussing a compilation issue which another developer also encountered and which could have taken several hours to resolve, the issue was resolved in the informal discussion.

Remediation:

To resolve the lack of chance exchange problem following GSD implementation, we implemented a process whereby bi-weekly informal conference calls were setup where all the software developers would give an update on their current assignments and discuss any items they wished to bring up. This was particularly useful for those developers who worked on their own and who would have had limited contact with the rest of the group. Weekly conference calls were also established with a more application-support focus with the aim of strengthening team links and fostering collaboration.

3.2 Coordination

3.2.1 Responsibilities

There is no substitute for experience. An interesting point is that the lack of "*Global Project Experience*" [14] (experience of working in a global team context) was something that was very observable in the early stages of moving to the global configuration. In MyOrg, a sort of "out of sight out of mind" attitude was prevalent simply by virtue of having operated as stand-alone departments for so long. This was exacerbated by the fact that MyOrg had multiple core-system instances around the world. For example, when a multi-site project was going live the configuration of the systems would have to be set-up in advance. On several occasions, last minute tweaks to the configurations were made in one site but not to the others, "*I never thought to tell the other sites*" was a common response.

Remediation:

This is something which, with persistence, gets better over time but which also required the implementation of a global project management process with tight coordination across the different sites. An important lesson learned was to delegate responsibility for managing important sub-tasks which would affect multiple instances.

3.2.2 Temporal Dispersion

Another difficulty experienced was the effect of temporal dispersion of GSD. It was very evident and something which really does require a big effort to circumvent. People like to operate within a structured timeframe, and time differences caused some problems, as it put pressure on the developers (and others) to work outside their comfort zone. As an IS support function, the IS team is obliged to support the local sites during their working times. When you are dealing with all 3 global geographical regions (US, EU and APAC[3]) it becomes problematic, especially when the US personnel need to work with people in APAC. In MyOrg's case this lead to three main outcomes:

1. Interaction between teams across the Atlantic got focused towards the end of the European working day and the start of the US day and thus skewed the rhythm of both teams
2. Interaction between Europe and APAC got focussed at the start of the European day and the end of the APAC day
3. Pressure on the US and APAC teams to work unsociable hours in order to overlap
4. Longer resolution times to issues

[3] US: United States of America; EU: European Union; APAC: Asia-Pacific

Remediation:

Temporal issues are extremely difficult to eradicate completely, but implementing processes around working arrangements can assist. For example, at times European developers worked the equivalent of US times to keep a project on track. Due to an asymmetry in knowledge and skills it took a long time before a more “follow the sun” approach could be implemented. Issue resolution was on average longer when dealing in the distributed environment but specific escalation paths were introduced in order to expedite special cases. Educating the internal business community was also required and performed by means of global email communications and site visits. It is also worth noting that having management located in Ireland (a ‘Bridge’) did help alleviate some of the issues since normal working days did overlap between APAC and Ireland and also the US and Ireland [15].

3.2.3 Inter-Site Contact

A source of much frustration for the developers was getting in touch with someone at a remote site. This was typically to help with things like clarifying user requirements, user-testing functionality or carrying out a local installation. This was exacerbated when a requesting party left the company at some point within the project lifecycle – it was often difficult to find a replacement who was knowledgeable enough to take up the project and ensure adequate resources to complete it.

Remediation:

The GSD project management process (described below) ensured that each site appointed a representative who could help get such situations resolved swiftly.

3.3 Control

One of the most problematic areas experienced was around the scheduling and prioritisation of projects. As we said before, operating as a shared service means that the group gets project requests from all corners of the organisation. In a global context that means that your internal customers can be located anywhere and also that different business managers and indeed regional presidents are vying with each other for development resources. Many times this led to conflict and internal management escalations in order to secure resources, sometimes external.

Remediation:

A project review board (PRB) was instigated which consisted of a representative from each site who would attend a weekly conference call and help set the priorities of development projects. It is important to say that these representatives had to have the authority to speak on behalf of their sites which is why it was crucial that the respective General Manager appointed them to the PRB. Overall this worked quite well but it still proved very difficult to get consensus on prioritisation when multiple sites were under pressure to deliver projects within the same timelines.

3.3.1 Code Control

SOX controls are quite strict when it comes to the control of source code and especially the access to deploy code to a system which could impact the financial statements. This ‘segregation of duties’ requirement meant that, for example, a developer was not allowed to have access to a ‘live’ system. As a result only a very limited number of people were allowed to deploy production code or even have administrator access to production systems. For traceability purposes, each code release had to be recorded and available for inspection during audits.

In a GSD context these points were problematic on a number of fronts. Below are some of them and the resolutions implemented:

1. There were certain skills required to check a code release for deployment and keep track of what was deployed in the context of the wider systems and then verify the deployment was successful. In some instances this skill was only available from senior developers.

Remediation:

DBAs (Database administrators) were trained up on the specialised deployment and trace techniques. However in one case, even though the DBA physically carried out the task, it still had to be under the supervision of a senior developer.

2. When second level support was required from a developer, they typically would have used a high level access login to troubleshoot the issue. This login gave them too many privileges on the system (contrary to the SOX guidelines).

   Remediation:

   When required, the application support group (who were allowed to have this level of access), would log the developer in with high level access and record the fact on a report which was archived. This did introduce delays into some issue resolution but this trade off was necessary to satisfy the SOX auditors.

3. The actual code which was being deployed had to be archived with the other project documents and signed documentation had to accompany it. This was then audited on an annual basis. This was problematic for remote developers who had to spend a lot of time getting forms filled and signatures from the relevant parties. This was also a new practice for many in the business community who were used to dealing with onsite developers where informal consent was common practice.

   Remediation:

   As a solution, the process we implemented ensured that we provided a shared network storage area (sharepoint) where all developers could deposit their project documents and from where all audit documentation could easily be retrieved. To ease the effort in obtaining signoffs, the business groups were educated in the requirements of SOX and therefore they understood the necessity and were much more compliant as a result.

3.4 Cost

We believe cost needs to be discussed in the context of regulation and GSD. The cynical amongst us might suggest that the financial consultancy fraternity was behind the implementation of the SOX regulations. A large amount of effort and cost went into developing, coordinating, controlling and auditing the internal practices of MyOrg to meet the SOX requirements. The direct external cost of hiring consultants for dry-run auditing followed by the official auditors has been significant. Typically this might take 1-2 weeks of an auditing company’s time. In our case costs increased dramatically as each of the larger sites had to be audited separately.

Remediation:

We introduced a process which established the position of a global compliance officer. MyOrg fine tuned the internal controls around the software development and support activities so as to only satisfy what was actually required by the SOX regulations. What had preceded this was that the *auditors* decided what they wanted to see and therefore how we had to operate. By reducing this to the exact SOX requirements and making all the documentation available on a central repository, MyOrg reduced the time the auditors needed to spend on site and also the number of controls which failed the tests. This single set of ‘Global’ controls was rolled out to all developers and support members and proved instrumental in moving towards a cross-regional IT function. However, one person spent a substantial amount of time ensuring that all team members, especially external contractors, were following the processes and maintaining the necessary documentation.

3.5 Economic Recession and the race to the bottom?

Thomas Friedman [16] wrote about how the technological advances of the 21st century had aided in ‘flattening’ the economic playing field of global business. For example, the ability of companies to join forces across the globe to create global supply chains that offer efficiencies, economies of scale and reduced operating costs. Even small companies can now compete for contracts which heretofore were only available to larger corporations. But the literature shows us that it sometimes is not so straight forward and the costs savings that are headlined in such outsourcing arrangements are often not reflective of the true costs incurred [17], [18].

When the global economic downturn started to affect MyOrg (2007/2008), international travel was reduced. There were many more areas which were affected such as equipment purchases, salaries, bonuses and support agreements, but we examine travel as a representative example. This resulted in the removal of regular trips to the central European facility for crucial face-to-face project meetings with the business managers. Even more so, these trips supported the process of bringing the distributed IT teams together by reinforcing the direction and structure being put in place. This had some negative consequences.

Due to the decreased management visibility “on the ground” for both team members and local business managers, we instigated processes whereby email communication, especially to the business managers, was intentionally increased, particularly in terms of project scheduling, status reports, initiatives and organisational changes. While there was less opportunity for team members to speak freely to their manager on a face-to-face basis we replaced this with much more frequent (formal and informal) phone conversations and instant messaging. We also appointed of a local (on-site) supervisor to act as a direct management point of contact. However, the reduction of human-to-human interaction, particularly between employee and manager, is a much underestimated and underexplored consequence of GSD in general.

Further, during the annual SOX audit time, IT management was unable to meet with the auditors, therefore audit interviews were performed over the phone. Local group members were appointed as contact points for the auditors and all evidence that the auditors requested was readily available electronically in a centralised location.

As the global recession set-in for a protracted period, senior management decided to implement an offshore outsource software development and support arrangement. An India based outsource provider was selected and a plan put in place to transition activities to them. Existing US and European based employees lost their jobs. It is not our intention to go into the already well published issues this can cause [15], [19], [20], [21] or how risky this can be to a company [22]. Instead, we feel what is worth disseminating is that quite a different version of the original outsource plan actually got implemented:

- More employees were retained than originally expected, mainly due to the gradual acceptance that it was much more difficult to find the exact skill-set required and also that the “tribal” knowledge that these people had built up over the years was quite substantial and probably impossible to transfer
- The transition time was severely underestimated partly due to difficulties in performing the knowledge transfer

An exercise which would be very insightful would be to attempt to calculate the actual cost of the outsourcing plan and what the planned versus actual annual savings turn out to be. The literature is well stocked with research and case studies on this [17], [18], [23] so an investigation into the hidden costs in this case would be enlightening.

4 Summary

The lessons and processes introduced and discussed in section 3 have been summarised into three main categories, Distribution, Regulation, and Recession (see Table 2).

Table 2: Issues and Processes introduced to resolve the problems

CONTEXT	ISSUE	PROCESS
Distribution	Loss of ad-hoc "chatting"	Bi-weekly informal calls
	"Out-of-sight out-of-mind"	Assign a global owner for specific tasks
		Persistence
	Time-Zone Issues	Shift working days as appropriate
		Define Global Escallation Paths
		Educate the Business teams
		Make use of a 'Bridge'
	Scheduling Conflicts	Create a Global Project Review Board (PRB)
	On the ground support	Local PRB Representative
Regulation	Segregation of Duties	Train up DBAs to perform deployments
		Coordinate with app support for 2nd level support access
	Auditability	Centralised repository (sharepoint)
		Educate the Business teams
	Cost	Precise mapping of SOX controls to internal controls
		A common 'Global' set of controls
		Centralised repository (sharepoint)
Recession	Travel Ban	
	-Less mgmt visibility	Consiously do more information pushing
	-Loss of team face-to-face	Consiously do more phone calls and instant messaging
		Appoint a local management contact
	-Audit difficulties	Interviews via conference calls
		Appoint local IT staff as contact points
		Centralised repository (sharepoint)

5 Conclusion

From reviewing our experiences we have shown how GSD introduces an assortment of interpersonal, procedural and organisational challenges. In particular we have shown, that within IS teams, there are different issues that emerge between the IT team and their internal customers. We found that while the existing literature concentrates a lot on the temporal aspects of GSD and how the team members and team management are affected, little focus has been given to how it affects the internal business community and the boundary/interaction with the IT team. We believe that further study is warranted here with a view to establishing an appropriate interface between the two groups in a GSD setting.

We have shown that regulation can be particularly problematic within a GSD setting, and time should be invested in developing and rolling out a common but minimised set of processes globally. The case presented has also been affected by the current recession and decisions to cut back on cost made it more difficult to carry out those activities which researchers have noted help alleviate the very issues faced by GSD teams. We feel that the cost factor has not been given enough attention with respect to how it influences strategies for GSD and the consequent effects on the wider organisation.

MyOrg subsequently undertook to outsource the software development and application support activities to a third party primarily located in India. At a further point in time we think it will warrant a revisit to analyse how this new dimension, that of organisational boundaries [14], has affected the organisation, what issues arose and how they were addressed over time.

Acknowledgements

This research is supported by Science Foundation Ireland (SFI) through Grant No. 03/CE2/I303.1 within Lero - the Irish Software Engineering Research Centre (http://www.lero.ie) and the SFI Principal Investigator Programme, grant number 08/IN.1/I2030.

References

1. Ågerfalk, P.J., et al., *A framework for considering opportunities and threats in distributed software development* in *International Workshop on Distributed Software Development* 2005, Austrian Computer Society: Paris, France. p. 47-61.
2. Lane, M.T. and P. Ågerfalk, *Experiences in Global Software Development – A Framework-based Analysis of Distributed Product Development Projects*, in *Fourth IEEE International Conference on Global Software Engineering*. 2009: Limerick, Ireland.
3. Boland, D. and B. Fitzgerald, *Transitioning from a co-located to a globally-distributed software development team: a case study at Analog Devices Inc.* IEE Seminar Digests, 2004. 2004(912): p. 4-7.
4. Carmel, E., *Global software teams: collaborating across borders and time zones*. 1999: Prentice Hall PTR. 269.
5. Herbsleb, J.D. and D. Moitra, *Global Software Development.* IEEE Software, 2001. 18(2): p. 16-20.
6. Eischen, K., *Andhra Pradesh: Lessons for Global Software Development.* IEEE Computer, 2003. 36(6): p. 31-37.
7. Martignoni, R., *Global sourcing of software development – a review of tools and services*, in *Fourth IEEE International Conference on Global Software Engineering*. 2009: Limerick, Ireland. p. 303-308.
8. Casey, V., *Software Testing and Global Industry: Future Paradigms*. 2009: Cambridge Scholars Publishing. 220.
9. Taylor, P.S., et al., *Do agile GSD experience reports help the practitioner?*, in *Proceedings of the 2006 international workshop on Global software development for the practitioner*. 2006, ACM: Shanghai, China. p. 87-93.
10. Carmel, E. and S. Sawyer, *Packaged software development teams: what makes them different?* Information Technology & People, 1998. 11(1): p. 7-19.
11. Morisio, M. and M. Torchiano, *Definition and Classification of COTS: A Proposal*. 2002. p. 165-175.
12. Lacity, M.C. and L.P. Willcocks, *Interpreting information technology sourcing decisions from a transaction cost perspective: Findings and critique.* Accounting, Management and Information Technologies, 1995. 5(3-4): p. 203-244.
13. Herbsleb, J.D. and R.E. Grinter, *Architectures, coordination, and distance: Conway's law and beyond.* IEEE Software, 1999. 16(5): p. 63-70.
14. DeLone, W., et al., *Bridging Global Boundaries for IS Project Success*, in *Proceedings of the Proceedings of the 38th Annual Hawaii International Conference on System Sciences (HICSS'05) - Track 1 - Volume 01*. 2005, IEEE Computer Society. p. 48.2.
15. Richardson, I., et al., *Having a Foot on Each Shore - Bridging Global Software Development in the Case of SMEs*, in *IEEE International Conference on Global Software Engineering*. 2008: Bangalore, India. p. 13-22.
16. Friedman, T.L., *The World Is Flat-A Brief History of the Twenty-first Century*. 2005: Farrar Straus Giroux.
17. Barthélemy, J., *The Hidden Costs of IT Outsourcing.* MIT Sloan Management Review, 2001. 42(3): p. 60-69.
18. Overby, S., *The Hidden Costs of Offshore Outsourcing.* CIO.COM, 2003.
19. Casey, V. and I. Richardson, *Virtual teams: Understanding the impact of fear.* Software Process: Improvement and Practice, 2008. 13(6): p. 511-526.
20. Cusick, J. and A. Prasad, *A Practical Management and Engineering Approach to Offshore Collaboration.* IEEE Software, 2006. 23(5): p. 20-29.
21. Vedder, R. and C.S. Guynes, *Social considerations for information technology offshoring.* SIGCAS Computers and Society, 2008. 38(4): p. 40-44.
22. Schwartz, E. *Painful lessons from IT outsourcing gone bad*. InfoWorld 2008 August]; Available from: http://www.infoworld.com/d/adventures-in-it/painful-lessons-it-outsourcing-gone-bad-032.
23. Hirschheim, R. and M. Lacity, *The myths and realities of information technology insourcing.* Commun. ACM, 2000. 43(2): p. 99-107.

Author CVs

Oisín Cawley

Oisín Cawley is a doctoral researcher within Lero – the Irish Software Engineering Research Centre in the Department of Computer Science and Information Systems at the University of Limerick, Ireland. He has over 17 years industrial experience in the area of software development and support. His most recent role was Global Applications Development and Support Manager with responsibility for globally distributed IT teams in a US multinational. He also holds an MBA from Dublin City University, Ireland.

Ita Richardson

Dr. Ita Richardson is a senior lecturer with the Department of Computer Science and Information Systems at the University of Limerick. She is Research Area Leader for Process, Practice and Methods within Lero – the Irish Software Engineering Research Centre. Ita's research focuses on software processes in particular domains, such as global software development, regulated industry, services and small to medium sized enterprises. She supervises both part-time and full-time post-graduate students. She was Research Co-Chair of EuroSPI in 2005 and 2006.