EuroSPI² 2011 Proceedings

Proceedings

The papers in this book comprise the industrial proceedings of the EuroSPI² 2011 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 EuroSPI² and the publisher.

DELTA Series about Process Improvement – ISBN 978-87-7398-153-5.

EuroSPI²

EuroSPI² is a partnership of large Scandinavian research companies and experience networks (SINTEF, DELTA, STTF), iSQI as a large German quality association, the American Society for Quality, and ISCN as the co-ordinating partner.

EuroSPI² conferences present and discuss practical results from improvement projects in industry, focussing on the benefits gained and the criteria for success. Leading European companies are contributing to and participating in this event. This year's event is the 18th 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.

Since 2009 we have extended the scope of the conference from software process improvement to systems, software and service based process improvement.

Roskilde University is the host of the EuroSPI² 2011 conference. At Roskilde University they think in innovative and creative terms about research, education and ability to innovate. EuroSPI² focuses on creating Europe-wide networks for innovation and improvement and thus we are proud of co-organizing this year's event at Roskilde University.

EuroSPI² Chairs

General & Workshop Chair	Richard Messnarz, ISCN, Austria/Ireland
EuroSPI² Marketing Chair	Miklós Biró, Dennis Gabor College, Hungary
Scientific Programme Chair	Rory O' Connor, Dublin City University, Ireland
Scientific Programme Chair	Jan Pries-Heje, Roskilde University, Denmark
Industrial Programme Chair	Jørn Johansen, DELTA, Denmark
Industrial Programme Chair	Risto Nevalainen FiSMA and STTF, Finland
Industrial Programme Chair	Stephan Goericke, iSQI, Germany
Industrial Programme Chair	Mads Christiansen, DELTA, Denmark
Industrial Programme Chair	Nils Brede Moe, SINTEF, Norway
Organizing Chair	Adrienne Clarke, ISCN, Ireland

Industrial Programme Committee

EuroSPI² Board Members

Editors of Proceedings

Richard Messnarz, ISCN, Austria/Ireland

Damjan Ekert, ISCN, Austria

Mads Christiansen, DELTA, Denmark

Jørn Johansen, DELTA, Denmark

Sonja Koinig, ISCN, Austria

Welcome Address by the EuroSPI² General Chair

Richard Messnarz

EuroSPI² is an initiative with 5 major goals (www.eurospi.net):

1. An annual EuroSPI² conference supported by Software Process Improvement Networks from different European countries.
2. EuroSPI² supported the establishment of a world-wide SPI Manifesto (SPI = **Systems, Software and Services Process Improvement**) 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 a web-based experience library based on hundreds of experience reports contributed to EuroSPI² since 1994 and which is continuously extended over the years and is made available to conference attendees
4. Establishing a European Qualification Framework for a pool of professions related with SPI and management. This is supported by Europe-wide certification for qualifications in the SPI area, exam systems, and online training platforms (European Certification and Qualification Association, www.ecqa.org).
5. Establishing a world-wide newsletter with articles from key industry and European key research associations helping to implement the SPI manifesto world-wide (newsletter.eurospi.net)

EuroSPI² is a partnership of large Scandinavian research companies and experience networks (SINTEF, DELTA, STTF), the iSQI as a large German quality association, the American Society for Quality, and ISCN as the co-coordinating partner. EuroSPI² collaborates with a large number of SPINs (Software Process Improvement Network) in Europe.

EuroSPI^2 conferences present and discuss results from systems, software and services process improvement (SPI) projects in industry and research, focussing on the benefits gained and the criteria for success. This year's event is the 18th 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 EuroSPI² was stated by a company using the following words:

> *"... the biggest value of EuroSPI² 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), GOSPEL (Financial SPICE Assessor and Demo Processes for COSO/COBIT SPICE), SOCIRES (Corporate Social Responsibility Expert), LSSA (Six Sigma Related Qualification), ResEUr (Research to Entrepreneurship Strategies), mLeMan (M-Learning Manager), BPM (Business Process Modeller), etc. A pool of more than 27 qualifications has been set up.

Join the community of cross-company learning of good practices!

Contact: Richard Messnarz, ISCN, Austria/Ireland, e-mail: rmess@iscn.com

Welcome from the Organization and Scientific Programme Committee Chair in Denmark

Jan Pries-Heje

Welcome to Roskilde. One of the oldest cities in Denmark located on the southern edge of a large fjord that connects with the North Sea. It was one of the most important Viking towns, no doubt by its unique strategic position and became the capital of what is now Denmark.

Also welcome to Roskilde University. The largest educational institution in Region Zealand, Roskilde University is Denmark's 'different' university. At Roskilde University, we are in a close dialogue with the world around us: locally, regionally, nationally and internationally. We receive input from the outside world about social and societal challenges that research can address, and we share our research results with the rest of the world, frequently and with great pleasure. Our desire to engage and be engaged, however, is not only aimed at the rest of the We also turn our focus inwards, seeking a strong engagement with and by all here at Roskilde University, both staff and students. We believe that together we make Roskilde University what it is, i.e. the attractive alternative to the rest of the Danish academic community.

I am Professor. I am head of a Research Group on IT Innovation; part of CBIT, the Department of Communication, Business and IT. I am the co-editor of the SPI (Software, Systems, Services Process Improvement) Manifesto which is basis for EuroSPI² strategies and which was created in an international workshop aligned with EuroSPI² 2009 in Alcala, Spain. My research focuses on designing and building innovative solutions to managerial and organizational IT problems. Previous and current projects explore quality software development @ Internet Speed, innovative capability in projects, the ability for an organization to improve, and how one can design a process for making better sourcing decisions. I serve as the Danish National Representative to IFIP Technical Committee 8 on Information Systems and I am currently the chair of that committee.

At Roskilde University, we think in innovative and creative terms about our research, education and ability to innovate. We are not afraid to try new things, and we constantly develop new offerings, just as we continually seek to improve.
This is why the European Systems and Software Process Improvement and Innovation initiative (EuroSPI²) is perfectly aligned with our strategic goals here at Roskilde University.

We are happy to welcome an impressive group of international experts in process improvement to the EuroSPI² conference at Roskilde University.

Contact: Jan Pries-Heje, Roskilde University, Denmark,
e-mail: janph@ruc.dk

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 EuroSPI² conference from the very beginning 16 years ago. We are now for the 4th time the publisher of the Industrial Proceedings from EuroSPI² 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 16 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 32 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 14 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

Table of Contents

Experience Session 1: SPI and Assessment

Experience Session 2: SPI and Systems (Medical/Automotive/Safety)

Experience Session 3: SPI and Organisational Strategy

Experience Session 4: SPI and Quality Models

Experience Session 5: SPI and Lean/Agile/Re-Use

Experience Session 6: SPI Outsourcing 1

Experience Session 7: SPI and People Issues

Experience Session 8: SPI and Implementation

Experience Session 9: SPI and Engineering

Experience Session 10: SPI Outsourcing 2

Selected Experience Workshop Papers

Med-Trace: Traceability Assessment Method for Medical Device Software Development

Valentine Casey, Fergal Mc Caffery,
Regulated Software Research Group
Dundalk Institute of Technology &
Lero - The Irish Software Engineering Research Centre,
val.casey@dkit.ie ,fergal.mccaffery@dkit.ie

Abstract

Traceability is central to medical device software development and essential for regulatory approval. To achieve compliance an effective traceability process needs to be in place. This process must ensure the need for clear linkages and traceability from software requirements - including risks - through the different stages of the software development and maintenance lifecycles. This is difficult to achieve due to the lack of specific guidance which the medical device standards and documentation provides. This has resulted in many medical device companies employing inefficient software traceability processes. In this paper we outline the development and implementation of Med-Trace a lightweight software traceability process assessment and improvement method for the medical device industry. We also present and discuss our findings from two industry based Med-Trace assessments.

Keywords

Medical Device Standards, Medical Device Software Traceability, Medical Device Software Process Assessment and Improvement, Risk Management, Assessment Method

1 Introduction

The role and importance that software plays in medical devices continues to increase [1]. With the demands for greater functionality in medical devices, software enables complex functional changes to be implemented without the requirement to change the hardware [2]. This has resulted in the complexity of medical device software and its development increasing [3]. This has also necessitated the requirement for effective traceability and risk management processes and tools to be in place to facilitate medical device software development.

Due to the safety critical nature of medical device software companies must comply with the regulatory requirements of the countries in which they wish to market their products [4]. This has resulted in governments defining regulatory requirements and establishing auditing bodies to ensure that only safe medical devices are placed on the market [5]. In Europe the requirements for medical devices are defined in the Medical Device Directive (MDD) (1993/42/EEC) [6] and amendment MDD (2007/47/EC) [7], In-Vitro Diagnostic Directive (IVDD) (98/79/EC) [8] and the Active Implantable Medical Device

Directive (AIMDD)[9] (90/385/EEC). The applicable directive depends on the type of medical device being developed.

In the United States the Food and Drug Administration (FDA) are responsible for medical device regulation and compliance. To provide assistance in achieving regulatory compliance the FDA have published guidance documents which address risk-based activities to be performed during software validation [10], pre-market submission [11] and when using off-the-shelf software in a medical device [12]. Although the FDA guidance documents provide information on which software activities should be performed, they do not enforce any specific method for performing these activities.

To achieve compliance national regulatory requirements also recommend conformance to a number of international standards which include: IEC 62304:2006 [13], ISO 14971:2007 [14], ISO 13485:2003 [15], EN 60601-4:2000 [16], IEC 80002-1:2009 [17], IEC 62366:2007 [18], IEC/TR 61508:2003 [19], and IEC 60812:2006 [20]. Given the need to address the requirements of these standards and regulations medical device software companies are compliance centric in their approach [21]. While this is essential to market their products it has resulted in a lack of focus on process improvement and the achievements of its associated benefits [5]. To address these important issues the authors are currently developing Medi SPICE [22, 23] a comprehensive process assessment and improvement model which is domain specific to medical device software development and incorporates regulatory compliance. In tandem with this work Med-Trace a lightweight assessment method has been developed which focuses on medical device software traceability.

2 Medical Device Software Traceability

In the context of software development requirements traceability refers to the ability to describe and follow the life of a requirement in both a forward and backward direction. This includes from its origins, specification, development, subsequent deployment and use and through periods of on-going refinement and iteration in any of these phases [24]. The focus of requirements traceability is identifying how high level requirements are transformed into low level requirements and how these are implemented and deployed in the software product. Traceability is also an important tool in the software development activities of project management, change management, risk management and defect management. The deployment of an effective traceability process is essential to facilitate the development of high quality software systems [25]. It is therefore not surprising that traceability is vital for critical systems which must satisfy a range of functional and non-functional requirements, including safety, risk, reliability and availability [26].

Medical device software development is a difficult and complex endeavour in comparison with other domains. Safety and risk management are two key areas which must be successfully addressed given the potential for harm that defective medical device software can cause. Software defects are an on-going problem for the medical device industry. This is highlighted by an analysis of the FDA medical device recalls from the 1st January 2010 to the 1st January 2011. Over that period the FDA recorded 80 medical device recalls and state software as the cause [27]. Effective traceability is important for increasing safety and reducing defects in medical device software. It is also an essential requirement for regulatory compliance.

In order to comply with the regulatory requirements of the medical device industry it is essential to have clear linkages and traceability from requirements - including risks - through the different stages of the software development and maintenance lifecycles. The regulatory bodies require that medical device software companies clearly demonstrate how they follow a software development lifecycle, without providing detailed guidance on how that can be achieved. This is further compounded by the requirement to adhere to numerous standards without guidance on how they can be implemented. Given the lack of guidance and importance that traceability plays in medical device software development it was recognized that this was an important area which needed to be addressed.

3 The Development of Med-Trace

One of the main objectives of the Regulated Software Research Group (RSRG) in Dundalk Institute of Technology is to provide assistance for the improvement of medical device software development. Therefore, as traceability is central to the development of regulatory compliant software we decided to develop a lightweight assessment method specifically to assist companies to adhere to the traceability aspects of the medical device software standards and regulations. This decision was taken in consultation with a number of multinational and indigenous medical devise software organizations who all highlighted the value of the development of such a method given the importance the traceability plays.

Emanating from the Adept method [28] and Med-Adept [5], both previously developed by the authors, and based upon the CMMI® [29] and ISO/IEC 15504-5:2006 [30] software process reference models, Med-Trace has been developed. Med-Trace is a lightweight assessment method that provides a means of assessing the capability of an organization in relation to medical device software traceability. It enables software development organizations to gain an understanding of the fundamental traceability best practices based on the software engineering traceability literature, software engineering process models (CMMI®, ISO/IEC 15504-5), and the medical device software guidelines and standards. Med-Trace may be used to diagnose an organization's strengths and weaknesses in relation to their medical device software development traceability practices.

When developing Med-Trace it was very important that the requirements for traceability in the context of software development and in particular of medical device software development were defined and addressed. To achieve this, an extensive literature review was undertaken which focused on the following areas:

- Generic software development and traceability
- Safety-critical software development and traceability
- Medical device software traceability requirements

As part of the generic software development and traceability review the CMMI® [29] and ISO/IEC 15504-5:2006 [30] were reviewed in respect of how they addressed traceability. In the area of safety-critical software traceability the DO-178B [31] standard for the aerospace industry and Automotive SPICE [32] for the automotive industry were among those reviewed and analysed. While all the relevant regulations and standards were reviewed with regard to medical device software. Of these specific emphases was placed on the following as they proved to be the most relevant with regard to traceability: IEC 62304:2006 [13]; MDD (1993/42/EEC) [6] and Amendment MDD (2007/47/EC) [7]; FDA Center for Devices and Radiological Health (CDRH) Guidelines [10-12]; ISO 14071:2007 [14]; IEC/TR 80002-1:2009 [17]; and ISO 13485:2003 [15]

The results from the literature review identify the key aspects of the software development process that Med-Trace had to focus on and address. It also highlighted the limited amount of published material regarding implementation challenges and advances in the field of traceability in medical device software development. This was in contrast to other sectors in the same context, which include automotive and aerospace software development. It was therefore not a surprise to discover that while there is a requirement to address traceability, and undertake traceability analysis, there is limited guidance available to help implement traceability effectively in medical devise software organizations. This finding is in line with a review of guidance for all aspects of medical device software development which took place in 2009 [22].

Based on the results from the literature review, the relevant areas of the CMMI® [29] and ISO/IEC 15504-5:2006 [30], and previous experience of developing lightweight process assessment methods Med-Trace was developed. The goal of a Med-Trace assessment is not certification, but to assist medical device software organizations to gain an understanding of the fundamentals of traceability and best practice with the objective of improving their software development process.

An important aspect of Med-Trace is the lightweight nature of the assessment. Med-Trace is light in the number of personnel both internal and external to the organization that are required to undertake an assessment. It is light in regard to the resources of both time and effort of those involved. It is light in the time it takes to undertake and report the results of the assessment. It also provides clear agreed guidelines which can be achieved in a short time period which will facilitate process improvement with tangible results with regard to traceability.

3.1 Stages of the Med-Trace Assessment Method

The Med-Trace assessment method contains eight specific stages. The assessment team normally consists of two assessors who share responsibility for conducting the assessment. **Stage 1**, a preliminary meeting between the assessment team and the company wishing to undergo a Med-Trace assessment takes place. At this meeting the assessment team discuss the main drivers for the company embarking on a Med-Tace assessment and an assessment schedule is agreed. During **stage 2,** the lead assessor provides an overview of the Med-Trace assessment to members of the organization who will be involved in subsequent stages of the assessment. At **stage 3**, a review is undertaken which provides a brief insight into project documentation. The first three stages are normally performed on the company's premises, but the sample documentation collected in stage 3 is, normally taken off-site as it can then be used to assist with the generation of additional questions for stage 4. The assessment team return onsite to commence **stage 4** when key staff members from the organization are interviewed. A set of scripted questions are used for these interviews which are based on the software traceability literature review, the CMMI® [29] and ISO/IEC 15504-5 [30] models, and traceability practices that are required by the medical device industry. Each Interview is normally scheduled to take 1.5 hours approximately and the number of interviews that take place is limited to a maximum of 4.

Stage 5 is a collaborative exercise which the assessors jointly undertake to develop the findings report using their respective interview notes. **Stage 6** involves presenting the findings report to participating staff in the organization. The focus of **stage 7** is the collaborative development with the staff of a pathway towards achieving highly effective and regulatory compliant traceability practices. The findings report will provide guidance to the assessed company and will focus upon practices that will offer the greatest benefit in terms of the company's business goals and objectives, in addition to quality and compliance. **Stage 8** involves revisiting and reassessing the company approximately 3 months after the completion of stage 7 and reviewing progress against the recommended improvement path. The outcome of this stage is an updated improvement path and a final report detailing the progress that has been accomplished along with additional recommendations.

4 The Implementation of Two Med-Trace Assessments

In this section we outline our observations from undertaking two Med-Trace assessments, one in an Irish company and the other in a company based in the United Kingdom. The process improvement objectives that were collaboratively agreed by both organizations to improve their respective traceability practices are presented. We also discuss our observations from our findings from undertaking both assessments.

4.1 Med-Trace Assessment in Medical Electronic

The first assessment took place in a small to medium sized (SME) Irish medical device organization, Medical Electronic (a pseudonym). Medical Electronic develop electronic based medical devices that are marketed in the US and Europe. To sell their products they require compliance with both the FDA and the MDD. The importance traceability plays in medical device software development was recognized by Medical Electronic and they sought a lightweight assessment method to obtain guidance as to how they could improve their traceability process. Having been introduced to Med-Trace and having discussed what was involved they requested a Med-Trace assessment.

4.1.1 Medical Electronic Med-Trace Assessment Recommendations

Based on the analysis of the results from the Med-Trace assessment undertaken in Medical Electronic and in collaboration with their staff, an improvement plan was developed with the following recommendations:

1. The organization will undertake steps to measure the time spent on traceability and evaluate its effectiveness.
2. In future projects the task of performing traceability will be identified as part of the project plan and adequate time and resources will be allocated to undertake this task.
3. Good practices which are employed while performing the traceability process will be documented in an efficient format and will be available for dissemination to relevant parties as and when required.
4. Project managers will mandate the use of traceability while conducting impact analysis, promoting its usage as a management tool and enabling the capture of information for management use.
5. Milestones will be put in place in the software development lifecycle which will not permit advancement to other phases/stages of the lifecycle until the requirements for traceability are satisfied.
6. A mechanism for tracing open bugs/known issues to the safety/hazard/risk management system and linking them to the requirements will be put in place and utilised.
7. The organization will evaluate and select a tool for the process of automating traceability and requirements management.

4.2 Med-Trace Assessment in North Medical UK

The second Med-Trace assessment took place in a United Kingdom based medical device organization, North Medical UK (a pseudonym). North Medical UK is an SME and they develop electronic-based medical devices that require compliance with both the FDA and the MDD. North Medical UK also sought a resource-light assessment method to obtain guidance as to how they could improve their software development traceability process. Having heard about the Med-Trace assessment method they contacted the authors and after discussions regarding what was involved they requested an assessment.

4.2.1 North Medical UK Med-Trace Assessment Recommendations

Having analysed the results from the Med-Trace assessment and in collaboration with North Medical UK staff, an agreed pathway for improvement was developed:

1. The software development traceability process will be formalised and documented.
2. Meetings between the various parties involved in traceability will be scheduled as part of the development life cycle
3. A formal training program will be introduced to facilitate the adoption of best traceability practices for requirements and risk management.
4. The current MS Office based traceability application will be replaced with an appropriate automated traceability tool.
5. Terminology usage with regard to traceability will be standardised and a formal definition of both risk and hazard agreed. A formal method for quantifying probability of harm will also be introduced and deployed.
6. A traceability and validation procedure will be developed, implemented and monitored to verify the activities of the staff that perform the traceability and validation function.

7. A formal procedure will be developed and implemented to facilitate mapping from the design documentation to the software code.
8. Resources will be allocated to enable the full implementation of an automated tool. This tool has been purchased to allow digital signatures to be recorded at each development stage, but it has not been properly implemented.

4.3 Observations - Implementing Two Med-Trace Assessments

The organizations assessed both recognized the importance traceability plays in medical device software development. This was reflected in the fact that in both organizations a member of the management team was responsible for its implementation. The difficult and complexity involved in successfully tracing requirements and managing risk and hazards were appreciated by both organizations. The lack of detailed guidance on how to implement traceability was highlighted by the management of Medical Electronic and North Medical UK. While these organizations both employed a process for traceability they recognized this needed to be improved and formalized. The requirement for relevant training and the ability to record and leverage best practice with regard to traceability also emerged.

The serious limitations of utilising manual tools such as MS Office to manage traceability was recognized as a problem. As was the requirement for the procurement of automated tools to address this issue. It was understood that this had to be undertaken with due care and within the financial and temporal constraints of both organizations.

Both organizations welcomed the opportunity to participate in a Med-Trace assessment. The fact that it is lightweight and specifically addressed key issues was considered very relevant and valuable. The findings from the assessments identified important areas where improvements were required and this was confirmed in consultation with the management and staff of both organizations. The adoption of the development pathway provided realistic goals and the collaborative process provided motivation for their achievement. Both organizations are implementing their respective development pathways and have agreed to be reassessed as part of stage 8 of the Med-Trace assessment method.

5 Conclusions and Future Plans

Due to the critical nature of medical device software and the potential harm failure can cause, the implementation of an effective traceability process is essential. Therefore, to ensure validity, software requirements traceability analysis needs to be conducted to trace software requirements to (and from) system requirements, and to risk analysis results. While this is mandated by the medical device guidelines and standards it is recognized as difficult to accomplish. The lack of detailed guidance and direction as to how this can be successfully achieved has been highlighted as a particular problem in this context. While the need to provide requirements traceability cannot be underestimated, the necessity to provide traceability for each identified hazard is of equal importance. Risk management is a key activity for medical device software development and hazards have to be traced to risk analysis, risk evaluation and the implementation and verification of the risk control measures.

Med-Trace helps to address these issues by providing a lightweight traceability centric assessment method that organizations can utilise. The focus is on a resource light assessment that can pinpoint specific areas for improvement with regard to traceability that will provide tangible results in a short time period. The need for the collaborative development of the improvement path is essential to ensure relevance and buy in within the assessed organization. The opportunity for reassessment provides an updated improvement plan and the final report contains additional recommendations and highlights what improvements have been achieved. The RSRG at Dundalk Institute of Technology will continue to refine Med-Trace based on the experience gained in undertaking future assessments, interaction with medical device software organisations and medical device regulatory bodies. It is envisaged that further research will be under taken for the development of similar lightweight assessment methods. These will deal with other important aspects of medical device software development which will build on and leverage the experience gained in undertaking this work.

6 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 (the funding of this project was awarded by Science Foundation Ireland under a co-funding initiative by the Irish Government and European Regional Development Fund), and supported in part by Lero - the Irish Software Engineering Research Centre (http://www.lero.ie) grant 03/CE2/I303_1

7 Literature

1. Méry, D. and N. Kumar Singh. Trustable formal specification for software certification. in The 4th international conference on Leveraging applications of formal methods, verification, and validation. 2010. Heraclion, Crete: Springer-Verlag.
2. Lee, I., G. Pappas, R. Cleaveland, J. Hatcliff, B. Krogh, P. Lee, H. Rubin, and L. Sha, High-Confidence Medical Device Software And Systems. Computer, 2006. **39**(4): p. 33 - 38.
3. Rakitin, R., Coping with defective software in medical devices. Computer, 2006. **39** (4): p. 40 - 45.
4. Burton, J., F. Mc Caffery, and I. Richardson. A risk management capability model for use in medical device companies. in International Workshop on Software quality (WoSQ '06). 2006. Shanghai, China: ACM.
5. Mc Caffery, F. and V. Casey. Med-Adept: A Lightweight Assessment Method for the Irish Medical Device Software Industry. in European Systems & Software Process Improvement and Innovation Conference, (EuroSPI). 2010. Grenoble, France.
6. European Council, Council Directive 93/42/EEC Concerning Medical Devices. 1993, Official Journal of The European Communities: Luxembourg.
7. European Council, Council Directive 2007/47/EC (Amendment). 2007, Official Journal of The European Union: Luxembourg.
8. European Council, Council Directive 98/79/EC On in vitro diagnostic medical devices. 1998, Luxembourg: Official Journal of The European Communities.
9. European Council, Council Directive 90/385/EEC On the approximation of the laws of the Members States relating to active implantable medical devices. 1990, Official Journal of The European Communities]: Luxembourg.
10. US FDA Center for Devices and Radiological Health, General Principles of Software Validation; Final Guidance for Industry and FDA Staff. 2002, CDRH: Rockville.
11. US FDA Center for Devices and Radiological Health, Guidance for the Content of Premarket Submissions for Software Contained in Medical Devices. 2005, CDRH: Rockville.
12. US FDA Center for Devices and Radiological Health, Off-The-Shelf Software Use in Medical Devices; Guidance for Industry, medical device Reviewers and Compliance. 1999, CDRH: Rockville.
13. IEC 62304:2006, Medical device software—Software life cycle processes. 2006, IEC: Geneva, Switzerland.
14. ISO 14971:2007, Medical Devices — Application of risk management to medical devices. 2007, ISO: Geneva, Switzerland.
15. ISO 13485:2003, Medical devices — Quality management systems — Requirements for regulatory purposes. 2003, ISO: Geneva, Switzerland.
16. BS EN 60601-1-4:2000, Medical Electrical Equipment, Part 1 - General requirements for safety. 2000, BSI: London.
17. IEC/TR 80002-1:2009, Medical device software Part 1: Guidance on the application of ISO 14971 to medical device software. 2009, BSI: London.
18. IEC 62366:2007, Medical devices - Application of usability engineering to medical devices. 2007, IEC: Geneva, Switzerland.
19. IEC/TR 61508:2005, Functional safety of electrical/electronic/ programmable electronic safety related systems. 2005, BSI: London.

20. IEC 60812:2006, Analysis technique for system reliability - Procedure for failure modes and effects analysis (FMEA). 2006, IEC: Geneva, Switzerland.
21. Mc Caffery, F., J. Burton, V. Casey, and A. Doring, Software Process Improvement in the Medical Device Industry, in Encyclopedia of Software Engineering, P. Laplante, Editor. 2010, CRC Press Francis Taylor Group: New York. p. 528 - 540.
22. Mc Caffery, F. and A. Dorling. Medi SPICE: An Overview. in International Conference on Software Process Improvement and Capability Determinations (SPICE). 2009. Turku, Finland.
23. Mc Caffery, F., A. Dorling, and V. Casey. Medi SPICE: An Update. in International Conference on Software Process Improvement and Capability Determinations (SPICE). 2010. Pisa, Italy: Edizioni ETS.
24. Gotel, O. and A. Finkelstein. An Analysis of the Requirements Traceability Problem. in First International Conference on Requirements Engineering. 1994. Colorado Springs USA.
25. Espinoza, A. and J. Garbajosa. A Proposal for Defining a Set of Basic Items for Project-Specific Traceability Methodologies. in 32nd Annual IEEE Software Engineering Workshop. 2008. Kassandra, Greece.
26. Mason, P. On Traceability for Safety Critical Systems Engineering. in 12th Asia-Pacific Software Engineering Conference, 2005. 2005. Taipei, Taiwan.
27. FDA, Medical & Radiation Emitting Device Recalls. 2011, http://www.accessdata.fda.gov/scripts/cdrh/cfdocs/cfres/res.cfm accessed 10/02/2011.
28. Mc Caffery, F., I. Richardson, and G. Coleman. Adept – A Software Process Appraisal Method for Small to Medium-sized Irish Software Development Organisations. in European Systems & Software Process Improvement and Innovation (EuroSPI 2006). 2006. Joensuu, Finland.
29. CMMI Product Team, Capability Maturity Model® Integration for Development Version 1.2. 2006, Software Engineering Institute.
30. ISO/IEC 15504-5:2006, Information technology — Process Assessment — Part 5: An Exemplar Process Assessment Model. 2006, ISO: Geneva, Switzerland.
31. DO-178B, Software Considerations in Airborne Systems and Equipment Certification. 1st December 1992, RTCA: USA.
32. Automotive SIG, Automotive SPICE Process Assessment V 2.2. 21 August 2005.

8 Author CVs

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.

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.

Reflecting on ten years of Software Process Assessment and Improvement initiatives by the European Space Agency

Marie-Odile Devic, David Escorial Rico, Sven Richter

Abstract

The European Space Agency (ESA) has been involved in researching, developing and deploying software process assessment and improvement methods across the European software space industry since 1999. An updated European Cooperation for Space Standardisation (ECSS) handbook for space software process assessment method compliant with ISO 15504 has been published in 2010. This paper presents a recapitulation of the major events in the last decade in relation to ESA sponsored activities in process assessment and improvement, reflecting on the experiences undertaken, as well as a look into the future in the context of the European space software industry.

Keywords

Space domain, ISO 15504, European Cooperation for Space Standardisation (ECSS), Risk mitigation, Product Assurance

1 Introduction

The mission of the European Space Agency (ESA) is to provide for and promote cooperation among European States in space research and technology and their space applications (ESA 2010). In practice, ESA works as an international procurement agency, placing large development contracts for space related systems with leading industries located in its 17 Member States. Among all the space segments, software is a key component in flight, ground, testing and user systems.

For the last four decades, software in space systems has evolved to implement a number of mission-critical functions: satellite precision manoeuvres, autonomous navigation of rovers on a planetary terrain or operational control in ground centres. For example, ESA developed the Automated Transport Vehicle (ATV) as an unmanned spacecraft for supplying the International Space Station with propellant, water, air, payload and experiments. The ATV docks automatically to the International Space Station using complex avionics software systems.

In the frame of this kind of mission-critical software systems, in which a failure can result in loss of life or impossibility of completing a mission, the promotion of best practices and a continuous software process improvement became a real need. This was evident more than 15 years ago, when the first initiatives in software process improvement demonstrated that the quality, cost and in-time delivery (cycle time) of software systems is highly influenced by the maturity of the processes used to develop it (Humphrey 1995) (Harter *et al.* 2000). ESA, as a procurement agency, is not just interested in the quality of the software product required for the criticality of the implemented function, but also in delivering it in time and within the budget.

Process improvement needs to be supported by the objective measure of process capability, by means of process assessment methods. These methods are generic and based on best practices proven across the entire industry. In the context of space software, an adaptation of the assessment method to reflect the practices, specific characteristics and even culture of the space industry was required in order to obtain a clearer picture of the processes' maturity.

The present paper describes the evolution of the initiatives taken by ESA to promote software process assessment and improvement within the European space software industry. It also reflects on the issues found during the last decade, the current status of those initiatives, and the expected future approach.

2 Timeline

The following table compiles as reference the timeline of relevant events related to the development of Software Process Initiatives for the European space industry. As ESA opted to base its software process improvement initiative on ISO 15504 only milestones related to the development of this standard, the space software standardisation process and other space-related process improvement initiatives are presented.

Year	Event	Comments
1993	SPiCE (Software Process Improvement and Capability Evaluation) working group is formed	Name later changed to Software Process Improvement and Capability dEtermination
1996	The European Cooperation for Space Standardisation (ECSS) secretariat publishes the ECSS-Q-80A Space Product Assurance - Software product assurance	Applicable ESA software standards are organised as requirements imposed on processes derived from ISO 12207:1995 Process metrics are required to assess the quality of the development process.
1998	Publication of ISO/IEC TR 15504:1998 – Software process assessment	The first draft of this Technical Report is published in 1995
1999	ESA publishes the draft results of the study:	This method will be known as SPiCE for

	ISO/IEC TR 15504 Conformant Method for the Assessment of Space Software Processes	Space (S4S)
2000	Drafts of new ECSS-E-40B and ECSS-Q-80B are made applicable for space software projects	A number of explicit requirements for Assessment and Improvement of processes are included.
	ESA starts the S4S trial programme	Eight space software suppliers are assessed.
2001	ESA publishes the methods: SpiCE for Space-Risk (R4S) and SpiCE ISO 9001:2000 for Space (S9kS)	A number of methods related to risks management and quality management in the space domain are derived from S4S
2003	Publication of ISO/IEC 15504 Part 2:2003– Process Assessment – Part 2 performing an assessment	The rest of ISO/IEC 15504 will not be published until 2008
	ECSS-E-40B ECSS-Q-80B officially published	
	3rd International SPICE Conference on Process Assessment and Improvement organised at ESA-ESTEC	
2004	Galileo Software Standard published Issue 7	Contractors supplying mission-critical software shall demonstrate certain capability levels for a number of processes (target profile). S4S is “strongly encouraged” as the assessment method, however other methods are acceptable.
2005	Automotive SPICE is published	Two domain-specific assessment and improvement methods.
	MedeSPI (Software Process Improvement model for the medical Device industry based upon ISO15504) is published	
2008	EADS Astrium On-board Software Entity appraised CMMI Maturity level 3	EADS Astrium and Thales Alenia Space are two of the main prime contractors of ESA
	EADS Space Transportation appraised CMMI Maturity level 2	
	Thales Alenia Space assessed CMMI Maturity level 3	
2009	ECSS-E-ST-40C Space engineering - Software and ECSS-Q-ST-80C Space Product Assurance - Software product assurance are published	This new version of the space software standards is published taking into account the latest version of ISO/IEC 12207:2008
		ISO/IEC 15504 Part 2:2003 is made explicitly a normative reference. Assessment performed and assessment models used shall be in conformance with ISO/IEC 15504 Part 2:2003.
		CMMI model applying SCAMPI A assessment method is considered as acceptable.
2010	ECSS-Q-HB-80-02 Space product assurance – Software process assessment and improvement is published	This handbook is the natural evolution of the S4S method. It takes into account the last ISO/IEC 12207 and is conformant with ISO/IEC 15504.

3 Beginning (1996-2000)

The Product Assurance (PA) and Safety Department of the Directorate of Technical and Quality Management (D/TEC-Q) of ESA has been systematically providing support to ESA projects to ensure that adequate quality is built into software, by monitoring the application of best practices as defined in their applicable standards. In particular, the ECSS (European Cooperation for Space Standardisation) standards for software for engineering (ECSS-E-40A) and product assurance (ECSS-Q-80A), considering a processes approach, directly taken from the Standard Software life cycle processes, ISO/IEC 12207 (1995), and complemented with space specific requirements. The product assurance standard also establishes requirements about the monitoring of processes and their quality.

In parallel, the ESA PA & Safety Department has been carrying out research on software engineering methods and processes to anticipate tools, guidelines and policies for Software Product Assurance. Having the strong belief that high-quality software products are influenced by high-quality software processes (development, maintenance, operation, support, organisational and acquisition), the ESA PA & Safety Department launched a research study in 1998 for establishing a software process assessment and improvement method for space software processes. The result of this study was the definition of the method Spice for Space (S4S) (Völcker *et al.* 2001), validated with four pilot assessments during 1999. The main characteristics of this method are:

- It follows the exemplar assessment model from Part 5 of ISO/IEC TR 15504 (1999). The decision to follow the ISO approach and logic was determined by the orientation of the software ECSS software standards, which followed ISO12207 (1995), as well as the flexibility offered by ISO15504 in order to expand and modify the process dimension.
- It covers all the requirements established in the ECSS software standards, by extending the process dimension to space software oriented processes, base practices, work products and additional explanatory notes. In particular, it adds four new processes: “Safety and dependability assurance“, “Independent software verification and validation”, “Information management” and “Contract maintenance”.
- The capability dimension was adopted without modification from ISO/IEC 15504.
- Three modes of assessment are established: process improvement, capability determination and ECSS conformance.

Eight European space software companies in Austria, Belgium, Denmark, Germany, Great Britain, Italy and Sweden, were assessed using this method during the year 2000. The trials were conducted by ESA, Synspace and Interspace trained assessors. Details on the characteristics of the assessed projects and the assessment outcomes can be found in (Cass *et al.* 2004).These assessments provided the first picture, though rather incomplete, of the typical capability levels for software processes of the European space software industry.

4 Application to the real world (2001-2006)

During the next 5-year period mixed teams of ESA and third party assessors performed a total amount of 15 voluntary assessments in projects of space system integrators and space software suppliers, covering companies in most of ESA Member States. The initial goal of performing assessments on the important suppliers at that time was reached within two years, after which a permanent goal was set up: to request software process assessments as a risk mitigation tool before the start of the development of a complex and critical software system. This goal was made explicit in the new version of the ECSS software product assurance standard (ECSS-Q-80B) as well as in the Galileo[1] software standard (2004).

In particular, the scope of application of the Galileo software standard covers all software developed

[1] Galileo is a global navigation satellite system backed by the European Union and European Space Agency.

for the Galileo programme, and as assessment requirements, it assigns target capability levels per process. This capability profile is tailored for each software development assurance level (SW-DAL)[2] – e.g. for SW-DAL B (Software whose anomalous behaviour would cause or contribute to a failure resulting in a critical event) a capability level 3 is required for all applicable processes (except organisational alignment, Human Resource Management and Audit).

Moreover ESA project management and product assurance support teams also suggested assessments to be performed in companies with little or no experience in the development of software for space or ECSS requirements.

In the meantime the research activity did not stop and several related assessment methods were published:

- Spice for Space – Risk or Risk for SPACE (R4S) a method for analysing and estimating the likelihood of process-related risks, and in a second step providing guidance for process improvement by means of a comprehensive set of generic improvement actions (Oured *et al.* 2001).
- SPICE 9000 for SPACE (S9kS), a quality management assessment model derived from the quality management system requirements of ISO 9001:2000 and European requirements for space product assurance (Cass *et al*. 2002). This method will be simplified into S9k or SPICE-9000, considering only ISO 9001 requirements (Dorling 2002).
- Software Product Evaluation and Certification (SPEC) method (Aldea *et al*. 2003) extended from ISO 9126, considers as part of its generic quality model the hypothesis that the software process influences the final quality of the product, and uses S4S method to evaluate and measure this property.

These experiences and the experimental results of their application were reported in a the SPICE user group conference and organised by ESA in 2003.

5 Consolidation (2006-2011)

From 2006 until today, the assessment and improvement approach of ESA is consolidated by supporting the quality assurance function on space project and the associated software development process. For confidentiality reasons, and also due to statistical disparity (different project characteristics at different stages of the project are not easy to compare), the results of these assessments are not provided. However, the assessments performed in the last five years result in the accumulation of expertise by ESA and associated assessment organisations, as well as a clearer picture by ESA PA Managers on what the results of the assessments are: a starting point for process improvement in two directions:

- Project-short level: the main goal is to anticipate problems, reduce risks while at the same time paying attention to budget constraints: it is impossible to carry out a full improvement plan that will produce immediate benefits to a project in the areas of concern. Project stakeholders have a real need to get high priority improvements put in place in the software development organisations.
- Corporate-long term level: in this case, the objective is to reduce uncertainties in selecting suppliers, while promoting and disseminating best practices already proven across industry. The determination of the capability to comply with space standards also provides a first contact of the standards to companies working in the space environment allowing them to receive detailed first hand information of what the customer expects from them.

ESA PA & Safety Department has maintained an assessment and improvement programme since the beginings sustained in a madatory induction training of new software Product Assurance engineers on ISO15504 Software Process Assessment (IntRSA - International Registration Scheme for Assessors – certified). This training is also open to other ESA software engineers as well as Product Assurance managers. This has allowed to maintain a pool of assessors available and a responsible staff member

[2] The SW-DAL of a software component is determined by establishing how an error in a software component relates to the system failure condition(s) and the severity of that failure condition(s).

in order to coordinate external suport by specialists (for the definition and development of the S4S method and lead some of the assssments), manage the administrative aspects of the assessments (record keeping and distribution of assessment records) and perofrm periodical presentations internally and externally.

It is worth noting that in all the cases the host organisation whose processes have been assessed commits voluntarily to hold the assessment. More importantly, the organisation commits to establish the improvement plan in order to remove the resulting gap and allocate the necessary resources to implement it.

The current situation allows ESA and contractors responsible for space project to choose from a range of exceptional risk mitigation evaluation methods in particular: project audits and process assessments. Although, they maintain many similarities, since they:

- Provide a snapshot of the organisation: showing shortcomings and weak points
- Can be used for improvement
- Are evidence-based (interviews, documentation review)
- Focus on process, not people or technology
- Maintain confidentiality
- Are based on reference practices (standards, models)

It must be stressed that they have some key differences, but more importantly both methods have a different status at ESA, being audits an official means to resolve contractual differences in some cases.

The methodological differences:

Assessments	Audits
• Focus: Capability levels	• Focus: Requirements
• Rating 0..100 % achievement	• Conformance / Non-conformance (Y/N)
• Anticipates (how capable is it?)	• Current status (is it compliant?)
• Captures complexity (Effectiveness, Efficiency)	• Straightforward (Success / Failure)
• Improvement based on strengths, weaknesses, risks	• Improvement based on corrective/preventive actions
• Longer (typically 1 week, depending on the number of processes to be assessed)	• Shorter (typically 2-3 days)

On the other hand, the evolution of the S4S method and its transformation into an integral part of the ECSS standardisation system was delayed until the approval and publication of the new ISO 15504 (2003-2008) process assessment standard and the new version of the ECSS software engineering standards (ECSS-E-ST-40C) (ECSS-Q-ST-80C). The latter introducing explicit requirements to “monitor and control the effectiveness of the processes used during the development of the software” and using “process assessment models in conformance with ISO/IEC 15504 (Part 2)” for all software criticality categories excluding the lowest one. However, the CMMI model and the application of the SCAMPI A method for the assessment are also accepted. The inclusion of this compromise is caused by two factors: the foreseen convergence of the CMMI and ISO 15504 assessment models, and the initiatives of three main European space prime contractors, EADS Astrium Satellites and EADS Astrium Space Transportation and Thales Alenia Space, to appraise their software processes to CMMI level 3.

With both sets of the new standards, published the software ECSS and the ISO/IEC 15504, the ECSS handbook on software process assessment and improvement is also published (ECSS-Q-HB-80-02), maintaining the alignment with both of them. It is composed of two parts:

- Part 1: Framework

It provides an overview of the handbook and the architecture of the method, presenting the process dimension (with the mentioned modifications from ISO15504 Part 5 – exemplar process assessment model) and process capability dimension (adopted from ISO15504).

It also describes the three purposes or modes of the assessments: improvement, capability determination and ECSS standards compliance, and the assessment process definition - how to perform the assessment - for each mode. It also includes templates for assessment plan, report, and assessor record.

The handbook aims at providing practical guidance in the context of ESA projects therefore it also includes proposed target capability profiles for different dependability and safety classes according to the ESA schemes.

This part provides guidance on the implementation of a process improvement cycle, including key success factors and common failures.

Finally it contains the instructions to be met by an assessment in order to claim recognition with respect to the S4S model.

- Part 2: Assessor Instrument

  This part provides assessors with a number of instruments needed to perform software process capability assessments using the method described in Part 1. In particular these instruments are:

  - The Process Assessment Model required to perform ECSS-Q-HB-80-02 assessments including process descriptions and process attribute indicators.
  - Conformance statement to the requirements in ISO/IEC 15504 Part 2.
  - A definition of the Process Reference Model on which the ECSS-Q-HB-80-02 Process Assessment Model is based (defined in ECSS-Q-HB-80-02 Part 1).
  - Detailed traces from base practices in the ECSS-Q-HB-80-02 PAM to ECSS standards clauses and from ECSS-Q-HB-80-02 work products to ECSS expected outputs.

6 Future

After last ten years of ESA's involvement in software process assessment and improvement activities, we can identify two types of organisations being assessed: the ones new to the space domain, and the ones assessed ten years ago – during the first wave of assessments of S4S – which are applying for a re-assessment. The reasons behind this tendency are:

- Most of these companies are ISO 9001 (2008) certified which requires setting up "processes for the continual improvement of the [quality management] system".
- Improvement plans derived from the initial assessment have become obsolete (e.g. due to change in the organisation of their processes or lacking impulse and stimuli) or its objectives have been reached and the firms would like to go beyond.
- Demonstration to ESA with regard to the companies' capability level in relation to their competitors

Moreover, this tendency has also a strong correlation with the current stage of the space software industry. The industry has reached a maturity stage where firms are focused to enhance their competitive advantage by means of process innovations and improvements - reducing cycle time and executing their processes flawlessly.

However, there is no specific or explicit long-term goal regarding the expected software capability level for the European space software industry, or targeting particular kinds of firms or organisations. ESA PA and Safety Department approach is to maintain the provision of an assessment service in order to reduce risks in ESA projects and long-term programmes that require it.

Nonetheless, since changes in the domain are slow, the space software process assessment method, ECSS Q-80-02 is valid:

- It provides a flexible, results-oriented and ISO-compliant method, providing a high level of rigour to perform assessments.
- It is in line with software development and quality assurance regulations, in particular in the mission critical domains, e.g. NASA (NPR 7150.2) or Air Traffic Management (ED-153).

7 Literature

Aldea F, Gomez E, Jones M, Shäbe. 2003. Practical Application of S4S for Product Evaluation: SCOS-2000 Evaluation according to SPEC. *Joint ESA – 3rd International SPICE Conference on Process Assessment and Improvement*, 17-21 March 2003, ESTEC, Noordwijk, The Netherlands.

Cass A, Völcker C, Ouared R, Dorling A, Winzer L, Carranza JM. 2004. SPICE for SPACE Trials, Risk Analysis, and Process Improvement. *Software Process Improvement and Practice* 9:13-21.

Cass A, Völcker C, Stutter P, Stienen H, Dorling A. 2002. SPiCE in Action — Experiences in Tailoring and Extension, *Euromicro*, pp.352, 28 th Euromicro Conference (EUROMICRO'02).

Dorling A, Volcker C, Cass A, Secchi P. 2002. The Application of SPICE to Quality Management Systems, *International Conference on Software Quality, Ottawa, ON, Canada,* 12:0 1-12.

ECSS-E-40A: Space engineering – Software. ESA-ESTEC Requirements and Standards Division. 13 April 1999.

ECSS-Q-80A: Space product assurance – Software product assurance. ESA-ESTEC Requirements and Standards Division, 19 April 1996.

ECSS-Q-80B: Space product assurance – Software product assurance. ESA-ESTEC Requirements and Standards Division, 10 October 2003.

ECSS-E-ST-40C: Space engineering – Software. ESA-ESTEC Requirements and Standards Division, 6 March 2009.

ECSS-Q-ST-80C: Space product assurance – Software product assurance. ESA-ESTEC Requirements and Standards Division, 6 March 2009.

ECSS-Q-HB-80-02: Space product assurance – Software process assessment and improvement. ESA-ESTEC Requirements and Standards Division, 16 April 2010.

ED-153 EUROCAE Guidelines for Air Navigation Software Safety Assurance, August 2009.

ESA 2010. *ESA Convention and Council Rules of Procedure*. ESA Communications, SP-1317/EN.

Galileo Software Standard. 2004. Issue 7, Galileo Program, 24 May 2004.

Harter DE, Krishnan MS, Slaughter SA. 2000. Effects of Process Maturity on Quality, Cycle Time, and Effort in Software Product Development. *Management Science* 46(4): 451-466.

Humphrey WS. 1995. *A Discipline for Software Engineering*. Addison-Wesley: Reading MA.

ISO 9001:2008. *Quality Management Systems – Requirements*. International Standard Organisation, 2008.

ISO/IEC 12207:1995(E). *Information Technology – Software life cycle processes*, International Standard Organisation, 1995.

ISO/IEC TR 15504-5:1999(E). *Information Technology – Software Process Assessment*, Part 5: An Assessment Model and Indicator Guidance. Type 2 Technical Report.

ISO/IEC 15504(E):2003-2008. *Information Technology – Process Assessment*, International Standard Organisation, 2003-2008.

NASA NPR 7150.2 NASA Procedural Requirements - NASA Software Engineering Requirements. 27 September 2004.

Oured R, Völcker C, Stienen H. 2001. PASCON / W06 CCN5/TN7 Part C. R4S – Risk Reduction with SPiCE for SPACE, Issue 1.1, 19 December 2001.

Völcker C, Cass A. 2001. PASCON / W06 CCN5/TN7, ISO/IEC TR 15504 Conformant Method for the Assessment of Space Software Processes. Technical Note No. 7 Part A: SPiCE for SPACE Assessment Model and Part B: SPiCE for SPACE Method, Issue 1.1, 19 December 2001.

8 Author CVs

Marie-Odile Devic

Software Product Assurance Engineer at the European Space Agency, Product Assurance and Safety Department, European Space Research and Technology Centre (ESTEC), The Netherlands.

David Escorial Rico

Software Product Assurance Engineer at the European Space Agency, Product Assurance and Safety Department, European Space Research and Technology Centre (ESTEC), The Netherlands.

Sven Richter

Software Product Assurance Engineer at the European Space Agency, Product Assurance and Safety Department, European Space Research and Technology Centre (ESTEC), The Netherlands.

Technical Issues in Test Process Assessment and their current and future Handling in TestSPICE

Tomas Schweigert, Michael Philipp
SQS Software Quality Systems AG, Stollwerckstr. 11, D-51149 Köln
tomas.schweigert@sqs.de; michael.philipp@sqs.de
Andreas Nehfort
Nehfort IT-Consulting KG – www.nehfort.at
andreas@nehfort.at

Abstract

Looking at normal test activities like test preparation or test execution setting up a Process Reference Model or a Process Assessment Model seems to be quite easy, but when discussion focuses technical issues, it becomes a real challenge.,. This paper describes the current handling of those issues in the TestSPICE approach, and also some potential pitfalls.

Keywords

TestSpice, Assessment, Improvement,

1 Motivation – Why TestSPICE

In the last decade Testing has become a more and more professional discipline in Software Engineering. In line with this an increasing part of test activities is transferred from the development team to specialized test teams: to internal test departments or to external test service providers (local, near-shore or offshore).

This creates increasingly the need to assess the capability of test services and test processes. Existing test reference models like TPI® NEXT and TMMI® provide testing best practices but do not satisfyingly support test process assessments.

The SPICE assessment model (ISO 15504-5) on the other hand has established as software process assessment approach but does not support test specific best practices in enough detail. The SPICE standard (ISO 15504-2) supports the development of domain specific Process Reference and Process Assessment Models.

So it seemed obvious for us to combine best of two worlds: The SPICE Assessment approach and Testing best practices represented by the ISTQB Certified Tester syllabus, TPI® NEXT and TMMI® to crate TestSPICE – Test Process Reference- and Assessment Model. TestSPICE shall be applicable as “stand alone” assessment model for test service providers or as complementary refinement to existing SPICE based assessment models like ISO 15504-5 (SW Engineering), ISO 15504-6 (Systems Engineering) or Automotive SPICE.

2 Technical Issues in Software Testing

2.1 Test Automation

When an organisation starts to completely automate its tests, it soon faces the issue that in fact this means to develop the application a second time. There are several questions that have to be answered before a test automation approach should be chosen and those questions must be discussed during a test process assessment:

- Do we really need all test cases to be automated?
- Is the application under test so stable that a native capture replay approach will work?
- Do we face change in a volume that can’t be handled with a capture replay approach and needs to be handled with more sophisticated methods?

2.2 Test Environments

Lots of organisations are able to handle the test environment delivery as some type of service, but still many organisations are not able to fulfil the “do it right the first time” approach when delivering a test environment to the test team. Here we find some key questions that shall be answered and be discussed during a test process assessment.

- What has to be delivered with the test environment (Applications, internal and/or external interfaces)?
- To what degree must the test environment be similar to the production environment (Drivers, stubs, mocks, server database, firewall …)?
- How will the support for the test team be organized?

2.3 Test Data

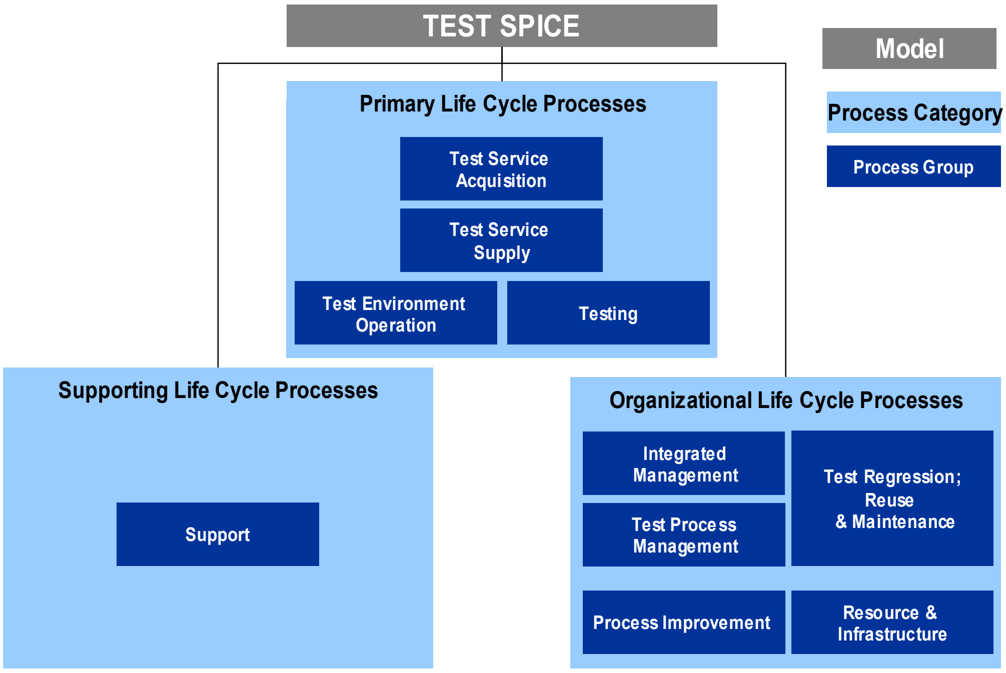
In earlier times most organisations used synthetic test data or test conserves. This approach was more and more replaced by an approach to produce test data by using the application. We found some drivers for this trend:

The rising complexity of databases,

The emerging integration of organisational data – and the growing dependency from enterprise resource management systems like SAP. This integration does not allow to run test systems without approved data from SAP

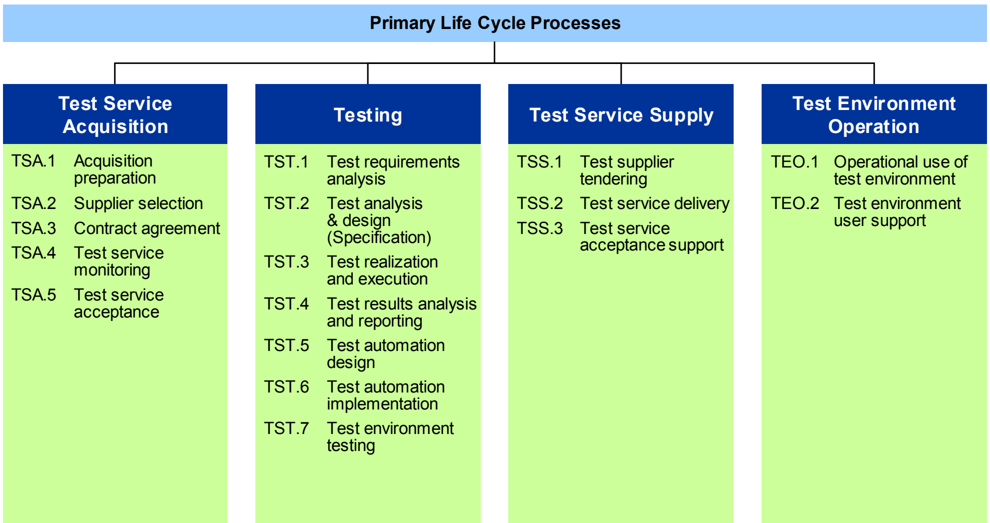
Data privacy that does not allow organisations –especially in the finance and health care sector- to test with production data.

3 *The current answer of TestSPICE V1.0*



TestSPICE uses a structure similar to ISO/IEC 15504 Part 6:

We see that there are primary life cycle processes, supporting life cycle processes and organisational life cycle processes. The TestSPICE approach also uses the measurement framework of ISO/IEC 15504 Part 2.



When we look at the primary life cycle processes we get the following picture:

We can see that test automation is handled as a real issue with the processes

TST.5 Test automation design

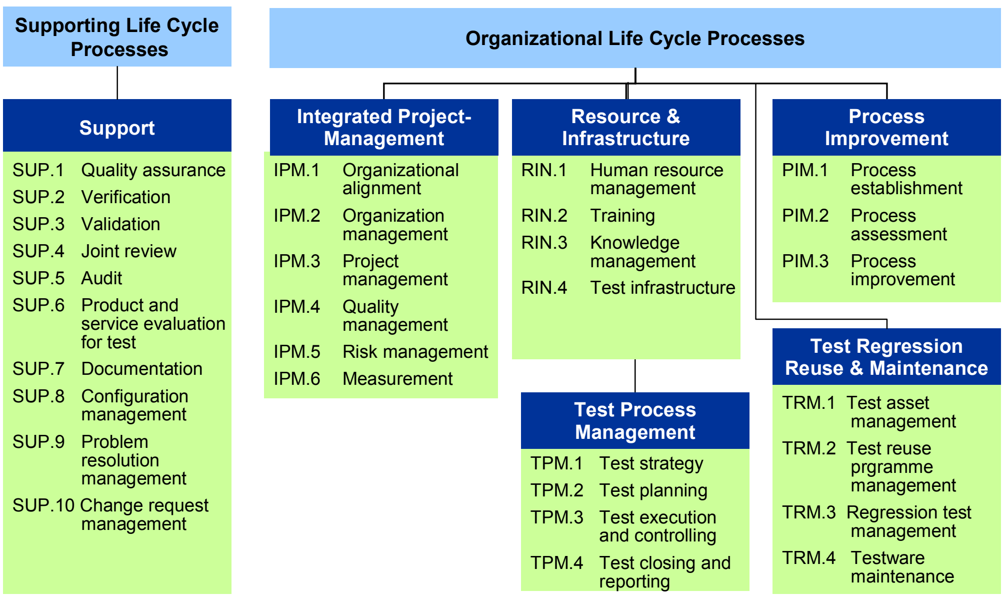
TST.6 Test automation implementation and

TST.7 Test environment testing

TEO.1 Operational use of test environment

TEO.2 Test environment user support

Looking at the Support and Management Processes Groups, we can also see that some technical aspects are addressed:



We see technical aspects addressed in the following processes:

RIN.4 Test Infrastructure

TRM.1 Test Asset Management

TRM.3 Regression Test management

TRM.4 Testware maintenance

3.1 An insight in some tactical questions

3.1.1 Definition and delivery of test environments This topic is addressed in several processes and Practices

From a workflow perspective the preparation of a test environment is a question of test preparation. This is described in the TST.2 "Test Analysis and Design" Process where we find the practices

- Analyse the requirements for the test environment
- Implement the test environment
- Establish a hand over procedure for the test objects.

We also find the process TST.9 Test Environment Testing

Looking at the TST.3 "Test execution" process we find also 2 relevant practices:

- Ensure that the test environment fulfils the requirements
- Check the delivery of the test objects to the test

From an Infrastructure perspective we find the RIN.4 Process that contains the following practices:

- Identify the test infrastructure scope
- Define the requirements for the infrastructure
- Acquire infrastructure
- Establish the infrastructure
- Provide support for the test infrastructure
- Maintain the infrastructure

Looking at this lot of information we also must be aware the supporting processes like Documentation, Quality Assurance, Configuration management or Joint Review are important.

We find also that proper testing depends on proper use and support of the Test Environment:

- TEO.1 Operational use of test environment
- TEO.2 Test environment user support

Last but not least we also find some practices in the measurement framework:

- GP 3.1.4 Identify the required infrastructure and work environment for performing the standard process
- GP 3.2.5 Provide adequate process infrastructure to support the performance of the defined process.

3.1.2 Test Automation

Every process model has to deal with the fact that test automation becomes more and more complex on the one hand but also has options how to perform. As described, the major options are native capture replay or parallel implementation. TestSPICE tries to support both approaches:

Parallel implementation:

- TST.5 “Test automation design”
- TST.6 “Test automation implementation”

Native capture replay:

The solution for native capture replay is hidden in the TST.3 “Test Realization and Execution” Process where we find the practice “Automate the tests”

From a Reference Model perspective we notice, that the existing SPICE conformant Reference Models are not work flow oriented and the RIN.4 Process is much to generic to reflect the importance of test environment and test automation properly.

Last but not least we have the issues with test data:

First we have to check the test case workflow: Logical Test Case, Concrete Test Case, and physical Test Data.

If we search for this workflow we find it in the TST.3 “Test Realization and Execution” Process.

- Design the concrete test cases
- Request the physical test data
- Check the test data for completeness and appropriateness

After all we see that technical issues are multi-dimensional and complex.

4 The practical experience with TestSPICE

In the practical assessment business the issues with technical topics is not really critical. Normally the approach of the organisation to test environments, test automation and test data can easily be evaluated and a proper scoping is possible. But TestSPICE assessments are currently performed by very experienced assessors, so that up to now no problem was reported.

5 The future of TestSPICE

Currently the TestSPICE SIG is reviewing the technical issues to find a less complex solution.

One outcome from the last reviews is a proposal to pool all test environment related issues into a separate process group called “Test Environment & Automation” which could for example integrate the following processes:

TEA.1 Test Environment & Automation Specification

TEA.2 Test Environment & Automation Design

TEA.3 Test Environment & Automation Implementation

TEA.4 Test Environment Testing

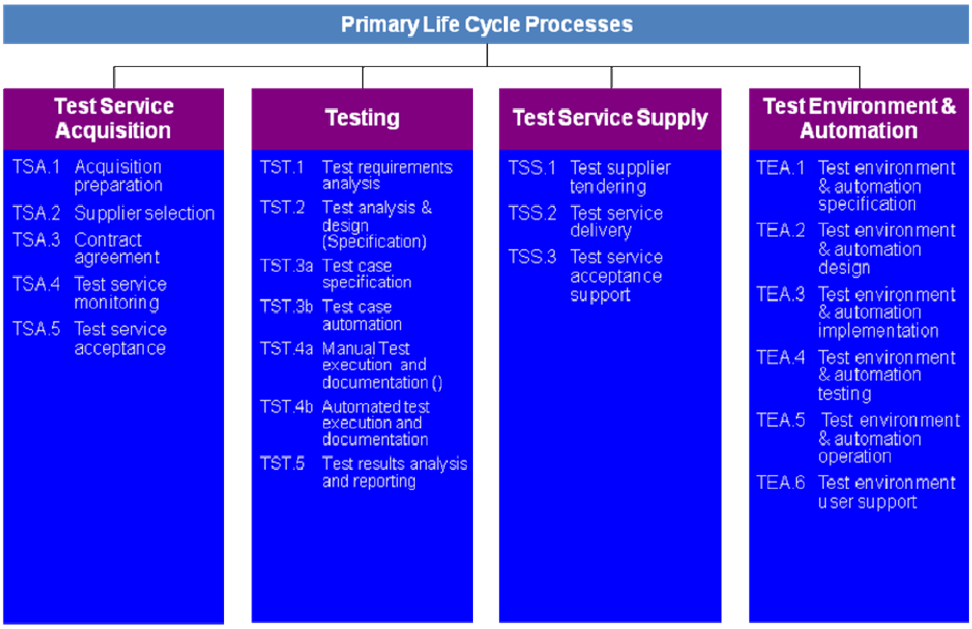
TEA.5 Test Environment Operation

TEA.6 Test Environment User Support

Therefore most test environment related issues could be eliminated from the TST-processes – only the (business/user) requirements for the test environment and test automation should remain in TST.1 Test requirements analysis.

On the other hand it could make sense to distinguish in the TST-processes between manual and automated tests for test case specification and test execution – because their performance indicators could look quite different.

Following these ideas a future TestSPICE-version might look like this:



6 Literature

1 Christian Knüvener, TestSPICE – SPICE für Testprozesse, SQ-Magazin 17/2010 S. 26-27

2 Michael Steiner et al, Make test process assessment similar tot o software process assessment - the TestSPICE approach(Journal of Software Maintenance and evolution, published online at Wiley online library(wileyonlinelibrary.com)DOI: 10.1002/SMR 507

3 Blaschke Monique et al, The TestSPICE approach, Test Process Assessments follow in the footsteps of software process assessments, Testing Experience 12/2009 S. 56ff.

7 Author CVs

DI. Andreas Nehfort

Nehfort IT-Consulting (since 1986) □ www.nehfort.at
Principal Consultant for IT & Software Process Improvement (SPI)
Focus: Process Models for Software & Systems Engineering and Process Capability □ CMMI / SPiCE; (Automotive) SPICE Principal Assessor (intacs & VDA certified)
Andreas Nehfort has developed SPICE based assessment models for IT Service Management (□ ISO 20000) and Information Security Management (□ ISO 27000ff)

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.TestSPICE

Michael Philipp (Dipl.-Kfm),

Senior Consultant at SQS since 1998. He has a broad experience of software quality management, project management, software testing and process improvement. He is particularly characterized by experience in international projects (testing, improvement, IT Life cycle), conducting assessments (ISO15504) and as a trainer for the SQS training program

Report on an assessment experience based on ISO/IEC 29110

Motoko TAKEUCHI, Yumi KOISHI, Kazunori SHIOYA, Naohiko KOHTAKE, Seiko SHIRASAKA

Abstract

January 2011, a new standard ISO/IEC 29110[1] - Life Cycle Profiles for Very Small Entities (VSEs) has published. As we understood, this is a new process model intend to apply readily not only for very small entities (VSE) but also for a division/project of big and medium enterprises.

Since around 2000, SRA has been promoting in-house standardization of software development process toward the quality improvement as goal. However, it is still in the process of institutionalization, hence the processes and the workproducts varies for each project so far, as a result, the product quality varies also. To make a new step for a breakthrough, it is needed to know actual project status correctly, identify the problems, and address the identified problem firmly, thus the improvement cycle must be implemented to establish the quality activities. To achieve the goal, we have chosen the ISO/IEC 29110 standard for the small project as a basis for a process assessment method.

To evaluate effectiveness of the effort of the assessment using ISO/IEC 29110, we have performed a series of mini-assessment using a task checklist according to ISO/IEC 29110-5. On the trial, a material from the VSE Center of Networks[2] which promotes ISO/IEC 29110, we customised it for our organization and developed an assessment method based on ISO/IEC 15504[3] process assessment framework.

In this paper, we introduce an overview of our assessment method and explain our experience of process improvement using this method with ISO/IEC 29110 applied for several trial assessments, its results and considerations.

Keywords

ISO/IEC 29110, software process model, software process assessment, software process improvement, Very Small Enterprise, Very Small Entity, VSE

1 Introduction

Since SRA has been promoting in-house standardization of development process toward the quality improvement. However, the processes and the workproducts are not actually standardized in every project, hence the product quality varies as resulted. To improve the quality, by understanding on the characteristics of the project, it is needed to know actual project status and performance correctly, identify the problems, and address the identified problem, this way the improvement cycle must be implemented and performed continuously. Meanwhile, ISO/IEC 29110 standards for Very Small Entities (VSEs), a software development process model has published on January 2011. Furthermore, this standard can be applied for the mini self-assessment and the third party assessment[4]. We supposed that this mini assessment could be applied for our in-house projects performing readily and continuously, as an improvement cycles performed, then the quality improvement might be achieved.

Therefore, we performed the series of the mini-assessments, to evaluate the effectiveness of the effort of the assessment using ISO/IEC 29110.

1.1 Consideration of assessment using ISO/IEC 29110

ISO/IEC 29110 process should be assessed using ISO/IEC 15504 compliant approach. As Varkoi described[5], current ISO/IEC 29110-4-1[6] is insufficient to perform formal capability assessment. However our purpose is to seek better way toward the improvement, we need to know the software development project statuses with less effort, hence we have chosen a mini-assessment method based on the ISO/IEC 29110 as a basic software development process with following conditions:

- Common business practices and conventions are shared

  The assesses at the project side and assessors at assessment side are belonging to same organization.

- Understand the software process lifecycle (ISO/IEC 12207)

  Project members have enough knowledge of the SLCP and enough experiences to perform tasks of software development.

- ISO/IEC 15504 Provisional Assessor have enough knowledge and experiences

  Assessment team members have qualified as ISO/IEC 15504 Provisional Assessor

- Supported by ISO/IEC JTC1/SC7 WG24 expert

  Although project members did not have any knowledge on ISO/IEC 29110 and also assessors had not enough knowledge of the 29110, a WG24 expert supported to help understand the 29110 tasks and workproducts. However, both members understand basic knowledge of SLCP, so a task based checklist for self-assessment downloaded from the VSE Center of Networks were considered to use with japanization and customization.

For above conditions, the task based checklist tool for mini-assessment based on ISO/IEC 15504-5 defined tasks are used for our trial.

Note: A Very Small Entity (VSE) is defined as an enterprise, organization, department or project having up to 25 people. ISO/IEC 29110 provides a set of standards and guides have been developed according to a set of VSEs' characteristics and needs. The guides are based on subsets of appropriate standards elements, referred to as VSE profiles. The purpose of a VSE profile is to define a subset of International Standards relevant to the VSE context. (Cited from ISO Catalogue of ISO/IEC 29110)

2 Mini trial process assessments

2.1 Method of mini-process assessments

2.1.1 Purpose of the trial

ISO/IEC 29110 is a lightweight process model develped for very small entities (VSE) and it is composed of 2 processes: Project Management and Software Implementation. In addition, it defines minimum activities and workproducts that are thought to be need for VSE specified by the International Standardized Profile mechanism.

We consider that the these two processes are the common and essentials of the software development which defines a minimum set of activities and workproducts to be performed during any software development activities. Moreover, we consider that this essentials can be common for every kind of project. So we expect that confirming the essentials to be performed firmly in every project, can be helped to raise the bottom of the basic capability of developement up through by the mini process assessments.

Therefore, we decided to perform the trial in order to evaluate the following issues;

(1) Can ISO/IEC 29110 be applied to every kind of project as a common and minimum process?

(2) How much the effort to be estimated?

(3) What kind of effects can we expect by the minimum process assessments?

Since the small number of projects we tried at the trial stage, we chose questionnaires analysis as a qualitative analysis to evaluate the results.

2.1.2 Work Team

An assessment team icluding authors was formed with 4 members who had the qualification of iNTCAS Provisional Assessor (ISO/IEC15504).

The Assesor of the assessment team assessed projects, and evaluated based on the assessment outcomes.

2.1.3 Project selection conditions

To evaluate whether ISO/IEC 29110 is effective for what kind of project, the projects are chosen by the combination of the following categories;

- Project size
- Project phase
- Outsourcing existence

2.1.4 Assessment policy

We declared the following policies in odrer to aim the mini assessments.

- Minimum cost and effort
- Minimum effort for project member

2.1.5 Assessment procedure

We followed the customized assessment procedure according to the one in ISO/IEC15504 process assessment framework as illustrated in Fig.1.

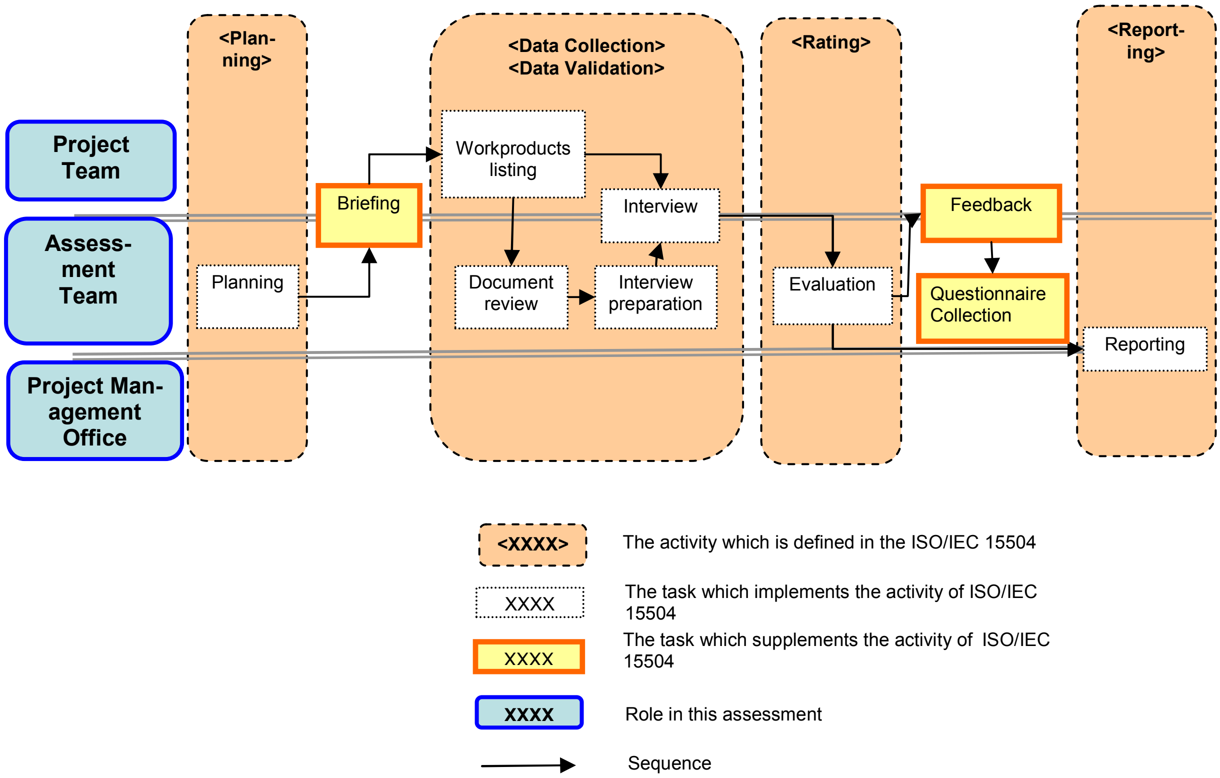


Figure 1: Assessment procedure

Following tasks are originally devised for this trial;

- Project understanding for assessment

In the past, SRA had forced CMM Level 3 like activities to entire organization. But it required heavy efforts to the projects and not many effects were gained, then the activity stopped. After that, there is a general refusal on any assessment over the company.

Considering this situation, we set briefing with project at the beginning of the assessment. In “Briefing“, the assessment team explained the purpose and procedures of the assessment, and expected efforts for the project, then got project understanding for assessment.

- Time restriction for meetings

In order to hold down the project cost, time restriction was set for certain meetings. (“Briefing“ and “Feedback“ is maximum 60 minutes, “Interview“ is maximum 90 minutes.)

- Assessment material customization

Although we translated a checklist from the VSE Center of Networks into Japanese and customized it to adopt for our organization, also several materials taken from ISO/IEC 29110 are customized aiming the materials to be comprehensible to the projects.

In order to hold down the assessment cost, we provided the mapping of workproducts defined by ISO/IEC 29110 and the must be made workproducts in our company at “Workproducts listing“. The project documentation that corresponded to the ISO/IEC 29110 was filled in by the project.

- Access to project repository

The assessment team allowed to access project folder directly, so “Document review” was performed only by the assessment team to reduce the efforts of the project member.

2.2 Experiments

According to the project selection conditions, 8 in-house projects were selected in all.

The following table shows the selected project by combination of three categories.

No.	Project size	Project phase	Outsourcing existence
1	Large	Software Architectural Design	Exist
2	Middle	Software Architectural Design	None
3	Large	Software Requirements Analysis	Exist
4	Large	Software Requirements Analysis	None
5	Small	Software Architectural and Detailed Design	None
6	Small	Project closed	Exist
7	Middle	Project closed	None
8	Small	Software Construction	Exist

The assessment team performed the mini assessments to these selected projects at part time. It took about 3 to 4 weeks to accomplish 1 project assessment.

After we performed the assessment for all projects, we collected the assessent results and analyzed effect of applying ISO/IEC 29110 as an improvment model.

3 Effects observed

In this chapter, we describe effects of the trial from 3 viewpoints; assessment results, cost measurement and summary from questionnaire.

- Assessment results

We obtained the evaluation results in NPLF according to the checklist used by mini-assessment. In addition, we obtained 97 observed findings from “issues and problems”, “things that project devises”, “things to be introduced to other projects” and “issues in organization”. Average 12 findings per project are extracted. (Minimum 10, Maximum 17)

Common issues to the projects, foreseen risks in continuing process and identified issues in organization could be extracted. Especially, many things that the project had not considered were identified as the foreseen risks.

For examples:

“Common issues”

- Concept of “baseline” is not properly understood
- Importance of “traceability” is not understood

“Foreseen risks”

- Operation and maintenance are not willing Requirements Analysis Process

  - No consideration for rebuild software development environment
  - Intensions and reasons to determine requirements and architectures are not documented

“Identified issues in organization”

  - Some company standards are not used or disregarded
  - Mismatch between company standards and ISO/IEC 29110
  - Poor communication and sharing information/know-how among projects

- Cost measurement

The total assessment cost for a project became total 8 hours (about 0.05man/per month) when participating in two people from a project.

- Summary from questionnaire

We found that the mini assessment did not put a great strain on project and was able to extract issues and risks that the project had not considered. Moreover, opportunity to give the advice from the assessor on the worry in the project, or opportunity to inform study case in other project was able to urge awareness on the project. From this, many positive reactions to the assessment that the project wanted to expect the mini assessment continuously were seen.

From the above mentioned, it has been understood that any project was able to find 10 observed findings or more, and common issues ware found among projects by the assessment. And also it has been understood that the mini assessment required few efforts to the project. Moreover, we found that the mini assessment was an acceptable way for the project and was able to extract issues and risks in the project that urge awareness on the project. And also, we found the effect that the advise from the assessor may activate the project.

4 Conclusions

4.1 Considerations

Through the trial, we found that the 29110 consist of minimum basic processes common to most projects from small to large in various domains. And it is useful to identify problems, issues and risks in low efforts also.

Regarding above, the mini-assessment using ISO/IEC 29110 has two benefits followings:

- Firstly, it is useful for the readiness checking for the forth-coming tasks and workproducts.

  This means that the diagnosis on the readiness status for not only the current or performed process but also for the forth coming not yet performed near future process. Therefore, the certain risks are identified in early stage.

- Secondly, the mini-assessments are performed iteratively and repeatedly performed several times in low efforts.

  This means that it makes possible to confirm performed actions for the identified problems found by prior assessment by following mini-assessment within short period. Therefore, identified problems are handled certainly without the risk of being neglected.

To take advantages of the above mentioned, mini-assessment should be performed, not for the evaluation but as a daily routine tool to obtain the correct status of management and task performing, identify the problems and solve the problems. This way the ISO/IEC 29110 mini-assessment is useful we understood.

Meanwhile, a process is needed to ensure the identified problems and the issues are firmly addressed after mini-self-assessment by project itself to facilitate information and know-how useful for other project.

We would pursue that to facilitate the quality improvement activities by combining the combination of the self-assessment by project itself and the third party assessment, and providing the mechanism to help the project could be performed process improvement by itself.

4.2 Future plan

Following issues to be considered:

To facilitate self-assessment by project itself, project member must understand the assessment method and use of self-checklist is needed. Therefore, an appropriate guide or the guidance system should be provided. Currently, a tool for this purpose is under development to be built into the existing risk checking tool SSISQET [7] developed by our group company. We would use this tool as self-assessment tool to help project member to perform self-assessment and also assessment assistance for the assessors.

5 Literature

[1] ISO/IEC29110 Software engineering -- Lifecycle profiles for Very Small Entities (VSEs)

[2] VSE Center of Networks < http://profs.etsmtl.ca/claporte/English/VSE/VSE-network.html >

[3] ISO/IEC 15504-2:2003 Information technology -- Process assessment -- Part 2: Performing an assessment

[4] ISO/IEC DTR 29110-1:2011 Software engineering -- Lifecycle profiles for Very Small Entities (VSEs) -- Part 1: Overview. See 7.2 Assessment Concept.

[5] Process Assessment in Very Small Entities, Timo Varkoi, 7th International Conference on the Quality of Information and Communications Technology, Oporto, Portugal, September 29 - October 2, 2010.

[6] ISO/IEC ISP 29110-4-1:2011 Software engineering -- Lifecycle profiles for Very Small Entities (VSEs) -- Part 4-1: Profile specifications: Generic profile group

[7] SSI Software Quality Examination Tool <http://www.sqet.jp/>

6 Author CVs

Motoko TAKEUCHI

Ms.Takeuchi is a project manager at Software Research Associates (SRA), Inc. She has worked in the software engineering field for more than 15 years, and is engaged in the area of software process improvement for 4 years. She is ISO/IEC15504 certified provisional assessor and PMP. She is a researcher of VSE Center in Japan which is established in 2011.

Yumi KOISHI

Ms.Koishi is an engineer in the process improvement group at Software Research Associates (SRA), Inc. She has experience in software development of the enterprise system and software process improvement. She is ISO/IEC15504 certified provisional assessor and has experience with process assessments in the space domain. She is a researcher of VSE Center in Japan which is established in 2011.

Kazunori SHIOYA

Mr. SHIOYA is a manager at Software Research Associates (SRA), Inc. He has been working in the software engineering field for over 30 years. His interests are in technologies and practices around Software Development activities, especially the Software Process. He also serves as a Japanese delegate of ISO/IEC JTC1 SC7 over 10 years and a member of WG24 since 2006. He is a researcher of VSE Center in Japan which is established in 2011.

Naohiko KOHTAKE

Dr. Kohtake is an associate professor at graduate school of System Design and Management, Keio University. He worked in research and development on avionics systems for rockets, satellites and International Space Station at Japan Aerospace Exploration Agency, and on-board software at European Space Agency. He is ISO/IEC15504 certified provisional assessor and has experience with process assessments in the space domain.

Seiko SHIRASAKA

Mr. Shirasaka is an Associate Professor at Graduate School of System Design and Management at Keio University. He earned a Master's degree in Astronautics from University of Tokyo in 1994 and immediately joined in Mitsubishi Electric Corporation. Since then he had worked on space development for Mitsubishi Electric Corporation. He participated in the development of the ETS-VII (engineering test satellite) and HTV (H-II transfer vehicle). In particular, he was involved with the HTV project from the initial designs until the completion of the first mission. He joined Keio University in systems engineering in AY2004. He appointed to current post in AY2010.

Maturity Differences between Customer and Supplier - Challenges, Problems and Possible Solutions

Tim Kasse & Jørn Johansen
DELTA Axiom
Venlighedsvej 4, DK-2970 Hørsholm, Denmark
kassetc@aol.com and joj@delta.dk
www.deltaaxiom.com

Abstract

In a globalized world, success in business today is linked to success with outsourcing. Several factors have a strong influence in this “game”. In our opinion organizational maturity and process capability are very important.

Low maturity in general is a problem in a Customer-Supplier relationship. It is obvious, that differences in maturity and process capability between Customer and Supplier will cause challenges and eventually problems in a collaboration.

The insight into the difference in maturity between Customer and Supplier is important during the daily work – maybe even more important, when a Customer has to select a Supplier. It is here the “landscape of problems” is formed.

We have based our performance analysis on CMMI® 1.2 for each possible combination of differences in maturity between the Customer and the Supplier. The problems caused by the differences are described along with our best suggestion for possible solutions based on our experiences from working with process improvement in companies worldwide.

It is not possible to present the entire set of results of the performed analysis in this paper. Only the main results and conclusions will be included in this paper. Further work will go on, and the goal is to develop a simple tool to assist companies in their use of outsourcing support.

This work is part of the 3-year and 18 man-year Danish SourceIT research project focusing on outsourcing and maturity, which is supported by the Danish Agency for Science Technology and Innovation under the Ministry of Science Technology and Innovation. Danske Bank, CSC Scandihealth, Nets, Roskilde University and DELTA have been part of this project.

Keywords

Maturity, process capability, customer, supplier, outsourcing.

1 Introduction

Subcontracting, Sourcing, Outsourcing, or simply using suppliers outside the project boundaries has become a way of life. Companies insisting on not to use suppliers one year ago suddenly find themselves in a position of trying to decide how to select a qualified subcontractor or supplier. In addition, the market trend in Western countries to outsource to companies in countries like India, China, Philippines, Taiwan, Thailand, etc. in order to get "cheap" development costs has reached an epidemic proportion. Outsourcing has become the word used to cover all use of suppliers much like Xerox stood for copying during so many years.

There are many important aspects to consider when selecting and managing a Supplier before signing a formal agreement. In our experience during the past 5 years, however, this guidance has not been heeded very closely. Instead, a more common scenario shows that a Customer seeks a Supplier, who claims to have achieved CMMI® maturity level (ML) 5, and the Customer often assumes that the Supplier will work for the good of the Customer simply as a result of the ML 5 rating.

This European Union project is intended to illustrate the importance of supplier management by examining the process capability levels of the Customer and its Suppliers. We should be able to look more closely at process capabilities and what the consequences of expectations to Capability Levels might be for both the Customer's activities for supplier management and the Supplier's ability to deliver adequate products (goods) and services. We will also try to point out what challenges / problems the Customer and Supplier may experience due to differences in process capability levels and suggest some possible solutions.

To set the stage, it is important to define Customers and Suppliers. A clear terminology is necessary to understand the importance of the role. A company may be a Supplier for its Suppliers and a Supplier for its prime contractor.

Customer: A project or organization that is setting up an agreement with an entity outside the boundaries of a project or an organization to develop a product or product component for delivery. Outside the boundaries of a project indicates, that the Customer normally has no control over the Supplier's resources.

Supplier: A project inside or outside the Customer's business unit or organization, that agrees to do the necessary product or product component development according to the requirements of the Customer and deliver within specified constraints such as cost, schedule, quality, and performance.

2 The concept

With those rather simplistic definitions of Customer and Supplier, the following map will be used to help us organize our discussion of differences in process capability between the Customer and Supplier and help us answer some of the following questions:

- What are the main differences in process capabilities between the two parties?
- What typical problems / challenges may be caused by the differences in process capabilities?
- *Are there any relevant solutions in relation to processes that would be the most beneficial to focus on?*
- Are there any innovations or special topics we should pay attention to?

This paper will only report on the italicized bullet points.

The Customer-Supplier Maturity Relationship Map is shown in Figure 1 below. Each field in the map represents a maturity level for the Customer and a maturity level for the Supplier, e.g. C2S4: The Customer at maturity level 2 and Supplier at maturity level 4.

Our work started many years ago at a workshop with the top management of a large Danish organization and was followed by several discussions in a Tecpoint knowledge exchange group (www.tecpoint.dk group 17, Process improvement with 60 members).

From the workshop and the following discussions it was clear, that the difference in maturity level created serious reflections on the basis for collaboration between a Customer and a Supplier. The main conclusions from the discussions are described below.

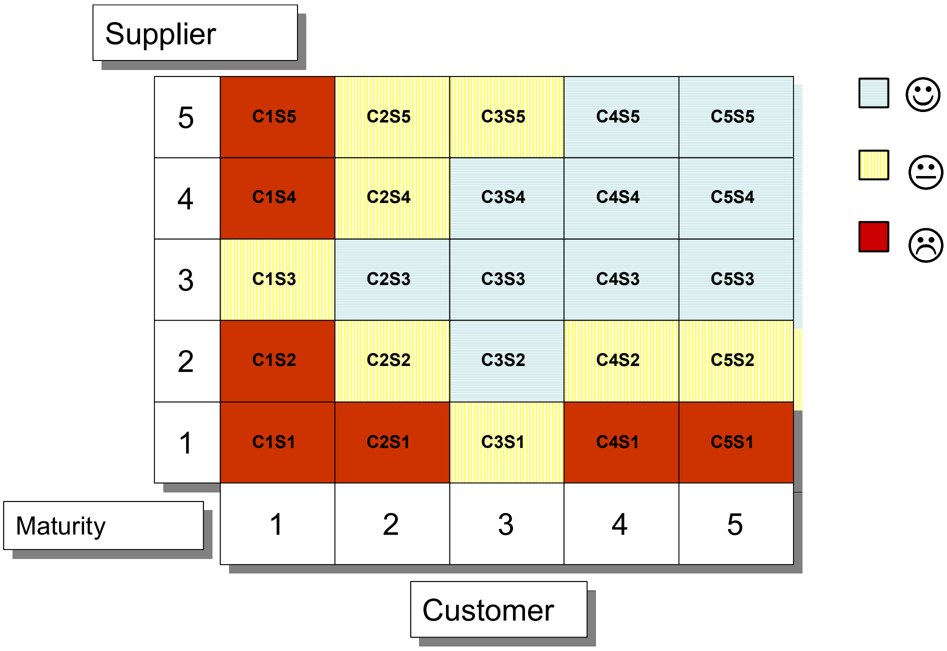


Figure 1. Customer – Supplier Maturity Relationship Map

If a Customer at maturity level 1 should select a Supplier, the most obvious choice would be a maturity level 3 Supplier. The customer could benefit from the supplier's process capability in learning and knowledge exchange. If the supplier is at a lower maturity level, say maturity level 2 or even maturity level 1, the respective maturity levels could actually bring chaos into the collaboration rather than result in an effective relationship. If the Supplier is at a higher maturity level, the differences will bring different types of problems into the collaboration, e.g. the supplier will try to "help" the Customer with the Customer's own processes (which may or may not work) or the Supplier (if unethical enough) could make use of the higher maturity to exploit the Customer.

If the Customer is at maturity level 2 the situation is nearly the same. A maturity level 3 Supplier would be the best choice. It is not a good idea to choose a maturity level 1 Supplier – the Customer still not being strong enough to "control" a maturity level 1 Supplier. Due to Customer maturity level 2 process capabilities it is now more relevant to work together with high maturity organizations, but it is still combined with serious problems and challenges.

If the Customer is at maturity level 3 it is convenient to work together with organizations at maturity level 2 to 4, and possible to handle the problems and challenges when working together with organizations at maturity level 1 and 5.

If the Customer is at maturity level 4 or 5 it is convenient to work together with organizations at maturity level 3 to 5, and possible to handle the problems and challenges when working together with organizations at maturity level 2. The difference is now so large for a Supplier at maturity level 1 that it will give serious problems and challenges to the relationship.

The same arguments can be seen from a Supplier point of view in relation to working together with a Customer.

The map is not symmetric (C3S5 and C5S3) because it is less risky for a high maturity Customer to collaborate with a Supplier at maturity level 3 than it is for a Customer at maturity level 3 to work together with a Supplier at maturity level 5.

3 *The analysis*

This work is based on an intensive study of the CMMI® model in relation to identify differences in process capabilities between two parties at different maturity levels. Based on these differences the typical problems and relevant solutions are deduced.

This work has been discussed with participants in the SourceIT project and in a Tecpoint knowledge exchange group and based on the response the work has been updated.

We will present our ideas by examining the Customer – Supplier process capabilities and their implications as we evolve towards increasing Customer maturity levels.

At the moment only half of the analysis has been performed. Figure 2 indicates the analyzed elements of the map. Although the analysis is not complete yet, the partial results are still usable for the reader.

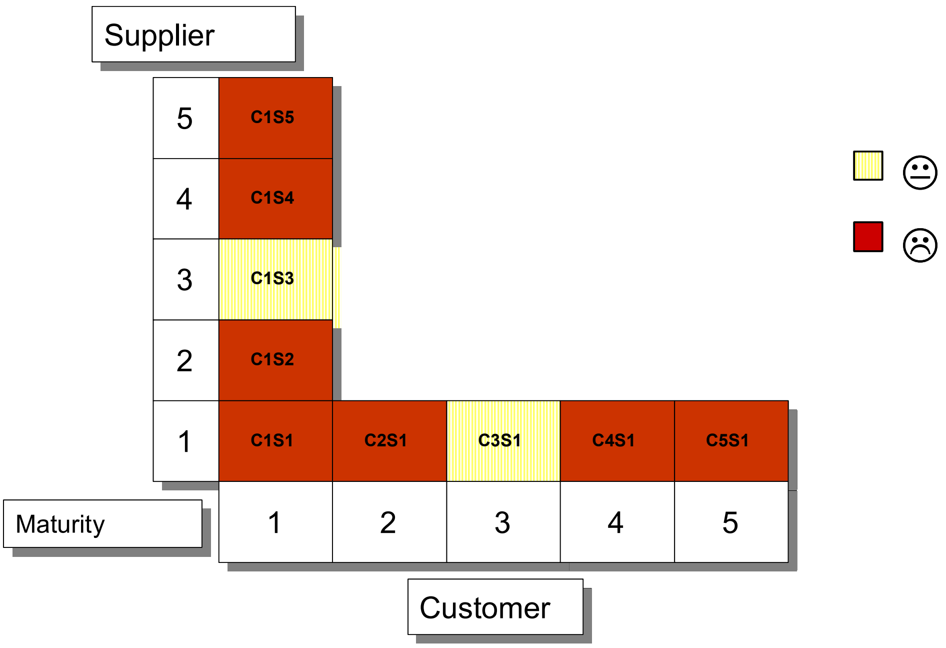


Figure 2. Analyzed Customer – Supplier Maturity Relationship Map elements

In the following text we refer to e.g. the capability of the Customer is “1”, which means the process capabilities indicate a maturity level 1.

3.1 Customer maturity level 1, Supplier maturity level 1

This square stands for the situation where the capability of the Customer is “1” and the capability of the Supplier is “1”. In some people’s eyes, this may well be the worst combination possible, and it is often viewed as the “blind leading the blind.” This may not be the case, however. Other combinations e.g. when Customer capability is “1” and Supplier capability is “5”, can be lethal as we will show later.

Typical Problems / Challenges

The typical problems / challenges with a C1S1 combination are quite simply that products do not meet the requirements and performance constraints, and they are hard to use if not impossible in the operational environment by the people who have to use them.

Cost and schedule overruns are expected to be high.

End Users are rarely satisfied and they are kept unhappy by promises made by the Customer and the Supplier about improvements in the next release.

Relevant Solutions / Special Topics to Pay Attention To

While the C1S1 combination is not desirable, some relevant solutions can reduce the typical problems / challenges described above.

- Choose a Supplier that does have strong application domain developers and sign a Supplier Agreement that identifies key personnel, who cannot be changed without severe penalty
- Choose a Supplier, who is close in proximity, so that daily face-to-face conversations can take place between the Customer and the Supplier instead of having a strong reliance on an effective requirements engineering process
- Eventually hire a consultant to train, coach, and support both the Customer and Supplier in the requirements development and requirements management process
- Ensure that acceptance testing is not left strictly up to the Supplier even if the Customer has to hire an external to mitigate the risk of leaving the acceptance testing criteria and actual testing up to the good will of the Supplier.
- Control the requirements and interface definitions throughout the Product Lifecycle through an adequate form of Configuration Management (CM).

3.2 Customer maturity level 1, Supplier maturity level 2

This square stands for the situation where the capability of the Customer is “1” and the capability of the Supplier is “2”. While the differences between the process and project management process capabilities of the Customer and the Supplier may not be so great depending on the “true” definition of C1, we find that, even in this combination the Customer wants the Supplier to be “good enough”.

Typical Problems / Challenges

The typical problems / challenges with a C1S2 combination are first and foremost when the Customer wants the Supplier to be “good enough”, so that the Customer only needs to spend nominal effort in order to manage the Supplier and get a quality product, on time, within budget and with the functionality and performance required. These constraints may or may not have been given to the Supplier in the product requirements.

Cost and schedule overruns are expected to be lower due to the increased capability of the Supplier. If, however, the Customer is at maturity level 1, the Customer’s own process and quality problems can overshadow the solid project management capability of the Supplier and result in an unsatisfied Customer and End Users.

Even when the Supplier’s capability level “2” is illustrated by solid project management activities, it may be hard or impossible for the Supplier’s schedule to be integrated into the Customer’s project schedule. This is most likely due to the inability of the Customer and the Customer’s unrealistic expectations to the Supplier.

Relevant Solutions / Special Topics to Pay Attention To

While the C1S2 combination can be problematic, some relevant solutions can reduce the risks described above concerning the typical problems / challenges, including those from C1S1.

- The Customer should schedule and ensure continuous communication with the Supplier’s representatives
- The Customer should certainly accept help and support from the Supplier in all aspects of planning, implementing, testing and delivering the product to the Customer / End Users

- The Supplier should invite the Customer's representatives to participate in Supplier quality and configuration management audits
- Customer testing representatives should be invited to participate or observe Supplier acceptance testing before the transition to the Customer or End Users
- The Customer should be encouraged to hire an external to mitigate the risk of leaving the acceptance testing criteria and actual testing up to the good will of the Supplier.
- Regularly review of interface descriptions with the appropriate Customer project representatives
- Supplier representatives for Quality, Configuration Management and Testing should communicate on a daily basis with their counterparts in the Customer's organization.
- The Supplier should work closely with the Customer Project Manager and other designated representatives to ensure periodic and continuous communication, including feedback to the Supplier on its progress and performance.

3.3 Customer maturity level 1, Supplier maturity level 3

This square stands for the situation where the capability of the Customer is "1" and the capability of the Supplier is "3". At this juncture, the differences between the process and project management process capabilities of the Customer and the Supplier start to be significant from the differences between the C1S2 combination. Unfortunately, if the Customer does not do an adequate job managing the level S2 Supplier, the probability that the Customer will do a good job managing an S3 Supplier is distinctly lower. The Customer often believes that because the Supplier is at the S3 Capability Level, it will require even less time and effort to manage the Supplier and get a quality product, on time, within budget and with the functionality and performance documented in the requirements capture and stated in the Supplier Agreement.

Typical Problems / Challenges

The typical problems / challenges with a C1S3 combination are first and foremost when the Customer expects and even requires that the Supplier is "good enough", so that the Customer only needs to spend nominal effort in order to manage the Supplier and get a quality product, on time, within budget and with the functionality and performance documented to some level in the requirements specification.

The Customer does not expect cost or schedule overruns and even expects system performance that may be unrealistic based on the quality of the requirements provided to the Supplier.

The Supplier's organizational process focus, measurement capability and limited predictability process capability may be hard to assimilate by the Customer and misunderstandings may be more common than expected.

The Supplier may feel pushed into level 1 behavior.

The Customer's level "1" capability can diminish the Supplier's level "3" capability and result in a better but less than sterling product from a functionality, quality and performance point of view and still result in an unsatisfied Customer and End Users.

Relevant Solutions / Special Topics to Pay Attention To

While the C1S3 combination can be as problematic as the C1S2 combination, clear solutions can reduce the risks described above concerning the typical problems / challenges, including those from C1S1 and C1S2.

- The Supplier should expect to spend more time and energy on the requirements elicitation and requirements management activities.
- Early phase validation and continuous validation of the requirements throughout the product lifecycle should be one of the highest priorities of the Supplier.

- Alternative solutions should be considered with the End Users, bearing the operational environment in mind
- Training and hand-holding support during the transition of the system from the Supplier into the Customer's / End User's operational environment should be planned and re-planned as the project evolves
- The Supplier should request visits to Customer and End User sites and invite Customer and End User representatives to reciprocal visits in order to mitigate the risk of poor requirements verification and validation.
- The Supplier should utilize process and product measurements to provide the clearest communication possible with the Customer about the product or product components, that are being developed, tested and placed into the End User operational environment.

3.4 Customer maturity level 1, Supplier maturity level 4

This square stands for the situation where the capability of the Customer is "1" and the capability of the Supplier is "4". The differences between the process and project management process capabilities of the Customer and the Supplier are now quite significant. One is based on chaotic behavior and the other is based on quantitative and statistical management. As with the C1S3 combination, if the Customer did not do an adequate job of managing the level S3 Supplier, the probability that the Customer will do a good job managing an S4 Supplier keeps getting lower.

Typical Problems / Challenges

The typical problems / challenges with a C1S4 combination occur when the Customer expects and believe that the Customer does not have to manage the Supplier at all. The Supplier will manage himself and deliver a quality product, on time, within budget and with the functionality and performance documented in the requirements capture and stated in the Supplier Agreement. On top of that, the Supplier will always be able to use quantitative processes because of its Capability Level. Communication with the Supplier will be easy and accurate all of the time. The Customer does not often recognize its own role in contributing to the Suppliers success.

The Supplier's organizational process focus coupled with its use of Process Performance Baselines (PPBs) and Process Performance Models (PPMs) leads the Customer to believe that no matter what input is provided to the Supplier, the results will be statistical in nature, and that all known or heard about quantitative management techniques will be used somewhere in the Supplier's processes. The Customer does not expect cost or schedule overruns at all.

The Supplier may feel pushed into level 1 behavior or at best level 3 behavior since there is low probability that its quantitative process management capabilities will be understood or properly utilized. The Supplier, in all likelihood, will become frustrated that its capability is understood, appreciated, or adequately utilized!

The Customer's level "1" capability will diminish the Supplier's level "4" quantitative capability and result in a product, which is better but less than hoped for. The Supplier may still be able to produce a decent product from a functionality, quality and performance point of view, but still may fall short of having satisfied Customer and End Users.

Relevant Solutions / Special Topics to Pay Attention To

The C1S4 combination can be as problematic as the C1S3 combination. There are clear solutions which can reduce the risks described above concerning the typical problems / challenges, including those from C1S1, C1S2 and C1S3.

- The Supplier should utilize process and product measurements to provide the clearest communication possible with the Customer about the product or product components, which are being developed, tested and placed into the End User operational environment.

- The Supplier should strive to manage the expectations of the Customer with regards to its quantitative and statistical process management capabilities.
  - While the Supplier may indeed have a Process Performance Database or multiple databases and be able to utilize Process Performance Models, it must also be able to calculate the effect of the Customer's requirements input and requirements change requests that will certainly occur due to the low Capability Level of the Customer.
  - Bayesian Networks could be helpful in not trying to over optimize processes that are trying to deal with poor quality data and understanding how the system is really supposed to work in the operational environment.

3.5 Customer maturity level 1, Supplier maturity level 5

This square stands for the situation, where the capability of the Customer is "1" and the capability of the Supplier is "5". The differences between the process and project management capabilities of the Customer and the Supplier are even more significant than in the C1S4 combination. This S5 behavior is based on the Supplier's already established quantitative and statistical management foundation.

In addition S5 Suppliers will be focused on overall organizational performance management to ensure that the Supplier's quantitative quality and process performance objectives are always synchronized with the organization's business objectives.

As with the C1S4 combination, if the Customer did not do an adequate job managing the level S3 or S4 Supplier, it will most certainly not adequately manage a S5 Supplier.

Typical Problems / Challenges

The typical problems / challenges with a C1S5 combination include many of the same problems / challenges of the C1S4 combination. In some cases, the problems are exacerbated!

The C1 Customer will continue to expect and believe that it will not have to manage the Supplier at all.

From another point of view, the S5 Supplier may feel that he can do whatever he pleases and the Customer will never be in a position to challenge the processes used or the quality of the product and services delivered by the Supplier.

Relevant Solutions / Special Topics to Pay Attention To

The C1S5 combination can be as problematic as the C1S4 combination. There are solutions that can reduce the risks described above concerning the typical problems / challenges, including those from C1S1, C1S2, C1S3 and C1S4.

- The Supplier should set expectations for the Customer related to participation in process- and technology improvement activities.
- The Supplier may increase the incentive for the Customer to participate in improvement efforts across the supply chain, by offering process descriptions, training, tools, measures, etc. which would benefit the Customer and therefore the Supplier.
- Supplier representatives should visit the Customer and assist in any implementation of desired processes and document lessons learned from pilots executed in the supplier environment to an extent where it makes sense.
- The Customer should know about results achieved by the Supplier's optimized processes.

3.6 Customer maturity level 2, Supplier maturity level 1

This square stands for the situation where the capability of the Customer is “2” and the capability of the Supplier is “1”. This signifies that although the process capabilities of the Supplier are still not good, the increased Project Management capability of the Customer has improved and may be used with some increased benefit to both.

Typical Problems / Challenges

With the results experienced by the Customer based on the Customer’s Capability Level 2 processes, the Customer often forget once being a capability level “1” Customer and immediately expects the Supplier to change their culture and put project management and quality management practices in place to match those of the Customer.

This new attitude can result in the Customer putting unrealistic pressure on the Supplier for improved commitments and product quality without undergoing the process improvement experienced by the Customer. Actually, this may result in a more stressful relationship than the C1S1 combination.

The typical problems / challenges with a C2S1 combination may still be products, which do not meet the requirements and performance constraints and are hard to use if not impossible in the operational environment by the people who have to use them.

Cost and schedule overruns are expected to be high.

End Users will probably still be unsatisfied and unhappy with the promises made by the Customer and the Supplier about improvements in the next release, which the Customer cannot control with a capability level “1” Supplier.

Relevant Solutions / Special Topics to Pay Attention To

While the C2S1 combination is still not the most desirable combination, the project management and quality management strength of the Customer can be used to provide stronger guidance for the Supplier and even support the Supplier in his weak areas. This is not suggested as the optimal solution but as a possible assist.

- Provide slightly more detailed requirements from the Customer and End User point of view.
- Try to ensure that the Supplier understands the operational environment of the End User and even arrange to have the Supplier meet End User representatives.
- Develop a more detailed WBS, that includes estimated Supplier activities as a roadmap for the Supplier.
- Review knowledge and skills requirements with the Supplier to get them to evaluate the process capabilities of their own project members.
- Ensure that the Supplier understands the Milestones and the required deliverables for each milestone.
- Share the risks from the Customer point of view with the Supplier and brainstorm other possible risks.
- Control the lifecycle work products through the Customer configuration management system or send CM representatives to the Supplier’s site and conduct CM activities similar to those that are followed at the Customer’s site.
- Assist the Supplier in the transfer of the lifecycle product or product component into the Operations and Support environment.
- Conduct a quality assurance audit on the site of the Supplier with Supplier representatives
- In some cases, due to the criticality of the product or product components being developed by the Supplier, Customers have been known to place highly qualified Quality Assurance representatives on the Supplier’s site to provide daily control project.

- Suggest a few basic measures for the Supplier to consider and provide training to help its understanding as much as possible.
- Choosing a Supplier that is close in proximity so that daily face-to-face conversations can take place between the Customer and the Supplier can still be helpful to a S1 Supplier
- Ensure that acceptance testing is not left strictly up to the Supplier, even if the Customer has to hire an external to mitigate the risk of leaving the acceptance testing criteria and actual testing up to the good will of the Supplier.

3.7 Customer maturity level 3, Supplier maturity level 1

This square stands for the situation where the capability of the Customer is “3” and the capability of the Supplier is “1”. This signifies that although the process capabilities of the Supplier are still not good, the increased focus on process and measurement at the organizational level of the Customer puts the Customer in a strong position to provide process and product quality guidance to the capability level “1” Supplier and possibly to assist the Supplier in his own process improvement initiative especially for selected critical processes and work products.

Typical Problems / Challenges

With the uniform results experienced by the Customer based on his capability level “3” processes, the Customer might expect a lot more from the S1 Supplier than the S1 Supplier is able to handle.

However, the C3 Customer may also have a very good reason for choosing the S1 supplier. With their organizational knowledge how to handle acquisitions, it is doubtful that a C3 Customer would contract with a S1 Supplier and simply “hope” that the Supplier will do a better job than the S1 capability suggests.

Two major or even critical problems could evolve from this Customer – Supplier combination:

- The Customer might reach the point where he thinks that if he simply gives his processes to the Supplier, all problems will be solved or, at the very least, the Supplier will gain capability overnight
- The Supplier, understanding that they have something special, which the Customer wants, may try to play the “I am special” card, while they believe that the Customer would never be able to afford to terminate the contract.

Relevant Solutions / Special Topics to Pay Attention To

While the C3S1 combination seems to be even less desirable, the project management and quality management strength of the Customer, now combined with the Customer’s organizational process and measurement strength, can be used to provide focused guidance and support for the S1 Supplier. The solutions from C2S1 are also relevant.

- Select Supplier technical solutions for analysis and technical review to confirm technical progress criteria or the satisfaction of contractual requirements
- Offer the Customer’s process definitions along with defined product lifecycles to the Supplier
    - Training, mentoring and coaching would be expected to be provided to the Supplier to ensure that the Supplier is able to actually follow the Customer’s processes
    - Tools used at the Customer’s site to support the process use may also be offered to the Supplier
    - The Customer may have to help the Supplier to tailor the processes to make the Supplier the most efficient and effective for the Customer’s project

- The Customer may have to ensure that the processes provided to the Supplier were for actually identify the Customer's most critical activities in order for the Supplier to deliver the highest performance as well as products and product components of the highest possible quality
- The Customer may focus his Quality Assurance support activities for the Supplier on the effective use of the provided processes.

3.8 Customer maturity level 4, Supplier maturity level 1

This square stands for the situation where the capability of the Customer is "4" and the capability of the Supplier is "1".

When a Customer enters the realm of High Maturity, the way processes are examined together with the measures and measurement analysis techniques start to have a significant effect on the project's ability to not only manage its activities but also to be able to predict the quality and process performance achievements.

Typical Problems / Challenges

Clearly, this combination can be very stressful, unless the Customer has chosen a Supplier for a very clear purpose and is willing to support the Supplier in selected aspects of process, quality and measurement.

With the quantitative project management results experienced by the Customer based on his capability level 4 processes, the Customer would like a higher capability from his Supplier. However, here we will make the assumption, that the Supplier at capability level "1" has significant contribution, which the C4 Customer wants, and therefore the Customer is willing to deal with the more chaotic behavior of the Supplier than is normally seen in the Customer's organization.

Besides the two major or even critical problems in C3S1:

- The Customer may decide to try to apply his own statistical measurement techniques on existing Supplier processes without a strong knowledge of what the results will indicate and how they might be used in some positive way for the Supplier.

Relevant Solutions / Special Topics to Pay Attention To

While the C4S1 combination seems to be odd for sure, both parties must remember the criteria by which the S1 Supplier was chosen in the first place, and work towards a cooperative relationship.

Some ways to accomplished in addition to what was offered for the C2S1 and C3S1 combination above include:

- Take care when developing the Acquisition Strategy for the S1 Supplier and ensure that it is clear, concise and reflects the acquisition requirements such as:
  - Ensure that the requirements of Data Management are defined, understood and tracked very closely and give continuous feedback to the Supplier.
  - The Customer must determine how his processes interact with the Supplier's processes to enable seamless execution of the project and a successful Customer-Supplier relationship.
  - The Customer should look at measures, measurement objectives, and quantitative and statistical techniques and determine, what would make any sense to introduce to the Supplier.
  - The Customer's risk management plan must be very well detailed, if any Supplier risks could negatively affect the Customer's quantitative project objectives.

3.9 Customer maturity level 5, Supplier maturity level 1

This square stands for the situation where the capability of the Customer is “5” and the capability of the Supplier is “1”. The differences between the process and project management process capabilities of the Customer and the Supplier are even more significant than with the C4S1 combination. This C5 behavior is based on the Customer’s already established quantitative and statistical management foundation.

Typical Problems / Challenges

This combination, like the C4S1, can be very stressful, unless the Customer has chosen a Supplier for a very clear purpose and is willing to support the Supplier in selected aspects of process, quality and measurement.

With the optimizing processes results experienced by the Customer based on the capability level 5 processes, the Customer would like a higher capability level from the Supplier. However, here we will again make the assumption that the Supplier at capability level “1” has significant contribution, which the C5 Customer wants, and therefore the Customer is willing to deal with the more chaotic behavior of the Supplier than is normally seen in the Customer’s organization.

Besides the two major or even critical problems in C3S1 and C4S1:

- The Customer may offer advice to the Supplier, how to search for incremental and innovative processes and technologies without the quantitative base used by the Customer. This may lead the Supplier to believe, that it is nothing more than reviewing journals about new emerging technologies coming from Apple or Microsoft.

Relevant Solutions / Special Topics to Pay Attention To

While the C5S1 combination really seems to be odd, both parties must remember the criteria by which the S1 Supplier was chosen in the first place, and work towards a cooperative relationship.

Some ways that could be accomplished in addition to what was offered for the C2S1 and C3S1 combination and even C4S1 combination include:

- The Customer should set expectations for the Supplier related to participation in process- and technology improvement activities, and the sharing of associated costs and benefits could be documented in the Supplier Agreement
- The Customer may be willing to share the costs of their improvements, if the Supplier would try out selected parts of those, that were identified to be the most critical from the Customer point of view
- The Customer may increase the incentive for the Supplier to participate in improvement efforts across the supply chain, by offering process descriptions, training, tools, measures, etc. that would benefit the Supplier and therefore the Customer
- The Customer may send representatives to the Supplier to assist in any implementation of desired processes and document lessons learned from pilots executed in the Supplier environment to the extent. where it makes sense
- The Customer should inform the Supplier about results experienced by the Customer with the optimized processes.

4 Conclusion

Although the analysis has not yet been completed, we believe that we have shown that difference in maturity level and process capabilities between Customer and Supplier results in a lot of problems and challenges. The analysis shows that the number of problems and challenges is increasing with the number of differences (adding on problems) and our overall recommendation is to focus on the difference in maturity level – and in maturity level in general.

The collaboration is based on processes. The higher maturity level and the less differences the easier the collaboration becomes.

Of course this study is based only on a model and supported only by our knowledge and experience. However, we believe the many people, who have supported the development of the CMMI®, ISO 15504 and other maturity models, that professionalism is correlated with maturity and process capability.

Our work will continue and the map as well as a final analysis will be completed during the next quarter.

5 References

[1] Kasse, T., "Practical Insight into CMMI®, Second Edition, Artech House 2008.

[2] Chrissis, M. B., et all, "CMMI® Guidelines for Process Integration and Product Improvement", Second Edition, Addison-Wesley 2007.

Author CVs

Tim Kasse, DELTA Axiom

Tim Kasse's experience in the computer industry is outstanding and includes over 40 years of job tasks ranging from computer operations, managing an open systems development group to process improvement consulting. He has worked as Software Developer, Manager of software development, Manager of hardware and systems development, Software Quality Assurance Engineer, and Manager of quality management including software, hardware, systems, and manufacturing. Mr. Kasse is considered to be a world citizen from his work in 26 countries on three continents with multiple cultures and in three languages.

Mr. Kasse has spent four years at the Software Engineering Institute. Mr. Kasse is SEI-certified Lead Appraiser for CMMI-DEV v1.2 and CMMI-ACQ v1.2 SCAMPI Appraisals and certified High Maturity Lead Appraiser. He is also certified CMMI Instructor for CMMI-DEV v1.2 and CMMI-ACQ v1.2. He has participated in over 100 process assessments and consulting engagements throughout North America, South America, Europe, Asia, and the Middle East.

His latest book A Practical Insight into CMMI – 2nd Edition was published by Artech House in September 2008. Mr. Kasse holds a Master's degree in Computer Science and a Bachelor's degree in Systems Engineering. He focuses on helping companies balance the achievement of business objectives with planned process improvement.

Contact: Tim Kasse, Delta Axiom Alliance Partner – United States, Europe and Asia. 1428 Chase Oaks Drive, Keller, Texas 76248, United States. Phone +1 817-576-3142 or email kassetc@aol.com

Jørn Johansen, DELTA Axiom

Jørn Johansen is Manager of the DELTA Axiom department at DELTA. He has an M.Sc.E.E. from Aalborg University and more than 30 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.

For the last 16 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 man-year SPI project and Talent@IT, a 26 man-year project involving 4 companies as well as the IT University in Copenhagen and DELTA. Currently Mr. Johansen is the Project Manager of SourceIT an 18 man-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.

Mr. Johansen is also one of the main drivers behind the SPI Manifesto and Editor on the ISO/IEC 33014, Guide for process improvement.

Contact: Jørn Johansen, DELTA, Venlighedsvej 4, DK-2970 Hørsholm, Denmark, phone +45 72 19 40 00 or e-mail : joj@delta.dk

Interpreting CMMI-SVC v1.3 practices for a really small non-IT consultancy services organization

Arthur Maria do Valle[1]*, Bianca Maria do Valle*[2]

[1] *ISD Brasil, Av Cidade Jardim. 400,*
01454-901 São Paulo-SP, Brazil
arthur@isdbrasil.com.br

[2] *TRENDSET Consultoria Ltda, Av. Água Verde. 1575,*
80620-200 Curitiba-PR, Brazil
bianca@trendset.com.br

Abstract

Since CMMI for Services is a quite new model, some organizations are facing difficulties in order to adopt it. This presentation describes how CMMI-SVC v1.3 ML3 practices could be interpreted for a small service setting. We will present the case of TRENDSET RH, a really small unit of a Brazilian consultancy company that provides consultancy services for people that are looking for jobs and that is operated by an only one service-operator. This presentation also shows some examples of interpretation of selected CMMI-SVC practices to this scenario/context.

1 Introduction

As CMMI for Services (CMMI-SVC) is a quite new model, some service organizations, especially those that are not IT-related services and so probably, have not been previously in contact with CMM for Development (CMMI-DEV) or CMM for Software (SW-CMM), are facing difficulties to adopt CMMI for Services. Additionally, implement any of CMMI models in a small setting is really a challenge, especially because there are a lot of roles that should be performed by a few persons. Although it is difficult, is not impossible, and some successful cases has been reported.

In this paper, we intended to show some interpretation examples of ML2 and ML3 CMMI-SVC v1.3 practices, using a really small unit of a consultancy company as a case study.

TRENDSET is a Brazilian consultancy company. A unit of TRENDSET, internally called TRENDSET RH (Human Resources in Portuguese) provides consultancy services for people that are looking for jobs. Those services include:

- Curriculum Vitae analysis
- Curriculum Vitae elaboration
- Cover letter elaboration
- Job Interview simulation

Since TRENDSET RH operates in a home office manner, all these services are provided virtually, on the internet, using email, MS-Office, MSN and Skype. TRENDSET RH has only one service provider individual and there is also the CEO of TRENDSET that involves himself in the TRENDSET RH operations.

2 CMMI-SVC v1.3 interpretation

In order to share some examples of interpretation, some process areas were previously chosen, basically all the service-related PAs and two CMMI core PAs that are usually hard to implement to a service operation context: PPQA-Process and Product Quality Assurance and CM-Configuration Management. In addition, a sample of generic practices was also selected.

Taking this into account, as examples of practices interpretation of those selected CMMI-SVC v1.3 process areas, we have the following:

2.1 SD - Service Delivery

For the first PA (Process Area) interpretation analysis, we will start with SD - Service Delivery, by far the most essential PA in a service-context, which has three specific goals. The first goal is SD SG 1 – Services agreements are establish and maintained.

After some research about the market that TRENDSET RH aimed to act, including analysis of competitors and target-public (SD SP1.1 - Analyze existing service agreements and service data to prepare for expected new agreements), TRENDSET RH had established some standardized service agreements (SD SP1.2 - Establish and maintain the service agreement). Those agreements are available on the company website for everyone interested on TRENDSET´s services.

For the second goal, SD SG 2 – Preparation for service delivery is conducted, we have the following analysis:

TRENDSET RH established its approach for service delivery, which comprises the way services would be provided; personnel, systems and processes to be used (SD SP2.1 - Establish and maintain the approach to be used for service delivery and service system operations). Once all services would be provided on the web, TRENDSET RH established its website that includes a request management system that uses a web form that allows clients to request one or more services. A payment system and another internal system, which uses Excel spreadsheets, were also implemented in order to manage requested services (SD SP2.2 - Establish and maintain a request management system for processing and tracking request information). Before TRENDSET RH started to publicized its services on the web, using Google AdWords, all service system components had been established and confirmed (SD SP2.3 - Confirm the readiness of the service system to enable the delivery of services),

And finally, we have SD SG 3 – Services are delivered in accordance with service agreements:

As an effect of TRENDSET RH publicity, clients started to request services, using the form available at the company website. So, the service operator registers and processes those requests in accordance with services agreements, providing, this way, the requested services (SD SP3.1 - Receive and process service requests in accordance with service agreements and SP3.2 - Operate the service system to deliver services in accordance with service agreements).

2.2 IRP – Incident Resolution and Prevention

Another service-related PA analyzed was IRP – Incident Resolution and Prevention, which also has three specific goals:

IRP SG 1 – Preparation for incident resolution and prevention is conducted.

Most of TRENDSET RH services incidents are client complaints about their CVs and Cover-letters. For example, some client could complaint about some relevant information, in his/her opinion, that is missing in the delivered CV, such as previous work experiences and courses. Typically, in such cases, incidents are tracking using email applications like MS-Outlook and status of them are recorded in the same spreadsheet used to manage services requests (IRP SP1.3 - Establish and maintain an incident management system for processing and tracking incident information)

IRP SG 2 – Incidents are identified, controlled, and addressed

As mentioned above, incidents, after being analyzed, are addressed by discussing with the client, via email or telephone, if the information to be included in the CV is really a “must-have” information in a CV in order to attract HR analysts attention (IRP SP2.1 - Identify incidents and record information about them, SP 2.2 - Analyze individual incident data to determine a course of action, and SP 2.3 - Resolve incidents).

IRP SG 3 – Approaches to address selected incidents are defined to prevent the future occurrence of incidents or mitigate their impact.

After some complaints about missing information had been reported by clients, TRENDSET has elaborated a CV template with the most relevant topics that should appear in any CV. After that, complaints about missing information have been significantly decreased.

Another kind of incident that TRENDSET was facing was when web conference system, such as Skype or MSN, did not support the job interview simulation in a reasonable speed connection. After being recognized that the underlying cause was the use of a wireless internet connection, a cable connection was always used when job interview simulations were done (IRP SP3.1 - Analyze the underlying causes of selected incidents, SP 3.2 - Establish and maintain solutions to respond to future incidents and SP 3.3 - Establish and apply solutions to reduce the occurrence of selected incidents).

2.3 SCON – Service Continuity

For SCON – Service Continuity, we have the following analysis:

SCON SG 1 - The essential functions and resources on which services depend are identified and documented.

First of all, TRENDSET should identify which functions and resources are essential in its business.

SCON SG 2 - Preparations are made for service continuity.

Although TRENDSET kind of services is not so relevant that any disruption should be avoid at any cost, there are some aspects that TRENDSET should consider, as part of a service continuity plan, such as (SP2.1 - Establish and maintain service continuity plans that enable the organization to resume performing essential functions):

- Have an alternative space to work when home-office facilities could not be accessed, in case of fire or flood.
- Have a backup website if the main one is inaccessible.
- Have alternative ways, like telephone, mail or fax, to collect requests from clients, if the solution in the company website goes down.
- Have someone else capable to provide the requested services, if the service operator gets sick, for instance.
- Have alternative ways to establish a communication, like a telephone line, when providing job interview simulation.

So, TRENDSET RH could write a service continuity plan and the service operator has to be aware of procedures to be triggered when needed, as well as, TRENDSET should put on air the backup systems and solution that were defined in its continuity plan. As TRENDSET RH is a one-person company, simulations are more applicable than training (SP2.2 - Establish and maintain training for service continuity, and SP 2.3 - Provide and evaluate training in the execution of the service continuity plan).

SCON SG3 - The service continuity plan is verified and validated.

After simulation of service continuity plan had been performed, TRENDSET could analyze the results in order to validate and verify if service continuity plan are appropriate to address disruption on TRENDSET services, or if something else is needed (SP3.3 - Analyze the results of verifying and validating the service continuity plan).

2.4 STSM – Strategic Service Management

For analysis of STSM – Strategic Service Management, we consider:

STSM SG1 – Strategic needs and plans for standard services are established and maintained

Based on its strategic plan, market and competitors, TRENDSET RH analyzed data (SP 1.1 - Gather and analyze data about the strategic needs and capabilities of the organization) in order to establish a plan to define its standard services (SP 1.2 - Establish and maintain plans for standard services).

STSM SG 2 – A set of standard services is established and maintained.

As mentioned earlier, standard services of TRENDSET RH are: Curriculum Vitae analysis; Curriculum Vitae elaboration; Cover letter elaboration and Job Interview simulation. For each service, TRENDSET RH had determined that there would be a service description that includes some attributes like the service description itself, the way of acquisition, way of payment, information that customer should provided, expect result/deliverable, maximum delivery time, price, etc (SP2.1 - Establish and maintain properties of the organization's set of standard services and service levels). After that, a service catalog was created and made available on the company website (SP2.2 - Establish and maintain descriptions of the organization's defined standard services).

2.5 CAM – Capacity and Availability Management

For CAM – Capacity and Availability Management, that has two specific goals, we have the following:

CAM SG1 – Preparation for capacity and availability management is conducted.

TRENDSET RH could define its strategy for capacity and availability management that could cover, for instance, using subcontracted HR analysts in order to provide services when service provider individual gets unavailable or in situation of high demand (i.e low capacity) (CAM SP1.1 - Establish and maintain a strategy for capacity and availability management).

CAM SG2 – Capacity and availability are monitored and analyzed to manage resources and demand.

In order to manage capacity and availability, TRENDSET RH could collect and analyze some measures (such as WIP-Work In Progress, Lead-Time, Takt time, queue time, etc) in tools like MS-Excel or Minitab (CAM SP2.1 - Monitor and analyze capacity against thresholds and SP 2.2 - Monitor and analyze availability against targets).

2.6 SST – Service System Transition

Analysis for SST – Service System Transition:

SST SG1 – Preparation for service system transition is conducted

Sometime ago, TRENDSET RH decided to offer its service in Portugal. For this, TRENDSET RH staff analyzed all characteristics of its existing services to identify impacts on its set of services aiming to start to offer them in Portugal. Attributes like language, price and service system components were considered in such analysis (SST SP1.1 - Analyze the functionality, quality attributes, and compatibility of the current and future service systems to minimize impact on service delivery). A plan was established in order to conduct the actions resulting from impact analysis (SST SP1.2 - Establish and maintain plans for specific transitions of the service system)

SST SG2 – The service system is deployed to the delivery environment

All identified actions were conducted, as planned (SST SP2.1 - Systematically deploy service system components into the delivery environment based on transition planning). After this, all service system components were checked to see if all things are OK in TRENDSET RH “operation” (although remote) in Portugal (SST SP2.2 - Assess the impacts of the transition on stakeholders and service delivery, and take appropriate corrective action).

2.7 SSD – Service System Development

For SSD – Service System Development, that is a kind of optional process area, we analyzed all three specific goals, listed below, together, getting the following results:

SSD SG1 – Stakeholder needs, expectation, constrains, and interfaces are collected, analyzed, and transformed into validated service system requirements.

SSD SG2 – Service system components are selected, designed, implemented, and integrated.

SSD SG3 – Selected service system components and services are verified and validated to ensure correct service delivery.

In fact, when TRENDSET had defined its services, this PA was not considered in order to define and develop the HR services. In the other hand, TRENDSET, since that, did not introduced brand-new services neither had significantly changed the current services.

So, if another service comes along, or a significant change in a existing service, TRENDSET would use these practices in order to develop consistent services.

In such case, needs, expectations and constrains should be collected, analyzed and transformed to customer requirements and be refined into service system requirements (SSD SP1.1 - Collect and transform stakeholder needs, expectations, constraints, and interfaces into prioritized stakeholder requirements and SP 1.2 - Refine and elaborate stakeholder requirements to develop service system requirements). After validation of these requirements (SSD SP1.3 - Analyze and validate requirements, and define required service system functionality and quality attributes) service system solutions should be analyzed and selected; designed, implemented and integrated (SSD 2.1 to 2.5), as well, validated and verified (SSD SP 3.3 - Verify selected service system components against their specified requirements and SP 3.4 - Validate the service system to ensure that it is suitable for use in the intended delivery environment and meets stakeholder expectations).

Imagining that TRENDSET want to change the current TRENDSET service “CV elaboration” to an automatized solution that uses client´s Linkedin profile information to automatically generate a CV using pre-defined CV templates.

In this case, SSD would be performed in order to identify, analyze, verify and validate such requirements. For this, TRENDSET should write down all the requirements related to this improved service, such as, final appearance to the client, time to generate the electronic CV, requirements on how to convert one resulting webpage to a PDF file interface, interface requirement concerning to LinkedIn application, etc (SSD SP 1.1 - *Collect and transform stakeholder needs, expectations, constraints, and interfaces into prioritized stakeholder requirements* and SP 1.2- *Refine and elaborate stakeholder requirements to develop service system requirements*). All requirements also should be analyzed/verify (using tests, for instance) and validate (using feedback surveys) to see if they will meet customer requirements and expectation of use (SSD SP 1.3-*Analyze and validate requirements, and define required service system functionality and quality attributes*). After this, some alternative service solutions should be identified and analyzed (SSD SP2.1-*Select service system solutions from alternative solutions*), and the selected solution, for example, that contains besides the service-specific part (i.e. service description, service catalog), a web-based software, such as a API that integrates with LinkedIn Application, should be specified and implemented (SSD SP2.2-*Develop designs for the service system and service system components* and SP2.4-*Implement the service system design*). This solution, for example, also should be integrated itself, be integrated with the others services and services components already in place, such as the payment system, and be integrated with LinkedIn web-system, in order to get information from them (SSD SP 2.3-*Manage internal and external interface definitions, designs, and changes for service systems* and 2.5-*Assemble and integrate implemented service system components into a verifiable service system*). In addition, the whole service solution, that covers more than the software-related aspect, should also be verified/validated (using pilots, testes, peer review, customer surveys, etc) before being transitioned to delivery (using SST for that) (SSD SP 3.1 to 3.4).

2.8 CM – Configuration Management

For CM – Configuration Management, that is a CMMI core PA, we have the following interpretation to TRENDSET RH Scenario:

CM SG1 – Baselines of identified work products are established

Since TRENDSET had defined which are the deliverables when each service are being delivery, this is the reference of which configuration items should be generated along with the services (CM SP1.1 - Identify configuration items, components, and related work products to be placed under configuration management). TRENDSET RH also defined its configuration management system that uses MS-Windows system of files and folder (CM SP1.2 - Establish and maintain a configuration management and change management system for controlling work products).

CM SG2 – Changes to the work products under configuration management are tracked and controlled

As mentioned above, each service has a folder and each customer is a subfolder, where files received or created by TRENDSET RH are placed on version control, using increasing numbers in front of names of files (ex: 1-; 2- and so on)(CM SP 2.2 - Control changes to configuration items). In cases of services that also have a deliverable product, such as a cover-letter, a configuration baseline could be created before been sent to the client (CM SP 1.3 - Create or release baselines for internal use and for delivery to the customer). If the customer requests a change in already released artifact, an impact analysis could be done in order to maintain consistency, for example, between the changed CV and the cover-letter (CM SP2.1 - Track change requests for configuration items).

CM SG3 – Integrity of baselines is established and maintained

TRENDSET RH could use self-audits in its configuration management system, in order to identify and address configuration non-compliances (CM SP3.2 - Perform configuration audits to maintain the integrity of configuration baselines).

2.9 PPQA – Process and Product Quality Assurance

For PPQA – Process and Product Quality Assurance, that is also a CMMI core PA:

PPQA SG1 – Adherence of the performed process and associate work products to applicable process descriptions, standards and procedures is objectively evaluated.

TRENDSET RH has some services that also involve products that are provided to the client, such as cover-letters and CVs. Before delivering those artifacts to customers, an evaluation of content and format/standard could be performed by service operator, as a kind of a self-quality assurance (PPQA

SP1.2 - Objectively evaluate selected work products against applicable process descriptions, standards, and procedures)

PPQA SG2 – Noncompliance issues are objectively tracked and communicated, and resolution is ensured.

Since TRENDSET RH has only one level of management, only non-compliances that cannot be solved at operational level could be reported and then solved by TRENDSET CEO (PPQA SP2.1 - Communicate quality issues and ensure the resolution of noncompliance issues with the staff and managers).

2.10 Generic Practices

As example of interpretation of generic practices, we have selected this sample:

For GPs 2.8 – Monitor and Control the Process: In order to the execution of processes be managed, the service provider individual could use a schedule or even a spreadsheet in order to monitor and control him/her activities.

For GPs 2.10 – Review status with Higher Level Management: In order to review status with higher level management, the service provider individual could make reports or even meeting to discuss with TRENDSET CEO the performance, issues and results of all TRENDSET RH processes.

3 Conclusion

As a conclusion, we would say that interpret CMMI SVC v1.3 practices for a non-IT service organization is a quite different from what would be considered in a traditional IT-related services context. In addition, adopt service best practices in a really small unit is, probably, the biggest challenge.

References

1. (SEI, 2010) CMMI Product Team. “CMMI for Services, Version 1.3” (CMU/SEI-2010-TR-034). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2010. http://www.sei.cmu.edu/library/abstracts/reports/10tr034.cfm.

How amendments to the Medical Device Directive affect the Development of Medical Device Software

Martin Mc Hugh, Fergal Mc Caffery, Valentine Casey
Regulated Software Research Group
Dundalk Institute of Technology
martin.mchugh@dkit.ie, fergal.mccaffery@dkit.ie, valentine.casey@dkit.ie

Abstract

A recent revision to the European Medical Device Directive (MDD 2007/47/EC) made fourteen amendments to the original directive (93/42/EEC). A number of these changes directly affect the development of software for use in healthcare. The most significant change in relation to medical device software development is that standalone software is now seen as an active medical device. Prior to this amendment medical device software was developed in accordance with the IEC 62304 standard. However, IEC 62304 is not sufficiently comprehensive to provide guidance in the development of standalone software as an active medical device. Medi SPICE is currently being developed to fill the gaps left by IEC 62304 in developing standalone software as an active medical device and to provide medical device software developers a single point of reference for developing software for use in healthcare.

Keywords

IEC 62304, Medical Device Directive, MDD, Software Process Improvement, 2007/47/EC, AIMD, Medi SPICE

1 Introduction

Software is ubiquitous in daily life however, it is more critical in some areas than others. Failures in medical device software can have severe or fatal consequences. Between June 1985 and January 1987, four people died and two were left permanently disfigured by a software controlled radiation therapy machine known as Therac-25[1]. Therac-25 used software to control a beam spreader plate which reduced patient's exposure to radiation. However, due to software malfunctions the plate was not in place when required and patients received massive doses of radiation. This case highlighted the need for adequate safety measures to be put in place to protect patients and third parties i.e. clinicians, using medical devices controlled by software.

All medical devices used within the European Union (EU) must conform to the current MDD to achieve the CE conformance mark. The current directive i.e. 2007/47/EC[2], amends European directives MDD (93/42/EEC)[3], AIMD (90/385/EEC)[4] and the Biocides Directive (98/8/EC)[5] [6]. The revision to the MDD (2007/47/EC) covers all areas relevant to medical devices including risk and quality management. This latest amendment allows for standalone software to be used as an active medical device. With this amendment, incidents involving medical devices such as Therac-25 become more relevant, as now software can be the only element in a medical device subject to conformance requirements.

Consequently methods used to ensure that software is safe and fit for purpose must be reviewed. A number of benefits can be gained by manufacturers employing Software Process Improvement (SPI) techniques, one of which is a reduction in software faults that could potentially result in device recalls [7]. SPI is a continuous cycle of performing an assessment, implementing the recommendations of the assessment and restarting the cycle [8]. This process of continuous assessment and improvement can help reduce the amount of defective software being developed. Essentially the safety of medical device software is determined through the software processes followed during development [9].

IEC 62304[10] is the current medical software lifecycle process standard. This standard contains a number of processes for medical device software development which medical device software developers should follow in order to implement best practice medical device software processes. IEC 62304 is used in the development of software when software is part of a medical device system which consists of hardware plus software. It provides the minimum regulatory medical device requirements within a specified group of processes. However, it excludes all system level processes. As a result, IEC 62304 hands off the system processes to other standards such as ISO 13485[11], IEC 60601-1[12], IEC 61010-1[13], IEC/ISO 90003[14], IEC 61508-3 [15], ISO 14971 [16] and ISO 12207[17] [10].

Section 2 examines the revision to the MDD and highlights what this means with respect to medical device software development. This will include particular reference to the development of standalone software as an active medical device. In section 3, we discuss the existing standards that are appropriate to the development of medical devices with emphasis on satisfying the requirements of the MDD (2007/47/EC). In section 4, we discuss the importance of SPI techniques and recommend a specific SPI model to follow. Section 5 contains the conclusions from this research and our plans to progress this research further.

2 European Medical Device Directive Amendment 2007/47/EC

The recent MDD [2] revision has made a number of amendments to previous directives i.e. [3-5].The MDD (2007/47/EC) came into force on March 21st 2010. In total there are fourteen changes introduced within the MDD (2007/47/EC)[18]. There are four areas within the amendment of the MDD (2007/47/EC) with important significance to medical device software development:

- Standalone Software as an active medical device;
- Validation of software as an active medical device;
- Software localisation;
- Safety Classification.

2.1 Standalone Software as an Active Medical Device

Prior to the release of the MDD (2007/47/EC) provision had been made within the MDD (93/42/EEC) for software to be used as a medical device. However MDD (2007/47/EC) Article 1 Section 2 makes explicit reference to software being a medical device.

"any instrument, apparatus, appliance, ***software****, material or other article, whether used alone or in combination, including the software intended by its manufacturer to be used specifically for diagnostic and/or therapeutic purposes and necessary for its proper application"*

To accompany this change provision has also been made for standalone software to be used as an active medical device. Within the MDD (2007/47/EC) Annex IX Section 1.4 amendment M5 states:

"Stand-alone software is considered to be an active medical device"

This can be difficult to understand particularly in relation to when software is or is not a medical device. An example of software as an active medical device is a software package which is used to calculate treatment doses for oncology treatment devices. A caveat has also been included into MDD (2007/47/EC) to avoid ambiguity in determining if a software package is a medical device.

“software for general purpose when used in a healthcare setting is not a medical device”

This caveat provides some clarity surrounding particular software used in healthcare. In Ireland we await a formal document from the Irish Medicines Board to define exactly what types of healthcare software applications will now be defined as a medical device. This document is expected in the spring of 2011 but in its absence there is still some confusion regarding what standalone software will be classified as a medical device.

2.2 Software Validation

As standalone software is now classified as an active medical device under the MDD (2007/47/EC) guidelines must be put in place to ensure that such software is safe and fit for purpose. To ensure this the MDD (2007/47/EC) Annex I Section 12.1a (amendment M5) states;

“For devices which incorporate software or which are medical software in themselves, the software must be validated according to the state of the art taking into account the principles of development lifecycle, risk management, validation and verification.”

“State of the Art” is used to here to mean what is generally accepted as good practice. Since this requirement was introduced, developers must now validate the software integrated or standalone, regardless of device class. IEC 62304 and its aligned, standards, ISO 13485, IEC 62366 [19], IEC\TR 80002-1 [20], 80001-1 [21], EN 60601 and ISO 14971 are harmonised under the MDD (2007/47/EC) and are seen as a good place to start when validating software.

Whilst these standards are generally accepted and are harmonised under MDD (2007/47/EC) they do not necessarily cover the requirements introduced within the MDD (2007/47/EC).

2.3 Software Localisation

Another change which affects the development of medical device software is software localisation. Under the MDD (2007/47/EC), software sold or used within the EU must be localised into the language of each of the EU countries that the product will be marketed i.e. MDD 2007/47/EC, Article 4.4. Essentially, if an Irish medical device manufacturer wishes to market a medical device into France, the graphical user interface (GUI) must be available in French. A number of difficulties can arise when attempting to perform a software translation [22]:

- Differing file formats;
- Different character encoding;
- Character size constraints;
- Parsing may be required;
- Errors caused in code caused by repossessing.

There are currently a number of organisations that provide translation services to medical device manufacturers such as Global Translations [23].

3 IEC 62304 – Software Lifecycle Processes

Medical device manufactures wishing to achieve regulatory conformance are advised to follow the relevant applicable standards. Evidence of following the applicable standards can improve the process of achieving regulatory conformance. Medical device software developers typically follow IEC 62304 and its aligned standards.

3.1 Processes

IEC 62304 is a medical device software development standard. It was created to provide assistance in terms of the safe design and maintenance of medical device software. Software developed using the IEC 62304 standard is founded upon the assumption that the software is developed in accordance with a quality management standard (ISO 13485), a risk management standard (ISO/IEC 14971) and a product level standard (EN 60601-1). This standard provides a framework of processes divided into activities which are further divided into tasks.

IEC 62304 provides guidance on the development of software as part of a medical device. However IEC 62304 does not provide guidance on all of the necessary processes required to develop stand-alone software as an active medical device. IEC 62304 states;

"This standard does not cover validation and final release of the medical device, even when the medical device consists entirely of software"

With the MDD revision i.e. (2007/47/EC) a medical device can consist only of software. As validation is required to ensure reliability and safety another method of validating standalone software as a medical device is required. One suggestion is to modify the scope of EN 60601-1 to include medical software systems. Another suggestion is to convert IEC 62304 to a product oriented standard in order to make it applicable to standalone software as an active medical device. All of these suggestions have advantages and disadvantages. A favourable outcome would be for IEC 62304 to remain as a software lifecycle standard and that a new standard be developed to specifically facilitate standalone software as an active medical device.

3.2 Safety Classification

IEC 62304 classifies all software based on the risk posed to the patient or user. The devices are classified as follows:

- Class A: No injury or damage to health is possible
- Class B: Non-serious injury is possible
- Class C: Death or Serious Injury is possible

This classification is similar to that of ISO 14971 Clause 4.4, 5 and 6.1. Safety critical software systems can be divided into items running a different software element each with its own safety classification. These items can be further sub divided into additional software elements. The overall software system assumes the highest classification contained within all of the software elements. For example if a software system contains five software elements, four of which may be classified as Class A, but one may be classified as Class C and therefore the overall device receives a classification of Class C. This can be seen in figure 1.

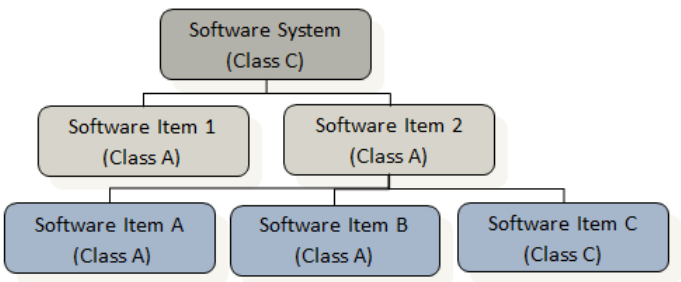


Figure 1. Classification of software items within complete software system

4 SPI and Medi SPICE

4.1 Importance of SPI

SPI is an important element within any software development lifecycle. Many organisations have difficulty in consistently developing high quality software. There are many benefits to be gained by using SPI including [24]:

- Improvements to overall quality
- Increased on-time delivery
- Budget consistency
- Reduced development costs

SPI places the emphasis on defining processes that are appropriate to the project and ensures that these processes are consistently followed. SPI maturity models focus on what has to be done, rather than how it should be done. The benefits of utilising SPI can be seen in many companies e.g. Siemens[25], Alcatel [26], NASA[27] and Motorola[28].

In order for SPI to be successful within an organisation, it relies on a number of critical factors. In 2005 a survey of one hundred and twenty software organisations identified six organisational factors as being crucial to ensure the success of SPI [29]:

- Business orientation;
- Involved leadership;
- Employee participation;
- Concern for measurement;
- Exploitation of existing knowledge;
- Exploration of new knowledge.

Research carried out by Embedded Market Forecasters in 2010[30] provided a comparison between software developed by the embedded industry and software developed by medical device producers. This research showed that 12.9% of medical device projects were cancelled, whilst 11.2% of embedded industry projects were cancelled. This research also revealed that on average 19.4% of the overall budget is wasted due to months lost during project development. The primary reason cited for these problems occurring is incomplete or vague requirements.

An empirical study in 2007 revealed how much importance medical device software developers place upon SPI [31]. The study surveyed organisations developing software for medical devices and medical information systems with the majority of respondents coming from Germany, USA and Sweden. 71% of respondents came from small and medium companies with between ten and two hundred and fifty staff. The remainder of the respondents came from organisations with over two hundred and fifty software developers. Of the respondents, 98% rate software as very important or an important part of their products. However, only 14% of the respondents have either a CMMI (Capability Maturity Model Integration) or ISO 15504 (SPICE) rating. Also only 50% of the respondents follow a defined process to perform requirements engineering, software architecture and design activities and implementation, over 50% of the time. This survey also asked participant’s which process or activities cause the most issues for a software development project.

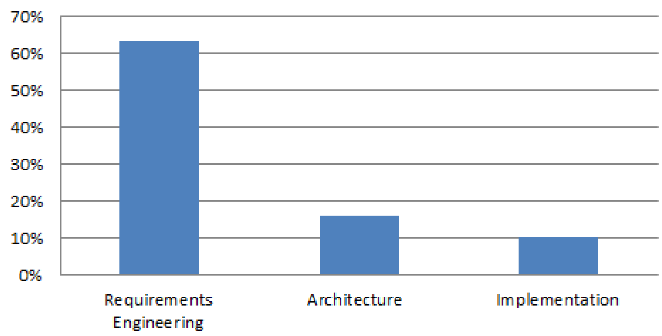


Figure 2. Activities & Processes most difficult to software development projects [31]

Figure 2 shows, the requirements engineering process is seen as causing most of the issues in the development of software for medical devices. Typically organisations following a defined set of processes or activities contained within an SPI model will have minimal problems with areas such as requirements engineering as they are provided guidance on all areas of development. This survey shows an inversely proportional relationship between importance placed on SPI rating/activities and difficulties caused by specific process areas.

Both of these pieces of research [30, 31] show medical device software developers have the most difficulties at the requirements elicitation/engineering stage of development..

4.2 Medi SPICE

Currently medical device manufactures developing standalone software as an active medical device have no single point of reference specific to the development of software for medical devices. To this end Medi SPICE[32] is currently being developed by the Regulated Software Research Group (RSRG) at Dundalk Institute of Technology (DkIT) in association with the SPICE User Group, to help in achieving regulatory conformance and improve overall device safety by employing SPI techniques through the entire development lifecycle.

Medi SPICE provides medical device software manufacturers with a Process Assessment Model (PAM), and a Process Reference Model (PRM), to help implement high quality medical device software processes that will enable seamless conformity to the medical regulatory standards. The PAM is being developed in accordance with the requirements of ISO/IEC 15504-2:2003, and is based on ISO/IEC 15504-5:2006. However ISO/IEC 15504 is not specific to the medical device industry. To overcome this, Medi SPICE incorporates the practices contained within IEC 62304 processes and the other relevant standards into appropriate the relevant ISO/IEC 15504 based processes. Due to the safety critical nature of medical device software, additional safety processes have also been incorporated from +Safe[33] and ISO 14971.

Medi SPICE aims to minimise the volume of software documentation and provide global harmonisation for all medical device software manufactures. Medi SPICE assessment results can be used to indicate the state of a medical device supplier’s practice in relation to regulation. Medi SPICE results can also be used as a criterion for supplier selection. As discussed in section 4.1 many medical device software development companies cite the requirements engineering process as the most problematic of all of the processes. Medi SPICE will provide clear guidance on how to perform this process in addition to other processes of the medical device software development and maintenance lifecycle.

5 Conclusions

The recent revision to the MDD allows for standalone software to be an active medical device and states that software must be validated in accordance with state of the art practices. However IEC 62304 does not provide comprehensive guidance on the development of standalone software as an active medical device as it is purely a software development standard. An example of a process beyond the scope of IEC 62304 is the requirements elicitation stage of development. However research has shown that the requirements elicitations stage causes medical device software developers the most issues during development. This omission can be overcome by employing an SPI model specific in the area of development of medical device software. To this end Medi SPICE is being developed to provide guidance in all stages of development of standalone software as an active medical device.

6 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 (the funding of this project was awarded by Science Foundation Ireland under a co-funding initiative by the Irish Government and European Regional Development Fund), and supported in part by Lero - the Irish Software Engineering Research Centre (http://www.lero.ie) grant 03/CE2/I303_1.

7 Literature

1 N. Leveson, Safeware: System Safety and Computers: Addison Wesley, 1995.
2 European Council, 2007 "Council Directive 2007/47/EC", September 2007
3 European Council, "Council Directive 93/42/EEC Concerning Medical Devices", 14-June-1993
4 European Council, 1990 "Council Directive 90/385/EEC", 5 September 2007
5 European Council, 1998 "Council Directive 1998/8/EEC", February 1998
6 European Council, 2009 "Implementation of Directive 2007/47/EC amending Directives 90/385/EEC, 93/42/EC and 98/8/EC ", 5 June 2009
7 P. Patra and S. Bartaki, "Productivity Improvement Using Ten Process Commandments," PMI Virtual Library, 2009.
8 F. McCaffery and G. Coleman, "The need for a software process improvement model for the Medical Device Industry," International Review on Computers and Software, vol. 2, pp. 10-15, 2007.
9 P. L. Jones, et al., "Risk Management in the Design of Medical Device Software Systems," Biomedical Instrumentation & Technology: July 2002, vol. 36, pp. 237-266, 2002.
10 ANSI/AAMI/IEC 62304, Medical device Software - Software life cycle processes, Association for the Advancement of Medical Instrumentation 19-July-2006
11 EN ISO 13485:2003 Medical Device: Quality Management Systems. Requirements for the Regulatory Process, 24-July-2003
12 EN 60601-1 Medical Electrical Equipment. General requirements for basic safety and essential performance. Collateral standard. Usability 31-May2010
13 IEC 61010-1:2010 Safety requirements for electrical equipment for measurement, control and laboratory use, June 2010
14 ISO/IEC 90003:2003 Software engineering -- Guidelines for the application of ISO 9001:2000 to computer software, February 2004
15 IEC 61508-3:2010 Functional Safety of electrical\electronic\programmable electronic safety related systems, May 2010
16 EN ISO 14971:2009 Medical Devices. Application of Risk management to medical devices 31-July-2009
17 ISO/IEC 12207:1995 Information Technology, Software Lifecycle Processes, 28-February-1995
18 Emergo. http://www.emergogroup.com/newsletters/europe-2007-47-ec-oct2007 2007.
19 IEC 62366 Medical Devices - Application of usability engineering to medical devices, 13-November-2007
20 AAMI/IEC TIR 80002-1:2009 Medical Device Software 1: Guidance on the application of ISO 14971 to Medical Device Software 31-May-2010
21 IEC TIR 80001-1 Application of risk management for IT networks - networks incorporating medical devices
22 Global Vision International Inc. http://www.globalvis.com/mdd-200745ec-and-software-localization/ 2010.

23 Global Translations http://www.gts-translation.com/medical_device_translation.asp. 2011.
24 G. O'Regan, Introduction to Software Process Improvement Springer, 2010.
25 T. Messer, et al., "Siemens Process Assessment and Improvement Approaches: Experiences and Benefits," in 22nd International Computer Software and Applications Conference, Vienna, Austria, 1998.
26 C. Debou, et al., "Alcatel's Experience with Process Improvement," in Better Software Practice for Business Benefits: Principles and Experience ed: CS Press, 1999, pp. 281-301.
27 V. R. Basili, et al., "Lessons learned from 25 years of process improvement: the rise and fall of the NASA software engineering laboratory," in 24th International Conference on Software Engineering, Orlando Florida, 2002, pp. 69-79.
28 M. K. Daskalantonakis, "A Practical View of Software Measurement and Implementation Experiences Within Motorola," IEEE Trans. Softw. Eng., vol. 18, pp. 998-1010, 1992.
29 T. Dyba, "An Empirical Investigation of the Key Factors for Success in Software Process Improvement," IEEE Trans. Softw. Eng., vol. 31, pp. 410-424, 2005.
30 Embedded Forecasters - Embedded Market Forecasters Survey (2010)
31 C. Denger, et al., "A Snapshot of the State of Practice in Software Development for Medical Devices," in First International Symposium on Empirical Software Engineering and Measurement, 2007. ESEM 2007, Madrid, 2007.
32 F. McCaffery and A. Dorling, "Medi SPICE: An Overview," Journal of Software Maintenance and Evolution: Research and Practice, vol. 22, pp. 255-267, 2010.
33 +SAFE, V1.2: A Safety Extension to CMMI-DEV, V1.2, Defence Materiel Organisation, Australian Department of Defence, March 2007,Software Engineering Institute, TECHNICAL NOTE CMU/SEI-2007-TN-006http://www.sei.cmu.edu/pub/documents/07.reports/07tn006.pdf

8 Author CVs

Martin Mc Hugh

Martin received his B.Sc. (Hons.) in Information Technology Management in 2005 and M.Sc. in Computer Science in 2009, from Dundalk Institute of Technology. He is now undertaking research for his Ph.D. in the area of software process improvement for medical devices as part of the Regulated Software Research Group in Dundalk Institute of Technology.

Fergal Mc Caffery

Dr Fergal Mc Caffery is a Lecturer with Dundalk Institute of Technology. He is the 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.

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.

Using Eclipse Process Framework in a medical development environment for process communication

Péter Balázs Polgár[1], *Ferenc Kazinci*[2]

[1] *Eötvös Loránd University*
sirpepe@elte.hu
[2] *Budapest University of Technologies and Economics*
kazinczi.feri@gmail.com

Abstract

This paper describes a case study on using the Eclipse Process Framework in a medical software development organization to consolidate existing process descriptions. While on non-medical fields the EPF was successfully used for process modeling and aiding process improvement efforts, the complex nature of the regulatory and process standard requirements on the medical software field needs special attention. In this environment compliance to the regulatory is more important than improvement, so the process modeling has to implement these requirements in a very strict way, while supporting the business goals of the organization and enabling the improvement of the processes.

Keywords

Process modeling, Eclipse Process Framework, medical software development

1 Introduction

In most development organizations, applying processes and using process improvement is a recognized way to reduce costs and increase the quality of the produced software. In the medical software development field processes are also used, but their aim is to comply with the complex regulations the developers are required to apply and ultimately prevent patients to come to harm. To cope with new technological challenges and business goals development organizations constantly strive to advance and enhance their processes through Software Process Improvement (SPI). To support SPI existing processes are usually organized into process models.

This case study describes the creation of such a process model with a dedicated process modeling tool, the Eclipse Process Framework (EPF) composer [3]. The organization implementing the solution is a small Hungarian development unit (approximately 25 developers with one person supporting the processes) working on the implementation of software for several active medical devices as part of a larger organization within a multinational company. While the processes and process modeling is primarily meant to be used for the smaller software unit, the whole development organization will use parts of it to support their workflows. While the unit used all related standards before, these were implemented as paper based, and formed no coherent model. To enhance the development processes, to better handle the application of standards and to make SPI easier to apply, the creation of a digital process model was proposed. The authors of this paper were predominantly responsible for the implementation between June 2010 and September 2010.

This case study's contribution to the advancement of the state of the art is the implementation of a process model in the medical environment with its strict regulatory requirements. These requirements provided an added challenge to implementation and to the usage of the resulting model. We will describe these challenges in this paper as well as some examples of the usage of the ready model.

Before the implementation a literature study was conducted to identify the challenges associated with process modeling in general and in the medical environment; also the potential advantages.

One of the problems we met and looked into was the alignment of regulatory requirements and the used processes. [14] describes a method on how to keep tailored process models aligned over time with the standards originally used to produce them. This method is based on the comparison of XML data structures. This suggested that keeping the processes up to date with consistency is much easier with a specialized modeling tool.

Before employing a digital, model based approach at the development unit; paper-based process descriptions were used. Historically, this is the most used method to record process requirements and communicate to team members, but in practice to apply process documents is problematic. For example, paper-based process descriptions are not very user-friendly; it is difficult to find relevant information which may cause difficulty for process performers wanting to find required information quickly. It is also difficult to distribute up to date process knowledge in printed form [1][8].

The process descriptions produced would be needed to be more accessible and in particular editable, tailorable and extensible to accommodate live processes. This is possible with a meta-model based process infrastructure as showed by [5]. Also arguing in favor of accessible process descriptions, [4] states that next to the suitability of the user interface of the tool, the underlying meta-model should meet the requirements.

In summary we needed a solution with the following features:

- Use a standardized meta-model suitable for medical development environment;
- Accessible both for process developers and team members;
- Editing tool with collaboration support.

The EPF platform, based on the SPEM [10][11] was chosen, based on its ability for these features.

The remainder of this paper is structured as follows. The second chapter contains a brief overview of the regulatory requirements and the description of some of the problems originating from these requirements. The third chapter introduces the SPEM and the EPF highlighting the useful features. The forth chapter describes the details of the implementation, and we provide an analysis of the results in the fifth chapter. In the sixth chapter we draw some conclusions and present some ideas on further uses of the implemented solution.

2 Regulatory requirements for medical software development

When medical software is concerned there is always some level risk involved, either for the patients (for example too much medication) or for the operators (for example chemicals or radiation from a device). To mitigate this and ensure patient, operator, i.e. user safety in general, manufacturers are required to comply with a number of standards.

Medical software development in general, and in particular the development of devices with embedded software (active medical devices) is a well regulated field. Process and design standards exist for most levels of development. The regulation spans from the quality system down to the individual development tests. Besides international standards, several national authorities published their own guidance which the medical device or the development process should comply with if the manufacturer intends to sell on that market. A brief overview with some of the more important standards is in Table 1.

International Standards	Guidance, national standards
ISO 9001:2000, Chapter 7.3. Design and development, Chapter 8.3 Control of nonconforming product	EC-Directive 93/42/EEC for Medical Devices and derived National Laws
ISO/IEC 90003:2004 Software engineering – Guidelines for application of ISO 9001:2000 to computer software	FDA 21 CFR 820 Quality System Regulations
ISO 13485:2003, Chapter 7.3. Design and development, Chapter 8.3 Control of nonconforming product	FDA Design Control Guidance for Medical Device Manufacturers, March 1997
ISO 14971:2007 Application of risk management to medical devices	FDA Guidance for the Content of Premarket Submissions for Software Contained in Medical Devices, May 2005
IEC 60601-1:2005 Medical electrical equipment - Part 1: General requirements for basic safety and essential performance	FDA General Principles of Software Validation, Final Guidance for Industry and FDA Staff , 2002
IEC 60601-1-6:2006 Medical electrical equipment - Part 1-6: General requirements for basic safety and essential performance – Collateral Standard: Usability	FDA Guidance – Do it By Design (An Introduction to Human Factors in Medical Devices) - 1997
IEC 62304 Medical Device Software – Software life cycle processes	FDA Guidance - Medical Device Use-Safety: Incorporating Human Factors Engineering into Risk Management - 18,July,2000
IEC 62366:2007 Medical Devices – Application of usability engineering to medical devices	JUS-97-623-01 Canadian Medical Devices Regulations (SOR/DORS)
IEC/PAS 62508 Guidance on human factors engineering for system life cycle applications	IEEE Std 1028-1997 IEEE Standard for Software Reviews
	ANSI/AAMI HE74:2001 Human factors design process for medical devices
	ANSI/AAMI HE75:2010 Human factors engineering – Design of medical devices

Table 1: Brief overview of international and national standards and guidance for the development of active medical devices

While the organizations involved with the creation of standards are striving for making the standards consistent through new editions, the evolving safety and technological changes constantly introduce new inconsistencies and new terms not covered by older standards.

Manufacturers apply the design standards through creating guidelines and the process standards through creating Standard Operating Procedures (SOPs). The organization in question implements these SOPs as seen in Figure 1. The information included in SOPs can be grouped into four broad categories:

- Definitions: Terms and abbreviations used in the SOP
- Process: The process execution described in the SOP
- Requirements: Minimal execution and outcome requirements for each step in the process
- Forms: Document templates for the SOP deliverables

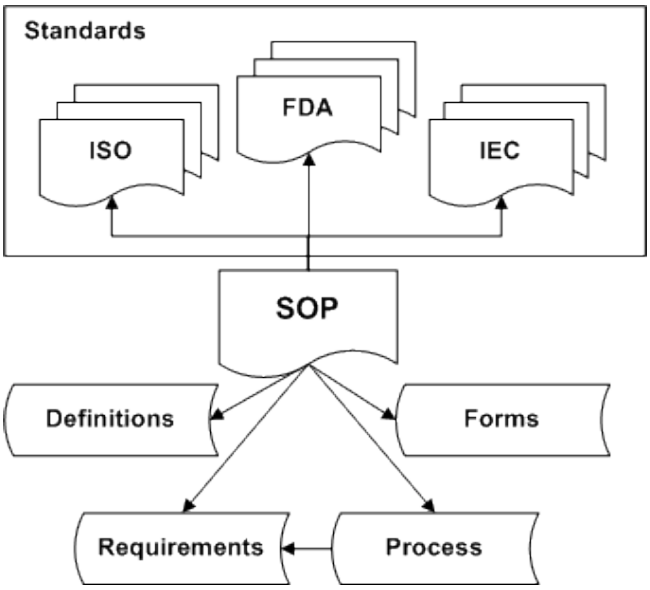


Figure 1. General structure of Paper based documents

The changes over time and the large number of standards results in several problems:

- Prob1: The application of standards is not coordinated in deliverables. The same concept has different names in different processes. The same artifact has different format (document template) in different processes.
- Prob2: The application of standards is not coordinated in tasks. The same process steps have to be executed with different input or output. The related tasks are executed independently.
- Prob3: Different organizational units take ownership of different process areas, if efforts are not well coordinated the standards' inconsistencies harden coordination.
- Prob4: With the inconsistencies of the different process standards, connections and sequencing the relevant process steps results in lots of questions.
- Prob5. Roles have overlapping content in different standards resulting in conflicting team member tasks and further organizational problems.
- Prob6: Team members have a hard time getting a clear view on tasks and outcomes from specific processes.

The problems described here are the most important hindrances of efficient development and software process improvement. If a manufacturer recognizes these problems, he may not have the proper resources to cope with them, as for any medical software development the compliance with the standards is the ultimate importance (as it enables entering the market with the product at all.)

3 Overview of the Eclipse Process Framework

The Eclipse Process Framework (EPF) is an open-source tool for creating process models. It is based on the Software Process Engineering Meta-model (SPEM). This gives the EPF a tool independent basis.

The SPEM is a meta-model used for describing processes and process parts. It uses UML for notation, thus readable to software practitioners. SPEM is most suitable to describe process models with strong collaboration.

The EPF is the open source version of an IBM tool, the Rational Method Composer (RMC) [12], both having the ability to open each others data files. One of the advantages of both products is the innate ability to visualize the text based processes flows. As a picture is worth a thousand words, for most users a visual approach is much easier to understand than a pure, text-based one. The EPF is able to not just present the described processes visually, but also in an interactive way, so the objects contained in the process displayed can be selected and opened, making the related objects available. This way, the complexities of a process can be explored without getting lost in the complete picture.

The EPF enables process authors to re-use all process elements. This is provided by separating the definition of the elements and their use. This makes it possible to use for example role artefacts and other elements consistently in all processes contained in the process model.

The EPF consists of three views:

- EPF Composer: used for authoring processes. The composer comes with sample process models (Scrum, XP, OpenUP). Users can customize these, or start their own model.
- Web view: used for exploring the model. This is intended for team members using the processes in their day-to-day work.
- Wiki view: used for collaborating on the process descriptions.

EPF is not just a modelling tool, the content produced by it can be used for other tasks, for example the IBM's Rational Team Concert tool (RTC) [13] uses process models as a mean to configure other project tools. This ability was another point for choosing this tool; in the future it may provide a connection point for other systems supporting the development process.

4 Moving from paper to modeling

Because of the high number of required standards, we realized that the complexity of the system makes process applicators unable to understand what is required of them from the ordinary paper based SOPs. Paper based process descriptions are difficult to understand throughout. The complexity of the process descriptions makes developers less eager to understand them, so process execution can deviate from its intended path.

In order to solve the problems described previously the EPF platform was chosen. In the first step, we only modelled the software life-cycle of the active medical devices and the related SOPs. This implementation immediately proved that it was easier to use models for process development. (See Figure 2. for implementation details.)

The two most important breakthroughs were:

- It reflected on how general our processes were. Most of them were simply concentrating on the medical regulations not including improvement thinking. With EPF we could find where to develop our processes, where the inconsistencies are and where they are coming from.
- We created an interactive interface which can be reached via the intranet by the developers. We gave them the possibility to search for the needed information on how to perform the tasks, what kind of documents must be made during the tasks, who is the responsible and so on.

In order to achieve this we had to adapt the EPF to our needs, we had to find out how to use the pre-defined objects to satisfy and describe medical development processes. Some of the EPF Composers features helped in the design management of the processes.

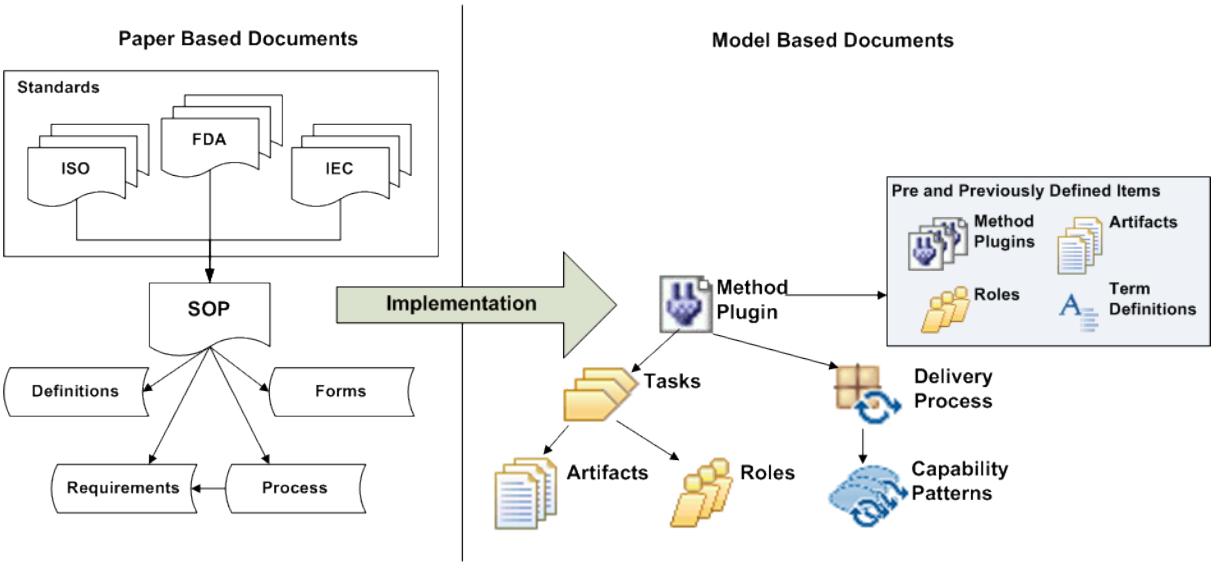


Figure 2. Implementation of the paper based documents into model based documents

The following list represents the most important objects, which are the heart of our model.

- Method plug-in: The method plug-ins are the main objects in the EPF. We decided that one method plug-in should act as one SOP. So one method plug-in contains one paper based SOP, making the transfer from the older system easier. There are also separate method plug-ins for those objects which can be reused in several processes such as Roles or Definitions. In this way it offers a good overview for the hierarchy of the model.
- Role: It represents a responsible person or a group of responsible persons who has to perform the specified task. Each role has its own permissions and functions during the whole life-cycle (for example Head of the R & D, Tester, Approval, Quality manager etc.).
- Task: Each task represents a process step. It describes how to perform the task. If the task is complex it can be divided into smaller steps. These objects have other useful attributes: the participant roles, the input documents which are needed to perform the task and the output documents which are the outcomes of the finished task.
- Artifact: It represents the documents or other pieces of outputs which are created during the whole life-cycle. In the medical software industry, documents are important because they are to fulfill regulations, e.g. they form the evidences for compliance during audits and inspections. (For example Requirement Specifications, Test Protocols and Reports, Verification Plans and Reports)
- Term definition: This object represents the terms which are used in the SOPs to help the developers in interpreting the processes and task descriptions. The collection of the Term definition objects forms the Central Glossary.
- Capability pattern and Delivery Process: These are both a type of process descriptions. The difference between them is that in our context the Delivery processes follow the whole medical software life-cycle from the beginning to the end of the life-cycle. In contrast, the capability patterns can be simply collections of tasks in an implementation order with different relationships or they can represent smaller processes during the life-cycle. It means the capability patterns are reusable and they can build up a delivery process.

There are others built in EPF objects too, which were useful during the development predefining collections of tasks, documents, definitions, enabling different views for the best understanding. One of the more useful features was the inclusion of document form templates next to the processes.

The “Pre and Previously Defined” box in Figure 2. represents the biggest advantage of developing processes in a modelling tool, the reusability of pre and previously defined items. These results faster and better process development. It enables the process developers to use the same term, definitions etc. to keep the model consistent. The users will not be confused by the ambiguities and it gives an easier way of understanding the processes.

The EPF has a built in publishing engine which can generate published HTML or Java websites from a given collection of method plug-ins. After implementing the processes into the EPF, we published the process model via the intranet (see Figure 3.). Since then team members can access it opening its URL in a browser. They can browse the process model based on different views (browsing per roles, documents, etc). All elements are clickable for further information. The central glossary and the search function provide additional options.

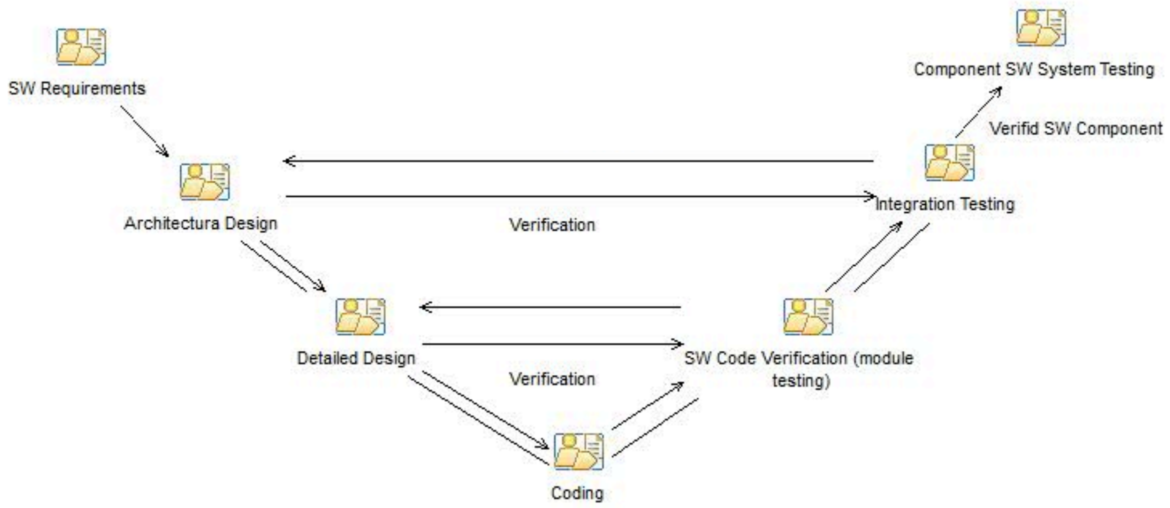


Figure 3. Software V-model from the implemented structure as it appears in the web-view

5 Analysis of results

With the solution described in the previous chapter the team members can act faster on each task as the steps and the deliverables are much clearer. Previously process steps had to be found out from the paper based documents, which required understanding the whole complexity of a certain process, not just the steps the team member is involved in. Deliverables were provided as general templates, not specific to a process steps (for example template for design requirements) which confused team members, often the wrong template, or out of date template was used. With the view available from the process model, the right deliverable template can be easily chosen.

Also crucial for the process success [15] is that the process instantiation is much easier as the deliverable templates (checklist forms etc.) are included in an integrated view with the process steps.

From the problems presented previously Prob1 was solved by creating common definitions, merging terms with the same content. Prob2 and Prob5 were partially solved, while same process steps were identified, the processes have to be changed to apply the findings. By EPF's collaboration abilities the Prob3 can be solved through the joint editing of the processes. Prob4 were resolved, as all processes were placed in a common model. By the visualization abilities of the tool, Prob6 is much easier to deal with; new project members can get a quick overview on the relevant processes to their tasks.

The model based approach also resulted in some new challenges. Some of the process related workflows have to be redesigned to accommodate the usage of the process (for example process release). As with all new tools, staff needs basic training to use, in this case the process applicators only need a minimal training, as the web-view is self explanatory. Process authors have to learn the new tool.

Currently the model is used parallel with the paper-based process and the signed paper-based process documents for inspections being official in the Quality Management systems, with the EPF used for training and modeling purposes. The consistency between the two systems is preserved through manual updates. Later on the whole document creation process will move to the EPF as changes will be controllable. While SOPs will still be needed as printed and signed documents, the whole workflow will be simplified.

6 Conclusions and future work

The case study presented will be useful specifically for practitioners in the medical field, and also more generally in any software project with extensive regulatory requirements. We have showed that process modeling with the EPF tool is viable and useful in identifying shortcomings, solving existing problems and providing a solid foundation for further improvement projects.

While the first phase of the implementation is complete, there are two directions for further work.

First, we are concerned about the problems generally associated with reading on computer screens [6], which is still a problem with the better and less tiring screen technologies. We are planning to have as a solution the ability to print documents from selected elements of the model, choosable from all elements, like role, task, deliverable or process. Printed documents are also desirable for audits.

Second, the EPF presents new opportunities in capturing relevant process experience, and re-use best practices which in turn is relevant for process improvement. A method for introduction of software project management practices based on Reusable Project Patterns is in [9], although that paper is based on a specialized tool, the EPF is able to capture patterns through the Capability Pattern element.

7 Literature

1. Becker-Kornstaedt, U., & Verlage, M. (n.d.). The V-Modell guide: experience with a Web-based approach for process support. In STEP '99. Proceedings Ninth International Workshop Software Technology and Engineering Practice (pp. 161-168). Presented at the STEP'99. Ninth International Workshop. Software Technology and Engineering Practice, Pittsburgh, PA, USA.
2. Chiam, Y. K., Staples, M., & Zhu, L. (n.d.). Representation of Quality Attribute Techniques Using SPEM and EPF Composer. Presented at the EuroSPI 2009, Delta. Eclipse Process Framework Web site. http://www.eclipse.org/epf/. Accessed on: 2011. 02. 22.
3. Eclipse Process Framework Project (EPF) homepage: http://www.eclipse.org/epf/ Accessed on: 2011.02.22.
4. Heidenberg, J., Holmström, P., & Porres, I. (2009). Tool Support for Collaborative Software Process Authoring in Large Organizations. Presented at the EuroSPI 2009, University of Alcala, Spain: Delta.
5. Henderson-Sellers, B. (2002). Process Metamodelling and Process Construction: Examples Using the OPEN Process Framework (OPF). Ann. Softw. Eng., 14, 341–362.
6. Hornbæk, K., &, Frøkjær, E. (2003). Reading patterns and usability in visualizations of electronic documents. ACM Trans. Comput.-Hum. Interact., 10, 119–149.
7. Humphrey, W. (1995). A discipline for software engineering. Reading Mass.: Addison-Wesley.
8. Kellner, M.I., Becker, U., Riddle, W., Tomal, J., & Verlage, M. (1998). Process guides: Effective guidance for process participants. In Proceedings of the Fifth International Conference on the Software Process (pp. 11-25). Chicago, IL, USA,: ISPA Press.
9. Martín, D., Garcia, J., Amescua, A., & Llorens, J. (2007). Reusable Project Patterns to enhance Software Process Improvement. Presented at the EuroSPI 2007, Delta.
10. Object Management Group: Software Process Engineering Metamodel (SPEM) Version 1.1 (2005)
11. Object Management Group: Software Process Engineering Metamodel (SPEM) Version 2.0 (2008)
12. Rational Method Composer (IBM) homepage. http://www-01.ibm.com/software/awdtools/rmc/ Accessed on: 2011. 02. 22.
13. Rational Team Concert (IBM) homepage. http://www-01.ibm.com/software/awdtools/rtc/ Accessed on: 2011. 02. 22.
14. Soto, M., & Münch, J. (n.d.). Maintaining a Large Process Model Aligned with a Process Standard: an Industrial Example. Presented at the EuroSPI 2007.
15. Wang, Y. (2000). Software engineering processes : principles and applications. Boca Raton Fla: CRC Press.

8 Author CVs

Ferenc Kazinczi

Ferenc Kazinczi is a BSc student at the Budapest University of Technologies and Economics in the faculty of Computer Engineering on the Department of Telecommunications and Media Informatics. He writes his thesis in the field of developing effective medical diagnostic and decision support systems for patients having different pathological vocal tract diseases. From the summer of 2010 he works for a medical corporation as an intern. He works with Péter Balázs Polgár on developing and visualise the strictly regulated corporation specific ISO and FDA medical processes and he works on automatism the documentation management system.

Péter Balázs Polgár

Péter Balázs Polgár has an MSc degree from computer science and is currently a PhD student at the Eötvös Loránd University, in the Basics and methodology of informatics program. His research interests include process improvement, usability and computer-human interactions. Besides the university research he is working at medical corporation as usability engineer.

Adapting the FMEA for Safety Critical Design Processes

O. Bachmann [1], B. Messner[2], R. Messnarz [3]

[1] SIBAC GmbH, Germany, info@sibac.de,
[2] Performing Facilities GmbH, Austria, b.messner@performing-facilities.com
[3] I.S.C.N. GesmbH, Austria & Ireland, rmess@iscn.com

Abstract

Functional safety standards (ISO 26262, IEC 61508) require a safety life cycle which demands additional design and engineering tasks to be managed. This paper addresses how the existing FMEAs have to be extended and refocused to address and overview signal paths throughout the system. The safety standards require to classify signals with a SIL (Safety Integrity Level) and the higher the SIL the more parallel controls and checks must assure that the signal is correctly calculated, used, and monitored. This paper illustrates this extension of the FMEA using the FMEA to investigate the effect of false sensor signals resulting out of failures in software monitoring functions and false failure reactions on system level resulting out of either false sensor signals or failures within the diagnostic software. AS a complementary activity to the FMEDA a FMEA method is introduced that allows an analysis during the development process that is performed prior to the "in-use" FMEDA.

Keywords

Functional safety, FMEA, signals, extended safety design process

Reference

O'Connor R. V., Pries-Heje J., Messnarz R. (eds.): Systems, Software and Services Process Improvement. 18th European Conference, EuroSPI 2011, Roskilde, Denmark, June 27-29, 2011. In: Communications in Computer and Information Science Series CCIS, Vol. 172, June 2011.

Extending Automotive SPICE to cover Functional Safety Requirements and a Safety Architecture

Richard Messnarz[1], Ivan Sokic, Stephan Habel[2], Frank König[3], Ovi Bachmann[4]

[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]SIBAC GmbH, Mittelbiberach, Germany

Abstract

This paper discusses (based on Automotive industry examples) how the functional and requirements traceability concepts in Automotive SPICE had to be extended to cover the criteria and content demanded by ISO 26262. In a second section the paper describes how these new concepts are considered in the integrated Automotive SPICE and Safety assessment approach which was proposed by the SOQRATES initiative (www.soqrates.de) where more than 20 leading German firms collaborate in cross company task forces. See previous papers about the integration approach and assessment method proposed in [4]

Keywords

Automotive SPICE, Functional Safety Requirements, Safety Architecture

Reference

O'Connor R. V., Pries-Heje J., Messnarz R. (eds.): Systems, Software and Services Process Improvement. 18th European Conference, EuroSPI 2011, Roskilde, Denmark, June 27-29, 2011. In: Communications in Computer and Information Science Series CCIS, Vol. 172, June 2011.

A Proposed Framework for Software Quality in the Healthcare and Medical Industry

Vispi Shroff, Louise Reid, Ita Richardson

Abstract

In this paper, we propose a theoretical framework for software quality within the healthcare sector. The aim of this framework is to improve the quality of software produced within the healthcare and medical device industries, while assisting the manufacturers in achieving compliance with existing regulation. To develop this framework, we undertook an evaluation of the healthcare and medical legislation. This was followed by an evaluation of existing software quality standards and models. The CMMI model was chosen as the model on which to base this framework. An initial mapping between CMMI and the US FDA Code of Federal Regulations was undertaken. We present some of the examples the mapping here.

Keywords

CMMI, 21CFR820, medical software development, healthcare software development, software quality

1 Introduction

One of the prominent issues in the software industry is the development of high-quality software [1]. When implementing software within the healthcare and medical industry the issues of software quality become more serious. As software in used increasingly within medical devices and in hospitals, any software quality issues can have significantly detrimental effects on a person's recovery, health and well-being. Some of the earliest and most prominent cases were the Therac-25 accidents between 1985 and 1987[2]. There is also a cost involved for organizations and companies when medical device recalls need to be made. Between 1999 and 2005, it was found that "one in every three medical devices, making use of software for their operation, has been recalled due to failure in the software itself" [3].

Compliance with legislation is another major hurdle which companies have to overcome at present. Many companies are utilizing software development methodologies such as waterfall and V-model and are reticent to embrace agile methodologies. Having achieved certification and compliance many companies find the potential risk of failing compliance with regulation due to extensive changes in practices and processes to be an inhibiting factor in adopting new processes.

In this paper, we have proposed a software quality framework which looks at software quality, software standards, healthcare regulation and implementation within a software development life cycle which can facilitate continuous software process improvement. This framework has been proposed after carrying out a literature review and a study of existing practices in a clinical environment. It is developed within the context of implementation within a European healthcare system research project, the aim of which is to develop a "rapid learning healthcare system' driven by advanced computational infrastructure that can improve both patient safety and the conduct and volume of clinical research in Europe" [4].

2 Research Methodology

To develop an understanding of the domain, case study research was undertaken on a number of systems in an Irish Hospital. This provided an insight into the quality issues that are unique to the clinical environment. This case study research was further strengthened by reviews into high profile hospital investigations such as the Lourdes Hospital Inquiry [5], the Shipman Inquiry [6], the Tallaght Hospital Review [7] and the investigation into the Therac 25 incidents. Findings from our research pointed to the importance of quality processes to ensure the success of existing systems. The Bristol enquiry found that the hospital was 'awash with data but that there was very little information'; this sentiment was echoed in the Lourdes Hospital Enquiry. We found systems that are in place but underutilised. We found cases where the staff that should be making use of these systems did not trust them and that this led to loop of quality degeneration due to underuse of existing information. All of the investigations highlighted the fact that things can and do go wrong in the clinical environment but that the risk can be offset by robust quality processes.

Through interviews with Irish medical device manufacturers and software vendors for the healthcare industry, we learnt that ensuring compliance with healthcare regulation can be a significant barrier to entry and a driver of effort and costs within the industry. A literature review was then carried out to study the current state of healthcare legislation. The focus of this review was primarily on US and European legislation followed by a study of Irish regulation. Some issues around relevance and ambiguity were identified within the legislation. As increasing the quality of the products manufactured appeared to be the purpose behind a lot of the regulation requirements, software quality standards and models such as the ISO 15504 and the various CMMI models were also studied.

The initial research has focussed around creating a mapping between selected process areas within the CMMI model and the Code of Federal Regulations Title 21 Part 820 regulatory requirements. The aim of creating this mapping is to provide software vendors to the healthcare industry a set of software processes which when undertaken, may not only improve the quality of the software manufactured, but also assist them in becoming compliant with regulation.

2.1 Healthcare Legislation

The US FDA Code of Federal Regulations was the core legislation studied. In particular, Subchapter H which focused on Medical Devices and Part 820 Quality System Regulation was studied in great depth [8]. This regulation is used as the foundation upon which the software quality models are mapped. European Council directives such as the Council Directive 90/385/EEC on Active Implantable Medical Devices (AIMDD), Council Directive 93/42/EEC on Medical Devices (MDD), Council Directive 98/79/EC on In Vitro Diagnostic Medical Devices (IVDMD) and subsequent amendments via Council Directives 98/79/EC, 2000/70/EC, 2001/104/EC, 2003/32/EC and 2007/47/EC were also studied.

Historically, the legislative requirements in the healthcare industry revolved around monitoring the processes and practices involved in the development of physical devices, not software. However, over the last couple of decades software has played a larger and increasingly significant role in the operation of medical devices. While the legislation was primarily developed to suit physical and hardware manufacturing processes, it has struggled to keep up with the changes in the manufacturing practices and the different problems and challenges which software development presents.

The EU Council Directive 2007/47/EC has come closest to including standalone software into legislation by adding “Stand alone software is considered to be an active medical device” and “It is necessary to clarify that software in its own right, when specifically intended by the manufacturer to be used for one or more of the medical purposes set out in the definition of a medical device, is a medical device. Software for general purposes when used in a healthcare setting is not a medical device” to the regulation. [9]

Some of the other key issues we found within the medical legislation included

- The lack of clear lines of distinction between hardware and software, and the impact of changes in software are not outlined.
- The focus of the healthcare industry is on patient privacy and security with very little focus on integrity and accuracy.
- Medical devices at present are categorized into three (US) or four (EU) different categories based on the potential impact they may have on patient health. These categories are shown in the table below, and there remains ambiguity in the definitions of these risk factors.

Table 1. Risk-based Medical Device Categorization in Legislation

EU Legislation		**US Legislation**	
Class	**Risk**	**Class**	**Risk**
Class I	Low	Class I	Low
Class IIa	Low-Medium	Class II	Medium
Class IIb	Medium-High	Class III	High
Class III	High		

Also, the proliferation of smartphones and the wide-spread usage of the application stores have seen a large increase in healthcare and medical applications becoming readily and openly available. At present, there is no means of implement and monitoring any regulatory or quality guidelines on these applications and this is becoming a major area of concern.

2.1.1 Code of Federal Regulations Title 21Part 820 (21 CFR 820)

As mentioned earlier, the 21CFR820 regulation document was adopted for the initial draft of the framework. To provide systems and solutions in the healthcare industry in the United States, products manufactured by companies need to be compliant with this regulation. The focus of this regulation is on the quality systems utilised in the manufacturing processes. There are certain artefacts which need to be created according to the 21CFR820 regulation. These artefacts are similar to and complement certain software engineering artefacts that are created as part of the software development lifecycle. Some examples of 21CFR820 artefacts include:

- Design history file (DHF) means a compilation of records which describes the design history of a finished device.
- Device history record (DHR) means a compilation of records containing the production history of a finished device.
- Device master record (DMR) means a compilation of records containing the procedures and specifications for a finished device.

From the descriptions, we can see that these artefacts are similar to the requirements specifications, design documents and other material which may be created during the software development life-cycle. So, if software development best practices are correctly followed, the creation of these artefacts and the documentation of the procedures and methods should take place, thereby assisting the software vendor in achieving compliancy with the regulation. This is where recommendations by software quality standards and models can be of assistance in developing good software development practices which assist a vendor in becoming compliant with regulation.

2.2 CMMI

For the purposes of this software quality framework, the Software Engineering Institute's Capability Maturity Model Integrated (CMMI) for Development [10] was utilized. The CMMI model provides flexibility as it allowed us to focus on very specific process areas. Also, this modularity would theoretically allow this framework to be implemented with any software development methodology. As the CMMI has its origins in Software Quality, and is freely and openly available it is widely adopted and validated by numerous companies and across various different industries. While it is heavily used in the software industry, there are also case studies available on the implementation of CMMI in the health care services industry [11]. We also found that the CMMI model and its ability to measure capability and maturity of certain processes mapped very easily to the Quality and Risk Management Standard (QRMS) adopted by Irish Health Service Executive[12]. Apart from this, CMMI's lays focus on and facilitates continuous SPI as compared to the IEC 62304 and other ISO standards. All these factors contributed towards the selection of CMMI as the primary model to base our framework on.

To reduce the scope of the investigation, the six process areas within the CMMI were focused on for mapping to the 21CFR820 legislation. These included the process areas of Requirements Development (RD), Requirements Management (REQM), Technical Solution (TS), Verification (VER) and Validation (VAL) and Configuration Management (CM). These process areas were chosen following discussions with the software developers and health informatics personnel involved in the healthcare system research project. These process areas were explored in depth and the specific practices and work products were mapped to the 21CFR820 legislation to see how they would satisfy regulatory requirements.

3 The CMMI – 21CFR820 Mapping

The fundamentals of the software quality framework lie in the mapping between the software models and the legislation. In the sections below, we have provided the mapping between the chosen process areas and how they satisfy different sections of the 21CFR820 legislation. Each process area within CMMI has a set of specific goals, each containing a set of specific practices. These specific practices are designed to assist in the accomplishment of the goal. The adoption of each of these specific practices creates certain outputs and deliverables. These outputs and deliverables can then be utilised to achieve compliancy with the existing regulations.

We will illustrate this mapping using the specific goal of Analysing and Validating Requirements within the Requirements Development Process Area.

3.1 Requirements Development Process Area

The purpose of Requirements Development is to elicit, develop and analyse customer, product and product component requirements. It involves studying and understanding the customers need, expectations and constraints which take into consideration all the stakeholder needs. Customer requirements are further developed into product requirements and product component requirements which are consistent with customer requirements.

CMMI provides three specific goals; develop customer requirements, develop product requirements and to analyse and validate requirements. The latter goal of validating and analysing the requirements is designed to assist in the accomplishment of the former two goals.

An example of this mapping, the CMMI-21CFR820 mapping is shown in the table below.

Table 2. Analyze and Validate Requirements

No.	Specific Practice	Deliverables	Legislation
SP 3.1	Establish operational concepts and scenarios	Operational concepts of the systems	DMR
SP 3.1	Establish operational concepts and scenarios	Product or product component installation, operational, maintenance and support concepts	DMR
SP 3.1	Establish operational concepts and scenarios	Disposal concepts	DMR
SP 3.1	Establish operational concepts and scenarios	Use Cases	DMR
SP 3.1	Establish operational concepts and scenarios	Timeline scenarios	DMR
SP 3.1	Establish operational concepts and scenarios	New requirements	DMR
SP 3.2	Establish a definition of required functionality	Functional Architecture	DMR
SP 3.2	Establish a definition of required functionality	Activity diagrams, use cases	DMR
SP 3.2	Establish a definition of required functionality	Object oriented analysis; services and methods defined	DMR
SP 3.3	Analyse Requirements	Key Requirements	DMR
SP 3.3	Analyse Requirements	Technical performance measures to measure progress	DMR
SP 3.3	Analyse Requirements	Requirements defects reports	DMR
SP 3.3	Analyse Requirements	Proposed Requirements Defects Reports	DMR
SP 3.4	Analyse Requirements to Achieve Balance	Document assessing risks related to requirements defined	21CFR820 Sec 820.30
SP 3.5	Validate Requirements	Records of requirements validation analysis and results	21CFR820 Sec 820.65
SP 3.5	Validate Requirements	Change reports for requirements which need changes based on validation results.	21CFR820 Sec 820.65

In this case, when following the specific practices for achieving the Requirements Development goal of analysing and validating requirements, the software manufacturer may create use cases, timeline scenarios, a functional architecture and much more. Many of these artefacts can be then be utilised in the creation of the Device Master Record (DMR) of the software for compliance with 21CFR820.

4 Future Research

There is great scope for future research to enhance the framework. The key challenges at present lie in addressing the applicability of such a framework within industry. We will be using the healthcare system research project as a test bed to evaluate the applicability of such a framework within software development teams. For evaluation purposes, it will be necessary to utilize a software development methodology to implement the framework.

Other areas of research would include extending the framework to include European legislation primarily, and other regional legislation may follow. European and US legislation tend to be leading the world as far as healthcare and medical devices are concerned, encompassing these bodies of legislation should lead to a framework which relatively encompasses universal legislation.

After extending the framework to include other legislation, one may look at other software quality models and standards such as the ISO standards to enhance aspects which may be missing from this model. For example, when compared to IEC 62304, CMMI was found “lacking specific safety related requirements that have been derived over time though regulatory monitoring of software intensive medical devices.”[13]

Finally, due to the intensive nature of the quality framework, as well as the amount of regulation involved, it would be worthwhile creating different models of the framework based on risk assessment of the medical device of software being produced. If a software or medical device is evaluated to be a low risk device, i.e. Class I device, then it may not be necessary to follow all the recommendations within the framework. In such situations, it might be worthwhile having a model of the framework which is designed for the different classes of risk. However, interviews with medical device manufacturers have shown us that the default class assumed by regulators is the highest risk class, and the onus lies on the manufacturers to demonstrate the reduced risk. Also, even though individual components may be fall under a lower risk category, the final evaluation is based on the overall risk classification of the medical device or software. So, even though software may play a miniscule and trivial role in the overall functioning of a medical device, if that medical device is classified as a Class III device, the software will need to pass Class III regulation guidelines.

Finally, one of the prominent issues which arose during the investigation into the case studies and subsequent interviews with hospital clinicians was that many hospital systems remained underused or stagnant due to the lack of trust in the systems and the general sentiment that the systems were not ‘fit-for-purpose‘. After the framework has been developed in further depth, the study of software development lifecycles may be carried out with a particular focus on agile processes. Within the European healthcare system research project, the Scrum methodology was proposed for implementation and will be observed for effectiveness against the scale and the nature of the project. We will also observe how the Scrum methodology facilitates this framework with the least challenges while allowing modification and improvement of the framework over time.

5 Conclusion

We have laid the foundations for a software quality framework for the healthcare and medical device industry. In a world where software is playing an increasingly pivotal role in the delivery of healthcare and patient well-being, the quality of the software being used is paramount. While there has been extensive work done on the patient privacy and data security side of the spectrum, there has been little focus on the data integrity and accuracy within systems which are becoming increasingly integrated. In this framework, our focus has not been on the outputs of the software development processes, but rather on the software development processes themselves. The proposed framework provides flexibility as a single component of it can be used by itself, or it can be implemented in its entirety. As the medical and healthcare industry is a heavily regulated industry, an equal amount of focus was laid on the existing legislation. It is the authors’ goal to provide a framework which is flexible, practical and easily implementable. We believe this framework allows the manufacturers to be compliant with regulation while developing high quality software.

6 Acknowledgments

This work is partially supported by Science Foundation Ireland grant 3/CE2/I303_1 to Lero. This work is also partially supported by TRANSFoRm which is funded by the European Commission – DG INSFO (FP7 247787).

7 Literature

[1] A. C. Gillies, *Software Quality: Theory and Management*. London, UK, UK: Chapman & Hall, Ltd., 1992.
[2] N. Leveson and C. Turner, “An Investigation Of The THERAC-25 Accidents,” *Computer*, vol. 26, no. 7, pp. 18 - 41, Jul. 1993.
[3] Z. Bliznakov, G. Mitalas, and N. Pallikarakis, “Analysis and Classification of Medical Device Recalls,” in *World Congress on Medical Physics and Biomedical Engineering 2006*, vol. 14, Springer Berlin Heidelberg, 2007, pp. 3782-3785.
[4] “TRANSFoRm.” [Online]. Available: http://www.transformproject.eu/Home.html. [Accessed: 25-Feb-2011].
[5] M. Harding Clark, *Lourdes Hospital Inquiry : An Inquiry into peripartum hysterectomy at Our Lady of Lourdes Hospital, Drogheda*. Our Lady of Lourdes Hospital, 2006.
[6] J. Smith, *The Shipman Inquiry, Fifth Report - Safeguarding Patients: Lessons from the Past - Proposals for the Future*. UK: , 2004.
[7] M. Hayes, *Report of the review of radiology reporting and the management of GP referral letters at Adelaide and Meath Hospital (Dublin), incorporating the National Children’s Hospital, (AMNCH) [Tallaght Hospital]*. Tallaght: Health Service Executive, 2010.
[8] US Food & Drug Administration, “Code of Federal Regulations Title 21 Part 820 Quality System Regulation.” Apr-2010.
[9] European Council, *Directive 2007/47/EC of the European Parliament and of the Council of 5 September 2007 amending Council Directive 90/385/EEC on the approximation of the laws of the Member States relating to active implantable medical devices, Council Directive 93/42/EEC concerning medical devices and Directive 98/8/EC concerning the placing of biocidal products on the market Text with EEA relevance*. 2007, pp. 21-55.
[10] SEI, “CMMI for Development : Version 1.2.” Aug-2006.
[11] E. Forrester, “CMMI for Services - Overview Presentation.” Software Engineering Institute, 2008.
[12] HSE, “Quality & Risk Management Standard.” Health Service Executive, Nov-2007.
[13] D. W. Walker, “CMMI and Medical Device Engineering.” Software Engineering Institute, 29-Sep-2009.

8 Author CVs

Vispi Shroff

Vispi Shroff is currently a part-time postgraduate researcher in Lero – The Irish Software Engineering Research Centre in the University of Limerick. Vispi has numerous years of experience in industry in the Irish SME sector and was responsible for project management and quality assurance of the software products manufactured. Having completed an MSc in Software Engineering from the University of Limerick in 2005, Vispi is also a Certified Scrum Master and has an interest in software process improvement, agile methodologies and software project management. Vispi is also pursuing a part-time MBA with the University of Liverpool.
Email: vispi.shroff@ul.ie

Louise Reid

Louise Reid is a Quality Auditor with the Health Service Executive, the national body providing healthcare in Ireland. A qualified nurse, Louise has worked in Irish hospitals over the past two decades and has first-hand experience of the software usage and quality within the Irish hospitals and healthcare system. Louise has completed a MSc in Health Informatics from Trinity College, Dublin and is currently doing her PhD with Lero – The Irish Software Engineering Research Centre in the University of Limerick.
Email: louise.reid@hse.ie

Ita Richardson

Ita Richardson is currently the project leader on numerous Irish and EU research projects within Lero – The Irish Software Engineering Research Centre in the University of Limerick. With almost two decades of lecturing and research experience, Ita has an interest in the areas of medical device software engineering, software quality, software processes, requirements engineering and global software development amongst others.
Email: ita.richardson@ul.ie

Exploring the requirements process for a complex, adaptive system in a high risk software development environment

Sheridan Jeary[1], Katarzyna Musial[2], Keith Phalp[1]
[1] Software Systems Research Centre, [2]SMART Technology Research Centre,
Bournemouth University, Fern Barrow, Poole, Dorset BH12 5BB.
sjeary, kmusial, kphalp @ bournemouth.ac.uk

Abstract

This work ties together research from a number of different areas to show how the development of a complex adaptive system for an industrial company has a number of difficulties given the current state of the art. The INFER system which is a Complex Adaptive System (CAS) has a number of attributes which mean that current requirements and indeed development processes are not able to cope with them adequately. A CAS can be recognised by the fact that it consists of a number of agents acting together dynamically resulting in emergent behaviour. This emergent behaviour cannot be predicted and thus, along with other phenomena such as reaction to and with the environment and deciding the different responsibilities of the components means that the requirements process for such a system is a current research area. A retrospective case study is underway to capture the rich data available from the experiences of building such a system.

Keywords

requirements process, development process, complex adaptive systems, adaptive software systems, risk

1 Introduction

Traditionally, a successful software system is one that meets or exceeds the needs or requirements of the client, users or other stakeholders. There has been considerable discussion over the last few years about the difficulty of obtaining requirements from the user when they have the IWKIWISI (I will know it when I see it) issue [1] and how the capturing of their requirements is a problem for requirements engineers.

Alongside requirements risk, the technical complexity of a project and the distributed nature of the personnel are also likely to be high risk elements of a project. The INFER project brings together all these elements and, therefore, the aim is to explore the requirements process in a technically complex project that is distributed between across three locations. Thus a retrospective, qualitative and exploratory case study is being run to examine both the successes and difficulties of creating a specification. The creation of the specification is for an exploratory complex, adaptive predictive system and has required academics to articulate their ideas to allow them to identify the requirements and to produce a specification prior to building the system. The personnel were distributed among three countries within Europe.

This work explores the background to the requirements process, complex systems adaptive systems, the INFER system, risk and the requirements process for adaptive systems and explains the importance of future work. Section 2 looks at the traditional requirements process. In Section 3 complex, adaptive systems are explained along with the phenomena they demonstrate. Section 4 explores the INFER project and Section 5 discusses how the INFER system is an adaptive system. Section 6 describes the attributes of the INFER project that are considered a risk factor in the project management community before Section7 explores the requirements process for complex adaptive systems and a system like INFER. The work then concludes and discusses current work in carrying out a retrospective case study before highlighting future work in Section 8.

2 The traditional requirements and development process

For the purpose of this study the measure of success of a software system is how that system broadly matches the purpose for which it is intended [2]. In addition, the way that the requirements process is considered is in the spirit of the ideas put forward by Jackson [3] who considers that the requirements engineer, the work that they do and the artefacts that they produce are all clearly situated in the problem domain. The further elements of the software engineering process are considered to be a part of the solution domain. The specification is at the interface of the two domains.

Requirements traditionally have been about completeness, consistency, traceability and testability, with an emphasis on early and accurate determination, often motivated by the high cost of fixing requirements errors later in the life cycle [4].

Hence, plan-based approaches, such as defined by Boehm and Turner [5] have proved to be both a popular and useful process particularly when it was possible to specify requirements in advance, and have also been extended to include processes for distributed (or even global) development [6]. Systems, such as embedded systems or safety critical systems for example, work well on this premise. The systems need to be specified sometimes formally and often with an accuracy that ensures that the resulting inspections and tests are passed with very few errors. The Capability Maturity Model was later developed with the idea of improvement of the software process making it both repeatable and measurable [7].

However, there are many systems that do not fit this ideal. Examples include development of web applications where the business is reacting to both business and technology pressures, where there is a lack of time to create full specifications and thus an iterative, incremental approach has been recommended to requirements with a prototyping approach [8]. In the post methodological era many developers are turning to a more informal way of development and using ‘more flexible off-the-cuff approaches’ [9]. There is some evidence that the backlash against the traditional processes and methods has focussed developers minds about they work and increased the search for helpful methods and techniques [9]. It is in this development environment that agile methods have become increasingly important and popular. This change in the development process has affected the gathering of the requirements by the creation of techniques such as user stories [10] and the production of a specification is no longer necessarily part of the development process. The agile group of methods encourage the stakeholder to become part of the development team and, as such, can act as a bridge between the problem and solution domains.

A further broad group of systems that are problematic when it comes to use of the waterfall process are complex predictive, adaptive systems that are described in Section 3.

3 Complex Adaptive Systems

The area of Complex Adaptive Systems (CAS) is a wide research field, which brings together specialists with backgrounds in economics, physics, social sciences, biology, computer science, and others. CAS's are systems that consist of populations of interacting agents that are able to adapt. The interactions of those agents lead to complex non-linear dynamics, the results of which are emergent system phenomena [11, 12]. The complexity means that the behaviour of the whole system cannot be described by a simple sum of behaviours of the entities that create this system [13, 14]. Examples of CASs include social insect and ant colonies, ecosystems, the brain and the immune system, cell biology and genomics, manufacturing business and various human social group-based endeavours in socio-cultural systems such as health and related educational support communities. However, not only natural systems can be complex and adaptive; software is an artificial system that needs to adapt.

The increasing complexity of software systems means that new approaches for developing and maintaining them are required. They need to adapt to the changing external environment and to new users' requirements; in addition to integrating large amounts of information and using the different technologies that they need to employ. Thus complexity and adaptivity of software systems has become an extensively researched area [15-17]. Self-adaptive software systems are the next generation of systems that are able to evaluate their own behaviour. If as a result of evaluation the system does not perform in the way it should, then the system adapts its behaviour [18]. Whilst completing its' analysis, a self-adaptive system will also need to take into account the external environment because the behaviour of the system is modified in response to changes in the environment in which it operates [19] and also to the changes internal to the system itself [15]. Modelling dimensions for self-adaptive systems can be grouped in four main groups: (i) self-adaptation of the system goals, (ii) causes of self-adaptation, (iii) mechanisms to achieve self-adaptation, and (iv) effects of self-adaptation on the system [15], [16]. Their aim is to support creation of models for representing important aspects of the self-adaptation and these factors should be considered when building this kind of system.

In the next Section, an example of a self-adaptive system that is being built within the Computational Intelligence Platform for Evolving and Robust Predictive Systems (INFER) project is presented.

4 The INFER Project

INFER is a major EU-funded project involving 25 academics from organisations in three different countries. This includes Evonik Industries from Germany, Research and Engineering Centre (REC) from Poland and Bournemouth University. INFER is a project funded by the European Commission within the Marie Curie Industry and Academia Partnerships & Pathways (IAPP) programme.

The project focuses on pervasively adaptive software systems for the development of a modular platform applicable in various commercial settings and industries. The main innovation of the project is a novel type of environment in which the "fittest" predictive model for whatever purpose will emerge – either autonomously or by user high-level goal-related assistance and feedback. Such a system is beneficial for businesses relying on accurate predictions of any type (e.g., customer behaviour, market conditions) and, at the same time, requiring an automated ability to react to changes in market or operational conditions.

4.1 System Goals

The INFER system aims to address current challenges in developing adaptive predictive models such as (i) how to build many models that can also be a combination of other models, (ii) what and how the adaptation mechanisms can be introduced, and (iii) which parts of the process and in what way they can be automated. The high level aim of the INFER project is to build a modular software platform for predictive modelling applicable in different industries. In order to achieve this, the predictive system that consists of a population of predictive models will be developed. The 'fittest' predictive model for whatever purpose will be chosen and used. Once applied, the predictive system will exploit any available feedback for its performance monitoring and adaptation. Based on the above description it can be summed up that the main goal of the software system is to deliver accurate predictions all the time.

4.2 High Level System Architecture

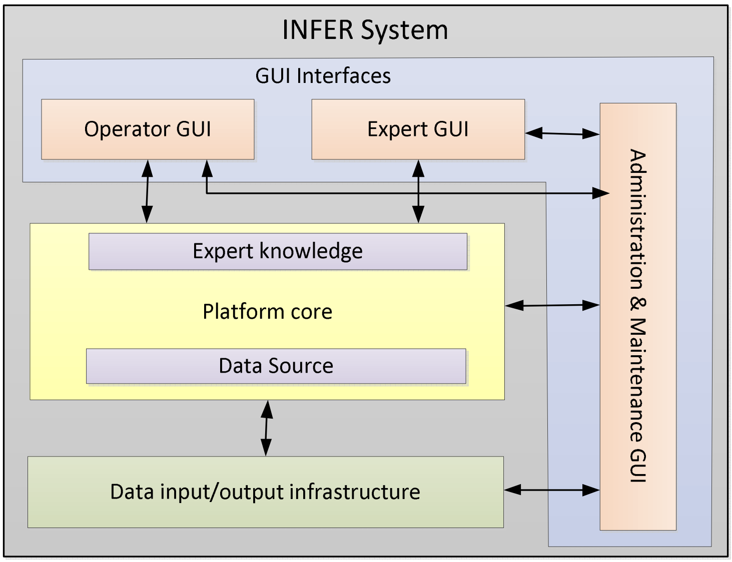


Figure 1: The main INFER system components.

The main components of the system are presented in Figure 1. The data input/output infrastructure enables the integration of the external data sources, i.e. import to the system of data in different formats and export of output data in the format that any defined external system can work with. The GUI interface is the second main component. Three GUI interfaces will be developed and the layout and functions available for users will differ depending on the access rights that a given user has. In general, these interfaces will provide an Administration and Maintenance interface that enables the user to change each part of the system; an Expert interface that has access to an expert knowledge module within the Platform core; and an Operator GUI from which a user can monitor the work of the system. This component will evolve to provide the user access to the newly developed functionalities of the system.

The Platform core will enable the processing of input data, build the predictive models that can compete or collaborate with each others, self-assess built models (also based on the meta-knowledge about previous predictive models), adapt the models, and in a result of all above deliver accurate predictions.

5 The INFER system as an Adaptive System

In this section the INFER system will be presented in the context of CASs. It has been done using four groups of modelling dimensions for self-adaptive systems [15], and a full description according to these dimensions is made available at [16]. The following now considers the main issues for requrememnts determination with INFER with respect to System Goals, Changes within INFER and the impact of adaption.

5.1 INFER system Goals

The INFER system has one main goal and three sub-goals as described in section 4.1. They are static and will not change during the system lifetime. However, the system is innovative and there are no solutions for some of the issues in modelling self-adaptive predictive systems. This means that one cannot say that the list of system goals is final and there will not be any additional goals. The continuous evolution of the requirements specification means that it is possible that new detailed aims may emerge during system analysis and design.

5.2 Changes in the INFER system

The next phase is to identify what kind of changes can occur in the system and in the environment in which it operates. The environment external to the system will change when the INFER system is applied in different application areas and this will mean that different input data will be sent to the system. Even within the same industry the input data can vary and change continuously. As a consequence, models the system builds need to be adapted or redeveloped. There can be also changes in the internal system environment when the system develops a better predictive model for a given problem. In such a case the system will reconfigure its behaviour to create and start using the more accurate model. Whilst these changes are all functional there will also be changes that are non-functional because for different industries different quality requirements can be important. In addition, the changes can occur very often in some application areas and extra adaptation will be required to cope with it e.g., if the system operates in a very dynamic environment. The types of dynamic changes in the system can be predicted in terms of their types but not when they will occur. In general, the more dynamic environment, the more unpredictable changes. In contrast, if there is a static external environment the system will continuously try to find a better solution for a given problem and data.

5.3 Mechanisms to Cope with Changes in INFER system

The mechanisms describing the reaction of the system to the changes can vary. thus for all dimensions presented in [15] there is no one concrete value in the case of INFER system. The adaptation can be related to both the parameters and the whole components of the system. In general it should be an autonomous process but it can also be assisted by the end user or be a mixture of both approaches. In system design the adaptation will be centralised however in the process of system development it may turn out that some decentralised mechanisms will also be needed. Also the scope of adaptation can vary from a local component to several components or even the whole system. Depending on the application area and the complexity of the models the time of adaptation can be short, medium or long. Regarding the timeliness, in the case when the predictive model does not deliver accurate enough predictions then the time for adaptation must be guaranteed. Finally, the adaptation is event-triggered. It means that it is done when the predictions are not accurate enough or there are free resources to improve working models. Thus the range of adaptation is very wide and different adaptation mechanisms can have different scope, duration and can be organised and administered in a variety of ways.

5.4 The effects of adaptation on the INFER system

The effects of adaptation will vary depending on the application area and also the type of adaptation that has been done. Because there are a lot of possible mechanisms that can be applied and various industries in which the system can work, one cannot predict all effects of the various adaptations.

The important thing to emphasize is that the overhead of adaptation must be insignificant and it means that the system must be able to deliver predictions all the time. There cannot be situation in which the system does not produce the output required or demanded by the user.

6 The elements of risk in a project

Risk as a factor in a software project is considered important particularly with respect to the management of the project and to allow for mitigation against project failure. Barki [20] defines software development risk as the project uncertainty multiplied by the magnitude of the potential loss due to project failure. Wallace and others identify 27 software project risks clustered into 6 dimensions which are risk factors and will affect project performance. One of the dimensions is requirements and they identify continually changing requirements, unidentified, unclear and incorrect requirements as risk [21]. The use of new technology is also identified as a risk along with a high level of technical complexity, immature technology and technology that has not been used in other projects. Han and Huang [22] use the same risk factors and show that the requirements phase in the 115 projects they examined is the most important factor across all projects and they show that of the ten most important software risk factors, four of the top five relate to requirements and number eight relates to the use of new technology.

Zowghi [6] argues that the requirements process when carried out as part of a global software development will need to be different in order to handle the issues of a distributed team. These include such things as cultural differences, communications, coordination and control and time difference. Requirements elicitation is a highly interactive process and can prove more difficult when the ‘problem owing and problem solving communities’ have distance between them [6, 23]. This is highlighted in Bhat, Gupta and Murthy’s case study [24] involving real projects when conflicting requirements approaches by teams at a distance caused problems. Issues that could be resolved by face-to face meetings were not resolved at a distance. Hanisch and Corbett [25] agree and highlight that the use of structured requirements processes by distributed teams may affect the social aspects of requirements elicitation and this may cause misinterpretation and miscommunication of the requirements. There are also cultural issues that affect virtual teams because a shared vision of the requirements and the project objectives are required. These issues are based on a variety of factors such as hidden meanings and interpretation of requirements caused by different cultural backgrounds of team members [25].

7 The requirements process for a complex, predictive and adaptive system

There is general agreement that there needs to be a new way of handling the requirements process for complex, predictive and adaptive systems [15, 23, 26]. Shaw argues that the system ‘health’ is difficult to predict when the designer has to deal with requirements from the user that are incomplete and the dynamics of the system with emergent factors are such, that future behaviour is not known [26]. The CAS may be termed a ‘large-scale’ system in terms of its complexity or the number of components and requirements for it will come from varying sources and be at a variety of different abstractions [23]. CAS’s that are particularly difficult are those systems such as the INFER system that are reliant on the physical environment. There are issues involving scope and boundaries, assigning responsibilities to peer software systems and components, hardware interfaces and human operators [23]. The integration of all these elements is the weakest area of current RE knowledge and practice [23].

The INFER project is thus about the design and building of a CAS by a distributed team and has a number of high risk elements to it. There is no clear requirements process for this type of system certainly none known to industry at this time, the technology is new and complex, it has requirements attributes that are unknown and behaviour which cannot be predicted. The development process the team could follow is also unknown for this type of system and finally, the development team is distributed.

8 Conclusions and Further work

Complex Adaptive Systems such as INFER are a new type of system that consists of a number of interacting agents whose emergent behaviour cannot be described as the sum of the entities that make up the system. Behaviour of the individual and the system as a whole will change as the system performs. Some of the attributes of a CAS are emergent behaviour, self organising, co-evolution, continual evolution, sub-optimal, diversity, connectivity and systems of systems. This means that a CAS is constantly changing, is often complex and large scale and reacts to both its environment and human users often in ways that cannot be predicted.

The traditional requirements and development processes are not able to cope with systems such as CAS's and we have shown that there is a very high element of risk attached to the building of a system like INFER. This has most of the previously discussed attributes and there will be rich data available from the requirements process which has been completed and a retrospective case study is underway.

There is considerable further work to be done in this area. The requirements process needs considerable exploration, in addition the way that development is done and the process it follows is also interesting. Finally the elements of risk that are attached to building this type of system will need further work as this type of system becomes more common place.

Acknowledgements. The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement no. 251617.

9 Literature

1. Boehm, B.: Requirements that Handle IKIWISI, COTS and Rapid Change. IEEE Computer 33 (2000)
2. Nuseibeh, B., Easterbrook, S.: Requirements Engineering: A roadmap. International Conference on Software Engineering, Limerick, Ireland (2000)
3. Jackson, M.: Problem Frames: Analyzing and structuring software development problems. Addison Wesley/ ACM Press, Harlow (2001)
4. Fagan, M.E.: Design and code inspections to reduce errors in program development. IBM Systems Journal 15 (1976)
5. Boehm, B., Turner, R.: Balancing Agility and Discipline. Addison-Wesley, Boston (2004)
6. Zowghi, D.: Does Global Software Development Need a Different Requirements Engineering Process? : International Workshop on Global Software Development - ICSE 2002, Orlando, Florida (2002)
7. Humphrey, W.: Characterizing the software process: a maturity framework. IEEE Software 5 (1988) 73-79
8. Boehm, B.: A Spiral Model of Software Development and Enhancement. Computer 21 (1988) 61-72
9. Avison, D.E., Fitzgerald, G.: Where Now for Development Methodologies? Communications of the ACM 46 (2003) 79-82
10. Cohn, M.: User Stories Applied: For Agile Software Development. Pearson Education, Boston (2004)
11. Brownlee, J.: Complex Adaptive Systems. Complex Intelligent Systems Laboratory, Swinburne University of Technology, , Melbourne, Australia. (2007)
12. Holland, J.H.: Hidden Order: How Adaptation Builds Complexity. Basic Books, New York (1996)
13. Levin, S.A.: Ecosystems and the Biosphere as Complex Adaptive Systems. Ecosystems, Biomedical and Life Sciences and Earth and Environmental Science 1 (1998) 431-436
14. Arthur, W.B.: Introduction: Process and Emergence in the Economy in the Economy. In: Arthur, W.B., Durlauf, S., Lane, D.A. (eds.): The Economy as an Evolving Complex System II. Addison-Wesley Pub. Co, Reading, Mass, USA (1997) 1-4
15. Cheng, B.H.C., Lemos, R., de, Holger, G., Inverardi, P., Magee, J.: Software Engineering for Self-Adaptive Systems: A Research Roadmap. In: Cheng, B.H.C., Lemos, R., de, Holger, G., Inverardi, P., Magee, J. (eds.): Software Engineering for Self-Adaptive Systems LNCS 5525 Springer-Verlag, Heidelberg (2009) pp. 1-26
16. Dimensions for INFER; Available at: http://sjeary.co.uk/eurospi.html
17. Salehie, M., Tahvildari, L.: Self-adaptation Software: Landscape and Research Challenges. ACM Transactions on Autonomous and Adaptive Systems 4 (2009)
18. Laddaga, R., Robertson, P., Shrobe, H.: Introduction to Self-adaptive Software: Applications. In: Laddaga, R., Robertson, P., Shrobe, H. (eds.): Self-adaptive Software: Applications. Springer-Verlag, Heidelberg (2003)
19. Oreizy, P., Gorlick, M.M., Taylor, R.N., Heimbigner, D., Johnson, G., Medvidovic, N., Quilici, A., Rosenblum, D.S., Wolf, A.L.: An architecture-based approach to self-adaptive software. IEEE Intelligent Systems 14 (1999) 54-62
20. Barki, H., Rivard, S., Talbot, J.: Toward an Assessment of Software Development Risk. Journal of Management Information Systems 10 (1993) 203-225
21. Wallace, L., Keil, M., Rai, A.: How Software Project Risk Affects Project Performance: An Investigation of the Dimensions of Risk and an Exploratory Model. Decision Sciences 35 (2004) 289-321
22. Han, W.-M., Huang, S.-J.: An empirical analysis of risk components and performance on software projects. The Journal of Systems and Software 80 (2007) 42-50
23. Cheng, B.H.C., Atlee, J.M.: Research Directions in Requirements Engineering. Future of Software Engineering (FOSE 07). IEEE, Minneapolis, MN, USA. (2007)
24. Bhat, J.M., Gupta, M., Murthy, S.N.: Overcoming Requirements Engineering Challenges: Lessons from Offshore Outsourcing. IEEE Software 23 (2006) 38-44
25. Hanisch, J., Corbitt, B.J.: Requirements Engineering during Global Software Development: Some Impediments to the Requirements Engineering Process: A Case Study. 12th European Conf. Information Systems Turku, Finland (2004)
26. Shaw, M.: "Self-Healing": Softening Precision to Avoid Brittleness. First Workshop on Self-Healing Systems (WOSS02), Charleston, South Carolina, USA (2002)

10 Author CVs

Sheridan Jeary

Sheridan Jeary is a Senior Lecturer in Software Entrepreneurship at Bournemouth University and has research interests that span Software Development Methods and processes, Web Systems and business processes. Her PhD is in the area of requirements and web development methods.

She had several years of management and systems experience across a variety of domains before moving into academia. She was the BU Project Manager for the EU Commission funded VIDE project on Model driven development and has just completed a 3 month secondment with REC in Wroclaw, Poland as part of the INFER project.

Katarzyna Musial

Katarzyna Musial was born in 1982 in Poland. She received her M.Sc. degree in Computer Science from the Wroclaw University of Technology, Poland in 2006. In the same year she received her second MSc degree in Software Engineering from the Blekinge Institute of Technology, Sweden. She obtained Ph.D. in 2009 from the Institute of Informatics, Wroclaw University of Technology, Poland. She is a Lecturer in Informatics at the Bournemouth University, UK.
Katarzyna Musial focused her Ph.D. thesis on the calculation of individual's social position in the virtual social network. She is interested especially in complex social networks and dynamics and evolution of complex networked systems. Katarzyna Musial is a BU Transfer of Knowledge Coordinator in INFER project She is one of the founder members of the 'Social Network Group' at WUT (SNG@WUT) established in October 2006.

Keith Phalp

Keith Phalp originally read for a first degree in Mathematics, which he then taught for a few years, before completing a Masters in Software Engineering, followed by Ph.D. in Software Process Modelling. He then spent three years as a post-doctoral research fellow at the University of Southampton, again in the area of process modelling. In 1997, Dr. Phalp took up a lectureship at Bournemouth University, and has been there ever since in a variety of roles. He is currently an Associate Dean within the School of Design, Engineering & Computing and Head of Computing and Informatics.

Terminology Management for process improvement in eBusiness and eProcurement

Dr. Gabriele Sauberer[1], Reinhard Pohn[2],
Barbara Inge Karsch[3]
[1] TermNet, Austria
[2] Paradine, Austria
[3] BIK Terminology & TermNet, Austria

Abstract

Today business processes handle a lot of redundant data throughout the life cycle of a product or service. Standardization in the definition of product properties eliminates major sources of failure in data transfer and interpretation of data transferred. This paper shows examples of successful implementation of terminological principles and tools, such as the eCl@ss Service Portal (www.eclass-serviceportal.com) and the new ISO Concept Database (http://www.iso.org/iso/iso_concept_database_cdb) which supplies terminology, graphical symbols codes and other concepts.

Keywords

terminology, terminology management, eBusiness, eProcurement, product properties, eCl@ss, classification, product description, graphical symbols, codes, concept, concept database, data repository, property dictionary

1 Current Situation

“eBusiness” still is one of today´s buzzwords. It is often used when speaking about electronic procurement processes (eProcurement). However, eProcurement is only one small part of the process, visible to all customers and consumers using web-based services. In our highly globalized markets, eBusiness may be defined with a very general definition, such as “any business process that relies on an automated information system”.

During the life cycle of a product or a service its data will be covered by many different IT-systems. Starting with product specifications where the first product related properties are defined and continuing until the product is disposed of or recycled, a large amount of product related data is generated and has to be maintained throughout the entire product life cycle. The duration for maintaining these data depends on the kind of product and can be several decades, e.g. in the case of equipment for production plants or aircrafts.

Information to be exchanged includes language dependent as well as language-independent information. Examples include terms and definitions, product properties, units of measurement, classification systems, codes of all types, graphical symbols, etc.

A lot of this information is available redundantly in different systems. Information about a product (e.g. a washing machine) has to be available in design and product data management systems (CAD, PDM, PLM, …) as well as production planning systems (ERP) and systems for maintenance (e.g. spare-part catalogue). Today there are also many suppliers which deliver parts, components or complete systems to the manufacturer of a product. Therefore product information also has to be shared with suppliers. These suppliers may work with sub-suppliers. Information about the finished product also has to be sent to its distributors and sales partners. Maintenance and service partners also have to be provided with product information and spare-part catalogues. In short, product information has to be distributed throughout the extended supply chain.

2 Targeting Problems

Analyzing eBusiness processes throughout the extended supply chain shows that several dozens of different IT-systems are involved, where data about a single product or service is processed, stored and maintained. Costs for data transfer between these systems sum up to enormous amounts. When focusing to single business processes we find out, that in most of the cases data transfer between systems is specially configured for the respective business process (e.g. CAD to ERP, ERP to Marketplace, …) to connect two systems. Business processes before and after these processes are not being considered. This application to application data exchange results in many data islands and maintenance of redundant data costs the companies a fortune.

The global market adds the challenge of delivering data in different languages. The companies deliver data sheets or product catalogues to their customers and business partners. If translations are not perfect or specifications in localization are not met, misinterpretation and failures are the result. Simple errors in data transfer may lead to catastrophes. An example which demonstrates this very obviously was the NASA mission of the Mars Climate Orbiter in December 1998. The Mars Climate Orbiter was destroyed when a navigation error caused the spacecraft to miss its intended 140–150 km altitude above Mars during orbit insertion, instead entering the Martian atmosphere at about 57 km. NASA reported the reason for the loss of the spacecraft as a problem in transfer of information between the Mars Climate Orbiter spacecraft team in Colorado and the mission navigation team in California. One team used English units (e.g., inches, feet and pounds) while the other used metric units for a key spacecraft operation.

Usually, misinterpretation of data or errors in data transfer do not have these dramatic impacts, but in all cases additional costs are the result. If errors are not recognized before the product is out on the market, an increase of project costs, extended time-to-market or expensive recalls of products can be the results. In any cases, it decreases the competitiveness of the company.

Currently, an integrated eSupply Chain exists in theory only. In the majority of companies, product data is created redundantly in different systems. Also maintenance of data is performed redundantly in several systems. Often product catalogues and specifications delivered from suppliers cannot be imported into systems and are added manually. This work causes errors when creating data and can lead to expensive changes and redesigns of products. Another problem is that different companies have different naming for the same product attribute. Data transfer leads to different kinds of problems.

- Data is not correct — Data is wrong and there are no automatic procedures to verify correctness.
- Data is not clear — Misinterpretation of information because information is not understandable for receiver.
- Different naming with same meaning — Sender and receiver use different naming for the same information.

- Same naming with different meaning — Sender and receiver use the same naming for different information.

It is already difficult to avoid the problems listed above when data is transferred in the same language. In multilingual environments, it is an even bigger challenge.

3 Common understanding

"What is the height of a sheet of paper in A4 format?" I asked. The woman sitting next to me at the workshop answered immediately: "It's 297 mm". I pointed to a sheet in front of her: "Are you sure?" She seemed surprised by the second question, but then she replied: "How do you mean – height?"

This was the entry question at the beginning of a workshop at which we discussed product properties and product classification. There are plenty more examples where you think you are clear about your meaning, but someone else may not understand.

4 Need for Terminology Management

As markets become global and goods are sold worldwide, suppliers source products and services from all around the world In this environment, business success requires a clear and common understanding between the buyer and seller of product or service specifications. To achieve this, we have to focus on master data more than we did in the past. The introduction of terminology management and master data dictionaries with content which can be electronically referenced uniquely is an important first step.

4.1 Terminology for common understanding

Terminology, as a type of language resource, is a set of terms representing concepts of a specific knowledge domain. Increasingly, terminologies also include non-linguistic concept representations, such as graphical symbols or formulae. The implementation and use of a corporate-wide, multilingual terminology is essential for common understanding. Standards developed in ISO/TC 37, Terminology and other language and content resources, have contributed significantly to the harmonization of these methods.

4.2 Why Terminology Management

While misunderstandings in every-day language can lead to amusing outcomes at best, misapprehensions in languages for special purposes (LSPs) can result in production delays, in recalls and potentially in disasters. As the product moves through the different product life cycle stages, content is handed off to different people. R&D might have included writers and editors, for example, but even these content publishers are slightly more removed from the conceptual phase. They will hand off finished or semi-finished content to translators who are even further removed from the conceptual origin of a new device.

All of these "senders of communication" face the challenge of writing or speaking about product concepts clearly and consistently and in the language of the receiver, so that the message can be understood. Creating clear and consistent strings on a device with a user interface is a particularly daunting task, as space is limited. UX design expert, Everett McKay, not only stresses the importance of user-centered design, he says "[f]ocusing on effective communication is the single best user-centered design technique".

Without a terminology strategy, problems are inevitable. Misnomers and inconsistencies are two of the most common problems. Incorrectly labeled functions, inadequately abbreviated buttons or simply poorly motivated terms are examples of misnamed concepts. Users may be slowed down in the use of the product, may need to research before using the product, may not use the product or may use it incorrectly. Specifically the latter presents a high risk for manufacturers of life science products.

According to a recent survey conducted by the International Network for Terminology, TermNet, inconsistent terminology is the number one factor impeding writers, editors, translators, etc. in the US information and communication technology (ICT) sector. 83% of the survey respondents said that inconsistencies between documents or within documents are consuming their time. Three participants even said that they have experienced product recall due to terminology issues.

4.3 Accuracy – a must for e-business

Processes, products and services must be described very accurately in electronic business. It is not adequate to describe a product by defining a list of properties, where the property is defined only by its name. In addition to the preferred name of a property, multilingual definitions in several languages for the property must also be provided to make meaning clear. Further information, such as data formats, units of measurement, or icons, might also be needed to avoid misinterpretation. And reference classification data such as eCl@ss is needed to assign the product to the correct structure.

Specifications for the definition of product properties and classes are defined in ISO 13584-42:1998, Industrial information systems and integration – Parts library – Part 42: Description methodology: Methodology for structuring part families (PLIB). This data model is standardized, open and may also be used for the definition of company-specific properties and classes.

4.4 A streamlined approach

In the past, product data management was application-centric. This means that information was handled only when needed in the respective application system (e.g. product design). Product data management was conducted redundantly for each application. Data transfers between applications and to suppliers and customers were complex and expensive.

Today, we are moving toward a corporate-centric approach. Concepts of products and services are described only once in master data dictionaries from a corporate perspective. Also product classifications such as eCl@ss are implemented at this point. The master data dictionary includes such information as multilingual definitions remarks, references, icons, graphical symbols, etc. serving as a lexicon within the organization. This ensures that the product description is consistent throughout the entire product life cycle.

For application systems, such as computer-aided design (CAD), computer-aided engineering (CAE), enterprise resource planning (ERP) and customer relationship management (CRM), only information required for the respective system is transferred. Valuation of the product is done directly in the application systems. Data transfer between two application systems is less complex, as both systems build on the same properties. This streamlines business processes and reduces complexity.

Corporate master data systems must be built on reliable and stable terminology, product properties and classification systems. Industry has recognized that international as well as national standardization bodies are the right vehicle to deliver reliable master data, as they have mature and proven processes as well as extensive experience in development and maintenance of master data. Country codes and language codes are examples which are already provided from ISO.

5 Terminology Management and Data Repositories

Terminology management is the systematic research, documentation and reuse of concepts and their terms. When new concepts are first developed and described, a terminologist helps the subject matter experts, e.g. a team of cardiologists, coin designations and documents information about it in a terminology management system (TMS). Anyone, from the research staff to the writer to the translator, can then access the centralized database and use correct terminology consistently. The following is a brief overview over skills, tools and processes necessary for successful research, documentation and reuse of standardized terminology.

Skills. Terminologists fall into two categories: The subject matter expert who has the technical knowledge of the subject area as well as enough awareness of terminology methods and principles; and the trained terminologist who has the research skills to quickly get into a subject matter. There are two-year master's degree programs as well as EU-wide recognized certification programs (ECQA Certified Terminology Manager). The most important terminology (vs. subject matter) skills are described in the skills card of the ECQA Certified Terminology Manager_basic (www.ecqa.org) under "Certified Professions".

Companies, such as Scania, Volkswagen or Siemens, organizations, such as the European Patent Office or the European Commission, and national language centers in Finland, Sweden or Ireland have a terminologist or small teams of terminologists who support the content supply chain with expertise, for example, in term formation and terminology data management. To enable data interoperability, they are also familiar with terminology management standards set forth, for instance, in ISO documents created by Technical Committee 37.

Case Study (Example).

> *"**Giesecke & Devrient Banks on a Secure Future with Terminology Management from SDL***
>
> *As a leading technology supplier to banks and governments, Giesecke & Devrient requires high-quality user documentation delivered into multiple languages for its worldwide customers.*
>
> *As the number of deliverables increased, the Translation Department began to notice significant inconsistencies in product and part names. Investigating these issues took valuable project time, causing increased costs, missed deadlines, and impacts to revenue and customer satisfaction. Giesecke & Devrient realized they needed to rapidly find and implement a terminology management system to eliminate these inconsistencies.*
>
> ***Results***

- *Corporate terminology dictionary created, enabling consistency*
- *Online access to corporate terminology database for multiple users*
- *Development of unified style and language, consistency across the organization*
- *Cleansed source language"*
- *Translation costs reduced by 15-20%*
- *30% total cost savings realized*
- *30-40% reuse of translated text enabled for new projects*

From: Giesecke & Devrient Case Study, http://www.lspzone.com/en/landing/premium-downloads/giesecke-and-devrient-case-study.asp

Tools.
1. Terminology management systems (TMS)

TMS are software applications that run on top of a relational database. One of the main criteria for a TMS is that it is concept-centered. That means that a concept, indicated by its unique ID and a definition, is the main entry. Attached to it are the designators that stand for the concept. To stick with the medical theme, terms could be "sphygmomanometer" or "blood pressure meter," depending on the needs of the users and the agreed upon (=standardized) usage. It could also have the terms "Blutdruckmessgerät" for German or "tensiomètre" and "sphygmomano mètre" for French attached to it in a multilingual setting.

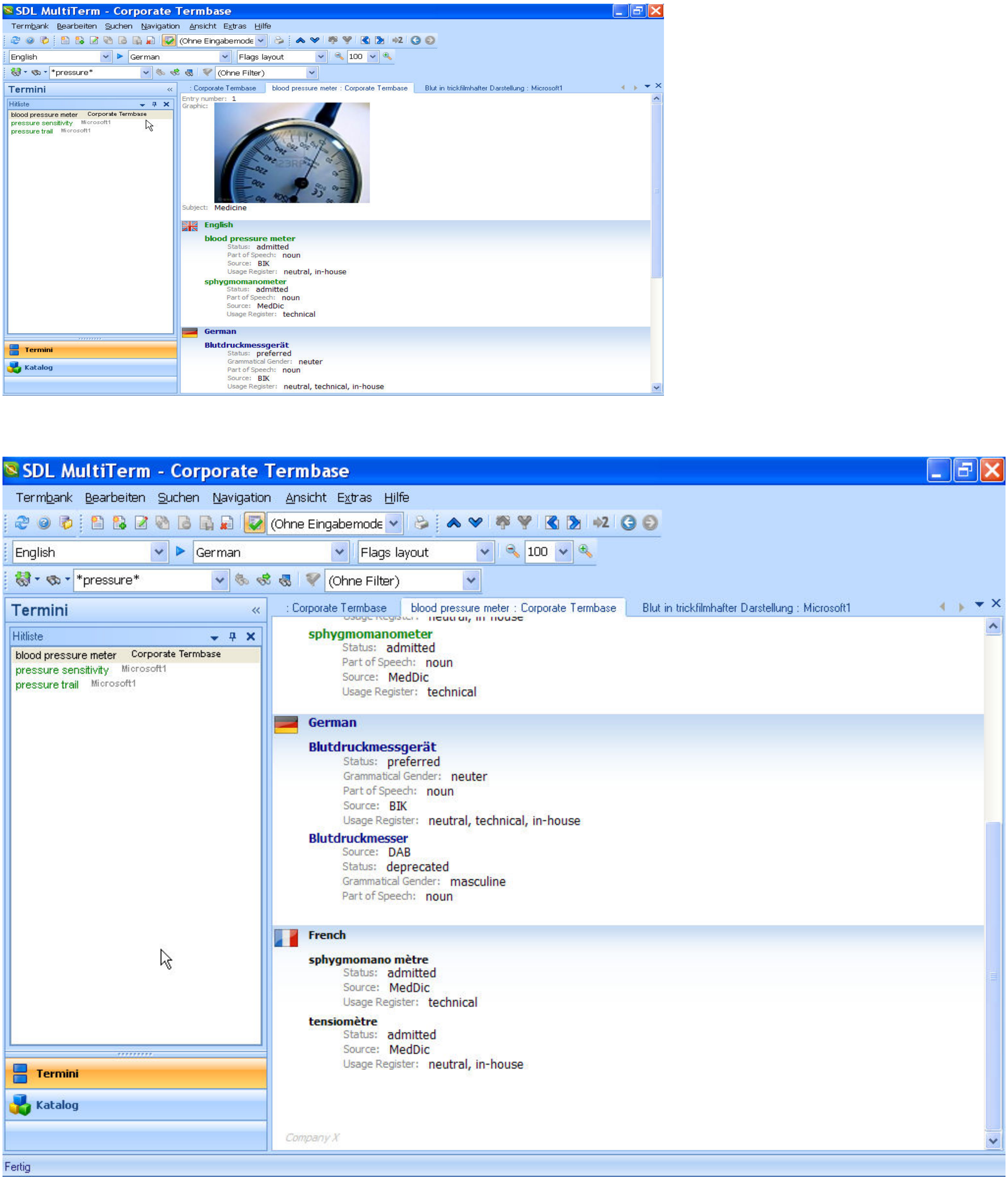


Figure 1: Terminological entry in SDL Trados MultiTerm

The software interface is generally designed for the terminology expert who prepares data which is then used in content management systems, translation memory tools, machine translation systems, etc. While there are several commercial tools available, particularly software companies, such as SAP, IBM, Microsoft or Oracle, created proprietary solutions.

2. Software Localization Tools

Many organizations developing software for global markets and face the challenge of releasing new software simultaneously into multiple markets. In order to guarantee that the mostly tight deadlines for new software releases are met, one of the key processes that have to be considered is localization.

There are various commercial tools available that help to speed up the translation of user interfaces (e.g. Alchemy CATALYST, SDL Passolo, Lingobit Localizer, MultiLizer, Visual Localize, etc. And with integrated Terminology Management Systems, it is possible to safeguard the consistent use of the correct terminology.

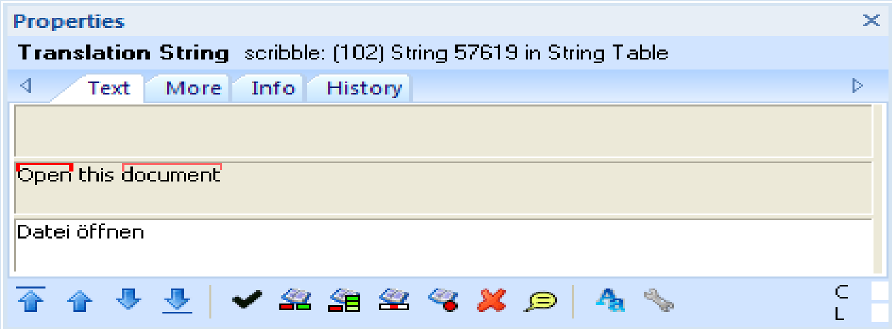


Figure 2: 3 a

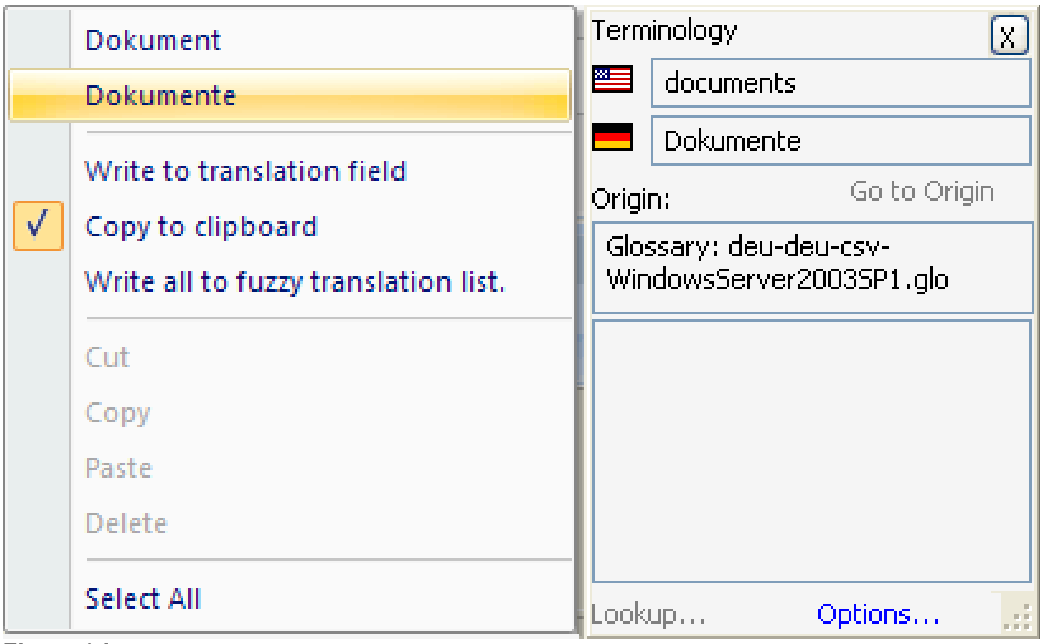


Figure 3 b

Figure 3 a and b: Example of a Software Localization Tool with integrated Terminology Management SystemTMS

Processes. There are various ways to break down the terminology management process. The expert team responsible for the best practice guide of the German terminology association, *Deutscher Terminologie Tag e.V.*, lists the following four phases Terminology production, preparation and distribution, use, and maintenance.

- Terminology production is characterized by the identification of existing and the creation of new concepts. Terms that represent these concepts are collected, and new terms or names are created.
- During the terminology preparation phase, concepts and terms are being standardized and documented in a TMS along with metadata, such as definition, part of speech, context, or subject.
- As soon as the terminological entries in the database are stable, they can be released, and writers, editors, and translators among others access and use them.
  Some terminological entries might need to be changed because errors are detected or new information is available. This happens during a maintenance phase of an entry.

The main outcome of the process, established by applying the skills and in the terminology management system, is entries, such as the example in Figure 12

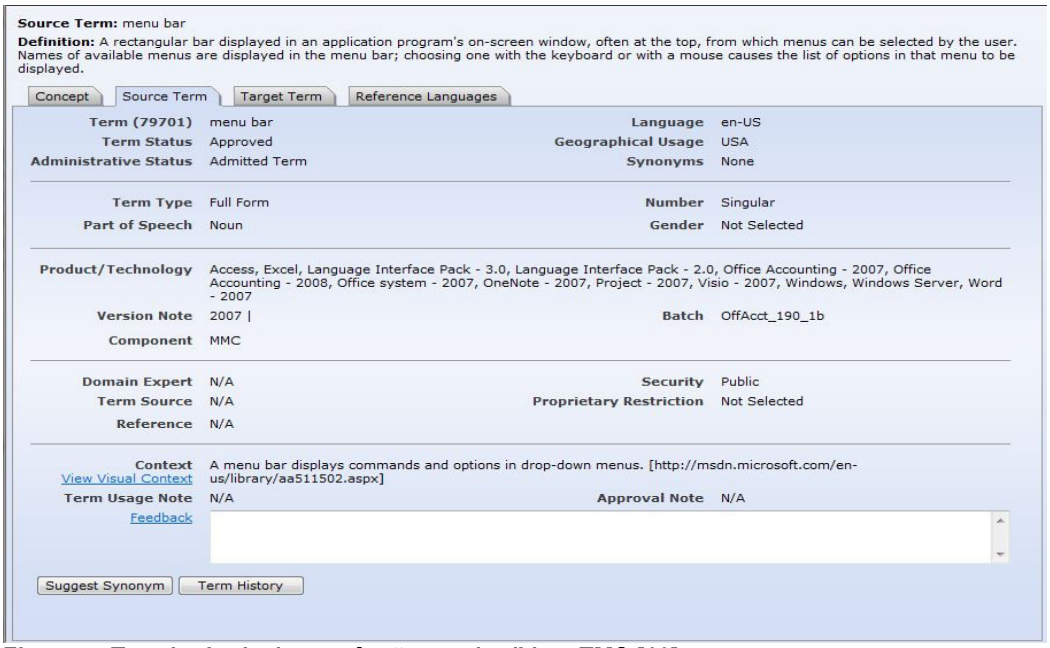


Figure 4: Terminological entry for "menu bar" in a TMS [12].

5.1 eCl@ss ServicePortal

eCl@ss is a leading international cross-industry classification system with more than 30.000 classes in the latest release 6.2. The classification is currently available in up to 16 languages. eCl@ss also adopted the ISO 13584-42/IEC 61360-2 data model. In previous releases of eCl@ss, the classification could be downloaded as CSV files. Maintenance and development of the classification system were made offline.

Going global and increasing coverage of industry segments demanded better services for both customers using the classification system as well as companies actively participating in development and maintenance of the classification. This resulted in the introduction of the eCl@ss ServicePortal.

The eCl@ss ServicePortal is an online-Database for development and maintenance of the eCl@ss classification. Registration to the eCl@ss ServicePortal is free of charge and can be made online at http://www.eclass-serviceportal.com.

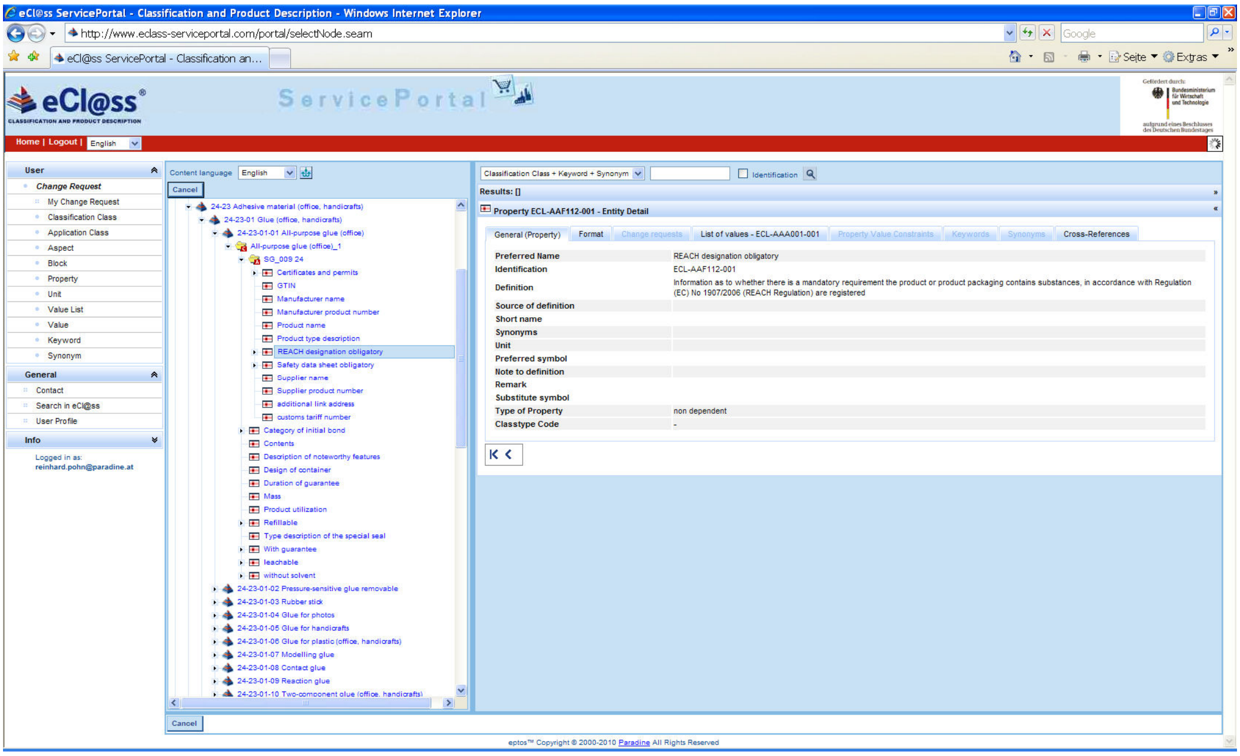


Figure 5: eCl@ss ServicePortal

eCl@ss started cooperation with DIN in 2004 to be able to use standardized properties from DIN within their lists of properties to describe products.

One of the major achievements of the eCl@ss ServicePortal is the possibility to “open doors” to a worldwide community to participate in development, maintenance and localization of the eCl@ss classification system. It is now possible to create change requests (e.g. add a new class, change the definition of a property, correct translations, etc.) on a 7/24 basis via the internet. A user can track the progress of all of her change requests online. Also, the system sends automatic notification mails if a change request reaches certain stages. This leads to faster processing of change requests and increase of content quality.

5.2 ISO Concept Database

Following a survey undertaken by the ISO Central Secretariat (ISO/CS) on the use of databases within ISO technical committees, the ISO Technical Management Board (TMB) decided, in 2005, to set up the ISO/TMB ad hoc group, AHG “Standards as databases”. The aim of this group was to address issues arising from the many database initiatives that were emerging in various technical committees.

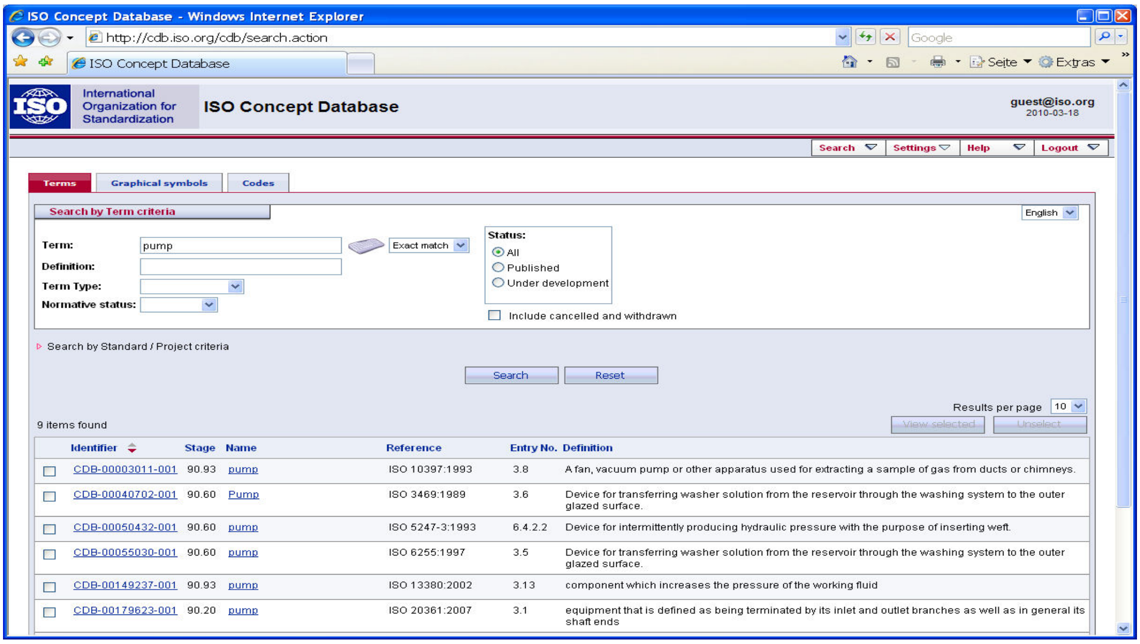


Figure 6: ISO Concept Database (ISO/CDB) – Search form

As a result of the work done by this ad hoc group, it was decided to create a common ISO Concept Database (ISO/CDB) which shall be available for all technical committees to develop, maintain and provide “concepts”. Concepts may be represented by terms, graphical symbols, product properties, codes, units of measurement and more. ISO/CS and Paradine have signed a long-term partnership agreement for development and operation of the ISO/CDB. Work began in October 2007. The first version of the ISO/CDB was presented to selected ISO technical committees in May 2008.

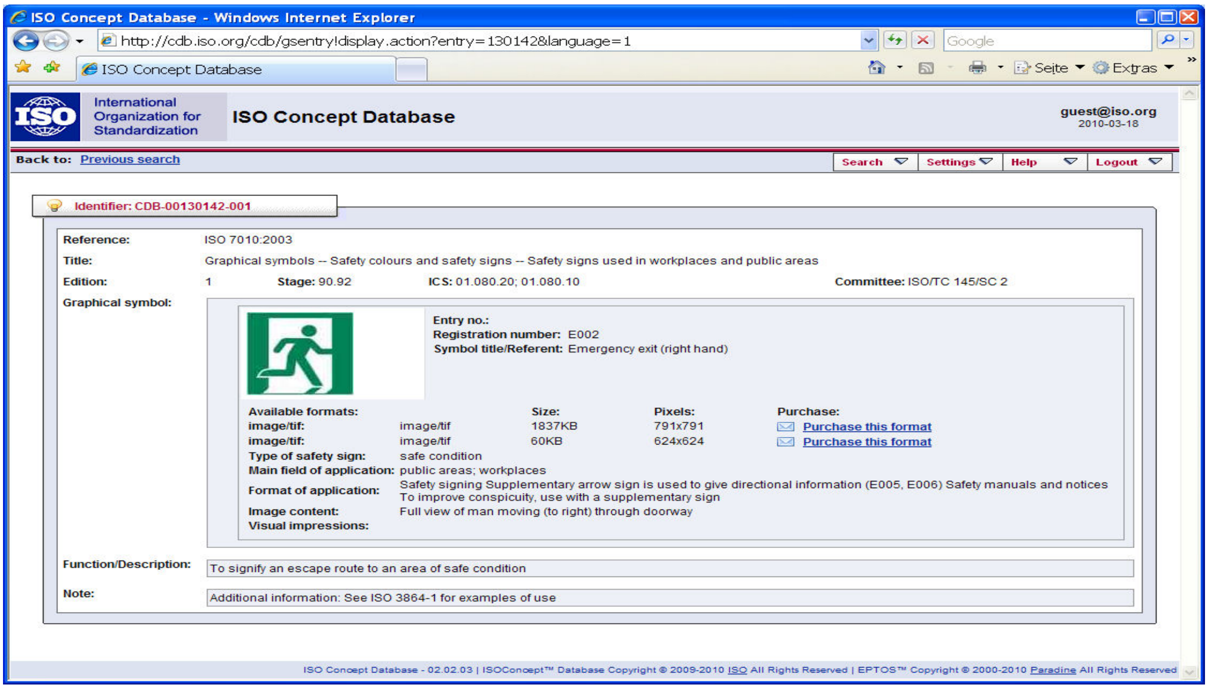


Figure 7: ISO Concept Database (ISO/CDB) – Entry for a Graphical Symbol

It is a major step for ISO to publish content in the form of an electronic database. The first version of the ISO/CDB was released to the public in October 2009. Currently the ISO/CDB contains terminology from all. 18.000 published ISO Standards which addsup to about 160.000 terms. Furthermore about 4000 graphical symbols and various codes (e.g. language codes, country codes, currency codes,…)

are available. Read-only access is free of charge. Further content, such as product properties and units of measure, are planned for later this year

6 Benefits for the Industry

Terminology management, standardized terminology, classification and properties offer substantial benefits for industry and trading partners. Product liability issues can be extremely costly, making clearly-defined product specifications and common terminology necessary to avoid misinterpretation.

The role played by standardization bodies and industry associations in providing this information is growing rapidly, because a key factor for industry is long-term security of investment in product description. Reliable and stable processes for creation and change management are highly appreciated.

Return on investmentof these projects for companies is usually achieved in less than 18 months, and the benefits are ongoing throughout the entire product life cycle. These benefits include:

- Consolidation in parts management, including spare parts
- Cost reductions in procurement
- Increased electronic data exchange to suppliers and customers
- Corporate-wide search for materials is enabled
- Reduced complexity of business processes and data exchange
- Increased turnover of stock and decreased fixed capital

7 Conclusion

Concept-based terminology management systems allow database users to select the correct term for the target audience of their message. That may be “sphygmomanometer” when experts are communicating about the gauge only; it maybe “blood pressure meter” when the content of a blood pressure kit is described in the user documentation; or it might be “the cuff” when a nurse refers to the measuring device to a patient. Precision is user-dependent.

When precision is not present, errors will occur, if not in the source language, very likely during the translation process. A translator relies on the source material to be precise and error-free. If the text reads “cuff” when the entire blood pressure kit is meant and it is not clear from the context, the translation will be faulty. It is of utmost importance that terms and names be tracked in a database from the concept phase on and used by everyone correctly and consistently.

The review of the ongoing developments and their potential for the future has shown that master data dictionaries may perform a foundational role in the emergence, structuring and operation of a global knowledge infrastructure. With its coverage of a multitude of technical and other subject fields, its wide range of stakeholders and open processes, the standardization bodies and classification system suppliers will make a key contribution to the evolution of such a widely shared knowledge infrastructure.

The use of reliable and uniquely electronically identifiable master data reduces the complexity of business processes. It helps companies to reduce costs and to increase product quality as well as reduce time-to-market.

Literature

Arntz, R., Picht, H., Mayer, F.: Einführung in die Terminologiearbeit. 4 ed. Studien zu Sprache und Technik, vol. 2, Georg Olms Verlag, Hildesheim, Zürich, New York (2002)

Engel, G., Picht, H.: Der Terminologe - Beruf oder Funktion? In: Fachsprachen - Languages for Special Purposes, Hoffmann, L., Kalverkämper, H., Wiegand, H.E. (eds.), pp. 2237-2244. Walter de Gruyter, Berlin, New York (1999)

Karsch, B.I: Anforderungen an Terminologen bei Microsoft. In: Best Practices in der Terminologiearbeit. Deutscher Terminologie Tag e.V., Heidelberg (2010)

Karsch, B.I.: Terminology Survey Results, TermNet: Vienna (2011)

McKay, E.: UI is Communication. Presentation, http://www.uxdesignedge.com/wp-content/uploads/2010/08/UI-is-communication1.pdf

Pohn, R., Weissinger, R. (2008) "Zooming in on the ISO Concept Database" ISO Focus, Nov. 2008: 18-20.

Reineke, D. (ed.): Terminologiearbeit - Best Practices. Deutscher Terminologie-Tag e.V., Cologne (2010)

SDL Passolo 2009, ReferenceManual.pdf

Weissinger, Reinhard (2007) "Standards as databases and the development of knowledge". ISO Focus, Nov. 2007: 36-38.

Weissinger, Reinhard (2010) "Platform for Progress: The ISO Concept Database" ISO Focus, Feb. 2010: 30-31.

Wilkes W, Ondracek O., Oancea M., Seiceanu M. (2007) "Web Services to resolve concept identifiers - Supporting effective product data exchange" Proceedings I-ESA 07 Conference, Madeira, March 2007

Author CVs

Gabriele Sauberer, TermNet

Gabriele Sauberer is the Executive Secretary (Director /CEO) ofof the International Network for Terminology (TermNet), headquartered in Vienna, Austria. She is an expert in various standardization committees, such as terminology, translation, knowledge and information management, diversity management, etc. She holds a PhD in linguistics and is an experienced manager in the language and content industries. Gabriele conceived and managed many projects at national, European and international level, such as the current project "ExcellenceTerm: International Center of Excellence for Terminology: Research, Technologies, Services.

ExcellenceTerm
RESEARCH • TECHNOLOGIES • SERVICES

Reinhard Pohn, Paradine GmbH

Reinhard Pohn is co-founder and managing director of Paradine GmbH, Vienna, Austria. He has been a member of several national and international standardization committees since 1987. Mr. Pohn received an education in mechanical engineering and marketing, and has 23 years of experience in the software business and mechanical engineering. He has extensive experience in the implementation of master data dictionaries in standardization bodies, industry associations and corporations.

Barbara Inge Karsch, BIK Terminology & TermNet

Barbara Inge Karsch holds a BA-equivalent and an MA in translation and interpretation. For 14 years, she worked as in-house terminologist for English and German at J.D. Edwards and Microsoft. During that time she designed and implemented two large-scale terminology management systems, and trained or mentored hundreds of translators, international project managers, content publishers and terminologists. In May 2010, she started her own terminology consulting and training business. She is also a US delegate to ISO TC 37 and writes a blog on terminology issues (www.bikterminology.com).

Configuration management of the organizational knowledge in software engineering

Antonio de Amescua, Alberto Heredia, Javier García

Abstract

The aim of this work is to obtain a system to support the configuration management of the organizational knowledge. This system will help software engineers to harvest, store, find and reuse process assets in projects improving in consequence the ease of learning and use of software processes. A feedback mechanism is proposed to gather new knowledge and to store it into a single repository enriching the pre-existing knowledge. The feedback mechanism allows software engineers to contribute new formal and informal knowledge to the organization during the project execution. Furthermore, the system allows presenting the same enriched organizational knowledge in different views depending on the needs of its users.

Keywords

Knowledge management, software engineering, configuration management, explicit and tacit knowledge elicitation, wiki.

1 Introduction

This paper proposes a practical application of Knowledge Management in Software Engineering to help the management and improvement of software processes. Knowledge is one of the main values that organizations own because, if managed properly, it can improve their competitiveness [1]. The current ICTs allow the Knowledge Management Systems to be capable of managing knowledge in an effective and efficient way [2]. Knowledge repositories are usually one of the base structures in every knowledge management system. A knowledge repository, also known as corporate memory or organizational memory, stores knowledge artefacts in such manner they can be retrieved and reused [3].

One of these corporate memories that can manage all the organizational knowledge is the Process Assets Library (PAL). A PAL is an organized, well-indexed, searchable repository of process assets that is easily accessible by anyone who needs process guidance information, examples, data, templates or other process support materials [4]. In every organization, a PAL provides the key infrastructure element that is required to support the process improvement effort [5]. A PAL not only contains knowledge about how to perform the processes but also should contain the lessons learned of those projects that have been successful in order to improve the processes.

However, this existing knowledge is difficult to find, and when found its reuse is difficult to achieve in practice [6]. The main reason is that most of the knowledge related to software processes is personal, context-specific and hard to formalise and to communicate among people [7]. This study focuses on how to establish mechanisms to enrich the organizational knowledge about the processes by adding that personal knowledge to the pre-existing one in an assisted and semi-automatic way, making then easier to spread the enriched knowledge through different platforms.

The main objective of the study is to obtain a system to support the configuration management of the organizational knowledge, allowing its users to improve their learning about the software processes. Such system must provide an easy mechanism to gather and store existing knowledge into a repository and encourage users to contribute and reuse knowledge within the organizational repository. To validate the solution, a wiki is used as a tool to implement a PAL and a feedback mechanism is proposed to gather the new knowledge into a repository that feeds the wiki.

The validation of this work was conducted in a training course to assess:

- The effectiveness of the feedback mechanism, in terms of determining the degree of use of the feedback mechanisms and the relevance of the new knowledge gathered from the practitioners.
- The enrichment of the knowledge in the repository, discussing the quality of the new knowledge elicited and the subjective evaluation of the usefulness of this new knowledge.

This paper is organized as follows. Section 2 contains the formulation of the problem. In Section 3 we propose a solution to the problem. Section 4 describes the validation carried out. Section 5 presents an analysis and discussion of the results. Finally, Section 6 presents our conclusions of the study.

2 Background

Nowadays, Knowledge Management (KM) has a large impact in Software Engineering, such as decreased time and cost for development, increased quality, and better decision-making abilities [8]. KM is also considered to be an essential part of the Software Process Improvement (SPI) efforts [9]. The experience at the Federal Aviation Administration (FAA) showed that KM complements SPI, resulting in positive interactions that benefit the organization and its process improvement programs [10]. Thus, KM turns into a critical success factor in a competitive and always changing software industry [11].

The knowledge that organizations have to mange can be classified in explicit and tacit knowledge [7]. The first one is formal and systematic, and can be expressed without ambiguities in words, data, numbers, and language. On the other hand, tacit knowledge is personal, context-specific, hard to formalise and to communicate among people, intuitive and derived from experience. Both types of knowledge must be managed properly to make knowledge available for software teams in order to improve the software development processes in an organization [8].

KM simplifies the process of sharing, distributing, creating, capturing, and understanding a company's knowledge to enhance value [8]. To do so, knowledge repositories store artefacts in such manner they can be retrieved and reused [3] in order to improve organizational performance [2]. Among these corporate memories that can manage the organizational knowledge is the Process Assets Library (PAL). A PAL provides the key infrastructure element required to support the process improvement effort [5]. A PAL not only contains knowledge about how to perform the processes, but also should contain the lessons learned of those projects that have been successful in order to improve the processes.

The Web 2.0 can be used as a technological platform for a collaborative management of all this software processes' knowledge. In the literature, wikis are used to support activities such as requirements elicitation [12], coding [13], and sharing experiences [14]. Although there are many studies on the use of wikis in Software Engineering, few of them use wikis as PALs. In [15] a PAL is proposed using a wiki in which stored process assets used CMMI as improvement process model. Amescua et al. [16] developed a PAL using a wiki as a lightweight repository of knowledge about agile software development, providing a more specific process structure for the PAL. Amescua also suggests that further research has to be done about the definition of mechanisms to feed back the tacit knowledge and the definition of mechanisms for the configuration management of the organizational knowledge.

In relation to the definition of mechanisms to feed back the tacit knowledge, Falbo et al. [17] described an initiative to capture informal knowledge by storing some information about lessons learned. Each lesson learned has to pass a knowledge filter to decide whether it should be available in the repository or not. The main drawbacks of this solution are that only the project manager can input a lesson learned in the informal knowledge base and the proposed solution has not been validated.

He et al. [18] developed a framework to capture process experiences, knowledge artefacts and personal skills from users, to store them in a repository and to make them reusable. Although this solution offers a Web portal, it doesn't support a clear collaborative framework and the authors didn't provide results about the use of the solution.

On the other hand, Process Framework (EPF) Composer [19] and IRIS Process Author (IPA) [20] are tools that allow publishing in a wiki the organizational knowledge stored in a repository, so users can provide feedback and suggest improvements by modifying the published version. However, these changes are not automatically propagated back into the repository.

Regarding the configuration management of the organizational knowledge, EPF and IPA rely on external revision control systems for version tracing [21]. Therefore, there isn't any solution in the literature which proposes built-in configuration management features in the system.

3 Proposed Solution

This work focuses on how to establish mechanisms to enrich and manage the organizational knowledge about software processes. The problem is that the existing knowledge is difficult to find, and when found its reuse is difficult to achieve in practice [6]. The main reason is that most of the knowledge related to software processes is tacit and very little organization's tacit knowledge can be captured and made explicit [7].

To reach this objective, a system is proposed to gather the existing knowledge, adding it to the pre-existing one in an assisted and semi-automatic way and then spreading the enriched knowledge through different platforms. This system supports the configuration management of the organizational knowledge, allowing its users to improve their learning about the software processes. The conceptual vision of the proposed solution is shown in Figure 1.

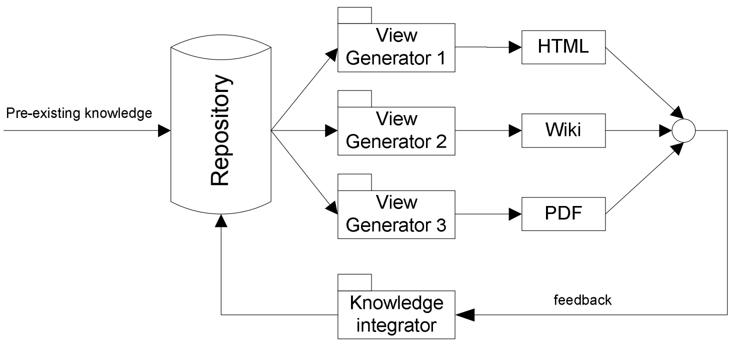


Figure 1: Conceptual vision of the proposed solution.

a) Knowledge Repository

The system has a repository acting as a version control platform that stores all the pre-existing knowledge about the processes in the organization. The pre-existing knowledge is obtained from the organizational software development processes, effective practices taken from software engineering literature and the experience of relevant organization's experts.

The system has a user-interface to provide the required information according to the internal structure of the organizational knowledge repository (see Figure 2).

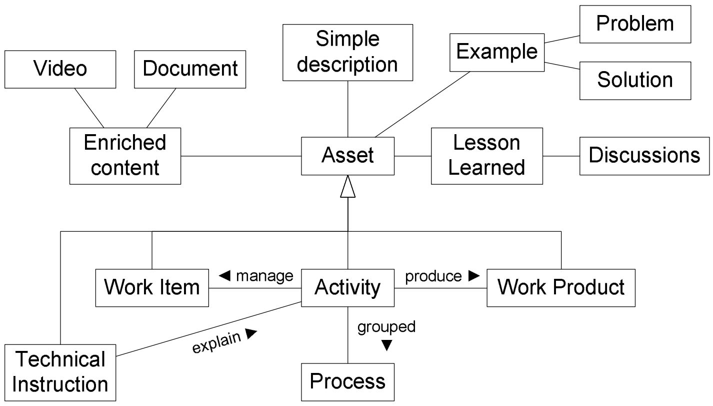


Figure 2: Repository structure.

An Activity is some work performed for a single purpose. Activities describe how things have to be done and are grouped in Processes. A Work Item is a register used to track the assignment and state of work. Activities manage Work Items to perform tasks resulting in deliverables or artefacts called Work Products. The Technical Instructions present information about how to carry out the tasks related to the Activities.

The repository stores the organizational knowledge in XML files and other multimedia files such as images, documents or videos. As Figure 2 shows, the assets can be stored in the repository using a simple description, as enriched content, or in the form of an example or a lesson learned.

The discussions correspond to the informal knowledge gathered during the project execution. Every discussion has a heading and its corresponding text. These discussions can be obtained in different ways, depending on the organization's actual platform (e-mail discussions, wiki comments or posts in a forum). When a discussion passes the feedback, it is stored in the repository as a lesson learned.

The examples correspond to the formal knowledge gathered during the project execution. Every example has a text describing the context of the example, a description of the problem, the artefacts associated with the problem, a description of the solution and the artefacts associated with the solution. These examples are the artefacts obtained when carrying out the processes in the organization.

b) Generation of Knowledge Views

The organizational knowledge can be presented in different formats depending on the generation mechanism used to create the view. The different generation mechanisms allow deploying an Electronic Process Guidance (EPG) in static HTML files, a PAL using a wiki and a PDF file downloadable from the wiki that compiles its contents.

These generation mechanisms are mainly composed of three tools. The first one is called "Integrator" and it combines all the XML files gotten from the repository in a single XML file, the Guidance file. The second one is the "Converter" and it transforms the Guidance file in another XML file ready to be imported in the wiki. The third tool is called "Generator" and it builds an EPG in static HTML files from the Guidance file obtained from the Integrator.

The system also has a feedback mechanism that allows gathering the existing knowledge and updating the organizational knowledge in the repository.

c) Knowledge Enrichment

Once an initial version of any knowledge view (i.e., a wiki) is created from the pre-existing knowledge stored in the repository, its users can make their contributions during the projects. As previously mentioned, these contributions will be mainly in form of examples, discussions or changes in the pre-existing knowledge.

The feedback mechanism allows gathering these contributions and enriching the organizational knowledge in the repository. In the particular case of the wiki, this mechanism is composed of three tools (Figure 3) and each one takes charge of extracting a different type of new knowledge from an

XML file dumped from the wiki.

When the feedback process begins, the pages of the wiki are dumped into an XML file. First, the “Changes Editor” detects the changes made in the wiki and updates the corresponding pre-existing knowledge in the repository. Then, the “Examples Parser” extracts the examples from the Dump file and stores them in the repository. Later, the “Lessons Learned Editor” extracts the discussions from the Dump file and shows them in a visual editor. Each discussion may be discarded, edited or directly accepted. In the last two cases the discussion becomes lesson learned and is stored in the repository.

When the repository is updated, the Integrator gets the bunch of XML files stored in the repository and combines them in a single Guidance file. This file can be then used to generate a new version of the static representation of the organizational knowledge or be converted into the special XML format to be imported into the wiki. Finally, this XML file is imported in the wiki offering its users a new version of the organizational knowledge.

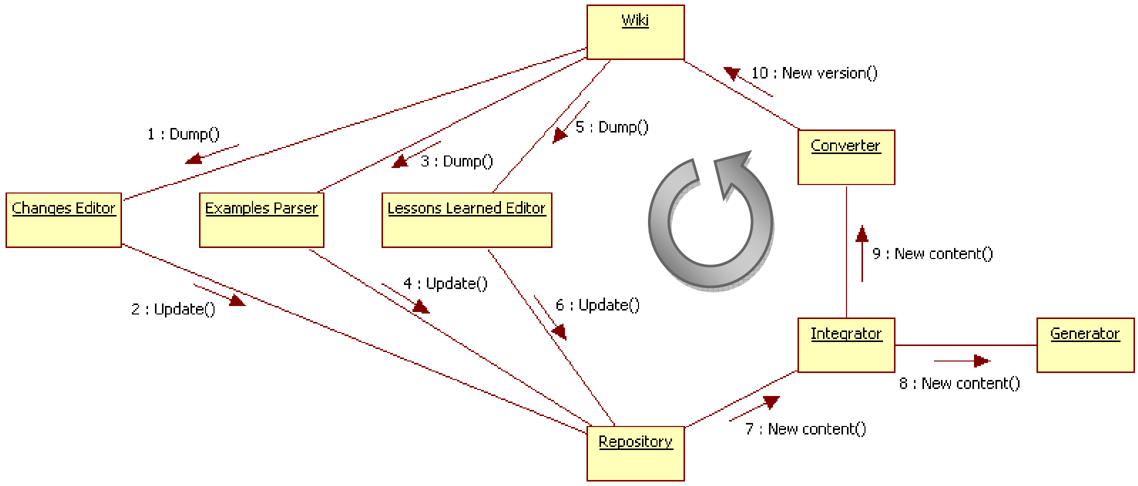


Figure 3. Collaboration diagram of the proposed solution.

4 Validation of the Proposed Solution

The main objective of the study is to obtain a system to support the configuration management of the organizational knowledge, allowing its users to improve their learning about the software processes. Such system must provide an easy mechanism to gather and store existing knowledge into a repository and encourage users to contribute and reuse knowledge within the organizational repository.

In order to evaluate the proposed solution, two validation objectives were defined:

- Determine the effectiveness of the feedback mechanisms to obtain upgraded knowledge. This effectiveness was analysed in terms of the use of the feedback mechanisms and the identification of useful knowledge items provided using each feedback way.
- Analyse the quality of the new knowledge elicited and the subjective evaluation of the usefulness of this new knowledge. This analysis will be done through the analysis of the knowledge items that were included as pre-existing elements in the subsequent versions of the organizational knowledge repository and the determination of the users’ satisfaction degree in relation to the repository and feedback process.

The validation was carried out in two phases and two different groups of students participated in it.

a) Phase 1: From September 2009 to January 2010

In this phase students used the PAL-Wiki presented in [16]. In this phase there wasn’t any feedback mechanism to enrich the organizational knowledge and the instructors used the e-mail to solve any question made by the students. Statistics about user accesses and contents accessed and downloaded were collected. This phase had a training stage and a project stage as Figure 4 shows.

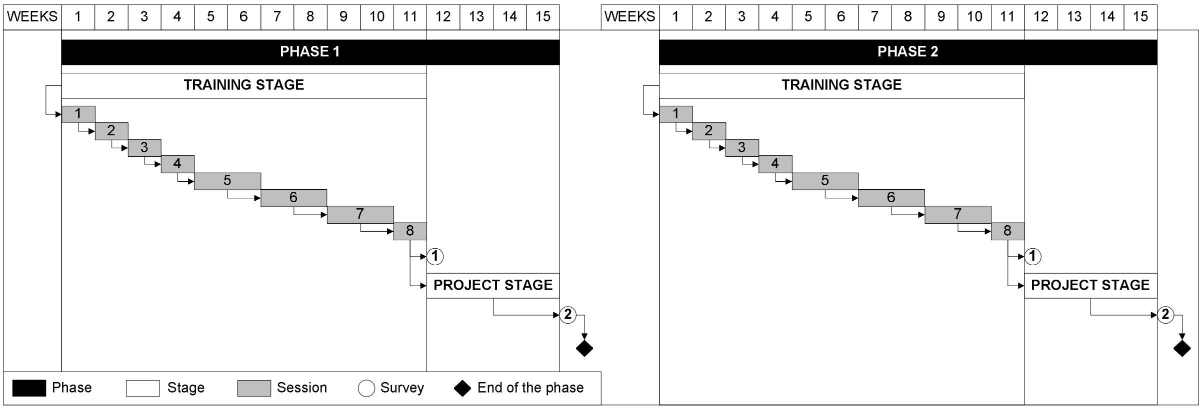


Figure 4. Timeframe for every phase in the validation.

The training 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 teams used the PAL-Wiki to perform the practices. The students were familiarized themselves with the PAL exploring its contents. Once completed the 8 sessions, a survey was carried out to collect data about the use of the PAL-Wiki.

The project 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 this stage, a second survey assessed the PAL-Wiki capabilities to provide information during the project stage.

b) Phase 2: From September 2010 to January 2011

In this phase students used the proposed solution in this paper that includes a feedback mechanism to enrich the organizational knowledge. This phase had the same stages than Phase 1 (Figure 4).

Three wikis were deployed in this phase to let the students access the same knowledge: a mediated wiki in which new knowledge were supervised by the professors in order to decide whether to add it to the repository or not, a non-mediated wiki and an EPG-Wiki in which students weren't allowed to directly contribute new knowledge.

During the training stage, the first two wikis gathered knowledge from the students while the knowledge from the students which used the third wiki was gathered by e-mail. At the end of the training stage, a new version of the three wikis was deployed. This new version contained the organizational knowledge in the repository at that moment, that is the pre-existing knowledge from the beginning of the training stage plus the new knowledge gathered during the training stage. Then, at the beginning of the project stage the three wikis offered the same organizational knowledge again.

Statistics about user accesses and contributions to the three wikis were collected during the whole phase. A survey was also carried out at the end of both the training stage and the project stage. The first one consisted of 5 closed questions about the use of the wiki, 8 opinion questions using a 4-point Likert scale and 27 semantic differential items using a 5-point Likert scale. The second survey consisted of 5 closed questions about the use of the wiki, 12 opinion questions using a 4-point Likert scale and 30 semantic differential items using a 5-point Likert scale. The closed questions asked the participant about his or her number of accesses and contributions to the wiki. The 4-point Likert scale questions assessed the ease of use of the mechanisms to gather new knowledge and the usefulness of the new knowledge. The semantic differential items assessed the usefulness and quality of the artefacts.

In order to assess the evaluation objectives defined at the beginning of this section, we analysed the collected data to verify if the new knowledge is relevant in quantity (validation objective 1) and if the new knowledge is relevant in quality and useful for the students (validation objective 2). The analysis of results and discussion is presented in the following section.

5 Analysis of Results and Discussion

The analysis of data obtained in the two validation phases was performed using descriptive statistics. The results and discussion for each objective are presented below.

5.1 Objective 1: Effectiveness of the Feedback Mechanism

To determine whether the feedback mechanism is effective of not, we will analyse the number of contributions, that is, changes in pre-existent knowledge, discussions and examples made by the students in the two phases of the validation (Table 1).

	Phase 1 (24 students)			Phase 2 (16 students)		
	Training stage (11 weeks)	Project stage (4 weeks)	Total (15 weeks)	Training stage (11 weeks)	Project stage (4 weeks)	Total (15 weeks)
Examples	240	221	461	160	147	307
Discussions	0	0	0	31	51	82
Lessons Learned	0	0	0	26	44	70

Table 1. Number of contributions.

The number of examples per student in both phases was nearly the same because in both phases the students had to do similar work. However, in Phase 1 there were no discussions in the wiki because the doubts of the students were attended personally or by e-mail, while in Phase 2 there were 82 discussions in the wikis. The reason for the increase in the number of discussions in Phase 2 was that students were motivated by the professors to publish their doubts as discussions in the wikis instead of sending e-mails and to participate in the discussions of their classmates within the wikis. This is also applicable to the lessons learned because they are obtained from the discussions.

Another interesting point is that students didn't make any change in the pre-existing knowledge neither in Phase 1 nor in Phase 2. When asked in the surveys during Phase 2, most of the participants considered that the pre-existing knowledge was correct and some of them stated not having enough knowledge about the process to make changes.

Therefore, the new feedback mechanism is effective in the sense that allows software engineers to contribute new knowledge to the organization in the form of discussions. What is more, these discussions may become lessons learned enriching then the pre-existing knowledge. In addition, although the number of examples didn't increase using the new feedback mechanism, the examples are now part of the system because they are no longer stored in an external repository.

Finally, this feedback mechanism has proven to be effective in gathering new knowledge from several sources and spreading the elicited knowledge through different platforms. Thus, the same enriched organizational knowledge can be presented in different views depending on the needs of its users.

5.2 Objective 2: Quality of the New Knowledge Elicited

In order to measure the quality of the new knowledge elicited we will analyse the levels of quality perceived by the professors and the levels of quality and usefulness perceived by the students about the discussions, the lessons learned and the examples. Data related to the quality and usefulness of the contributions was obtained in the surveys conducted at the end of the training stage and the project stage in the two phases of the validation.

The quality of the new knowledge perceived by the professors can be measured in terms of how much inefficient new knowledge was fed back into the repository. At the end of the training stage in Phase 2, all the 8 discussions in the non-mediated wiki were accepted and stored in the repository as lessons learned. However, only 6 of them should have been accepted, so there was some inefficient new knowledge introduced in the feedback. On the other hand, the amount of inefficient new knowledge introduced in the feedback during the project stage decreased because only 7 of the 32 discussions should have been discarded. With regard to the examples, at the end of the training stage in Phase 2, all the 50 examples were stored in the repository.

From the point of view of the students, the levels of quality perceived by them about the new knowledge elicited are presented in Table 2, rated in a 1-to-5 scale.

			Discussions	Lessons Learned	Examples	Average
Phase 1	Project stage	PAL-Wiki	2.08	n/a	3.88	2.98
Phase 2	Training stage	Mediated wiki	3.38	n/a	4.63	4.00
		Non-mediated wiki	1.50	n/a	2.00	1.75
		EPG-Wiki	n/a	n/a	4.33	4.33
		All the 3 wikis	2.75	n/a	3.87	3.31
	Project stage	Mediated wiki	3.50	3.13	4.25	3.63
		Non-mediated wiki	2.25	2.50	3.50	2.75
		EPG-Wiki	n/a	3.33	3.67	3.50
		All the 3 wikis	3.08	3.00	3.93	3.34

Table 2. Quality of the new knowledge elicited.

Examples were better valued than discussions in Phase 2 because they show how to perform the processes while discussions were just used to solve specific doubts. At the end of the project stage of Phase 2, ratings increased with respect to the training stage. The main reason was that the proposed system was new to the students so it required some time to get used to it and to get them involved.

Ratings from users of the mediated wiki were higher than ratings from users of the non-mediated one. As some students commented in the surveys, the lack of participation of some classmates in the non-mediated wiki affected negatively. The willingness of students towards they work affects their participation in the wikis, making more contributions and with better quality. Statistics support this statement because the students who contributed actively produced better quality products in the project stage while the students who didn't contribute actively produced average products in the project stage.

Considering all the 3 wikis in Phase 2, ratings were better than in Phase 1. There was a significant increase in the quality of the discussions perceived by the students due to the increase in the use of discussions as a mean to solve doubts.

We finally need to address the question of how useful was the new knowledge for the students. In general, the students considered the examples were useful to solve doubts, to optimise the time and to enrich their knowledge. Thus, usefulness of the examples was rated high in the surveys, especially in the mediated wiki and in the EPG-Wiki with ratings over 4.3 in a 1-to-5 scale (Table 3).

			Discussions	Lessons Learned	Examples	Average
Phase 1	Training stage	PAL-Wiki	2.86	n/a	4.47	3.66
Phase 2	Training stage	Mediated wiki	3.00	n/a	4.38	3.69
		Non-mediated wiki	1.75	n/a	3.00	2.38
		EPG-Wiki	n/a	n/a	4.67	4.67
		All the 3 wikis	2.58	n/a	4.00	3.29
	Project stage	Mediated wiki	3.38	3.00	4.38	3.58
		Non-mediated wiki	2.00	1.75	3.50	2.42
		EPG-Wiki	n/a	3.33	4.33	3.83
		All the 3 wikis	2.92	2.73	4.13	3.26

Table 3. Usefulness of the new knowledge elicited.

However, students didn't agree about the usefulness of the discussions and the lessons learned. On one hand, the users of the mediated wiki considered they were useful but not as much as the examples. On the other hand, the users of the non-mediated wiki considered the discussions weren't useful. The lack of participation in the non-mediated wiki affected the perception of the usefulness of the contributions, as affected the perception of the quality.

6 Conclusions

This paper proposes a system to support the configuration management of the organizational knowledge. The system uses a feedback mechanism to gather new knowledge and to store it into a repository enriching the organizational knowledge.

The feedback mechanism allows software engineers to contribute new informal knowledge to the organization in the form of discussions. These discussions can be stored in the knowledge repository becoming lessons learned and enriching then the pre-existing knowledge. In addition, formal knowledge can be gathered during the project execution in the form of examples and stored in the same knowledge repository.

During the validation, examples were better valued than discussions because examples show how to perform the processes while discussions are usually used to solve specific doubts. As the system is new to the software engineers, it requires some time to get used to it and to get them involved. Thus, the perception of the quality and usefulness of the different ways of contribution to the organizational knowledge repository improves with the time.

On the other hand, the lack of participation of some software engineers can affect negatively. The willingness of software engineers towards they work affects their participation, making more contributions and with better quality.

Finally, the new feedback mechanism allows gathering new knowledge from several sources and spreading the elicited knowledge through different platforms. Thus, the same enriched organizational knowledge can be presented in different views depending on the needs of its users.

7 Literature

[1] Halawi, L.A. et al.: “Knowledge management and the competitive strategy of the firm”. In The Learning Organization, 13 (4), 384-397 (2006).

[2] Maier, R.: “Knowledge Management Systems: Information and Communication Technologies for Knowledge Management”. 3rd edition. Springer (2007).

[3] Alavi, M.; Leidner, D.: “Knowledge Management and Knowledge Management Systems: Conceptual Foundations and Research Issues”. In MIS Quaterly, 25 (1), 107-136 (2001).

[4] Garcia, S.: “What is a Process Asset Library? Why Should You Care?”. Aimware Professional Services Inc., Boston, MA, USA (2004).

[5] Calvo-Manzano, J.A. et al..: “Process Asset Library to Support Software Process Improvement in Small Settings”. In Proceedings of the 15th EuroSPI, Dublin, Ireland, September 3-5, 2008.

[6] Komi-Sirviö, S. et al.: “Toward a Practical Solution for Capturing Knowledge for Software Projects”. In IEEE Software, 19 (3), 60-62 (2002).

[7] Assimakopoulos, D.; Yan, J.: “Sources of Knowledge Acquisition for Chinese Software Engineers”. In R&D Management, 36 (1), 97-106 (2006).

[8] Dingsøyr, T. et al.: “What Do We Know about Knowledge Management? Practical Implications for Software Engineering”. In IEEE Software, 26 (3), 100-103 (2009).

[9] Bjørnson, F.O.; Dingsøyr, T.: “Knowledge management in software engineering: A systematic review of studied concepts, findings and research methods used”. In Information and Software Technology, 50 (11), 1055-1068 (2008).

[10] Burke, D.; Howard, W.: “Knowledge Management and Process Improvement: A Union of Two Disciplines”. In CrossTalk – The Journal of Defense Software Engineering, 18 (6), 28 (2005).

[11] Aurum, A. et al.: “Investigating Knowledge Management practices in software development organizations – An Australian experience”. In Information and Software Technology, 50 (6), 511-533 (2008).

[12] Riechert, T.; Berger, T.: “Leveraging Semantic Data Wikis for Distributed Requirements Elicitation”. In 2009 ICSE Workshop on Wikis for Software Engineering (Wikis4SE), Vancouver, Canada, May 19 (2009).

[13] Xiao, W. et al.: "On-line collaborative software development via wiki". In Proceedings of the 2005 international symposium on Wikis (WikiSym), pp. 177-183, Montreal, Canada, October 21-23 (2007).

[14] Rech, J. et al.: "Using Wikis to Tackle Reuse in Software Projects". In IEEE Software, 24 (6), 99-104 (2007).

[15] Wongboonsin, J.; Limpiyakorn, Y.: "Wikipedia Customization for Organization's Process Asset Management". In Proceedings of the 2008 ICACTE, pp. 467-471, Phuket, Thailand, Dec. 20-22 (2008).

[16] Amescua, A. et al.: "Knowledge repository to improve agile development processes learning". In IET Software, 4 (6), 434-444 (2010).

[17] Falbo, R.A. et al.: "Using Knowledge Management to Improve Software Process Performance in a CMM Level 3 Organization". In Proceedings of the 4th QSIC, pp. 162-169, Germany, Sept. 8-9 (2004).

[18] He, J. et al.: "A Framework of Ontology-Supported Knowledge Representation in Software Process". In Proceedings of the 2007 ISKE, Chengdu, China, October 15-16 (2007).

[19] Eclipse Process Framework. http://www.eclipse.org/epf/, visited February 14, 2011.

[20] Osellus Company. IRIS Process Author. http://www.osellus.com/IRIS-PA, visited February 14, 2011.

[21] Holmström, P.: "Ideas for Next Generation Process Authoring Tools". Master's Thesis. Åbo Akademi University (2009).

8 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.

Alberto Heredia

Alberto Heredia has been teaching assistant in the Computer Science Department at Carlos III University of Madrid (UC3M) since 2006. His current research interests focus on Software Engineering, and more specifically on Software Project Management and Knowledge Management, carrying out his thesis in relation to the generation of Electronic Process Guides. Alberto has taken an active part in several R&D and innovation projects related to the Collaboration@rural (C@R) European integrated project, the Information Technologies and the services provision of knowledge management for software process improvement, among others.

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.

Improvement of Idea Generation by Capitalisation on Internal and External Stakeholders

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 information on the product during its complete life cycle. This applies especially to companies which obtain their competitive advantages by technological lead and quality. The automotive supplier industry is one such sector. Suppliers have to predict product strategies and technologies that decide on market success. Therefore their innovation management has to guarantee a sustainable idea generation to support the product development with the continuous flow and collection of new successful ideas. To this aim, all the stakeholders involved throughout the complete product life cycle are valuable resources of knowledge and ideas which are largely unexploited in current innovation management systems. Methods and tools have to be found that facilitate the systematic involvement of these stakeholders in idea generation without imposing considerable additional efforts on them. This paper presents a case study of existing idea management at the automotive supplier Kolbenschmidt Pierburg AG (KSPG), and points out improvement steps through the targeted enlargement of idea management to internal and external stakeholders.

Keywords

Idea Generation, Idea Management, Innovation Management, Stakeholder Concept, Automotive Supplier Industry

1 Introduction

Many companies think that innovation is the universal remedy for their economic problems. High flop rates of innovations [1] confirm that this is not correct. For example, only 20 percent of the R&D spending by both OEMs and automotive suppliers represents profitable innovation investment [2]. The development of new products and processes is an extremely complex procedure that many companies, despite extensive theoretical findings, control only in limited way [3].

Product innovations and associated innovation activities confront the innovation management with different challenges how to handle idea generation. In this context, the main purpose of all idea generation activities is to ensure that the company does not leave the exploration phase of new product development to chance [4]. Therefore companies should try to increase the quality of their ideas by the enlargement of its sources of ideas. All members of the innovation value chain should participate in a systematically and continuously organised idea generation process to guarantee sustainable innovation results and business success.

This paper discusses topics linked to the improvement of the existing innovation management at the automotive supplier Kolbenschmidt Pierburg AG (KSPG) [5]. Section 2 of this paper gives a short overview of the research question and is devoted to the idea sources KSPG already uses in its daily business. Section 3 presents some suggestions how to enlarge these idea sources to improve innovation management. This paper is closed in section 4 with a conclusion and a final evaluation.

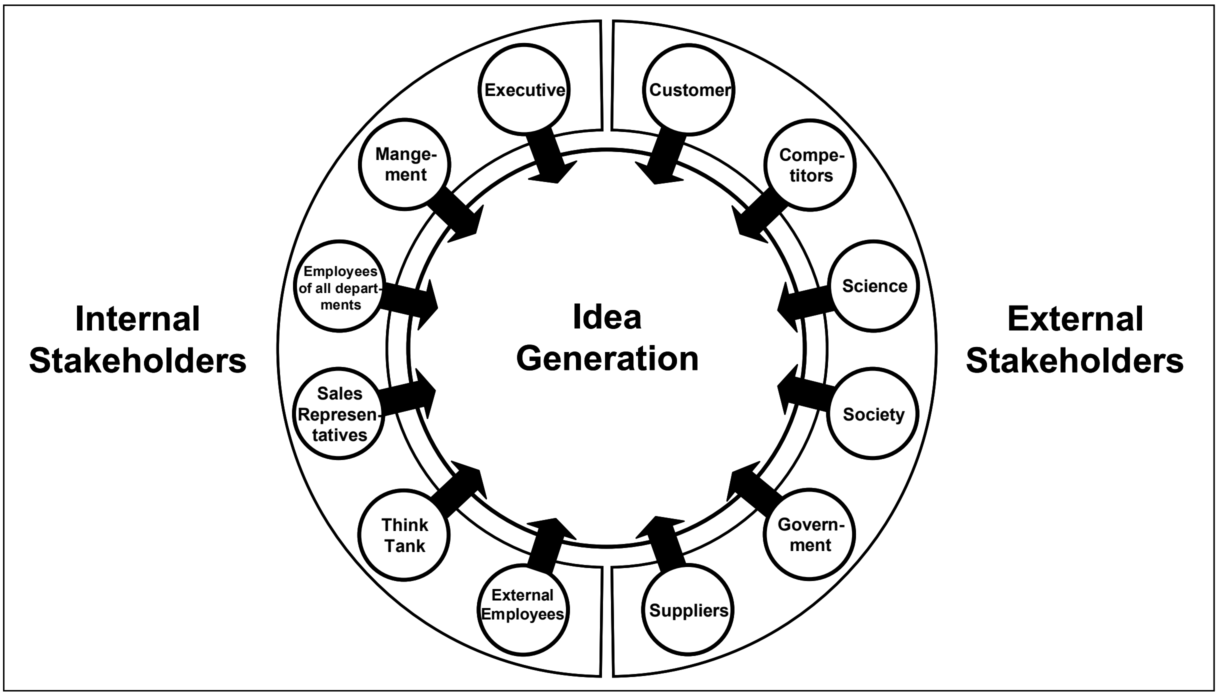
2 Idea Generation at Kolbenschmidt Pierburg AG

2.1 General Facts

A critical analysis of the existing innovation management system at KSPG reveals that idea generation is currently limited to a core group of employees who act as idea contributors. This situation represents a threat of idea stagnation, which is why the company's innovation management has declared the improvement of idea generation as one of its major strategic objectives.

The adopted solution approach is to enhance idea generation using the model of market stakeholders [6, 7]. The basic concept is that not only one group of idea contributors should be responsible for innovations; also other stakeholders of the corporate environment should be actively involved in the idea generation process. Figure 1 demonstrates this approach.

Figure 1. Idea Generation based on the Stakeholder Concept



To capture new ideas from different sources, it is essential to identify potential sources inside and outside the organisation. With this identification of additional idea sources it is necessary to analyse methods to extract and format their information and knowledge, and to evaluate it in form of short- and long-term aspects for the company's innovation development.

There is a very important consideration in the involvement of a large group of different stakeholders in the idea generation process. As was shown in research on integrated product and system design [8], integrating stakeholders of the complete product/system life cycle throughout the entire product/system development process from the earliest phases on, is the key to creating sustainable innovation. The sustainability aspect is leveraged by the fact that only the integration of different views on the product/system in terms of its functions and its economic, ecologic, and social environment allows to identify requirements and constraints on the product/system in a holistic manner, and therefore to take them into account both in the composition of development teams, as well as in the design and architecture of the product/system. The same issue applies to idea generation and assessment, which is part of the earliest upfront phases in the product/system life cycle. Figure 1 distinguishes among stakeholders who are internal to a particular organisation and those which are external to it. There is no unique grouping of related stakeholders; however concepts from social science help in clustering stakeholders as it has been done in the model represented in Figure 1.

This approach of integrating more stakeholders in the idea generation process has started with an as-is analysis of KSPG's current innovation management with special regard to the idea sources up to now and how these 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 has to be analysed 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 organisational framework and IT-infrastructure which makes the collection and evaluation of these ideas possible. This system is partly existent at KSPG thanks to the efforts of its two divisions Pierburg and Pierburg Pump Technology (PPT) which have implemented a so-called "Innovation Database". Innovation management uses this central IT-tool to provide internal stakeholders a facility to easily submit their ideas. The Innovation Database supports the innovation management in the collection, evaluation and selection of ideas. On the other hand, however, information from external stakeholders must not be ignored. These stakeholders must be observed and explored for usable ideas.

The following Section 2.2 gives an overview of the as-is analysis of the existing idea sources within KSPG based on the stakeholder model. Due to their key role concerning innovation management, it will focus on the divisions Pierburg and PPT.

2.2 Current idea sources

2.2.1 As-is Analysis of Internal Sources of Ideas

The central tool of innovation management to capture internal ideas at Pierburg and PPT is the Innovation Database. The Innovation Database is available via intranet for all Pierburg and PPT employees. In March 2011, more than 3900 users were registered. The database supports the innovation management in the collection, evaluation and selection of inventions and technical product ideas. The system is secured and thus protected against unauthorised access. The standardised process cycle of the Innovation Database ensures a simplification and a shortening of the operational workflow.

Since the introduction of the Innovation Database in 2006, this tool is primarily used by a core group of 5% of Pierburg's employees as idea contributors. Although the tool is available to all employees in all departments (like R&D, Sales, Purchasing, etc.), including management and executives, and in the case of PPT also in plants outside Germany in English language, input from not R&D related departments and from employees in leading positions outside the R&D department is very low.

The current implementation of the database mainly supports the evaluation and selection of inventions. Up to 95% (Status: March 2011) of the ideas already collected within the database are inventions. Intellectual property plays a major role in a technology-driven business environment like the automotive industry, but it is important that innovation is not to be confused with the term invention [3,9,10,11]. While invention describes the first technical realization of a new problem solution developed as a result of research activities leading to a legal basis for the utilisation of the results (for example in the form of patents), the term innovation denotes the successful conversion of inventions in marketable solutions. R&D is the basis for the development of innovations. It covers a set of specific processes that are created to gain knowledge and to discover new technical solutions to a problem [3,9,11].

Regarding the degree of novelty of the collected ideas in the innovation database, nearly 80% (Status: March 2011) of the ideas are application-related incremental innovations [9], and describe useful modifications of existing products for the day-to-day business. Only 20% are development-related radical innovations [9] and are of relevance to the advanced development department to guarantee sustainable success in the long term.

Another weak point of the Innovation Database is liked to its configuration. Although there is an idea pool included which should stimulate the users in that they see ideas and concepts suggested by their colleagues, there is no systematic use of this pool by the innovation management. Text Mining based tools supporting the clustering of ideas and identification of main topics and subjects would facilitate the systematic integration of the idea pool in innovation management activities. Vice versa, the innovation manager could broadcast a major product question as some sort of campaign within the database to stimulate and direct the input of idea contributors.

2.2.2 As-is Analysis of External Sources of Ideas

According to Figure 1 the as-is analysis of external sources of ideas will concentrate on following six main sources: customers, competitors, science, society, government and suppliers.

At KSPG a lot of activities and techniques exist which are partly directly connected with idea sources and the generation of product ideas, other actions are indirect idea sources and influence indirectly the generation of product ideas. Up to now these information sources help mainly management and business development. It has to be analysed how they can also be used for successful product idea generation. Table 1 shows the major existing external idea sources at KSPG.

Table 1. Overview of external idea sources

Stakeholder	Idea Source	Direct way to get ideas	Indirect ways to get ideas
Customers	• Core customer groups	• Customer submitted ideas • Interviews • Customer contracts negotiations	• Customer analysis • Satisfaction surveys • Customer database • Internal customer-related teams
Competitors	• Direct competitors	• Competitive Intelligence • Direct talks during international fairs and summits	• Market research firms
Science	• Universities	• Sponsoring of university chairs • Master thesis projects • Networking	• Scanning new technology releases, like PhD thesis or other publications
Society	• Groups of interests like Industry Associations • Media sources	• Working groups • Contact with editors	• Publications from associations • Scanning media, especially internet research or patent research
Government	• Governmental departments	• Visiting respective website • Scanning new technology regulations	• Attend in national and international fund programs for innovation projects • Scanning commentaries concerning new laws

Suppliers	1. Suppliers of physical goods, for example Tier 2 suppliers, etc.	• Supplier submitted ideas • Meetings • Contract negotiations	• Supplier analysis • Research for news from suppliers
	2. Suppliers of information, like consultants and research firms	• Direct talks • Visiting presentations • Networking	• Working with database of consultants • Use of provided automotive information services

Looking at Table 1, KSPG has access to a lot of possible external idea sources still to be capitalised on. Unfortunately the information about these external idea sources is widely spread within the company and no central storage of this knowledge exists. So the collection of information must be effected individually and it is necessary to know the right contact person within the company for the collection of the specific information from the external idea sources. No systematic knowledge management is implemented so far.

3 Enlargement of Idea Sources

The following section is based on the results of the as-is analysis in section 2.2 and reveal a starting point for the improvement of innovation management at KSPG.

3.1 Enlargement of Internal Sources of Ideas

The definition of internal stakeholders in Figure 1 is the basis for innovation management to whom in-house idea generation activities have to be addressed. In particular, employees are highly cited as sources of ideas [4,12,13]. Typically, mainly executives and R&D employees submit ideas, and the potential of idea generation by all employees in the company remains largely unexploited. An organisational framework to manage idea generation by actively involving all these internal stakeholders is essential. Table 2 summarises the direct and indirect ways to generate ideas from these internal sources.

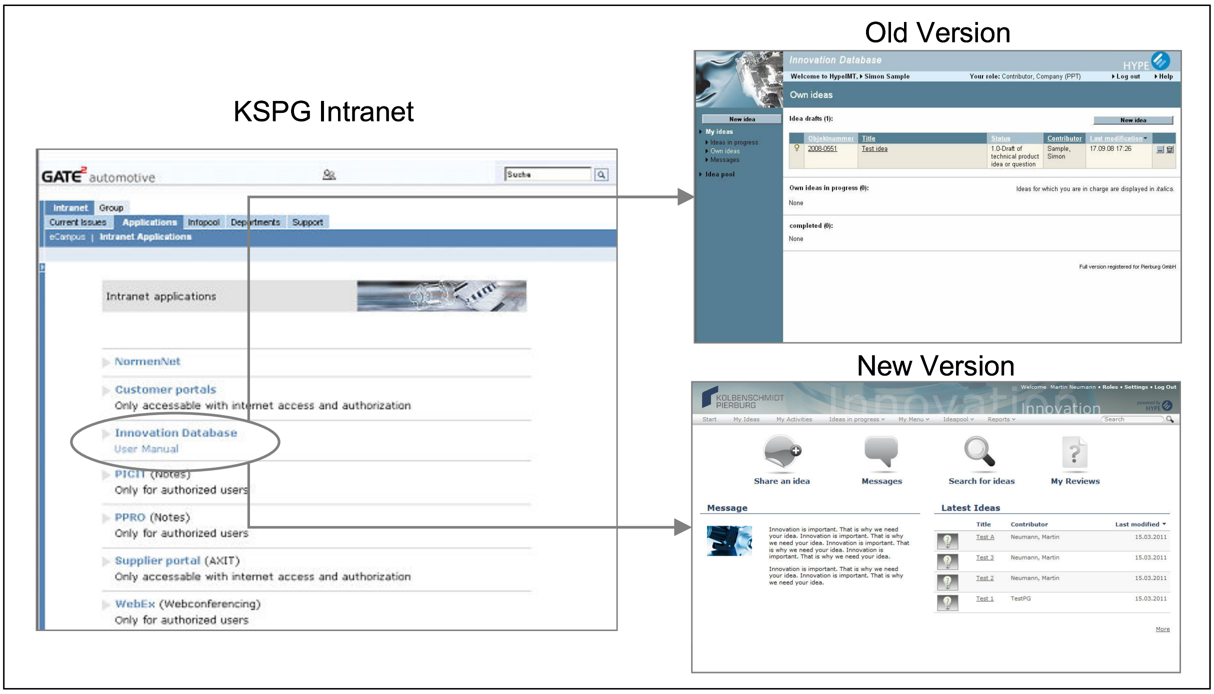
Table 2. Overview of internal idea sources

Stakeholder	Idea Source	Direct way to get ideas	Indirect ways to get ideas
Executive	Executive in the company	• Idea generation activities and processes • Idea/Innovation Database • Idea management or corporate improvement suggestion system • Continuous improvement process • Inventions	• General overview of external stakeholders' interests
Management	Management professional in the company	• Same as first source	• Same as first source
Employees of all departments	Departments includes R&D, sales, purchasing, customer service, quality control, production, controlling etc.	• Same as first source	• Especially sales and customer service should capture customer ideas • Purchasing should collect supplier ideas

Sales Repre-sentatives	Sales representatives promote and sell products directly to customer in the field; may have their office at the customer's site	• Contractual agreements • Direct solicitation	• Organisation of workshops or seminars at the customer's site, for example R&D can present and discuss new product solutions direct with the customer on-site
Think Tank	A group dedicated to coming up with new ideas, research and knowledge	• Outcome based innovations • Inventor circles	• This group can have members from all departments and so different aspects can be considered
External employees	Collective term for loosely affiliated employees, like project-based employees, temporary employees, freelancers or students	• Same as first source	• Stimulus from outside • Possible solution to avoid to be professionally blinkered

According to the as-is analysis at KSPG in section 2.2.1, the improvement of the "Innovation Database" as existing innovation management support tool, can help involve these different groups of internal stakeholders. Practically this necessitates overcoming technical obstacles by installing a new program version on a separate IT server to improve performance, and to give it a more user friendly interface. Also, idea contributors have to receive faster responses and a better qualitative feedback about their submitted ideas. This upgrade process has already been started. Figure 2 illustrates the much more attractive user interface of the new database version compared to the old version.

Figure 2. Comparative illustration of the old and new version of KSPG's Innovation Database



Apart from this improvement of the tool, innovation management has to create an environment for the promotion of innovation. In general innovation management has to motivate all employees to take part in the innovation activities, and in particular innovation management has to give the impulses for ideas by confronting for example inventor circles or especially formed integrated teams, with new market trends to stimulate their idea generation. This new market trends could be at one hand identified within the Innovation Database by using knowledge mining techniques or on the other hand discovered throughout external sources. For example, the regulatory environment may provide opportunities for customised products to meet regulatory requirements in areas such as pollution and emission control, a topic which influences automotive suppliers like KSPG significantly.

3.2 Enlargement of External Sources of Ideas

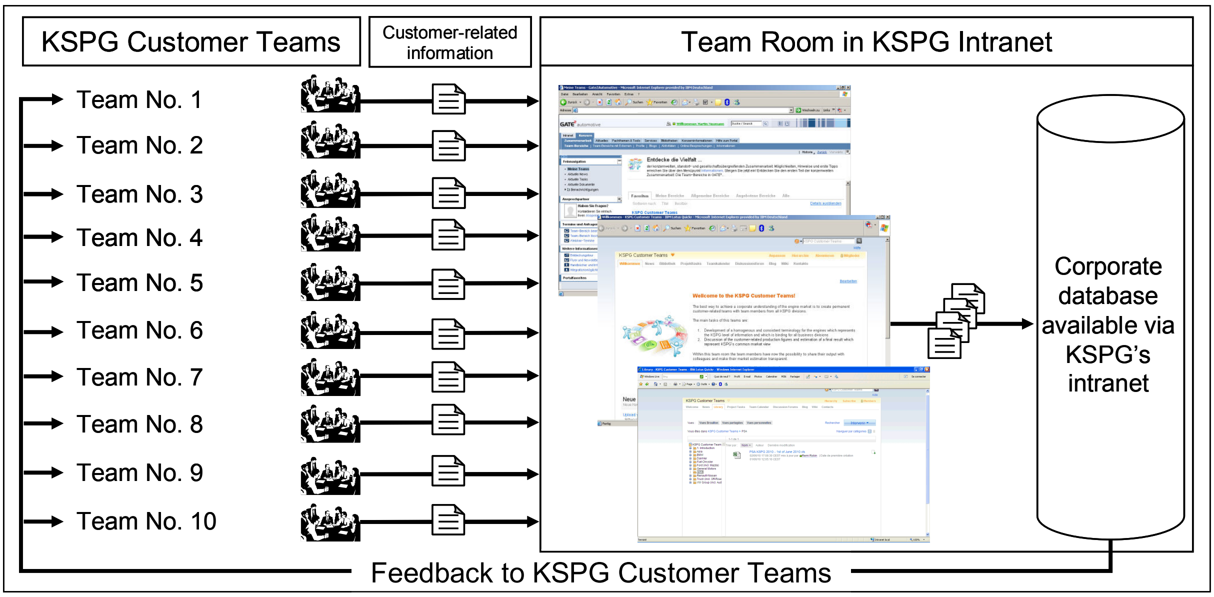
To achieve sustainable innovation success, it is important to obtain internal acceptance for the usage of external idea sources. One possible way is to use internal contact persons for the collection of external ideas first. However, it is important to minimise the individual work effort for the internal contact person to get information from external stakeholders, and to share this knowledge with other colleagues. When this approach is applicable it makes sense to widen the sources of ideas within the specific categories of external stakeholders, which are shown in Table 1 and which are not fully integrated in the idea generation process. Thus, the exploitation of external idea sources is first of all an internal step-by-step process.

For example, one efficient step in the direction of a better collection of customer ideas was the creation of permanent customer-related teams with team members from all KSPG sales divisions. The main tasks of these teams are:

1. Build-up knowledge about KSPG customers and share these customer insights with team members.
2. Development of a homogenous and consistent understanding of the customers' future production, which represents the KSPG level of information and which is binding for all business divisions.
3. Discussion of the customer-related topics and estimation of a final result, which represents KSPG's common market view.

Members of these teams have the possibility to share their knowledge with colleagues and make their market estimations transparent. For the management of these KSPG Customer Teams, an IT solution within the KSPG's intranet, a so-called team room, has been created and continuously improved. Since the introduction of this IT solution in the summer of 2010, it is possible to collect and to store systematically the data of the customer team members. This concept of KSPG Customer Teams is depicted in Figure 3.

Figure 3. Concept of KSPG Customer Teams



Further external idea sources for innovation purposes should be definitely used to obtain the targeted improvement of idea generation at KSPG.

4 Conclusion

This paper points out that idea generation has to be understood as part of the whole innovation process. To support new product development, idea generation activities may not only be based on well-known internal sources but has to be expanded to external sources. Many more ideas are generated by others outside the company which could be turned into new innovative products and services. It is a major mistake to think ideas can only come from inside the company. This error has been termed the not-invented-here (NIH) syndrome where companies reject ideas generated outside its walls because they think those ideas are inferior to their own [14].

Knowledge plays an important role in this context. It is essential to build up a fundamental understanding in the company for the utilisation of these new idea sources and the accompanying advantages of this new concept of innovation management. This may helps to avoid internal restrictions. Moreover, it can be said that this challenge of openness may provide an exceptional stimulus on innovation and sustainable success.

In conclusion, the described conceptual thoughts concerning idea generation outlined in this paper would be very useful for the improvement of innovation management in general, and for KSPG in particular.

Literature

[1] Brockhoff K. (1999). Forschung und Entwicklung: Planung und Kontrolle. 5th edition, Oldenbourg, München Wien 1999

[2] 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

[3] Stockmeyer B. (2001). Ansatzpunkte und Methoden zur Effizienzsteigerung im Innovationsmanagement der Ernährungsindustrie. PhD Thesis, Technische Universität München, Germany 2001

[4] Stasch S. F., Lonsdale R. T., LaVenka N. M. (1992). Developing a Framework for Sources of New Product Ideas. In: Journal of Consumer Marketing, 9 (2), pp. 5-15

[5] Neumann M., Riel A., Brissaud D. (2010). Innovation Management System for the Automotive Supplier Industry to Drive Idea Generation and Product Innovation. In: Proceedings of the 17th EuroSPI Conference, Vol. 17, pp. 7.13-7.24

[6] Freeman R. E. (1984). Strategic Management: a Stakholder Approach. Pitman, Boston, MA 1984

[7] Freeman R. E. (2004). The Stakeholder Approach Revisited. In: Zeitschrift für Wirtschafts- und Unternehmensethik (zfwu), Jahrgang 5, Heft 3, pp. 228-241

[8] Riel A., Tichkiewitch S., Messnarz R. (2009). Integrated Engineering Skills for improving the System Competence Level. Software Process Improvement and Practice (SPIP) 14(Nov/Dec 2009), pp. 325–335

[9] Pleschka F., Sabisch H. (1996). Innovationsmanagement. Schäffer-Poeschel, Stuttgart 1996

[10] Nagel R. (1993). Lead User Inovationen. DUV, Wiesbaden 1993

[11] Specht G., Beckmann C. (1996). F&E-Management. Schäffer-Poeschel, Stuttgart 1996

[12] Alam I. (2003). Commercial innovations from consulting engineering firms: An empirical exploration of a novel source of new product ideas. In: Journal of Product Innovation Management, Vol. 20, Issue 4, pp 300-313

[13] Belliveau P., Griffin A., Somermeyer (Eds.) (2004). The PDMA ToolBook 2 for New Product Development. John Wiley & Sons, Inc., Hoboken, NJ 2004

[14] Katz R., Allen T. J. (1982). Investigating the Not Invented Here (NIH) syndrome: A look at the performance, tenure, and communication patterns of 50 R&D Project Groups. In: R&D Management, Vol. 12, Issue 1, pp. 7-12

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 AG 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.

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.

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.

Glimmering Interface for Successfully Managing Outsourcing Partnerships

Masao Ito

Abstract

In offshore software development, communication between the outsourcer and the vendor is difficult. Especially in the case of reporting the status of a project, the outsourcer is affected by a lack of information that may not appear in periodical reports. For addressing this issue, We propose the concept of glimmering interface, which performs two important functions: 1) First, it uses the data that the developer in vendor company input in order to develop software, not for management purpose and 2) project progress data is subtly offered to the outsourcing manager so as not to interrupt his activities. If the manager decides to take action, a tool that supports the glimmering interface provides navigation towards detailed information on the progress data. This concept is also useful for development in the same site because timing and contents of the status information are important for monitoring the development process.

Keywords

Distributed software development, Project support tool, Awareness, Ambient Information System, Visual interface design

1 Introduction

In offshore development, the outsourcer monitors and manages projects undertaken by vendor companies in remote locations. In distributed software development, the actual conditions at the remote development site are difficult to determine because the main communication methods are mail, phone and progress reports, which do not contain the project's raw data. This lack of contact prevents offshore development outsourcing managers from making timely critical decisions (cf. in [1] most risky factor in offshore business is 'lack of top management commitment'). This paper proposes methods and tools for overcoming this complication by providing alerts of risky situations to the outsourcer, thus creating a trigger for making decisions.

2 Analysis

In offshore and distributed software development, inadequate communication is typical. To address this problem, multi-modal communication mechanisms such as mail, phone and instant messaging are used [2]. In the case of small onsite projects, signs of unusual activities can be detected rapidly and easily. However, in the case of distributed software development, the manager may be simultaneously handling numerous tasks and cannot devote his attention to a single project. In addition, simple, consolidated information that reflects actual conditions of each project may be difficult to attain in a remote setting. Therefore, a method for improving communication in offshore and distributed software development is necessary.

2.1 Problem in Gathering Information

There is a problem: 'Manager of outsourcer wants to know the (risky) situation of project, but he never gets it when he needs.'

A project that advances well provides adequate time to developers on the vendor side for gathering data that can be used to report its progress. In addition, the developer may be eager to report the project's satisfactory progress to the manager because of psychological reasons. On the other hand, a project with multiple flaws may not provide ample time for reporting critical information, and the developer may be reluctant to report it. Thus, although the manager requires timely updates, especially regarding problematic issues, he may only receive positive news.

2.2 Difficulties Faced by Managers in Gathering Information

Lack of communication not only exists at the vendor end but also at the outsourcer end. An outsourcing manager may not always be alerted to a project's risky status because of two reasons: 1) he may be working on numerous tasks that require his time and attention and 2) maintaining focus on a project that undergoes slight changes is difficult. If the situation changes rapidly, the manager may decide to monitor it; however, the changes do not usually occur in a short term, for example, within an hour or a day. A project that continuously requires his attention is not an effective use of his time. On the other hand, developers working cooperatively are faced with a different situation. For these developers, it is important to feel that they are in the same virtual proximity. In such an environment, there is continuous flow of information because the developers work on the same design and coding activities [3].

3 Solution : Mitigation of Problem

In this section, We offer a solution to these problems that provides project managers with adequate and timely information on the project.

3.1 Categorization of Data for Development

Recently, visualization techniques for collecting data, such as MIERUKA (visualization) in Japan, which is well known in hardware manufacturing [4], have been reported. In factories, we can apply numerous sensors for continuous collection of problematic data and analyse the information mathematically for optimization purposes. The sensors can automatically send the data without any delay, and the machine is only able to perform simple tasks. Hence, the above-mentioned problem does not exist in this field.

To mitigate the problem in software industry, we can use the data that are collected for the developer instead of those for the manager. The developer must apply the data to his activities such as writing test cases and their results in the testing process. As a result, these data also show his daily activities. This may resolve the communication problem, because these data are created regardless of the project situation; the data contain both positive and negative information.

However, managers cannot use these data directly for development purposes because the progress report pertains to only a part of the software lifecycle. The test activity does not usually show the design stage of the project. Therefore, we will determine the type of data the developer usually records and attempt to apply the information for management purposes. Our extensive experience with software support tools enables us to identify three such categories: scheduling, issues and verification.

(1) Scheduling

This category of data contains information on each task including due dates and ending dates of as-signed tasks. In addition, it shows deviation between planned performance and actual performance.

(2) Issues

Issue management pertains to problems to be solved in a project. Risk management is also included in this category.

(3) Verification

This category includes many types of data related to verification activities such as review, walk-through and testing. Defect management is a typical subcategory in project management.

To clarify the characteristics of each category, We show the typical amount of data produced according to the progress of the project (Fig.1). In the early phase of the project, the scheduling category contains a large amount of data. The number of issue reports increases towards the mid-point but decreases with the project’s progression. The final phase contains numerous reports belonging to the verification category.

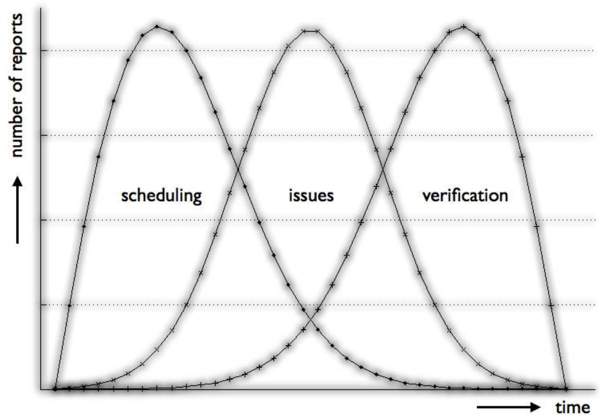


Fig. 1 Three curves showing quantitative changes in each category along the time axis. Each maximum number is normalized.

Usually by using the software reliability growth curve (it is the graph of the cumulative number of defect data and it is in the category of 'verification'), we can estimate the maturity of product's quality, but it is hard to estimate the situation in early or middle phase of software development only by this curve. Therefore, implementation of our three categories is useful in estimating the project status through its entire lifecycle.

We re-emphasize that the data are not collected for management purposes. For example, if you check the schedule data in order to decrease the difference between plan and actual time, developers are apt to set the long planned time or input the 'good' actual time, and we may fall into the problem. Therefore, we just collect the data for development purpose and we read the tendency of project status in it.

3.2 Glimmering Interface

In this subsection, we propose the concept of a 'glimmering interface' for providing a subtle alert to the outsourcing manager regarding the project status without disturbing his work. With this interface, we attempt to trigger the same awareness mechanism the manager uses when working on a small onsite project.

From the user interface aspect, the glimmering interface has two important advantages: 1) it provides simple information to the user so that he may judge whether an action is necessary and 2) it provides a navigation mechanism for easy access to detailed information on the project.

3.3 A Scenario

To explain the glimmering interface mechanism, we demonstrate a simple scenario: the developer inputs and modifies data to assist his and his colleagues' activities. These data are assigned a particular category (cf. 3.1), and each dataset is assigned its own status. Basic statuses such as 'open' and 'closed' are common in all datatypes and are important for judging the situation of a project. These data also contain the history of a status transition.

By employing this information, a project is subtly represented as a balloon on display (Fig. 2 (a)). If a problematic situation occurs—for example, a small amount of risk data in the issues category in the early phase or a large amount of data that did not close in the last phase—the system attempts to alert the manager by vibrating the balloon. After determining the problem and checking the detailed information, the manager may call to discuss the issue with another manager or a developer in a different location.

When checking the project status, timing is important in changing the status of the data. For example, although previously created risk data can be monitored and the problem resolved, it cannot be easily closed. Complete avoidance of risk is usually difficult; therefore, the tool that supports the glimmering interface must evaluate the situation and characteristics of the collected data. However, the tool never judges the project status; it just provides the trigger. An outsourcing manager makes the decision of contacting developers, and because he is often engrossed in numerous tasks, the tool's signal intensity must be sufficiently weak so as not to disturbance him.

4 Tool

Our concept is realized as a web-based tool xDTS; DTS refers to data tracking system and x is the symbol that represents the collected datatype, similar to an asterisk. This tool can manage numerous datatypes to support the developer in software development projects. The typical datatypes, also referred to as tracking units, and their categories (c.f. 3.1) are shown in Table1.

Table 1 Collected datatypes and their categories

Datatype (Tracking Unit)	Category
Change Management	Issues
Defect Management	Verification
Issue Management	Issues
Review Management	Verification
Risk Management	Issues
Schedule Management	Scheduling
Test Management	Verification
Test Schedule Management	Scheduling

To completely understand the status of a project, several facets must be evaluated. For example, a bar chart is used to check the schedule of a project. However, it shows only degrees of progression. If review data is absent, they may not review at all and we may need the extra time for rework. This tool automatically calculates the current status of the project by using those data from each category.

In this tool, the glimmering interface representation of a project is provided by colour and vibration. Fig. 2 (a) shows the top-level view of the several projects in which the user is currently involved. If an unusual situation is detected, the user can click the project symbol and several aspects of the project will be displayed (Fig. 2 (b)). The left bar displays the basic status of each datatype, and the right bar shows the desirable rate of close status calculated from the planned project period. An additional project status alert method can be compared to the indicators on an automobile dashboard (Fig. 2 (c)). In this diagram, the change rates of closed status reports. The speedometer only shows the current rate of the closed status and the upper three boxes represent the change rates of closed status reports. R signifies reverse; that is, the number of report data increases and that of new closed reports decreases. N signifies that the change rate of close is neutral (no change) and D indicates that the change rate is in drive (increases). Fig. 2 (d) shows the detailed history of cumultive number of reports. each curve shows the status of reports, however, (c) is a more intuitive representation and uses the manager's awareness as a trigger.

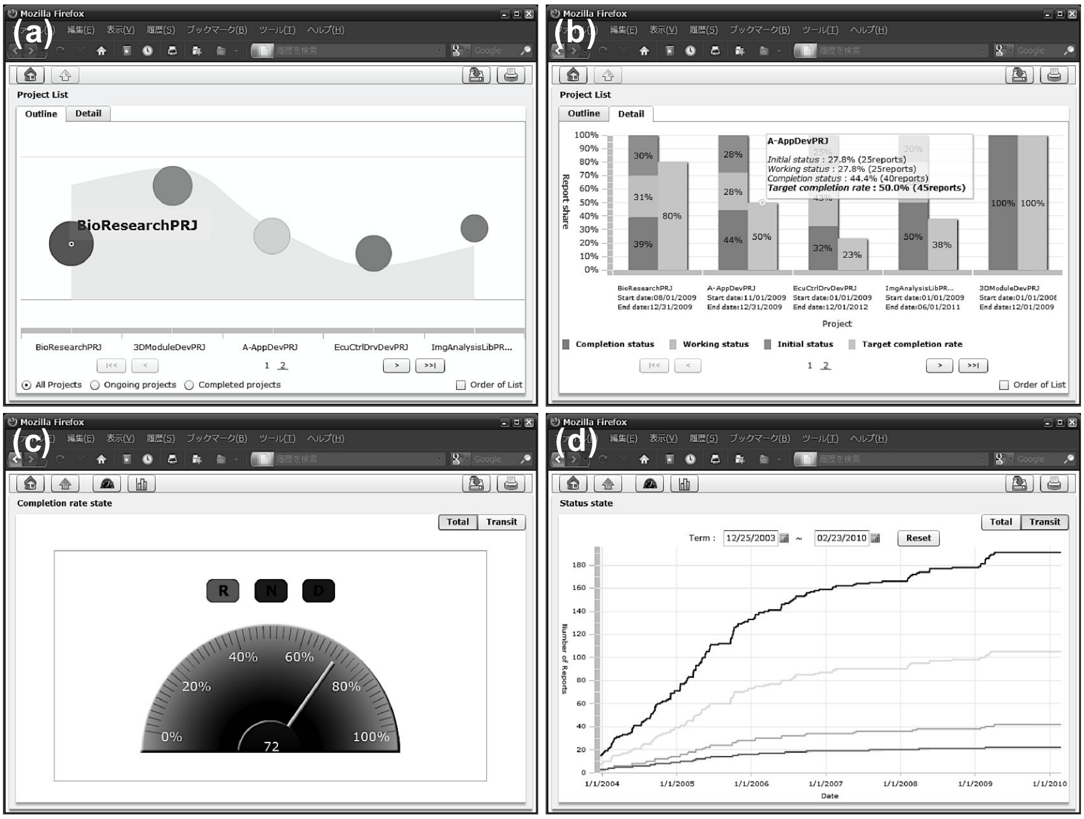


Fig. 2 Various figures offer project awareness to the manager.

Offshore developers, not managers, use the tool for their development activities. By using the basic information included in the report, such as status and category of data, the tool automatically creates a managerial view that supports the glimmering interface. Certainly, the automation does not always deliver the caution reflected in an actual situation; however, it does provide the manager with the appropriate trigger to communicate with the developer, thus offering a quiet solution to the common problem.

The glimmering interface alerts the manager in a subtle way. Because a project does not change rapidly for several days and the manager cannot continue focusing on the project, this tool offers an appropriate user interface. In addition, xDTS displays multiple project views, enabling the manager to observe more than one projects simultaneously.

5 Related Work

In the field of computer–human interaction (CHI), situation awareness (SA) is an important topic ([5-8]). SA originates from the study of a mission's critical system, such as that for an aviation or nuclear plant, and it provides the concept, method and metrics of observation depending on the circumstances. It was previously defined as an element in the environment within a volume of time and space; the comprehension of their meaning and the projection of their status in the near future [5], [6]. SA is used as a technique for alerting the user about a complicated situation and displays strong indicators for avoiding critical conditions. The consideration of human factor in SA is also important in our glimmering interface; however, software development does not present the same critical situations encountered by a pilot or nuclear plant operator. Only the method of notification is the basis of our method. However, the SA also has three levels: perception (Level 1), comprehension (Level 2) and projection (Level 3). We handle only perception; other levels depend on the user.

The association with ubiquitous computing presents an additional CHI issue: ambient information system, which is the concept of a calmer and more environmentally integrated approach for displaying information. This system has three types [9]: ambient display, peripheral display and notification system. We do not currently use the special equipment for purposes other than other usual display; therefore, the relating type in an ambient information system is the notification system. A previous study discusses the attention cost [10], and according to their classification our glimmering interface is in low interrutpion and low comprehension position. The level of reaction is the middle position because it depends on the manager’s degree of expertise.

Related studies originate from Weiser’s studies [11], [12, 13], which insist that the tool has to be transparent and calm. In our glimmering interface, information related to the project progress is automatically extracted from the designer’s activity; it does not stand out, unobtrusively displaying a project’s current situation.

In the software development domain, several examples similar to our study have been documented. One of these studies focuses on artefacts and the activities are explicitly shown in the Aguar visualization system [14]. However, the situation at an early stage of the project’s lifecycle cannot be captured by this method. Moreover, the result is shown by using an excellent display method; thus, it is far the from transparency that Wiser claims. On these points, the system differs from our method.

6 Conclusion

In this paper, We propose a mechanism that extracts useful data that originate from a system used by a software developer in his daily activities. This tool subtly displays current situations of projects for an outsourcing manager. Because a manager is involved in many tasks simultaneously, the glimmering interface is designed to provide information in a manner that does not interrupt his work. He can concentrate on his job and at the same time check the progress of the outsourced project.

If the developer performs his job casually without using the tool needed for his activity, the base data is not obtained, and the manager is not timely informed about the status of the project. However, information that indicates no progress in a project has its own meaning; that is, a problem exists in the project itself or the software is developed in an unusual way. With our tool, the manager has an opportunity to discuss the project with the developer on the offshore side.

Our proposed approach can be used with the existing progress report prepared on the vendor side with the glimmering interface’s endorsement. This proposal does not define the manager’s behaviour after receiving the notification about the project’s progress. This action depends on the outsourcing manager, who must possess sufficient experience about project management to enhance his capability of intuition.

7 Literature

1. Aron, R. and J.V. Singh, *Getting offshoring right.* Harv Bus Rev, 2005. **83**(12): p. 135-43, 154.
2. Thissen, M.R., et al., *Communication tools for distributed software development teams*, in *Proceedings of the 2007 ACM SIGMIS CPR conference on Computer personnel research: The global information technology workforce*. 2007, ACM: St. Louis, Missouri, USA. p. 28-35.
3. Panjer, L.D., D. Damian, and M.-A. Storey, *Cooperation and coordination concerns in a distributed software development project*, in *Proceedings of the 2008 international workshop on Cooperative and human aspects of software engineering*. 2008, ACM: Leipzig, Germany. p. 77-80.
4. Osono, E., N. Shimizu, and H. Takeuchi, *Extreme Toyota : radical contradictions that drive success at the world's best manufacturer*. 2008, Hoboken, N.J.: John Wiley & Sons. xiv, 306 p.
5. Endsley, M., B. Bolte, and D. Jones, *Designing for situation awareness: an approach to user-centered design*. 2003: London; New York: Taylor Francis.
6. Endsley, M.R. *Situation awareness global assessment technique (SAGAT)*. in *Aerospace and Electronics Conference, 1988. NAECON 1988., Proceedings of the IEEE 1988 National*. 1988.
7. Endsley, M.R. and D.J. Garland, *Situation awareness: analysis and measurement*. 2000: CRC.
8. Gutwin, C., R. Penner, and K. Schneider, *Group awareness in distributed software development*, in *Proceedings of the 2004 ACM conference on Computer supported cooperative work*. 2004, ACM: Chicago, Illinois, USA. p. 72-81.
9. Pousman, Z. and J. Stasko, *A taxonomy of ambient information systems: four patterns of design*, in *Proceedings of the working conference on Advanced visual interfaces*. 2006, ACM: Venezia, Italy. p. 67-74.
10. McCrickard, D.S. and C.M. Chewar, *Attuning notification design to user goals and attention costs.* Commun. ACM, 2003. **46**(3): p. 67-72.
11. Weiser, M., *The world is not a desktop.* interactions, 1994. **1**(1): p. 7-8.
12. Weiser, M. and J.S. Brown, *Designing calm technology.* PowerGrid Journal, 1996. **1**(1): p. 75-85.
13. Weiser, M. and J.S. Brown, *THE COMING AGE OF CALM TECHNOLOGY.* 1996.
14. Froehlich, J. and P. Dourish, *Unifying Artifacts and Activities in a Visual Tool for Distributed Software Development Teams*, in *Proceedings of the 26th International Conference on Software Engineering*. 2004, IEEE Computer Society. p. 387-396.

8 Author CVs

Masao Ito

Mr. Masao Ito is the CEO of Nil Software Corp, a Tokyo-based software tool development company. He has a long career in Integrated Process Support Environment (IPSE) construction and consulting. He is a board member of Software Engineers Association of Japan (SEA) and a core member of SEA-SPIN. Currently, he is involved in research activity of Volume CAD application development as a visiting scientist of RIKEN and the National Institution of Radiological Sciences (NIRS). In addition, he serves as the CEO of the venture company VCAD Solutions, Co., LTD.

The future of SPI Knowledge and Networking in Europe – A Vision

R. Messnarz[1], M. Biro[2], S. Koinig[1], M. Reiner[3], D. Ekert[1]
[1]I.S.C.N., Austria & Ireland, {rmess, dekert, skoinig}@iscn.com,
[2]Dennis Gabor College, Hungary, biro@gdf.hu
[3]University of Applied Sciences Krems, Austria, michael.reiner@fh-krems.ac.at

Abstract

Authors of this paper have been founders of the EuroSPI (1994 – now, www.eurospi.net) network with the first networking of SPI strategies published at CON'93 conference. They were also founders of the idea and establishment of a Europe wide certification network ECQA (www.ecqa.org) in 2005 (strategy development 2005 – 2007, online systems set up 2008 – 2009, Europe wide roll out since 2009).

In a think tank and network of leading SPI experts we have developed the idea of a future European knowledge networking strategy and how the existing SPI paradigms will shift into a new SPI world applying new principles for collaboration, networking, and using new media which became available in the last 3-4 years.

This vision will then impact about how we collaborate and implement SPI in the future.

Keywords

Process Improvement, Networked Organizations, Knowledge Strategies

Reference

O'Connor R. V., Pries-Heje J., Messnarz R. (eds.): Systems, Software and Services Process Improvement. 18th European Conference, EuroSPI 2011, Roskilde, Denmark, June 27-29, 2011. In: Communications in Computer and Information Science Series CCIS, Vol. 172, June 2011.

Collaborative Business Process Discovery: Method and Case Study

Paula Ventura Martins[1], Marielba Zacarias[1], Nuno Castela[2], and Paulo Dias[2]

[1] CIEO - Research Centre for Spatial and Organizational Dynamics,
University of Algarve, Campus de Gambelas, Faro, Portugal
{pventura, mzacaria}@ualg.pt
http://www.cieo.ualg.pt
[2] Escola Superior de Tecnologia IPCB, Castelo Branco, Portugal
{ncastela, paulo}@ipcb.pt

Abstract

Information flows across the organization are complex and procedures employed to understand, share and control organizational knowledge and experiences should be properly supported by collaborative environments. Nevertheless, few collaborative methodologies had been proposed to describe and evolve business processes. Existing tools don't provide the right methods for business processes discovery, modelling, monitoring and improvement. In the future, business processes models should be the result of cross-team and cross-departmental collaboration, with involved business people sharing their personal knowledge and formalizing it. This paper focuses on case study applying a collaborative process discovery method. Business Alignment Methodology (BAM) is a methodology that provides guidance about how organizational practices and knowledge are gathered to contribute for business process improvement against current BPM approaches.

Keywords

Business Process, Business Process Discovery, Collaborative Work, Modelling, Methodology

1 Introduction

The business world is continuously changing and is now aware of the need to improve interactions employee-employee and employee-client. Now organizations feel the urgent need to share, communicate and improve business processes descriptions that uncover such interactions. Information flows across the organization are complex and procedures employed to understand, share and control organizational knowledge and experiences should be properly supported by collaborative environments [1].

An adequate supporting system requires attention to continuous changes and business process descriptions should emerge and evolve based on knowledge sharing within an organization [2]. The time that senior managers defined and controlled business process models is coming to an end. In the future, business processes models should be the result of cross-team, cross-departmental and cross-organizational collaborations, with all the involved operational actors sharing their personal knowledge and formalizing it.

There is a vast literature about principles and theories behind collaborative processes and tools [3]. However, few collaborative tools had been specifically proposed to describe and evolve business processes. Existing tools don't provide the right methods for business processes discovery, modelling, monitoring and improvement. They don't allow gathering current experience and knowledge for business processes discovery and improvement purposes. They don't support the underlying collaboration mechanisms of operational actors that enable them to cope with changing contexts or react to existing problems. These are the main reasons for high rates of failure and disappointment in many approaches, methods, and techniques used for Business Processes Management (BPM).

There are several problems associated to current BPM approaches, such as: (1) existing BPM approaches are based on the top-down paradigm; (2) knowledge acquisition and modelling are time consuming activities; (3) existing BPM approaches don't provide means to react to change; (4) existing BPM mechanism don't allow identifying business practices which are diverging from its base business process description.

Business Alignment Methodology (BAM) is a methodology for business process improvement that sets out principles and strategies for improving the quality of business processes. This methodology provides guidance about how organizational knowledge and work practices are gathered and discussed to improve business process. BAM encompasses the following phases:(1) Business Process Discovery, (2) Business Process Supervision and (3) Business Process Analysis and Improvement.
A key driver of BAM, concerning business process improvement, is the ability to facilitate the alignment of business processes and daily work practices. In this regard, this paper focuses on the process discovery phase, particularly on the collaboration processes and tools that allow integrating local information about work practices into coherent and sound process definitions. A brief discussion of existing approaches for business process discovery [4] allows identifying weaknesses and proposing improvements included in this methodology. One of the contributions of this paper is to present a new collaborative tool to support the process discovery phase through the annotation and discussion of business process descriptions among all operational actors somehow related with such processes.

The remainder of this paper is organized as follows; section 2 presents the state of the art about business process discovery methods. Section 3 briefly describes the phases of the proposed methodology, emphasizing on the business process discovery phase. Section 4 presents the results of the process discovery phase in a case study performed in a Portuguese organization. Finally, Section 5 concludes and introduces future work.

2 Business Process Discovery Methods

Traditional Methods to discover business process requirements include: an initial meeting, interviews, questionnaires, observation and ethnography. However, these traditional methods are usually costly, take a great amount of time to complete and introduce a high percentage of human error [5]. There are several reasons why traditional BPM approaches do not achieve expected goals:

- Top-down approach that provides a broad organizational perspective, however with lack of detail.
- Different people have different viewpoints and it is difficult to achieve shared process definitions.
- Organizations structures, rules and practices are constantly changing, it is important to provide mechanisms to align business process descriptions with actual structures, rules and practices.

Ethnography represents a form of descriptive observation where researchers embed themselves into a work [6]. Within this field, a number of strengths and weaknesses have been identified. The strength of these insights is that they reflect actual work practices. Observation also uncovers features of the work that are not obvious to participants because they are deeply embedded in what they do on a day-to-day basis. The weaknesses are due to the fact that the wealth of information that is contained within an ethnographer's notes is highly unstructured and personalized.

Recently, several authors have proposed automated process discovery techniques [5, 7]. Nevertheless, these techniques are restricted to records created by Process-Aware Information Systems (PAIS), which are not always available.

From our perspective, Business Discovery (BD) requires hybrid techniques to detect process problems and the root causes for these problems [5]. Interviews allow gathering viewpoints from selected key actors, while collaborative methods allow discussing such viewpoints among a larger number of actors enabling asynchronous discussions that reduce the need of face-to-face discussions. Regarding business processes, main benefits of collaborative tools are: uncovering several viewpoints of the same activities, an enhanced visibility of business processes to operational actors and deeper analysis of business processes.

It is important to emphasize the difficulties common to ethnography and collaborative methods. In particular, when it comes to bridging the gap between business and IT view on business processes, results from these methods may have to be processed and structured to develop more accurate business models. We consider that a good collaborative tool that provides structured data of business processes for IT engineers should allow to discover, share, analyse and improve business process descriptions from non-structured and personalized information. Whereas the proposed approach allows non-structured annotations and discussions as well as personalized information, it provides the means to enable accurate, precise and structured representations of business processes.

3 Methodology

It is argued that existing business process (BP) methodologies do not offer the necessary flexibility or agility that new approaches require. Process-centric approaches tend to emphasize process (workflow, decision, information, activities) as the dominant dimension [8], but an activity-agent-product centric approach must also capture aspects about interactions between human, activity and informational components. Recent research in BPM pays more attention to flexibility as a way of coping with the unpredictability of business processes [9, 10]. Based on this new context, BAM methodology represents a multidisciplinary approach that allow business analysts to improve business processes discovery, monitoring and analysis, paying attention not only to process but to product, information and human dimensions through work practices.

Although business processes do involve different actor perceptions, this flexibility is important to adjust business processes in response to changes in the organization, in order to reach a global view of the real organization. BAM approach addresses the impact of individual knowledge, collaboration and knowledge sharing from a business perspective by examining actions at individual, group and organizational levels. To assists business analysts in creating, sharing and learning business process, the methodology proposes a two-dimensional approach. The two dimensions, Practice and Process, will ensure the proper structure to articulate individual, group and organizational knowledge with business analyst knowledge acquisition. The Practice dimension explores day-to-day work based on practitioners descriptions. The methodology proposes to assist organizations in its efforts to assess and manage problematic situations of specific daily practices, and develop and implement solutions that help manage these problems. The Practice dimension covers real information needed to systematically support or reject many of decisions about the business process models. In the Process dimension, business analysts based on information of the Practice dimension formulate business process reviews and iterative business process improvement efforts.

The present approach integrates collaborative methods into a methodology that allow discover and improve business descriptions. There are three phases outlined in the methodology: (1) *Business Process Discovery*, (2*) Business Process Supervision* and (3) *Business Process Assessment and Improvement*. *Business Process Discovery* phase main goal is an initial process specification through interviews and collaborative methods. *Business Process Supervision* phase involves provides assurance that daily practices follow the base business process model or reveals the need to take corrective actions because real activities performed by operational actors are different from those specified in the business process model. In *Business Assessment and Improvement* phase, initially, business analyst and managers analyse real business performance and produce assessments focused on business processes issues, respectively. After that, the results gathered during assessments enable improvements and consistent refinements of the base business process model creating a new version. Each phase includes specific working methods and goals that are presented in the following subsections.

Since the focus of this paper is on business process discovery methods, we only describe in detail the business discovery phase. The other two phases only have a brief description of methods and goals.

3.1 Phase I - Business Process Discovery

The first goal of a *Business Process Discovery* (BPD) is just to get personal descriptions of business processes. Operational actors have knowledge of their actions, nevertheless their knowledge is personal and to a certain extent tacit, hence it is hard to formalize. In fact, most organizations simply do not know their end-to-end processes accurately or in detail, since their process knowledge is tacit and decentralized [11].On the other hand, organizational knowledge crosses functional divisions and outside the organization (clients, suppliers). Therefore, organizational processes embody specific accumulated knowledge that is not confined to particular individuals or groups. This knowledge, which is explicit, must be transmitted but this is not enough. Business process modelling approaches should consider the effect of continuous business processes improvement as a reaction to fast-changing environments in the business world.

BPD phase aims at developing an organizational profile of people, activities, technology, and information in order to understand business processes. This phase includes two main sub-phases: (1) *Learning (Eliciting) Business* (LB) and (2) *Modelling Business* (MB). *Learning Business* is knowledge acquisition, a set of tasks that can be the most time consuming portion of BPD. Some of the major considerations include the choice of the methods to use for acquiring specific types of knowledge [12]. In order to serve this purpose, the methodology will not consist only on translating natural language descriptions of business processes but also includes guidance in the form of instructions, templates and examples. Modelling Business consists in an intensive interaction between actors of the two dimensions, operational actors and business analysts.

LEARNING BUSINESS Initially, the sub-phase *Learning Business* includes interviews where the business analyst interacts with operational actors as key to the success of business discovery. At an organizational level, the methodology proposes to assist organizations in their effort to assess and manage problematic situations based on daily actions and implement solutions related to these problems. The BAM methodology was designed for explicitly addressing the social dynamics of business process specification. It is based on social interactions as proposed in a two-dimensional space. The Practice dimension covers information needed to systematically support or reject many process decisions based on the result of daily experiences. The exact process representation concerns activities, resources, decision points and work flows (topology). In the Process dimension, business analysts capture best practices from the practice dimension that leads to business process reviews and improvement. The dynamic interplay between these two dimensions (practice and process dimensions) shows the synergy between activities performed by key operational actors and activities described by business analysts involved in BPD. Table 1 outlines the guidelines regarding method, actors, activities, milestones and goal.

Table 1: Learning Business Process guidelines

Guideline	Description
Goal	Find out organization's needs regarding business discovery, analyse and understand problems with senior managers, and define an initial business process view.
Purpose	The business analyst finds current process practices to include in an initial business process description.
Method	Initial meetings with senior managers mainly covered questions about the organization's mission and goals. Also important was information retrieved through interviews and questionnaires. The organization also delivered 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.
Actors	Business Analyst and operational actors.
Activities	Kick-off meeting, interviews and questionnaires. Business process specification.
Milestones/Outputs	Organization's goals list and initial. Business process specification.

MODELLING BUSINESS After an initial business process definition, business analyst starts modelling business processes. The sub-phase *Modelling Business*, from the knowledge management point of view, involves several stakeholders (business analyst, process owner, organizational unit responsible and operational actors) into four interrelated activities: (1) model construction; (2) model revision and evaluation and (3) model approval (table 2). Model construction is an activity that transforms tacit and implicit knowledge and specific contextual situations into more structured and documented forms. The model revision and evaluation activity results from a critical review of existing knowledge about business processes.

Table 2: Modelling Business Process guidelines

Guideline	Description
Goal	Support business process model discussions and negotiations to improve an initial proposal.
Purpose	Obtain a business process model accepted by operational actors.
Method	The business analyst constructs and initial business process model, which triggers business process model changes based on annotations of involved operational actors. The phase ends the approval of a final version of the business process model.
Actors	Business Analyst, Organizational Unit Responsible and operational actors.
Activities	Model construction, model revision and evaluation model approval.
Milestones/Outputs	Business process models.

All participants may present alternative proposals that result from associating different facts and new meanings. The updating process provides support to business process model discussions and negotiations to correct represented activities and other aspects. Finally, the model approval activity concludes the interaction process and collaboration among the parties involved in a business process model specification by approving or rejecting the model.

The model revision and evaluation activity will use the annotation as a mechanism to capture the updated proposals made by business actors. Annotations are used mainly to make proposals to correct

the model (corrective maintenance), to capture changes in action or interaction contexts (adaptive maintenance), to make free comments that could anticipate problems (preventive maintenance) and promote continuous process improvement (perfective maintenance). If business actors detect any misalignment between the model and their knowledge of activity current states-of-affairs, they can make a textual or graphic annotation with the correction proposal.
Booth dimensions (practice and process) involve the same actors (business analyst, organizational unit responsible and operational actors). Organizational unit responsible has modelling responsibility of the practices within his organizational unit. At process dimension, modelling responsibilities should be attributed to process owners. All actors have annotation and reviewing rights over practices of their organizational units. Nevertheless, they also have rights to annotate and review processes related with their work.

3.2 Phase II — Business Process Supervision

In the *Business Process Supervision* (BPS) phase, formal control mechanisms are designed in order to ensure that operational actors carried out real business activities as described by business models. Control mechanisms consist of three main activities: (1) compare real business activities with base business models (2) annotations/reviews and (3) identify new business descriptions.
Each annotation/review should be adding or validating new features to improve the business process. During this phase, the organizational unit responsible and operational actors will analyse improvements against oldest daily practices. In this context, operational actors will become more and more confident on suggestions of the business analyst. From iteration to iteration, confidence will increase and results will start appearing.
The milestone that marks the end of this phase and the beginning of the *Business Assessment and Improvement* phase is crossed when business analysts and operational actors agree that: business process model describe the detailed behaviour that address real needs, major problems have been solved, business process practices provides some useful value to the organization and these practices are stable enough to implement a new and improved business process version.

3.3 Phase III — Business Process Assessment and Improvement

Business Process Assessment is a mean for organizations to identify their strengths, weakness, existing improvement activities and key areas for improvement. It enables organizations to determine the current state of their business processes and to develop improved models. In the begin of the Business Process Assessment and Improvement (BPAI) phase, the business analyst analyse change proposals and through a comparison between base business process models and proposed changes, a new set of models is build to correct work that is not proceeding well, by showing where adjustments need to be made. In the end, the results gathered during assessments enable improvements and consistent refinements in order to produce an improved set of business process models. This phase ends when all the involved actors agree that: the objectives set during BPD (and modified throughout the second phase) have been met; and especially if all participants are satisfied with the new business process model version. BAM directs attention to organization's needs for communication, coordination and collaboration within and between operational actors and business analysts. The methodology is about how business processes are represented and how business analysts acquire and use knowledge to improve business process. BAM methodology includes model and control activities that intend to develop and implement organizational business processes (process dimension). Nevertheless, control activities also include monitoring and tracking of day-to-day work (practice dimension).

4 Case Study

This section summarizes the results of the BPD within a Portuguese organization. The objective was to uncover, discuss and improve Financial, Human Resource and Project Management processes in order to assure a proper implementation of SAP. The underlying goals were to (1) discuss current work practices in order to improve them; (2) synchronize business process descriptions with pretended work practices and (3) identify and/or cross-department flows. The organization is characterized by a highly hierarchical structure where procedures within each department, service or division

are explicit and registered in manuals but cross-departmental processes have never been defined and are executed in an ad-hoc fashion, resulting in inefficiencies, errors and great deal of confusion among the employees.
A core team of two employees were given the roles of business analysts with the responsibility of uncovering the business processes. Their selections stemmed from the fact that they had experience with business analysis tasks and were already familiar with traditional BPM approaches, as well as their limitations. The team decided to apply the BPD phase of the BAM methodology. The team started interacting with the head of the units and subunits but extended these interactions with other operational actors including selected members of the units at different hierarchical levels and two directors. Results are described in the following sections according to each sub-phase of the BPD phase.

4.1 Practice Dimension

LEARNING BUSINESS In the Learning Business Phase, the core team interviewed the Head of Financial, Human Resource and Project Management Departments. These interviews resulted in eliciting intra-departmental procedures and a list of problems and improvement areas according to the viewpoints of the interviewees. The information gathered, in this phase, encompassed list of activities, as well as their inputs, outputs and tools used. It was also gathered the names of the individuals responsible for performing such activities, as well as the names of other participants, descriptions of problems and improvement areas. All this information was collected in free-form text and organized in tables. In both cases, descriptions reflected local vocabularies and terms used to designate daily actions, information items and tools. The collaborative method used involved face-to-face interactions.

MODELLING BUSINESS Initial diagrams reflected intra-departmental work practices. Descriptions of work practice were represented with diagrams using the Business Process Modelling Notation (BPMN) [13]. Since practices are associated to specific individuals, actors were identified using actual names rather than roles, and activities included the tools employed by each actor (paper, pencil, calculators, excel sheets, etc.), diagrams were adjusted to fit this type of usage. The tools used in each activity were depicted using representative icons, which enriched the diagrams and allowed an immediate recognition from participants of the practices represented in such diagrams.

4.2 Process Dimension

LEARNING BUSINESS Weekly meetings enabled to achieve shared meaning around vocabularies and terms, to discuss work practices, select the best and discard the more problematic ones, they also enabled defining inter-department flows. Initial process descriptions were depicted in tabular forms reflecting activity inputs, outputs and steps, list of participating units, and in some cases, decision tables. The business analysts worked together with the other operational actors to elicit the required knowledge and depict the resulting diagrams.

MODELLING BUSINESS The information gathered from the practice dimension facilitated the discussion and depiction of process diagrams. The process dimension was supported by weekly face-to-face meetings complemented with the usage of a modelling tool. Meetings were integrated by the head of the organizational units and sub-units related to Financial, Human Resource and Project Management processes, and by a representative of unit employees. These representatives had the responsibility of offering the execution view of each activity. Two directors, one representing technical units and the other administrative units, also integrated the meetings in order to provide views of higher hierarchical levels, strategic and more comprehensive issues.
The modelling tool [14] not only was employed to depict BPMN diagrams of the uncovered process, but also to discuss the resulting diagrams, thus it also contributed to the Learning Business subphase. The tool enabled actors to continue discussions during the week, accelerating the process of uncovering undefined and changing processes, reducing the time required for regular meetings and allowing the participant actors to fit the discussions according their own schedules.
It is noteworthy that throughout the BPD, the business analysts worked together with the other operational actors to elicit the required knowledge and depict the resulting diagrams. The process owners

performed the modelling task rather than the business analysts, who are limited themselves to the role of facilitators.

5 Conclusions and Future Work

This paper described a methodology for Business Process Improvement with focus on Business Process Discovery phase, particularly on collaboration methods. Incorporation of multiple viewpoints into the business modelling process also contributed to learn and change business processes specifications. Embedding an annotation-review-approval process, the modelling tool improved participation, providing a structured support to disseminate business process models, express opinions and negotiate perspectives.
The proposed methodology is the result of several other case studies for business process improvement. However, this case study was a pilot experience regarding collaborative process modelling supported by the modelling tool. Preliminary results show that it stimulated the involvement of all actors (business analyst, organizational units responsible and operational actors) in validating (and thus updating) the enterprise model, aligning it with the reality in an interactive and shared way. So, annotations and reviews fit the requirement of being a suited mechanism to generate, debate and insight about the organization in order to propose changes and improve the organizational business model.
In the future, we will focus on incorporating new functional requirements in the modelling tool in order to: (1) model daily actions (practice model) and (2) align practice model (practice dimension) with business process models (process dimension). Future work will also include new efforts to gather empirical data on its use that will help refine the methodology.

6 Literature

1. Althoff, K.-D., Bomarius, F., Tautz, C.: Knowledge Management for Building Learning Software Organizations. Information Systems Frontiers 2, 349 (2000)
2. Mengoni, M., Graziosi, S., Mandolini, M., Peruzzini, M.: A knowledge-based workflow to dynamically manage human interaction in extended enterprise. International Journal on Interactive Design and Manufacturing 5, 1 (2011)
3. Huettner, B., Brown, M.K., James-Tanny, C.: Managing Virtual Teams: Getting the Most from Wikis, Blogs and Other Collaborative Tools. Jones & Bartlett Publishers (2007)
4. Ghose, A., Koliadis, G., Chueng, A.: Rapid Business Process Discovery (R-BPD). In: Parent, C., Schewe, K.-D., Storey, V., Thalheim, B. (eds.) Conceptual Modeling - ER 2007, vol. 4801, pp. 391. Springer Berlin / Heidelberg (2007)
5. Houck, C.: Breaking through with Business Process Discovery. Business Intelligence Journal 12, 30-39 (2007)
6. Ritter, J., Lyons, J., Swindler, S.: Large-scale coordination: developing a framework to evaluate socio-technical and collaborative issues. Cognition, Technology & Work 9, 33 (2007)
7. Aalst, W.M.P.v.d., Reijers, H.A., Weijters, A.J.M.M., Dongen, B.F.v., Medeiros, A.K.A.d., Song, M., Verbeek, H.M.W.: Business process mining: An industrial application. Inf. Syst. 32, 713-732 (2007)
8. Hollingaworth, D., Services, F.: The Workflow Reference Model: 10 Years On. In: Fischer, L. (ed.) Workflow Handboook 2004, pp. 295-312. Future Strategies Inc., FL USA (2004)
9. Reichert, M., Dadam, P., Jurisch, M., Kreher, U., Goser, K.: Architectural Design of Flexible Process Management Technology. In: Conference Architectural Design of Flexible Process Management Technology, pp. 26-28. (2008)
10. Mutschler, B., Weber, B., Reichert, M.: Workflow management versus case handling: results from a controlled software experiment. Proceedings of the 2008 ACM symposium on Applied computing, pp. 82-89. ACM, Fortaleza, Ceara, Brazil (2008)
11. Verner, L.: BPM: The Promise and the Challenge. Queue 2, 82-91 (2004)
12. Koschmider, A., Song, M., Reijers, H.A.: Social Software for Business Process Modeling. Journal of Information Technology 25, 308-322 (2010)
13. White, S.A., Miers, D.: BPMN Modeling and Reference Guide. Future Strategies Inc., FL, USA (2008)
14. Castela, N., Zacarias, M., Tribolet, J.: Business Process Model Dynamic Updating. In: Quintela Varajão, J.E., Cruz-Cunha, M.M., Putnik, G.D., Trigo, A. (eds.) ENTERprise Information Systems, vol. 110, pp. 158. Springer Berlin Heidelberg (2010)

7 Author CVs

Paula Ventura Martins

Paula Ventura Martins has a Master's degree in Computer Science and Engineering from FCT/UNL (2000) and a PhD degree in Computer Science and Engineering from IST/UTL (2008). She is an assistant professor of the Electronics and Computers Engineering Department of Faculty of Sciences and Technology, University of Algarve, and she is also a member of Research Centre for Spatial and Organizational Dynamics (CIEO). Her current interests are software process improvement, software development processes, business processes, modelling languages, and project management.

Marielba Zacarias

Marielba Zacarias is a Computer Engineering (1982) with a MSc degree in Systems Engineering (1996), and with a PhD in Informatics and Computers Engineering (2008). She is an assistant professor of Electronics and Informatics Engineering Department and member of Research Centre for Spatial and Organizational Dynamics (CIEO), at the Algarve University. She has worked mainly in organizational modelling and ontologies, focusing particularly in modelling organizational contexts.

Nuno Castela

Nuno Castela is currently an Enterprise Engineering and Information Systems PhD student at Instituto Superior Técnico, Technical University of Lisbon (IST/UTL), expecting to finish in 2011. He holds a Computer Engineering MSc degree from IST/UTL and an Electrical Engineering BSc degree from University of Coimbra. He is an Assistant Professor at Polytechnic Institute of Castelo Branco and an Invited Researcher at Centre for Organizational Design & Engineering (CODE/INESC-INOVAÇÃO).

Paulo Dias

Paulo Dias is an Invited Teaching Assistant at Polytechnic Institute of Castelo Branco. He holds an Information Technology Engineer Degree and an Information Systems and Technology MSc degree, from the University of Beira Interior, Covilhã, Portugal. Is present research interests are: Business Process Modelling, Workgroup support systems and Information systems.

Product and Risk Based Quality Models

Alexander Poth, Michael Winter

Abstract

Quality models are a systematic abstraction of quality factors. Quality factors are the most important quality drivers in a product. By abstraction it makes visible the core elements of quality factors in a product. Once quality factors become apparent they can be used to steer quality and help improving product development through the product life cycle. The core elements of the quality model presented in this paper are based on a product and quality risk based views, which are defined in the model. The two views are used to set up a structured and systematic value based approach and they help improving the quality factors. The presented quality model also includes quality planning and controlling features. The benefit of the presented model is to have a praxis oriented and systematically value driven quality model.

Furthermore it demonstrates how to integrate this quality model in projects and organizations with a SPICE-conform quality process and a SCRUM oriented product development context. Both are selected as representatives for more or less process oriented development environment.

Keywords

Quality model, quality management, risk management, quality measures, quality planning, quality controlling, value based quality management, software process improvement, key performance indicators, SPICE, SCRUM, balanced score card

1 Introduction into quality models

Quality models are a systematic abstraction of quality factors. Furthermore quality models have the objective to be useful to set up an understanding of what is quality in the specific or defined context. They set up ways to analyze the present stage of quality and give hints or show ways to improve the quality.

Different approaches for quality models exists and are well described. They can be grouped by their approach:

- generic ISO/IEC 9126 quality models [2] or [6] (superseded by the newer standard ISO/IEC 25000),
- a product based approach [1],
- an activity based approach [5],
- a metric based approach [7],
- an expert based approach [8] and
- an example for a method for organizational and domain specific refinement of quality models [10].

Each of these approaches above has potential for further improvement in

- systematic refinement of the quality model to improve the product
- the value case or the cost/benefit relation of the activities or tasks which are derived from the quality model
- concepts for the implementation of the quality model in companies with focus on rollout and transition in an organization.

The present approach of a quality model tackles these issues by combining different quality model approaches in one more holistic and applicable quality model.

2 A product based and quality risk based approach

The key elements of the product and quality risk based approach are presented in figure 1. It starts with a forward chain, orientated on [4], to look at the actual value, based on *quality risk aspects* to derive a *quality focus value*. The *quality risk aspects* and the *quality focus value* are deduced of the risk based testing concept of [15]. Based on this *quality focus value* the *validation measures* are derived. These *validation measures* are like an individual plan or set of activities for the validation of each function to reach the estimated quality expectations for the product functionality. The effectiveness is controlled by a backward chain which is based on the *product quality indicators*. To handle the product specific context an abstraction is needed to reduce the complexity of the product. A flexible element for abstraction of the product complexity is the *function*. The functions are used to identify mostly independent logical elements on the product view. These independent elements can be steered individual to reach an optimal quality. To be conform to standards and also flexible in selecting appropriate *validation measures* a set of validation measures is defined which makes it possible to define an adequate validation context. The *validation measures* in some cases have to be selected conform to specific standards of the product context.

The *product quality indicators* and the *quality risk aspect* are defined as the mentioned quality factors. The *product quality indicators* are the metrics on which the current and actual quality is checked. The *quality risk aspects* are used to identify the gap between quality expectations at the initial or current state of fulfilling them. The validation measures are the means to fill the gap.

Product A

						foreward chain					backward chain		
Function	Quality-Risk-Aspect1	Quality-Risk-Aspect2	...	Quality-Risk-AspectN	Quality-Focus-Value	Validation-Measure1	Validation-Measure2	Validation-Measure3	...	Validation-MeasureX	Product-Quality-Indicator1	...	Product-Quality-IndicatorY
Function1	low impact	middle impact	...	middle impact	8		x		...		N/A	...	N/A
Function2	high impact	middle impact	...	middle impact	23	x	x	x	...	x	low	...	low
Function3	middle impact	low impact	...	high impact	29	x	x	x	...		low	...	N/A
Function4	low impact	middle impact	...	low impact	12	x			...		N/A	...	middle
Function5	high impact	high impact	...	high impact	9		x	x	...	x	middle	...	low
Function6	middle impact	low impact	...	middle impact	7	x		x	...		low	...	N/A
...	...	...	...	...	...	...	...	...	...	...	..	...	...
FunctionM	high impact	middle impact	...	low impact	18	x			...	x	N/A	...	low

Figure 1: Presentation of a functional refinement of a product, the evaluation of the relevant quality risk aspects for specific validation measure definitions including the monitoring of effectiveness through product quality indicators

2.1 The product functionality aspect

Products are based on functionality. The functionality can be clustered in *functions* at an abstraction level. Based on the definition of a *function* the abstraction level is defined. For example a *function* could be defined as a product element with a stimulus, an action or implementation to realize the *function* and an output as described in [16]. The derived *functions* can be represented in an unstructured set of *functions* or more structured as in feature trees. With a more structured representation it is easier to identify equal or similar elements in a product or in products which are compared. For an organization it is useful to define different levels or a hierarchy of abstraction to cluster the products and their functionality. An example, orientated on [12], which supports a structure to refine the abstraction, could have the following abstract levels: the highest abstraction level will distinguish user and system functionality, the next level will distinguish function groups of a product family and the 3rd distinction level are the functions. In cases in which less abstraction is needed the refinement could be the feature or the requirement level. This detailed level could be interesting when the validation of the requirement is either very expensive or time intensive. It should not be used more often than absolutely necessary.

A benefit of using this quality model is that through the product refinement and identification of the *functions*, at least the representative stakeholders of the development and the quality department get a common understanding of the product.

2.2 The quality risk aspect

The quality risk aspect covers the product quality risks. Typically the product risks are often the same for similar products. If an organization has as an output of similar products the quality risks are similar or equal and differ for example more in the level of a quality risk. Furthermore there are risks which are more or less relevant for all software or systems which have to be considered always.

As described in [11] it is possible for organizations and businesses to identify a specific set of *quality risk aspects* which are on an abstraction level or are generic enough to be applied to all products in the organizational unit. Important at this point is that the *quality risk aspects* are independent. In case they overlap a weighting of *quality risk aspects* has to be introduced. In this case it is important to prevent too much emphasis on a cluster of similar *quality risk aspects* in comparison of the other *quality risk aspects*. An example for generic *quality risk aspect* is maturity of requirements or complexity which is applicable to all products and *functions*. In the organizational context it has to be defined what exactly is covered by a *quality risk aspect* and what are the criteria for ranking them like high impact, middle impact or low impact (compare figure 1). Based on the risks which are identified it is necessary to control the *quality risk aspects* in cycles, because aspects like the maturity of requirements are dynamic and can change during the product development phases.

All *quality risk aspects* of each function are integrated into the *quality focus value*. This *quality focus value* is a management indicator which makes *functions* comparable. But it is important that the derivation method of *functions* and the *quality risk aspects* are the same. Loop sidedness or scattering of the derivation of functions a can be avoided by a smart selection of *quality risk aspects* like complexity which handles internal and external complexity aspects of the *function*. This is a compensation for the divide-and-conquer-method because for example many small functions have less internal complexity but more (complex) interfaces than one big function which has to realize the same set of requirements.

The benefit of this approach in the presented quality model is that with this bottom up approach complex functionalities are broken into sizeable *functions*. For each *function* the risks are summarized and the resources necessary to mitigate the individual risks are transparent. Based on the *quality focus value* management has an instrument to allocate resources in a balanced manner according to *quality risk aspects* to the individual risk.

2.3 The derivation and definition of specific validation measures

Good quality models are compatible with the organizational quality strategy which derives from the strategy of the organization. To guarantee compatibility of the strategy with the product specific quality model it is useful to define a pool of accepted *validation measures*. Measures of this pool are selected by the project responsible for the product. They have to conform to the strategy of the organization and the specific demands of the project. Typical *validation measures* are for example reviews of specification, design, code artefacts etc or static code analyses, defined test stages and/or systematic test coverage.

To each *function validation measures* are assigned based on the specific *quality risk aspects*. Each *validation measure* allocates its necessary resources. By this assignment the specific resource allocation for *validation measures* is directly motivated by the specific *quality risk aspects* of each *function*. After *validation measures* have been in place for some time, there will be know how for each *validation measure* in the organization as to the effectiveness and resources needed. This is a special kind of a cost benefit which comes as a side effect of the quality model which is usually not considering cost benefit factors on a quantitative basis. Instead it uses a qualitative way with relations to risks and the effectiveness of *validation measures* which are based on resources. The presented concept is based on the amount of resources that can be spent to reduce the *quality focus value* compared to the initial value. An exact cost value relation is difficult to achieve. So the presented approach is based on a qualitative argumentation and is not directly quantifying. An example for quantifying could be an approach which uses the *quality focus value* and the *validation measures* to get the contribution margin or the return on investment (RoI). Basically this is not done because the initial effort to analyze this in detail, like for example for a worst case or an average case, is considerable and the added value to the qualitative approach for the product is not clear, because it is not easy to define adequate cases and calculate them with a high acceptance by all stakeholders. However, it is not ruled out completely. If there are management decisions or reporting needs the calculation can be done because the needed aspects can be separated.

An interesting aspect and benefit on this quality model is that aspects of the ISO 9126 and their specific refinement like for example in [10] could be defined as only one specific measure. The content of this measure is to get or derive more detailed information about the quality aspect for one or more *functions* in cases in which the known quality information is not enough. By this abstraction of details it is possible to use this approach as a kind of meta quality model which could be extended or refined by other approaches. The results of the refinement could also be defined as objectives for other measures.

In the context of an iterative development model it is possible to define different quality goals for iterations. In this case the iteration defines specific validation measures for a common measure iteration specific target values.

By the definition of the *validation measures* it is useful to involve the relevant stakeholders, for example the representative persons of the quality and development department. This could be for example the persons in charge for the measures, for confirmation that they have understood the activities and give their commitment to the validation activities.

2.4 The controlling of the validation measures

Effectiveness and efficiency of the *validation measures* is controlled by metrics. These metrics, the *product quality indicators*, are based on the context of the specific *validation measures* which are in our case the product element or function. In a first step measures are needed for the effectiveness of the *function* specific *validation measures* like for example problem reports (an indicator for incorrectness of a product element and rework) or change requests (an indicator for incomplete product elements). These measures indicate that the specific validation package for the function is effective. The next step is to collect data for the efficiency like resource allocations for the implementation of the quality measure. In a further step the optimization to the cost/benefit relation starts to optimize the value for the organization or projects.

When it comes to spending money or allocating resources to quality measures the quality helps to ensure that money and/or resources are spent adequately. Depending on the goals of the organization and project which is responsible for the product the *validation measures* can be systematically optimized to save either cost while maintaining the same quality level or to improve quality

Based on the controlling information which becomes available through the application of the presented approach refinements such as a quality cockpit or a balanced score card on the product level or on an organizational level become a possibility.

3 The approach in practice

This chapter describes implementations in different development environments. One is a process oriented SPICE development organization and the other is a SCRUM context. This shows the flexibility of the approach because the selected examples are used in different development paradigms.

3.1 A process for quality model application

A detailed process for application of the presented quality model is described in [11]. The process is based on the following generic steps:

- Identification of the *quality risk aspects*
- Identification of the *functions*
- Product specific weighting of the *quality risk aspects*
- Evaluation of the *functions* against the quality risk aspects
- Derivation of the *validation measures*
- Tracking and controlling of the realization of the *validation measures* and their results in the *product quality indicators*

Step one and two could be swapped if the quality model is new to an organization. In this case it is easier to start with the identification of the functions and then used the functions to derive and identify the product risks. Furthermore an appropriate set of *product quality indicator*s has to be identified or derived too in this case. For step three, the weighting of the *quality risk aspect* different methods are known as for example a pair-by-pair comparison.

3.2 A concept for a SPICE conform implementation

For an implementation in organisations a tailoring of the quality model for specific organisations or units and domains in which the development projects of products are running is needed. The challenge is to keep the balance between a very specific definition of the quality model in a specific context and its reusability for similar or other contexts. The presented approach solves this issue by using a flexible level of abstraction of the product by the functionality which is applicable in almost all domains and the flexible set of *quality risk aspects* and *product quality indicators*. Furthermore the approach fulfils normative requirements by its openness towards the *validation measures* which are being used. For example SPICE (ISO/IEC 15504) [3] requires that the *validation measures* are associated to processes. In a process environment conforming to SPICE the set of *validation measures* have to fit this requirement by subsets of *validation measures* which conform to the defined and applied processes. Equally important for an implementation in an organisation is the possibility of the refinement of the presented quality model approach which allows a step by step introduction. The presented approach provides this with an initial set up of *functions*, *quality risk aspect* evaluation and the *validation measures*. In a next step these could be refined or enhanced once more information becomes available on the product or the organisational constraints or other details.

A major constraint which has to be managed by an organisation is the training of the employees who have to be acquainted with the usage of quality models. Further constraints are the management of change and the controlling of the implementation progress during a rollout.

Figure 2 shows an example of an embedded system which refines a transmission controller in to functions and features. The functions/features are evaluated on *quality risk aspects* which are mostly motivated by the V-Modell 97 in this example but also extended with the *quality risk aspect* mission relevance. Based on the quality risk evaluation the quality focus value is calculated. The factor depends on the focus of the quality risk aspect at the project. The calculation uses the *quality risk aspects* which are weighed with a factor. The basic formulary for the *quality focus value* (QFV) for a function is:

QFV= QRAfunctionValue_1*QRAfactor_1+…+ QRAvalue_n*QRAfactor_n with QRAfunctionValue :={1=”low impact”|3=”middle impact”|5=”high impact”} and QRAfactor :={0,5…1,5} QRA is Quality Risk Aspect

Based on the *quality focus value* the *validation measures* are derived with a schedule and a resource allocation which is the responsible person. Further the quality target is defined via the measure like the MISRA conformances analysis which motivates the derivations of the standard or the unit-test objective which is a coverage target. The backward chain shows based on the *product quality indicators* the current quality status of each function. The measures can be mapped to processes of the automotive SPICE like the MISRA analysis as a part of the unit verification strategy of ENG.6 (software construction) which is controlled by the SUP.1 (quality assurance) process, which owns the presented quality model approach as part of its project quality assurance strategy.

Transmission Controller A

		1,5	1,5	0,75	0,75		1		foreward chain					backward chain		
Function	**Feature**	**criticality**	**mission relevance**	**internal complexity**	**external complexity**	**…**	**expacted change rate**	**Quality-Focus-Value**	**Review of systemrequirements**	**static code analyse according to MISRA**	**Unittests with >95% C1 coverage**	**…**	**100% traceability from requirements to test**	**open change requests**	**…**	**open bugs**
gear change	neutral	low impact	middle impact	middle impact	middle impact	…	middle impact	8		Mr. C, CW28		…		0	…	0
	foreward	high impact	middle impact	high impact	high impact	…	middle impact	23	Mr. A, CW24	Mr. C, CW28	Mr. A, CW29	…	Mr. C, CW29	2	…	1
	backward	middle impact	low impact	high impact	high impact	…	high impact	29		Mr. C, CW28	Mr. B, CW29	…		3	…	1
gear mode	automatic	low impact	middle impact	low impact	middle impact	…	low impact	12		Mr. C, CW28		…		1	…	0
	manual	high impact	high impact	high impact	high impact	…	high impact	9	Mr. B, CW24	Mr. C, CW28	Mr. B, CW29	…	Mr. C, CW29	5	…	1
diagnosic	rpm	middle impact	middle impact	high impact	low impact	…	middle impact	7	Mr. A, CW25	Mr. C, CW28	Mr. A, CW29	…		2	…	0
	…	…	…	…	…	…	…	…	…	…	…	…	…	..	…	…
	clutch	high impact	middle impact	middle impact	middle impact	…	low impact	18	Mr. C, CW25	Mr. C, CW28		…	Mr. C, CW29	0	…	2

Figure 2: Example product with derived quality measures which are schedule for calendar weeks (CW) and allocated with resources.

An example how to integrate the presented implementation concept of the present quality model approach, as a risk based and preventive quality strategy together with a package of analytic and constructive measures is shown in [9].

3.3 A SCRUM oriented implementation

This chapter describes a project which uses SCRUM. The project involves developing two separate software systems that will eventually be the base of a very large customer benefit program – one software system is being developed in India, the other in Germany. Both systems are interacting with more than 50 other systems or interfaces in order to deliver the business value. Due to the different business cases or functions which are handled with the system different teams for systems integration and user acceptance tests were set up. Three functional teams for earn, manage and spend topics are located on different locations. The team of nine members responsible for spend topics works according to SCRUM. The first important step is to define the right *quality risk aspects* for the context. An important aspect of SCRUM or rather agile project management is to work value and priority driven [14] which needs to map the value idea of agile and the *quality risk aspect*. One way is to define risk values in terms of money to get an equivalent value – which helps calculating the risk costs which are then multiplied with the factor of expected occurrence as shown by VaR [13]. As an alternative to the quantitative, money driven, approach a qualitative driven approach can be used. This requires the definition of a set of comparable units (like in figure 1 for each *quality risk aspect* the low, middle and high impact). The project defined 5 different types for value and risk. In the presented project the context of the systems function is the spend environment in call centres which are dealing with different spend use cases and their activities. *Functions*, stories in SCRUM wording, are activities of use cases. The first *quality risk aspect* is value and is measured in how often the *function* is used during the daily business (<=20%, <=40%, <=60%, <=80% and up to 100%) to make the contribution of its activities for the daily business transparent. The other risks of functions were split into two *quality risk aspects* of which one was the probability of loss occurring impact (not really an impact, impact with a customer invisible inexpensive workaround, impact with customer invisible expensive workaround, impact with a customer visible workaround, show stopper) and a special risk aspect of these company which makes transparent that some functions are important for a very small customer group which generates a very high revenue – which is a high risk if these customers could not be served or in the value view which generates a very high value.

With this set of *quality risk aspects* all relevant value and risk drivers are covered in the *quality focus value* for the presented project context. This approach brings together both the risk based quality and the value driven agile paradigm. Values are transformed in a risk view, via an accepted table of classification units for values and risks. The point behind is: losing high value functions due to defects provoke high quality costs – this leads to high quality risks. Vice versa: quality issues due to the risks are destroying the value of a *function*.

Based on these set of *quality risk aspects* a ranking or prioritization in a value oriented way of SCRUM of the *functions* was easy because adequate risk mitigation delivers more value per *function*. Furthermore the set of quality measures for the handling of the quality risks was derived – which leads to a mandatory and optional (and selectively used measures like pair working versus reviews or early regression testing) set of quality assurance activities. One *product quality indicator* is bugs per SCRUM story or *function*.

3.4 Benefits from using these quality model

Let's start with a formal point: In a SPICE assessment it is evaluated if the processes are adequate for the product development. This can demonstrate for the quality assurance process by the systematic derivation of *validation measures* from *functions* with their specific *quality risk aspects* and *quality focus values* which are controlled by the *product quality indicators*. No other quality model delivers this transformation on a same or more direct way.
Another benefit is the usability in different project environments like SCRUM, SPICE, V-model and product lines as described or referred in this paper. This give companies the opportunity to manage the quality of products in one way also if they are delivered by projects with different realization/development models. The referred quality models are mostly to process oriented or abstract for direct application in SCRUM environments or the models are not directly applicable in a SPICE environment.

An implicit evidence of the effectiveness of this quality model concept is that it is using the basic idea of the forward and backward chains of [4] and the risk based testing of [15] which are proven in many projects and accepted by project and quality managers. The additional benefit of this quality model concept is to combine the strengths of these two concepts.

Continuous improvement in the selection of *validation measures* during application of this quality model concept. Due to the controlling of effectiveness of the improvement of the *product quality indicators* the *validation measures* will be used in future for effective and efficient validation. The pool of accepted *validation measures* in the project and organization is growing over time. This is important for preventive quality actions. When they are carried out their effects on product quality are not yet visible. The effectiveness of a selection of *validation measures* is best shown in product line context [9].

4 Summary and perspectives

The presented proposal to derive a quality model is driven by a structured value based approach [1]. The functional breakdown which is described in this paper will help reduce the quality risk within an organization. This is achieved by setting up quality measures that will reduce quality risk.

Beyond the holistic approach of a quality model the approach can integrate all relevant stakeholders on order to guarantee the commitment for the *validation measures*. Furthermore the approach can be applied to any product or portfolios as the approach shown is flexible enough to cope with different-sized requirements.

The approach will ensure that quality aspects are considered at a very early project stage and become adequately estimated and integrated, as for example during the tender phase. The approach presented here shows that it is possible to extract *functions* from specification as a good planning and estimation tool for quality management.

The outcome of the presented approach is a more systematic optimization of the portfolio of *validation measures*. It supports a more targeted based steering of quality based on the balance of effort and costs. Furthermore it enhances and improves the identification of quality relevant aspects and the *product quality indicators* for a quicker set up, and a more specific continuous optimization in the organizational context. But there are limitations as every organization has specific quality aspects or risks and the balance of effort and costs also depend on the organizational context.

5 Literature

[1] David A. Garvin: What Does „Product Quality" Really Mean?, Sloan Management Review, Fall 1984, pp. 25–45

[2] ISO/IEC 9126:1991

[3] ISO/IEC IS 15504:2006

[4] R. Schmitt, C. Lenkewitz, C. Behrens: Unternehmerisch umgesetzt – Das neue Aachener Qualitätsmanagement Modell; MQ Management und Qualität 11/2007

[5] F. Deissenboeck, S. Wagner, M. Pizka, S. Teuchert, J.-F. Girard: An Activity-Based Quality Model for Maintainability; ICSM '07: Proc. of the 23rd IEEE International Conference on Software Maintenance 2007

[6] R. G. Dromey: A model for software product quality. IEEE Transactions on Software Engineering, 21(2), 1995

[7] V. Basili, G. Caldiera, H. D. Rombach: The Goal Question Metric Approach. In: Encyclopedia of Software Engineering. John Wiley & Sons, 1994, S. 528–532

[8] R. Plösch, H. Gruber, A. Hentschel, Ch. Körner, G. Pomberger, S. Schiffer, M. Saft, S. Storck, The EMISQ Method - Expert Based Evaluation of Internal Software Quality, Proceedings of 3rd IEEE Systems and Software Week, March 3-8, 2007, Baltimore, IEEE Computer Society Press, 2007

[9] A. Poth, J. Hohner, M. Pramsoler: Take advantage in degrees of freedom for process and product quality – a holistic view on the SW-development; ATZ Automobil Technische Zeitschrift (Volume 112 (2010) Issue 6); Springer Verlag 2010, ISSN 0001-2785

[10] S. Adam, F. Blucha, J. Doerr, A. Poth: High Quality in Elicitation and Specification of Nonfunctional Requirements - Lessons Learned from Applying this Method to the Automotive Domain; Conference on Quality Engineering in Software Technology (CONQUEST'08), Potsdam, Germany, ISBN 978-3-89864-567-6

[11] A. Poth, B. Arends, T. Link: Funktionsorientiertes QM - Funktionsorientiert zur Absicherung von Produktmerkmalen: Q-Brennpunkte im Fadenkreuz; QZ Qualität und Zuverlässigkeit (Volume 54 (2009) Issue 11, p 29-33), Hanser Verlag 2009, ISSN 0720-1214

[12] M. Bittner, A. Botorabi, A. Poth, M-O. Reiser, M. Weber: Managing Variability and Reuse of Features and Requirements for Large and Complex Organizational Structures; Requirements Engineering Conference 2005 (RE05), Paris, France

[13] Glyn Holton, Value-at-Risk: Theory and Practice, Academic Press (2003). ISBN 978-0123540102

[14] www.scrumalliance.org/articles/89-focus-on-value scrum value priority.

[15] J. Bach: Heuristic Risk-Based Testing; Software Testing and Quality Engineering Magazine, 11/99

[16] K. Buhr, F. Houdek, A. Poth: Systematic Improvement of System-Specifications for Automotive Systems; Conference on Quality Engineering in Software Technology (CONQUEST'09), Nürnberg, Germany, ISBN 978-3-89864-637-6 (p 207-222)

6 Author CV

Alexander Poth (alexander.poth@sqs.de)

He studied computer engineering at the Technical University of Berlin and the Universidad Politecnica of Madrid.
In 2004 he qualified as SPICE assessor, in 2006 as an isqi Professional for Project-Management and in 2007 as an ISTQB Test Manager and iREB Professional for Requirements Engineering. Furthermore he qualified as QAMP and SCRUM Master.

Since 2007 he is working as senior consultant for the SQS Software Quality Systems AG with focus on quality management. Since 2004 he has been working as a quality management consult and engineer. Previously he worked as a software developer for a telecommunications OEM and an ERP-System OEM.

Michael Winter (michael.winter@dlh.de)

He studied English and History at the Johann-Wolfgang-Goethe University in Frankfurt before becoming involved with the computer industry after taking his degree in 1986. Since 1989 he has been working for Germany's largest airline. In 1993 he published: MVS/ESA JCL which was mainly designed for professionals using large IBM mainframes.

After 1999 he became involved in the design and building of large applications designed for call centres. He focused on the design of user interfaces and on the introduction of quality management measures.

Metrics to improve control in outsourcing software development projects

Laura Ponisio
BE Software Design
Amsterdam, The Netherlands
laura@besoftwaredesign.com

Pascal van Eck
Department of Computer Science, University of Twente
Enschede, The Netherlands
p.vaneck@utwente.nl

Abstract

Measurements serve as vital instruments to control projects involving software development outsourcing. However, managers have found it difficult to develop and implement effective measurement programs, in part because guidelines for choosing among concrete measurements are scarce. We address this gap between research and practice by examining frameworks and guidelines in the software process improvement literature. Our contribution comprises a framework that provides a set of measurements (selected from the research literature) for control of software development in a cooperative setting and a set of principles and guidelines for the design of an information infrastructure that provides managers with control information. We have validated our approach by showing our framework to an expert in outsourcing projects, who confirmed its potential.

Keywords

Software process improvement, measurements, metrics, outsourcing

1 Introduction

Making decisions that are better for value creation in current fast-changing software development scenarios demands dynamic monitoring and control mechanisms. Control through metrics is a critical component of the success of software process improvement programs (SPI) [14,18]. Much has been written about metric program implementation in SPI efforts [20,7,13,1,11,8]. But despite the importance of metrics, few studies have been made of its role in enabling new networked outsourcing models or of the software process by which control is improved through metrics. To address this gap, in this paper we aim at selecting metrics that help managers of outsourcing projects to make software that helps clients to meet their business goals in the current networked context of development processes.

To examine these issues, this explorative study was designed to address the following questions: a) how can managers increase control without losing flexibility? b) what has been written about concrete metrics – their implementation problems, benefits and contexts – managers can choose from? and c) what challenges did companies face when implementing software development metrics?

To answer these questions, a systematic literature survey has been carried out to develop a picture of the shape of the measurements field through the lens of software project improvement in an outsourcing context. By developing such a picture, we expose the ubiquity of approaches, help practitioners to synthesise and reflect on existing work and contribute to focusing the direction of interest of project managers. The current paper presents a subset of the results of this study in the form of a framework (Section 3) that contains a set of organisational effectiveness measurements and an information infrastructure that collects and distributes principles, lessons learnt and measurement data in software development projects. This framework is distilled from research findings published over the last two decades at the intersection of three different, but related fields: software process improvement (SPI), metrics, and outsourcing. An overview of the relevant parts of this field is presented in Section 2.

2 SPI, Metrics and Outsourcing

2.1 SPI

The fundamental objective of software process improvement (SPI) approaches is to “change software practices in order to achieve improvements in quality and productivity” [1]. Attempting to improve software quality, stakeholder satisfaction and profitability, SPI techniques address a number of topics such as software processes, standardisation, software metrics and project management.

Aaen *et al.* made a survey of the SPI literature and experiences from SPI practice [1]. The examples from practice described in their paper illustrate that there is room to implement SPI plans in very different ways and that metrics must be adapted, at the time of implementing them, to the specifics of an organisational environment.

All in all, although some of the approaches are extensively applied and offer unquestionable benefits such as the possibility to evaluate an organisation against stable criteria, systematisation and prioritisation [12], there is almost no discussion about how current networking conditions impact metrics on SPI programs. In particular, how do current global development conditions affect the implementation of metrics supporting an SPI initiative?

2.2 Metrics

People use measurements to gather feedback regarding the state of a project (for instance, completeness, quality of requirements, and accuracy of project estimations), create orientation during the change process, and determine how much the organisation has benefited from the changes derived from improvement plans. Visible results are considered critical to success of any improvement plan:

they keep participants focussed and motivated. In particular, it has been argued that members of an organisation would put more effort in SPI activities if measurements prove a return on investment [14].

However, measuring involves several risks. Organisations experience difficulties gathering and applying measurements that are meaningful [1]. Having numbers to show does not mean per-se neither that the measurements are relevant and meaningful, nor that they are accurate and reliable. Opportunistic behaviour, for instance, might jeopardise any SPI effort on the grounds of protecting particular interests.

Iversen and Mathiassen report from a case study that analyses an engineering process in which a metrics program is constructed and put into use [13]. The program's goal was to test the effect of ongoing SPI initiatives within the organisation. This article is of interest to us because it describes a measurement program that went beyond the barrier of gathering data. Once implemented in the company Danske Data, the measurement program generated an "evolutionary cultivation process". Moreover, Iversen and Kautz [15] and Kautz [17] emphasise that to be successful, the metrics programs implemented should be defined according to the organisation's specific information needs.

All in all, measurements can be regarded as one of the means to gather feedback concerning the effect of the SPI effort, establishing baselines in the SPI programs and to demonstrating the extent to which the goals of a program are met. In spite of the clear importance of metrics, we have observed that outsourcing organisations are still lacking a portfolio of metrics defined to their specific information needs.

2.3 Outsourcing

Outsourcing today is mostly performed in an inter-organisational network rather than by a single organisation. This change in the context of software development makes managers face the challenge of having to control actions that are beyond traditional boundaries. Moreover, in this new context, open source and outsourcing development share challenges related to geographically distributed development such as project members working in arbitrary locations, rarely or never meeting face to face and coordinating activities exclusively via e-mails and bulletin boards.

Mockus, Fielding and Herbsleb examined the development process of an open source application by quantifying elements of software development such as developer participation, core team size, and code ownership for the Apache web server open source software development project [19]. The study shows that a large network of people (400 code contributors) cooperated to develop software and that most of the code was made by a small group of developers (approximately 15 developers). It was expected that these 15 developers arranged a partition of the code, to prevent making conflicting changes. But measurements proved otherwise: parts of the system requiring changes were worked upon by more than one developer, suggesting thus a healthy contribution coordination mechanism based on mutual trust and respect.

The practical nature of the coordinating contribution mechanism, however, remains to be researched. Specially viewing today's highly competitive outsourcing market and recent research which has shown that not only product quality is important: low customer perception of delivery quality may rule out a supplier for the next project [6]. We hypothesise that part of this project success was due to the well covered open source system's information needs; which was supported by the metrics they used.

3 Research findings

We present our research findings in a framework that consists of two parts: a set of *organisational effectiveness measurements* (Section 3.1) and a set of *information infrastructure principles* (Section 3.2). Organisational effectiveness measurements are software metrics in a broader context, as we explain below. The information infrastructure principles are the starting points for the design of a system of components that provides managers with information to control software processes.

Organisational effectiveness metrics				
		Metric	**Organisation**	**Reference**
Results	**Size**	Functional size	Hewlett Packard, Eclipse, Danske Data	Grady97*, Iversen00
	Code ownership	Developer participation: how is work distributed? E.g., can we recognise "partitioning" in the code?	Apache server	Mockus00
	Time	Adherence to schedule: variation from agreed time of delivery, absolute and relative to volume of project.	Danske Data	Iversen00, 03 and 06
	Cost	Adherence to budget: variation from estimated use of resources.	Danske Data	Iversen00, 03 and 06
	Quality	Number of error reports relative to size in function points.	Danske Data	Iversen00, 03 and 06
		Number of error reports, absolute.	Danske Data	Iversen00, 03 and 06
		Defect density	AT&T, Motorola, IBM, Apache server, Eclipse, Danske Data	Barnard94*, Rosenberg94*, Florac97, Mockus00
		Defects per line of documentation	Hewlett Packard	Grady86*
		Defects per testing time	Hewlett Packard	Grady86*
		Defects per thousands of non-commented source statements	Hewlett Packard	Grady86*
		Post-release defects per thousand lines of code added, instead of delivered.	Apache Server	Mockus00
		Non-commented source statements per engineering month	Hewlett Packard	Grady86*
		Cyclomatic Complexity	IBM Rochester SPI	Kan95*, McLoughlin10
		System paritioning	IBM Rochester SPI	Kan95*
		Fan in	IBM Rochester SPI	Kan95*, Forac97
		Fan out	IBM Rochester SPI	Kan95*, Forac97
Stakeholder satisfaction	**Customer satisfaction**	Survey in-person, phone and mail, random, systematic and stratified.	IBM Rochester SPI	Kan95*
		Satisfaction with the development process (questionnaire)	Data Dansk	Iversen00, 03, and 06
	Employee satisfaction	Satisfaction with the development process (questionnaire)	Data Dansk	Iversen00, 03, and 06
Industy Dynamics	**Market performance**	Any lost bids	Not applicable or unreported	McLoughlin10
		Loss of reputation to the firm	Not applicable or unreported	McLoughlin10
		Satisfaction with the development process (questionnaire)	Data Dansk	Iversen00, 03, and 06
Software process performance	**Productivity**	Number of individuals submitting reports (eg. bugs)	Apache server	Mockus00
		Size of the development community	Apache server	Mockus00
		Core team size	Apache server	Mockus00
		Cumulative distribution of contributions to the code base	Apache server	Mockus00
		Resources used to develop the system relative to volume of project hours (hours/FP).	Data Dansk	Iversen00, 03 and 06
		Gain per Year in Productivity	SEI Capability Maturity Model for Software	Herbsleb94
		Gain per Year in Early Detection of Defects	SEI Capability Maturity Model for Software	Herbsleb94
		Average fixed defects per working day	Hewlett Packard	Grady86*
	Time	Time used in review meetings	Small company	Iversen00
		Reduction per Year in Calendar Time to Develop Software System	SEI Capability Maturity Model for Software	Herbsleb94
		Time to resolve problem reports	Apache server	Mockus00
		Percent overtime per 40 hours per week	Hewlett Packard	Grady86*
	Cost	Resources used in coordination activities	Small company	Iversen00
		Thousands of Dollars per Year Spent on SPI	SEI Capability Maturity Model for Software	Herbsleb94
		Dollars per Software Engineer per Year Spent on SPI	SEI Capability Maturity Model for Software	Herbsleb94
		Average engineering hours per fixed defect	Hewlett Packard	Grady86*
	Quality	Reduction per Year in Post-Release Defect Reports	SEI Capability Maturity Model for Software	Herbsleb94
		Business Value Ratio of SPI Efforts	SEI Capability Maturity Model for Software	Herbsleb94
		Average fixed defects per working day	Hewlett Packard	Grady86*
		Perception of delivery quality	Large consultancy company	VanEkris08

*According to Rico 1999

Fig. 1: The hierarchical diagram of measurements.

3.1 Organisational effectiveness measurements

Information infrastructure (part 1)					
		Principles	**What *not* to do**	**Quote / Example**	**Reference**
Information management	**Start by determining goals**	Define clear outcomes to expect and collect the data based on clear objectives. An example of a clear outcomes to expect is ``to let all developers work on all parts of the product". Using Basili's Goal Question Metric method is a good way to design the metrics to include in the program, being based on the goal that stakeholders expect to achieve from the metrics program.	To implement a set of metrics that is not well suited to describe a concrete outcome. An example of a too general objective is ``to give information about the effect of improvement initiatives".	*``Improved procedures for documentation of source code should allow all developers to work in all areas of the companies products and should facilitate the extension of the development teams."*	iversen00
	Match your goals with measurable attributes	To have a clear goal such as improve efficiency by x % is important, but do neither procrastinate nor forget to choose the attribute to be measured to check if the goal is met. It should be clear to management that the chosen attributes to measure (and the metrics results) describe the program's goal.	To underestimate the importance of having process and product attributes that describe the goal, leaving the decision of what should be measured to the group responsible of the metric program.	*``We expect to gain a 10% improvement in efficiency has become an important focal point of the SPI project [...] However, neither the CEO nor the contract was explicit on what should be measured to show this 10% efficiency improvement."*	iversen03
	Establish incentive structures	Metrics programs seem to be more successful if people see that they bring improvements to the process or the product. It should be clear: 1) what data to report, 2) how to report that data, 3) why data they provide is important, 4) show results based on the data.	To not inform project managers about exactly what data they should report, how they have to report it, what part of the process and product will be better by analysing that data.	*``Those who report data to a metrics program need to see some form of advantage in the program."*	iversen00
	Establish a project	A metrics program consume resources and therefore to set it in the context of a project for its own sake should make the task of collecting and analysing the data easier. Moreover, you need the right staffing to carry on a measurement program.	To have people work on metrics as an extra task of their current projects, forgetting to recognise that collecting and reporting metrics consume resources (e.g., time).	*``Establishing a formal project made the program far more visible in the organisation and made it much easier for the participants to argue that adequate resources should be available."*	Dekkers99 iversen00 iversen06
	Start simple	In the beginning, collect a small set of goal-oriented metrics. For example, ``One company measured the number of fixed change requests delivered on time and the time used in review meetings and found that change requests delivery on time had increased from 45% to 77% and review meeting time was shortened by a factor 4."	Neither to start with a large metrics set, nor to start with too general metrics. Too many resources will be spent on them and people will not see accordingly advantages when the (analysis report containing the) results comes back to them.	*``six fairly complex factors should be measured, and all projects were required to report data from day 1. This was an extremely ambitious undertaking, and as of yet, all the factors have not been measured, and some have even been officially abandoned."*	iversen00
	Plan to throw one away	Some measurements will prove to be too difficult to get due to wrong initial assumptions and inaccuracy in the data access.	To underestimate measurement. For instance, planning that the measurement will be made in a completely automatic way.	*``The first measurement report only included 20% of the projects and only three out of six factors."*	iversen06, Rifkin 91
	Use organisational knowledge	Metrics programs must take into account the existing work practices in the organisation, and the needs of the stakeholders affected by a potential effect of the metrics application.	To not communicate clearly the advantages expected from implementing the program.	*``external consultants acted as analysts of the current practice and carried out interviews with the developers [...] This provided the knowledge necessary to define metrics and to gather data. "*	iversen00
	Consider potential problems when measuring size	Measure size accurately is critical because size is a key attribute to measure common goals such as efficiency and productivity.	To believe blindly the first result obtained from counting function points wihout checking that results match goal and match questions, i.e., checking that numbers obtained match the perceived size of the system.	*``Excluding a size measure seriously impeded reaching the original objective of measuring efficiency and productivity, as there was no longer a measure of the output of the software projects."*	iversen06
	Match measurement with your organisation's goal	Attributes to measure, measurement data and its results need to be recognised by management. Measurement must describe part of the project in the eyes of stakeholders. Otherwise measurements become unacceptable.	To see measurements as just gathering data without matching them to a business goal. For instance, to use function points count without normalising them when needed.	*``...after counting function points in several application systems, it was very difficult to see any relationship between the perceived complexity of the systems, and the number that the counting procedures had arrived at."*	iversen06
	Have a complementa-ry suite of measure-ments	The measurements you choose should be complementary. Each measurement should contribute to improve the picture of the system obtained from the measurement program.	For instance, to count code performed during the original development twice.	*``Enhancement projects that continued work on an existing system were accredited the entire function point count of each module they modified, even if the modifications were miniscule. This gave very few hours per function point, or in other words, unrealistically high productivity."*	iversen06 dekkers99
Information access	**Use improvement knowledge**	Implementing metrics programs involves several branches of knowledge such as SPI, software development and reverse engineering. One solution might be to include external consultants.	Disregard unfamiliar areas of knowledge (for instance, reverse engineering if management has background in software architecture) when making decisions about how to implement the metrics program.	*``The first measurement report [...] was criticized for being too academic."*	iversen00
	Use non-invasive measure-ments whenever possible	Facilitate collecting data by making it simple. Metrics from finished projects would be used as a baseline and metrics of finished parts of projects (when collecting them makes no harm) should be used when possible. Beware that some stakeholders will not provide data.	To have tedious mechanisms to report the data, especially with unclear questions.	*``There are some who simply do not enter data into the system. There are some that have misunderstood the definition of the field." ``...results from 13 out of 56 projects that were completed"*	iversen03
Communication management	**Publish Objectives and collected data widely**	People need to see that metrics they collected are used and that bring some advantage. Publishing realistic objectives is a way to secure gathering data of quality for the SPI program, and to improve the metrics program with employee's feedback.	To be vague in the objectives and to relate the metrics results to performance evaluations. The objective of using metrics is to improve the way we do things rather than to find who is to blame. Not informing the results of the metrics program might form undesirable rumours.	*(About why developers did not enter data) `` this information was not previously used for anything"* *``Data discipline was improved in the next report ."*	iversen00 iversen06
	Facilitate debate	Not only communicate the metrics, but incentivise stakeholders to discuss the metrics program and its results. Use their feedback to improve the measurement program	To not hear what employees have to say about the metrics program and its implementation. This might cause the loss of valuable improvements for the program (such as improving the input fields of the system used to report data).	*``Another problem was that questionnaires [..] covered questions relating to contractual agreements and to the entire course of the project, whereas those who answered the questionnaire were users who were only involved in acceptance tests."*	iversen00

Fig. 2: Information infrastructure: principles for the design of a system that provides managers with information to control software processes (part 1).

This section reports software development outsourcing measurements for SPI found in the existing

Information infrastructure (part 2)					
		Principles	**What *not* to do**	**Quote / Example**	**Reference**
Communication management	**Use the measurement feedback**	Empower the program with the feedback of the employees that develop and collect measurement data.	To enter in this vicious circle: ``*the data reported were of a poor quality, since those who reported them did not see any advantage in supplying accurate data in a timely manner. At the same time, the poor data quality caused management to be wary of making the results public.*"	*`` being able to recognise trends, even from imprecise or non-complete data sets, can be more helpful than having no data at all."*	iversen00
Presentation	**Facilitate feedback with good layouts**	Feedback empowers the measurement program. One of the mechanisms to facilitate feedback is to use the information gathered, for instance by emitting reports containing the results of the measurement program.	To have forms to fill in the measurement data that are difficult to understand. We don't have examples of 'bad' layouts, but examples of good ones can be found in Florac (1997)	*(about why developers did not enter data) ``the user interface for the application used to enter the data was highly confusing, giving rise to many wrong entries."*	iversen00 iversen06

Fig. 3: Information infrastructure (part 2).

literature. Why do we use the term 'organisational effectiveness measurements' instead of software metrics? As we have argued before, software metrics need to be applied in their organisational context, which, in current practice, often means globally distributed cooperative software development. The notion of *organisational effectiveness* as proposed by Applegate [2], which we explain below, provides us with the means to systematically identify the organisational context for software metrics. Our framework provides a collection of software (mostly process) metrics that instantiates Applegate's notion for the domain of SPI.

The software metrics that comprise our set of organisational effectiveness measurements are metrics that help managers to control software development projects. We are specifically interested in metrics that project managers can use to control software development performed in a cooperative context, such as outsourcing. Control in this context is the ability to develop an understanding of what is going on in the project online and make informed decisions. In particular, we want to understand how these measures can inform software development managers. This coincides exactly with Applegate's notion of organisational effectiveness, which "concerns what to measure to provide information upon which to base management decisions." [14]. The four areas of interest in measuring organisational effectiveness measurements adapted from Applegate [2], when instantiated with metrics we found in the SPI literature, are (i) **results**, which are needed to know how the software quality assurance process is performing, (ii) **stakeholder satisfaction**, (iii) **industry dynamics**, and (iv) **software process performance**, the set of "activities, methods and transformations that people use to develop and maintain software and the associated products, for example: product plans, design documents, code, test cases and user manuals" (SEI)).

Our selection of metrics is presented in Figure 1. Note that already in 1999, more than 487 metrics [4,10,9,16,22,23,5] for software process improvement had been identified [21]. Our selection comprises metrics that, according to existing literature, have been tried in real projects of real organisations, as is indicated in the column labelled 'Organisations'. (The name 'SEI Capability Maturity Model for Software' refers to organisations that have implemented that model.) This choice complies with our acceptance criterion: managers of outsourcing projects should find them useful for their software development outsourcing projects.

3.2 Information infrastructure

As stated before, we are interested in metrics that project managers can use to control software development performed in outsourcing. According to general models of control, a controlling system (in this case: a project manager) needs information about the system that it tries to control (in this case: a software development project in a cooperative context). The software metrics literature discusses the many different metrics identified in the software field that can serve as control information, and that we have presented in the previous section. This information, however, needs to be made available to the controlling system. The *information infrastructure* is the system that connects the controlled system to the controlling system and supplies the controlling system with information.

In Figure 2, we present a set of principles that can serve as a starting point for the design of such an information infrastructure. Like the set of metrics presented in the previous section, this set of principles is a selection of principles found in the research literature. We have selected those principles that, based on experiences of applying SPI programs, provide managers with information to control software processes. Thus, the principles are taken from related work describing actual experiences in SPI measurements program application. We illustrate how our principles can be used with the following examples. Consider a manager of an outsourcing project who wants to have more control. A significant part of having that control is to know exactly who to contact in the off-shore team, which can be seen by measuring developer participation (I.e. how is work distributed? Is there any partitioning in the code? Our framework points out which metrics can be used for that; see the Organisational effectiveness metrics, Results, Code ownership, Developer participation in Figure 1).

As another example, consider a manager who has to implement, for the first time, SPI in his project and who needs to do so with very limited resources. He plans to base size measurement on function points and, in order to reduce the burden of extra work for his people, plans to compute function points automatically. Our framework shows that functional size is a measurement that has been used many times (Figure 1 under Results, size, functional size), and that it is critical (Figure 2 under Information Management, consider potential problems when measuring size). It also points out an example where after using function points, managers could not see any relationship between the perceived complexity of the system and the obtained metric results (Figure 2, Information management, match measurement with your organisation goals). This challenge could be explained by a known problem with function points: maintenance projects being credited the entire function point count of each module they modify even for tiny modifications, thus, indicating unrealistically high productivity. This reference is pointed at by our framework. In this way, using our framework, the manager is warned of the risk of automatizing function point counting.

3.3 Using the framework to derive guidelines for cooperation in outsourcing

In this section, to show usefulness of our work, we present a number of guidelines for software development in an outsourcing context. We view these guidelines as new insights that we were able to find by comparing traditional industrial styles of development with cooperative development processes from the perspective of the sets of metrics and principles provided by the framework.

Distribution of work Measure how work is distributed within the project (for instance, employee’s participation). In off-shored outsourcing projects there is most likely little if any control on work distribution within part of the team, distance being a large obstacle to exercise control in this way. However, measuring the distribution of work enables managers to improve control in the development *process*.

Coordination mechanisms Consider improving traditional coordination mechanisms such as plans, system-level design, and defined processes. We have seen cases of successful development projects where non-traditional mechanisms are used, such as social networking-like notifications of commit information in the development of the Apache server. These mechanisms have the potential to help managers save valuable resources.

Feedback and frontier building Respect the position of those with experience in the area being worked on. Respecting the frontiers built on the ground of experience supports beneficial feedback.

Cohesion and coupling Analyse coupling and cohesion to improve development processes. By cohesion we mean the number of intra-unit activities that generate knowledge, such as the amount of time spent in introducing new members to project tasks. By coupling we mean work that crosses the boundaries of a working group. For instance, coupling increases with the interactions between working units. Consider how many resources are invested in tasks related to cohesion and coupling: measuring coupling and cohesion have the potential to paint a picture of where resources go in a project.

3.4 Evaluation of the proposed framework

We have validated our framework via two interviews with an expert in managing outsourcing project. Reflecting on the potential of the framework applied to general situations, our expert could relate to this approach, finding it useful to improve control in outsourcing projects. In particular, the interviewee was very enthusiastic about the guidelines and principles, recognising specially the principle “Plan to through one away” and strongly agreeing with the guideline that suggest to analyse cohesion and coupling.

Is our framework suited to address the outsourcing measurement challenges? The expert agreed that our framework would be useful in the current outsourcing context because it helps companies to meet the current outsourcing challenges such as lack of time, need for semi-automatization and competitiveness. In the opinion of this expert, our framework “is OK because metrics are important, but there is no time to report findings or searching for the best metrics: it is all about finishing a project on time. We would need some kind of automated system, so that there are fewer things to do.”

Surprisingly, our interviewee was concerned about the framework’s potential to replace managers. This was never the authors’ intention, who considers the thought of replacing people by frameworks totally unrealistic. The idea behind our framework is to empower managers by giving the right information.

For reasons of space, unfortunately we can neither detail our rigorous literature review and the metrics inclusion and exclusion criteria, nor compare this paper’s contribution with other existing methods in literature such as Basili’s Goal Question Metric method. However, we are aware of the importance of being rigorous in these regards and we are reporting these essential points in coming articles.

4 Discussion

This study addresses the question of what to measure to provide information upon which to base management decisions in software process improvement. We have based our study on research papers that report real experiences with metrics in organisations such as Rico’s a survey of almost 500 metrics [21]. Our framework is different from Rico’s in two ways. First, while Rico’s aim was to be complete, our aim is to provide a small set of metrics that (i) have proven to be useful in organisations and (ii) focus on cooperative software development, specifically outsourcing. Second, our selection contains a number of metrics that have been identified after the year in which Rico’s selection was published.

The results of our research suggest that organisations should develop mechanisms so that outsourcing companies can develop their own benchmarking standard to compare across the different kinds of projects in their organisation. Our work is, thus, in line with recent work in the area of software maintainability [3]. Our results suggest that the framework of Applegate has the potential to help us to find the right metrics. In that sense, our results are in line with Iversen and Ngwenyama [14], who, based on a longitudinal study of a change initiative in a Danish software company, tried to develop an understanding of measurements in SPI. Their organisational-change-theory-based measurement methodology can be considered an ancestor of our outsourcing measurement framework.

5 Conclusion

Despite an abundance of research literature on metrics for software process improvement, it is still difficult for managers to choose a set of measurements that enables them to control software development, especially for software development in a cooperative setting such as outsourcing. In this paper, based on a systematic survey of the research literature, we present a framework that provides a set of measurements and a set of principles and guidelines for the design of an information infrastructure. The set of measurements has been distilled from the research literature by selecting metrics that are well-defined, have been used in real projects in real organisations (as reported in the literature), and that we believe are most suitable for cooperative software development. The framework has been validated via expert interviews. The main conclusion is that the set of measurements has the potential to improve control in outsourcing projects, and that the principles and guidelines are potentially very useful for managers to apply the measurements in real-world projects, particularly in an outsourcing context. However, given the time pressure that is prevalent in today’s outsourcing market, it is important that any SPI project strives to automate as much measurements and reporting as possible.

6 Acknowledgements

Thanks to the expert who confirmed our findings, helping us to validate the results presented in this paper. We are also grateful for the helpful comments we received from the reviewers of this paper.

7 Literature

1. Aaen, I., Arent, J., Mathiassen, L., Ngwenyama, O.: A conceptual MAP of software process improvement. Scandinavian Journal of Information Systems 13, 81–101 (2001)
2. Applegate, L.M.: Managing in an information age: Transforming the organization for the 1990s. In: Proceedings of the IFIP WG8.2 Working Conference on Information Technology and New Emergent Forms of Organizations: Transforming Organizations with Information Technology. pp. 15–94 (1994)
3. Baggen: Standardized code quality benchmarking for improving software maintainability (2010)
4. Barnard, J., Price, A.: Managing code inspection information. IEEE Software 11(2), 59–69 (1994)
5. Dekkers, C.A.: The secrets of highly successful measurement programs. Cutter IT Journal 12(4), 29–35 (1999)
6. van Ekris, J.: Customer perception of delivery quality: a necessary area for attention for project managers. In: Pahl, C. (ed.) Proceedings of IASTED International Conference on Software Engineering as part of the 26th IASTED International Multi-Conference on Applied Informatics, Innsbruck, Austria. pp. 268–275. ACTA Press (2008)
7. Florac, W.A., Park, R.E., Carleton, A.D.: Practical software measurement: Measuring for process management and improvement. Tech. rep., CMU/SEI-97-HB-003 (April 1997)
8. Goldenson, D.R., Herbsleb, J.D.: After the appraisal: a systematic survey of process improvement, its benefits, and factors that influence success. Technical CMU/SEI-95-TR-009, Software Engineering Institute, Carnegie Mellon University (1995)
9. Grady, R.B.: Successful software process improvement. Saddle River, NH: Prentice Hall (1997)
10. Grady, R.B., Caswell, D.L.: Software metrics: Establishing a company-wide program. Englewood Cliffs, NY: Prentice Hall (1986)
11. Hansen, B., Rose, J., Tjrnehj, G.: Prescription, description, reflection: the shape of the software process improvement field. International Journal of Information Management 24(6), 457 – 472 (2004)
12. Herbsleb, J.; Carleton, A.; Rozum, J.; Siegel, J. & Zubrow, D.: Benefits of CMM-Based Software Process Improvement: Initial Results. CMU/SEI-94-TR-013 (1994)
13. Iversen, J., Mathiassen, L.: Cultivation and engineering of a software metrics program. Information Systems Journal 13, 1365–2575 (2003)
14. Iversen, J., Ngwenyama, O.: Problems in measuring effectiveness in software process improvement: A longitudinal study of organizational change at Danske data. International Journal of Information Management 26(1), 30 – 43 (2006)
15. Iversen, J.H., Kautz, K.: The challenge of metrics implementation. In: Svensson, L., Snis, U., Srensen, C., Fgerlind, H., Lindroth, T., Magnusson, M., stlund, C. (eds.) Proceedings of IRIS 23. Laboratorium for Interaction Technology. University of Trollhttan Uddevalla (2000)
16. Kan, S.H.: Metrics and models in software quality engineering. Reading, MA: Addison-Wesley (1995)
17. Kautz, K.: Making sense of measurement for small organizations. IEEE Softw. 16(2), 14–20 (1999)
18. McLoughlin, F., Richardson, I.: The rosetta stone methodology a benefits-driven approach to SPI. In: Systems, Software and Services Process Improvement, proceedings of the 17th European Conference, EuroSPI 2010, Grenoble, France. pp. 201–212 (2010)
19. Mockus, A., Fielding, R.T., Herbsleb, J.: A case study of open source software development: the Apache server. In: Proceedings of the 22nd international conference on Software engineering. pp. 263–272. ICSE ’00, ACM, New York, NY, USA (2000)

20. Paulish, D., Carleton, A.: Case studies of software process improvement measurement. IEEE Transactions on Software Engineering 27(9), 50–57 (1994)

21. Rico, D.F.: Using cost benefit analyses to develop a pluralistic methodology for selecting from multiple prescriptive software process improvement (SPI) strategies. Master's thesis (1999)

22. Rifkin, S., Cox, C.: Mesurement in practice. Tech. rep., CMU/SEI-91-TR-016 (August 1991)

23. Rosenberg, L.H., Sheppard, S.B., Butler, S.A.: Software process assessment (SPA). In: Third International Symposium on Space Mission Operations and Ground Data Systems (1994)

8 Author CVs

Laura Ponisio

Laura received her Ph.D. in the area of software reverse engineering from the University of Bern (Switzerland), her Master in object-oriented software engineering from both Ecole des Mines de Nantes (France) and Vrije Universiteit Brussel (Belgium), and her Licentiate degree in Computer Science from UNLP, the National University of La Plata (Argentina). She has work experience in industry and academia, including software systems development and database administration. Currently, she performs action research in The Netherlands focusing on international collaboration with private and public institutions. Her research focuses on inter-organisational networks, global software development and object-oriented systems.

Pascal van Eck

Pascal van Eck received his MSc degree in Computer Science from Vrije Universiteit Amsterdam in 1995. From 1995 until January 2000, he worked as a research assistant at the Artificial Intelligence Department of Vrije Universiteit. In 2001, he successfully defended his PhD thesis on a formal, compositional semantic structure for the dynamics of multi-agent systems. Starting February, 2000, he works at the Information Systems Group of the Department of Computer Science, University of Twente, as an assistant professor. His research interests include IT governance and alignment of business and IT architectures at the enterprise level.

The Design and Development of Software Process Reference Models – Experiences and Lessons Learnt

Padraig O'Leary[1], Ita Richardson[2],

[1] RiSE – Reuse in Software Engineering and the Federal University of Bahia, Brazil
[2] Lero – the Irish Software Engineering Research Centre, University of Limerick, Ireland
Padraig.oleary@rise.com.br
Ita.richardson@lero.ie

Abstract

Software process reference models can serve as a tool for simplifying process problem solving. Through a series of research phases using sources in industry and academia, we developed a software process reference model for the derivation of products from a software product line. In this paper we describe how empirical evidence was used in the development of that process reference model while following an evolutionary multi-method research approach. A discussion on the selection of research methods for construction of process reference models is included. We explain how the different phases of the research formed a continuum in which the model was continually adjusted. Finally, we document important lessons learnt on software process reference model construction. The goal of this paper is to contribute to both the improved understanding of real world reference model construction and to the practical implementation of reference model construction guidelines.

Keywords

Software product lines, product derivation, process reference models.

1 Introduction

The main objective of a reference model is to streamline the design of (particular) models by providing a generic solution that can serve as a template for defining a model for a particular enterprise [1]. It can thus serve as a tool for simplifying process problem solving, and enables users to have a degree of confidence that the process begins on a solid foundation.

A reference model is (usually) created to represent already existing processes, e.g. by observing current practice in industry and academia, and thus serves as a blueprint for others. Reference models have to be universally applicable for a certain situation, and serve as a recommendation on how to solve or organise that situation.

The possible benefits of reference models are to raise the quality of the models produced, a cost, time and risk reduction, the reuse of knowledge, and access to industry best practices [2]. Reference models accelerate the modeling and configuration process by providing a repository of potentially relevant models [3].

Although there is a lack of empirical evidence to support these claims, it is a software engineering assumption, that when you reuse, instead of developing from scratch, a positive influence can be observed on the time-to market, product quality and development cost of the intended product. It is based on this assumption that reference models are believed to bring benefit.

According to [4] there has been a growing interest in construction approaches for process reference models. However there is a lack of real world experiences on reference model construction.

In Lero, we developed research aimed to fill an identified gap in process support for software product lines through the development of a process reference model for product derivation (Pro-PD). Pro-PD has been well received by the software engineering community with both journal [5, 6] and conference publications [7-9], and an invention disclosure by the University of Limerick [10].

The objective of this paper is two-fold: Firstly, show how an evolutionary multi-method research approach that adopted best practice reference model construction guidelines was designed and applied; and secondly, present important lessons learnt for process reference model construction based on our experience.

The paper is organised as follows: In Section 2 the research design is presented including the selection of research methods. Section 3 presents Pro-PD, the process reference model for product derivation. Section 4 discusses the lessons learnt. Finally Section 5 presents the conclusion.

2 Research Design

To define the research steps needed to construct a process reference model, the research methods must be selected. No single research method however is universally applicable and “all research approaches may have something to offer” [11]. There is a considerable range of research methods available [12], all of which have distinct strengths and weaknesses. To compensate for these weaknesses, Franz et al. [13] recommend multi-method research design. Multi-method design is “the conduct of two or more research methods, each conducted rigorously and complete in itself, in one project” [14]. By triangulating between methods and data, more plausible interpretations can emerge.

According to [15] multi-method research may be conducted from a complementary or evolutionary perspective. In the development of our process reference model, an evolutionary approach was followed. An evolutionary approach is used when there is little research conducted on a particular phenomenon, as was the situation in our case. Rather than investigating an effect through two or more different empirical methods, seeking confirmatory power between them, an initial exploratory study gathering qualitative data is undertaken. At this early phase, the initial study is designed to explore a wide range of topics in the area under investigation. The collected data is then analysed, and the important findings from the initial study are refined and used in the study. This process is then repeated, usually using a different research method.

Therefore, we applied an evolutionary multi-method research approach. The research design adopted was influenced by an approach by Ahlemann et al. [16] and was focused on empirically grounded and validatable process reference model construction.

2.1 Overview of Research Design

In an analogy with systems engineering, the overall construction process was based on a cyclic structure to allow for model corrections on preceding construction stages via feedback-loops. Although the stages are dealt with sequentially, they contain cyclic sub-processes. The research design was compatible with common suggestions for qualitative research designs in process models [4]. Stages 1 and 2 were the primary construction steps. Stage 3 was both a development and an evaluation step. Finally, stage 4 was purely an evaluation step. An overview of the research design is presented in Figure 1.

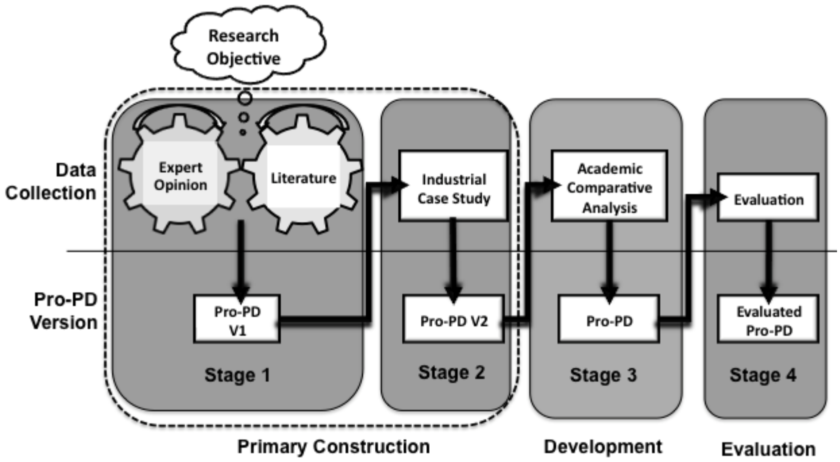


Figure 1 Overview of Research Design

Stage 1 entailed a literature review from which a preliminary version of the model was developed. The literature review aimed to identify the fundamental practices of product derivation, through studying existing identified product derivation approaches. Concurrent to the literature review, a series of iterative expert opinion workshops was organised. Participation by expert users in the core construction stage is emphasised by Rosemann and Schütte [17] and Schlagheck [18], as the users are the subject-matter experts of the problem domain. Furthermore, as the research is designed for use in both industry and academia, the selection of experts should reflect this. With this in mind, the selected participants were two academic SPL experts with 20 years' experience, an industrial SPL expert with 10 years' experience and a software process improvement expert.

Participants met twice a month for six months. At each workshop the reference model was presented to the experts and was evaluated using formal questions on model structure. The model was discussed amongst the group until a consensus was formed and the model was revised. After each workshop we returned to the literature and based upon the expert revisions and secondary research, iteratively developed the reference model.

Stage 2 was an industrial case study within Robert Bosch GmbH. This was carried out as an inductive, empirical validation [16]. We chose a case study as they are often considered to be the optimal approach for researching practice based problems, where the aim is to represent the case authentically "in its own terms" [19]. The reference model was mapped and compared to product derivation practices within the company. Robert Bosch GmbH was chosen for the case study because previous SPL efforts had been judged a success by their peers [20]. The case study was carried out in conjunction with the corporate research division. The case study was dual-purpose. In the first instance, we modelled the Bosch product derivation process for their internal use and then we updated the reference model based on our observations.

In conducting the case study, we analysed internal company documentation, which illustrated the existing process through completed projects. We then organised an onsite visit including a two-day workshop with the corporate research division of Robert Bosch GmbH. Attendees included selected product architects and developers from product line business units within the company. The primary researcher (O'Leary) was accompanied by two other researchers, one of whom had published extensively on case study research. After the workshop, a technical report [21] on the company's product derivation process was created and validated through feedback with Bosch SPL experts. Both the documentation analysis and the workshop output were used to identify what components should be included in subsequent versions of the reference model.

Stage 3 of the research, an academic comparative analysis, was carried out during a research collaboration with JKU (Johannes Kepler University Linz, Austria). JKU had previously developed the DOPLERP^{UCon} (**D**ecision-**O**riented **P**roduct **L**ine **E**ngineering for effective **R**euse: **U**ser-centered **Con-**figuration) approach. Based on initial discussions and existing documentation of our two approaches, a high-level mapping was created. This was done in a distributed manner using spreadsheets to visualize commonalities and differences between the two approaches. Using this mapping, the researchers met to analyse the first results, discuss open issues, and detail the comparison. We then conducted several telephone conferences with JKU researchers to work on the details of the comparison. Pro-PD was compared to the activities identified by DOPLER for Siemens VAI. Based on this comparison [5, 8] the final version of the model, Pro-PD, was developed.

Pro-PD was evaluated in two steps during stage 4 of the research. The first was an inter-model evaluation with the SEI PLPF [22] during which Pro-PD was reverse engineered and compared to the PLPF. According to Ahlemann et al. [16] process models that are compatible with such standards and norms can be are regarded as high quality. Then, we systematically evaluated Pro-PD by analyzing support for its activities in three independently developed, published and highly-cited approaches: COVAMOF [23], FAST [24], and PuLSE-I [25]. The approaches have been developed with different goals, for different purposes, and in different domains. Furthermore, in our literature review we identified that these three approaches were influential through their frequent citations.

Although a framework for evaluating product derivation approaches does not exist, we adapted a framework developed for the purpose of evaluating software product line architecture design methods [26]. We used this framework as a basis for our validation for two reasons. Firstly, it provided a simple tabular evaluation structure. Secondly, it had previously been published at ICSE, which ensures that it has been peer-reviewed.

3 Result of Research – Pro-PD

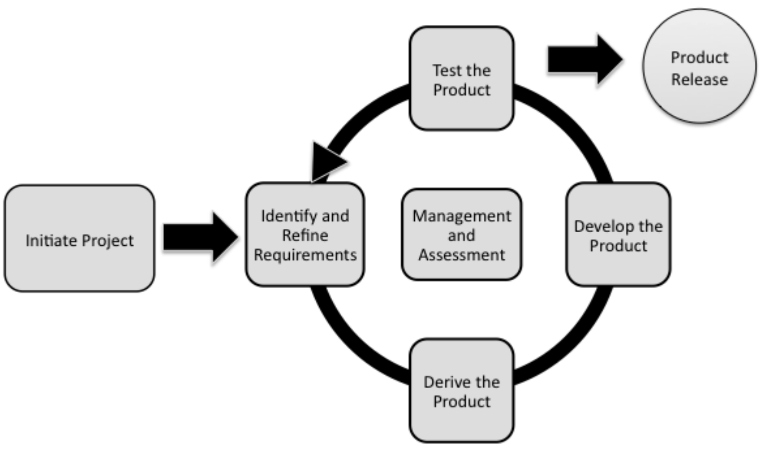


Figure 2 Overview of Pro-PD

As a result of the described research design, Pro-PD, a process reference model for product derivation was developed. Pro-PD focused on the roles, work products, tasks and activities used to derive products from a software product line. These elements represent the process building blocks of Pro-PD. **Fehler! Verweisquelle konnte nicht gefunden werden.**gives on overview of these Pro-PD activities and the iterative nature of the Pro-PD process.

3.1 Units of Work: Tasks and Activities

Pro-PD contains the following activities:

- Initiate Project - the preparatory tasks required to establish a product derivation project.
- Identify and Refine Requirements – the preparatory tasks required to commence a new iteration of the product derivation project.
- Derive the Product - creates an integrated product configuration that makes maximum use of the platform and minimises the amount of product specific development required.
- Develop the Product - facilitates requirements that could not be satisfied by a configuration of the existing assets through component development or adaptation.
- Test the Product - validates the current product build.
- Management and Assessment - provides feedback to the platform team and monitor progress of derivation project.

Table 1 lists the tasks performed for each of these activities:

Activity	Tasks performed in this activity
Initiate Project	Translate Customer Requirements; Coverage Analysis; Customer Negotiation; Create the Product Requirements; Verify the Product Requirements
Identify and Refine Requirements	Find and Outline Requirements; Create the Product Test Cases; Allocate Requirements; Create Guidance for Decision Makers
Derive the Product	Select Closest Matching Configuration; Derive New Configuration; Evaluate Product Architecture; Select Platform Components; Product Integration; Integration Testing; Identify Required Product Development
Develop the Product	Component Development; Component Testing; Product Integration and Testing
Test the Product	Run Acceptance Tests
Management	Provide Feedback to Platform Team, Monitor Project

Table 1 Pro-PD Activities and Tasks

3.2 Roles

We identified roles that represent the different responsibilities, which occur during product derivation: Customer, Product Analyst, Product Architect, Product Developer, Product Manager and Product Tester. These roles are assigned to specific tasks, which create and modify the different work products.

Role	Responsibilities
Customer	Customer Negotiation, Create Guidance for Decision Makers
Product Analyst	Translate Customer Requirements, Find and Outline Requirements,
Product Architect	Derive New Configuration, Select Closest Matching Configuration, Integrate and Create Product Build, Integration Testing, Provide Feedback to Platform Team
Product Developer	Select Platform Components, Develop/Adapt Components, Component Unit Testing
Product Manager	Coverage Analysis, Customer Negotiation, Create the Product Specific Requirements, Assess Results, Provide Feedback to Platform Team
Product Tester	Create the Product Test Cases, Integration Testing, Run System Tests

Table 2 Roles and Responsibilities

3.3 Work Products

Software Artefact	Platform Test Artefacts, Product Build, Product Test Cases, New Platform Release, Platform Architecture, Platform Components, Developed or Adapted Components, Existing Platform Configurations, Base Product Configuration, Integrated Product Configuration,
Documentation	Required Product Development, Translated Customer Requirements, Product Specific Platform Requirements, Product Requirements, Platform Feedback, Platform Requirements, Customer Requirements, Customer Specific Product Requirements, Negotiated Customer Requirements, Glossary,

Table 3 Work Products

A work product is an artefact, which is produced, modified or used by a task within the derivation process. The list of Pro-PD work products is listed in **Fehler! Verweisquelle konnte nicht gefunden werden.**.

3.4 Pro-PD as a Reference Model

Pro-PD is defined at a high level and is not to be used ‘as is’ but through specialization. In order to create a working company specific model the process needs to be specialized and a lower level of model abstraction needs to be constructed. Different instantiations of Pro-PD are created by following the adaptation rules and using the roles, tasks, activities and work products defined. We demonstrate the adaptability of Pro-PD as a reference model by proposing a waterfall instantiation [27] and an Agile instantiation (A-Pro-PD) [6].

4 Research Validity

All research only becomes valuable when it is first deemed credible. Many authors such as Creswell [28] however point out that there is no consensus or right way of verifying the credibility of qualitative research. In this context, Marshall [29] argues that the quality of research is dependent on honest and forthright investigations. This dependency means it is difficult to verify the quality of qualitative research. Another difficulty in qualitative research is the introduction of bias and the danger of multiple interpretations of data. Therefore, a self-critical attitude is essential in qualitative research. Demonstrating how you know is as important as demonstrating what you now.

4.1 Ensuring integrity, validity and accuracy of the results

Internal validity and credibility is achieved through prolonged engagement in the field, persistent observation and triangulation exercises [30, 31]. From their discussions on triangulation, Liamputtong and Ezzy [32] identify four types of triangulation, which may be used to strengthen a potentially weak case:

- Data Source Triangulation: The use of multiple information sources.
- Method Triangulation: The application of findings generated by different data collection methods.
- Researcher Triangulation: The inclusion of a variety of researchers in the research process.
- Theory Triangulation: we draw on multiple theoretical perspectives to provide new insights.

This research uses data source triangulation. In the initial framework development, multiple sources of literature were used as well as anecdotal evidence from SPL experts. Industrial practice was integrated directly through case study research and indirectly through the experiences of DOPLER^{UCon}. In the Robert Bosch GmbH case study multiple data sources were used. In the DOPLER^{UCon} academic comparative analysis, we had access to documentation and to the developers of the approach.

The research uses method triangulation. The research design includes case study research, expert opinion, facilitated workshops and SPL literature.

The research uses researcher triangulation. When it was possible the services of other researchers to facilitate the main researcher were engaged. In the facilitated industrial case study workshop session two other researchers assisted in the organisation and recording of results in the workshop sessions. During the collaboration with JKU, the research involved two members of the DOPLER^{UCon} team.

The research uses theory triangulation. In the development of the initial version of Pro-PD, we used a multitude of evidence from literature to theoretically validate aspects of the model. We synthesised this existing theory. This is a form of theory triangulation.

Another method of strengthening credibility is through respondent validation [33]. Lincoln and Guba state that respondent validation is “the most critical technique for establishing credibility” [31]. In respondent validation, we went back to the subjects with tentative results and refines them in light of their reactions. In Section 2 we describe how respondent validation was used during the iterative development of the model in the expert opinion workshop series. In the academic comparative analysis, the results were validated by the DOPLER^{UCon} team.

5 Lessons Learnt

Based on our experiences, we have detailed a number of important lessons for process reference model construction.

5.1 Choose an Appropriate Description Level

A common mistake in process reference model construction is over detailing the process. Often, due to over enthusiasm, process designers specify the most minute tasks to be performed – such process are virtually impossible to verify. Furthermore, once the process is implemented it is impossible to verify if these minute tasks were performed. Each process step should be defined at an appropriate level with clearly identified inputs and outputs, which can be used for performance verification.

A distinction should be made between a descriptive and prescriptive process description. In this regard, it is best to separate guidelines and checklists from processes. For example, detailed prescriptive process steps are better kept as guidelines or checklists. However, descriptive task listings should be applied and can be validated by an external auditor.

This makes the process reference models simple and more stable, while providing flexibility at the lowest level by providing different checklists and guidelines. Keeping the reference model simple makes it verifiable and minimizes the desire by practitioners to “fake it” by pretending to conform to some over strident process task requirements.

5.2 Documentation of Design Decisions

The construction process is normally not documented in reference modelling projects [16]. Therefore, it is often not clear how the final design of the reference model came to be. A reference model is only as strong as the design decisions taken in its construction; therefore documentation of design decisions is an essential element in proving the quality of a reference model.

Typically from reviewing the design decisions, the different flaws for each version of a reference model can be identified. The purpose of the different research development iterations is to sort out the weak as well as the strong parts of the process reference model. The important answers to capture are how, why and what impact these research stages have had on the model design.

5.3 Handling Refinements

Each stage of the research provides the basis for the revision or refinement of the reference model. A major challenge when performing development iterations is the evaluation of different suggestions with respect to each other. For example, before a correction is integrated it has to be determined whether the proposal can be characterized as being universally valid or whether it is tied to a specific context and therefore not suitable for model refinement.

Furthermore, improvement suggestions made by different persons are sometimes contradictory. There were two options to resolve these situations. First, one proposal is chosen over another if the source was deemed to be of a better quality, either through its experience or the location of the source. This evaluation is conducted by the researcher, and involves a degree of researcher interpretation as to the quality of the various sources. The alternative approach is to consider both suggestions and integrate them both into the model.

5.4 Generalizing the Findings

The ability to generalise is a key component of process reference model construction. However, this can often be challenging, particularly when your research design involves qualitative methods. As Patton [34] points out that the small size involved in qualitative methods make it impossible to generalise the results. This can be especially true in case study research where the focus on a particular case makes it unable to produce a general conclusion.

In an effort to counteract any weakness a multi-method research design should be adopted. By considering different sources such as expert opinion, literature, case studies and documented best practice, the generalisabitly of a reference model can improve and more plausible interpretations of the data can emerge.

6 Conclusion

This paper discusses how reference modeling projects, such as the development of Pro-PD, can be based on an empirically grounded and verifiable process. The discussion surrounding the selection of appropriate research methods is described, the application of the selected methods is detailed and the important lessons learnt are documented.

The paper contributes to an improved understanding of real world reference model construction through documenting our experiences and approach. This example from a completed research project contributes to the practical implementation of reference model construction guidelines.

7 Acknowledgements

This work is partially funded by IRCSET under grant no. RS/06/167 and by Science Foundation Ireland under grant no. 03/CE2/I303 1 and 07/SK/I1299.

8 References

1. Reinhartz-Berger, I., P. Soffer, and A. Sturm. A Domain Engineering Approach to Specifying and Applying Reference Models. in Proceedings of the Workshop Enterprise Modelling and Information Systems Architectures. 2005. Klagenfurt, Austria: German Informatics Society.
2. Winter, R. and J. Schelp, Reference Modeling and Method Construction: a Design Science Perspective, in Proceedings of the 2006 ACM symposium on Applied computing. 2006, ACM: Dijon, France.
3. Bernus, P., IFIP-IFAC Task Force on Architectures for Enterprise Integration, GERAM: Generalised Enterprise Reference Architecture and Methodology,
4. Fettke, P. and P. Loos, Reference Modeling for Business Systems Analysis. 2006: IGI Publishing.
5. Rabiser, R., P. O'Leary, and I. Richardson, Key Activities for Product Derivation in Software Product Lines. Journal of Systems and Software 2010. 84(2): p. 285-300.doi:10.1016/j.jss.2010.09.042
6. O'Leary, P., F. McCaffery. S. Thiel, I. Richardson, An Agile Process Model for Product Derivation in Software Product Line Engineering. Journal of Software Maintenance and Evolution, 2010. n/a.10.1002/smr.498
7. O'Leary, P., F. McCaffery, S. Thiel, I. Richardson. Preparing for Product Derivation: Activities and Issues. in 4th International Conference on Software and Data Technologies - ICSOFT 2009. 2009. Sofia, Bulgaria: INSTICC - Institute for Systems and Technologies of Information, Control and Communication.
8. O'Leary, P., R. Rabiser, S. Thiel, I. Richardson. Important Issues and Key Activities in Product Derivation: Experiences from Two Independent Research Projects. in Software Product Line Conference. 2009. San Francisco, CA, USA: Proc. of the 13th International Software Product Line Conference (SPLC 2009).
9. O'Leary, P., F. McCaffery, S. Thiel, I. Richardson, Towards Agile Product Derivation in Software Product Line Engineering. in 16th European Conference on Software Process Improvement (EuroSPI 2009). 2009. Madrid, Spain.
10. O'Leary, P. and I. Richardson.(2010), Invention Disclosure - Pro-PD: A Software Process Reference Model for the Derivation of Products from a Software Product Line, U.o. Limerick, UL2006125
11. Gill, J. and P. Johnson, Research Methods for Managers. 1991, London: Paul Chapman.
12. Galliers, R.D., Choosing Information Systems Research Approaches, in Information Systems Research: issues, methods and practical guidelines, R.D. Galliers, Editor. 1992, Blackwell Scientific: Oxford.
13. Franz, C.R., D. Robey, and R.R. Koeblitz, User Response to an Online Information System: A Field Experiment. MIS Quarterly, 1986. 10(1): p. 29-42
14. Morse, J.M., Principles of Mixed Methods and Multimethod Research Design, in Handbook of Mixed Methods in Social and Behavioral Research, I.T. Teddlie, Editor. 2003, Sage Publications: Thousand Oaks, CA:. p. 189-208.

15. Woods, M., et al., Multi-Method Research: an Empirical Investigation of Object-oriented Technology. Journal of Systems and Software, 1999. 48(1): p. 13-26.

16. Ahlemann, F. and H. Gastl, Process Model for an Empirically Grounded Reference Model Construction, in Reference Modeling for Business Systems Analysis, P. Fettke and P. Loos, Editors. 2006, IGI Publishing.

17. Rosemann, M. and R. Schütte, Multi-Perspective Reference Modelling, in Referenzmodellierung. State-of-the-art und entwicklungsperpektiven, J. Becker, M. Rosemann, and R. Schütte, Editors. 1999, Physica-Verlag: Heidelberg. p. 22-44.

18. Schlagheck, B., Object-oriented Reference Models for Process and Project Controlling - Foundation Construction Fields of Application. 2000: Wiesbaden: Deutscher Univ. Verlag.

19. Hammersley, M., R. Gomm, and P. Foster, Case Study Method: Key Issues, Key Texts. 2000, London: Sage Publications.

20. The SPLC Product Line Hall of Fame. [cited 3/02/2009]; Available from: http://www.splc.net/fame.html.

21. O'Leary, P., S. Thiel, and I. Richardson, Experience Report on Industrial Product Derivation Practices within a Software Product Line Organisation, Lero Technical Reports, Limerick, Ireland, Lero-TR-2010-03

22. Clements, P. and L. Northrop, Software Product Lines: Practices and Patterns. 2001, Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc.

23. Sinnema, M., S. Deelstra, and P. Hoekstra. The COVAMOF Derivation Process. in Proceedings of the 9th International Conference on Software Reuse (ICSR 2006). 2006. Turin, Italy: Springer Berlin Heidelberg.

24. Weiss, D.M. and C.T.R. Lai, Software product line engineering: A Family-based Software Development Process. 1st ed. 1999: Addison-Wesley Professional.

25. Bayer, J., et al. PuLSE-I: Deriving Instances from a Product Line Infrastructure. in 7th IEEE International Conference and Workshop on the Engineering of Computer Based Systems. 2000. Edinburgh, UK.

26. Matinlassi, M. Comparison of Software Product Line Architecture Design Methods: COPA, FAST, FORM, KobrA and QADA. in Software Engineering, 2004. ICSE 2004. Proceedings. 26th International Conference on. 2004. EICC, Scotland, UK.

27. O'Leary, P., Towards a Product Derivation Process Reference Model for Software Product Line Organisations, in Department of Computer Science and Information Systems. 2010, University of Limerick: Limerick. p. 277.

28. Creswell, J.W., Research Design – Qualitative & Quantitative Approaches. 1994, Thousand Oaks, CA: Sage Publications.

29. Marshall, C., Goodness Criteria: Are they Objective or Judgement Calls?, in The Paradigm Dialog, E.G. Guba, Editor. 1990, Sage Publications: Newbury Park, CA. p. 188–197.

30. Seale, C., The Quality of Qualitative Research. 2000, London: SAGE Publications. 224.

31. Lincoln, Y.S. and E.G. Guba, Naturalistic Enquiry. 1985, Beverly Hills: Sage Publications. 416.

32. Liamputtong, P. and D. Ezzy, Qualitative Research Methods. 2005, USA: Oxford University Press, 2 edition. 424.

33. Reason, P. and J. Rowan, Issues of Validity in New Paradigm Research, in Human Inquiry: A Sourcebook of New Paradigm Research, J. Rowan, Editor. 1981: Chichester: Wiley.

34. Patton, M.Q., Qualitative Evaluation & Research Methods. 2nd ed. 1990, Newbury Park, CA: Sage Publications.

9 Author CVs

Pádraig O'Leary

Pádraig O'Leary obtained a first class honours BSc degree in Computer Systems from the University of Limerick in 2005. He obtained his PhD in 2010. He worked as a post-doctoral researcher in Lero–the Irish Software Engineering Research Centre, on a software services project. Prior to joining Lero, Pádraig worked as a software developer in both Fineos and State Street Bank. His main research interests are software process improvement and software product lines. Currently, he is a Post-doctoral researcher with the Reuse in Software Engineering (RiSE) group based in the Federal University of Bahia (UFBA).

Ita Richardson

Dr Ita Richardson is a Senior Lecturer in the Department of Computer Science and Information Systems at the University of Limerick, Ireland, and a Research Area Leader for Process, Practice and Methods within Lero–the Irish Software Engineering Research Centre. Dr Richardson's main research interests are in software quality and software process improvement, focusing particularly on Global Software Development and regulated industry such as Medical and Health Systems. She publishes internationally and supervises full-time and part-time PhD students. She has received funding from sources such as Science Foundation Ireland, Enterprise Ireland and the European Community.

Lessons learned of a software process improvement using a multi-model environment methodology*

Muñoz Mirna[1], Mejia Jezreel[1], Jose A. Calvo-Manzano[2], Gonzalo Cuevas[2], Tomás San Feliu[2]
[1]Instituto Tecnológico de Orizaba, Av Oriente 9 no 852, 94300 Veracruz, México
{mirna.munozm, jez.mejia}@gmail.com
[2]Faculty of Computer Science, Polytechnic University of Madrid Campus de Montegancedo, Boadilla del Monte, 28660 Madrid, Spain
{joseantonio.calvomanzano, gonzalo.cuevas, tomas.sanfeliu}@upm.es
*****Authors listed by organization*

Abstract

Although many organizations are motivated to improve their software processes, very few know how best to do so. The result of this lack of knowledge is mainly reflected in the failures that often have the improvements efforts. These failures are reflected mainly because stakeholders feel frustrated and organizations are more convinced than ever that they must continue doing their work as before. Consequently the resistance to change in Software Process Improvement (SPI) increases. The software process improvement was done by applying a multi-model process improvement methodology (MIGME-RRC). This paper analyzes the results of a software process improvement in an organization focused on project management processes. Besides, the lessons learned are highlighted.

Keywords

multi-model environment, software process improvement, lessons learned, project management, MIGME-RRC methodology

1 Introduction

Software is becoming the core on any modern product or service; this has focused the software process as a critical challenge in organizations in order to create a competitive advantage respect to its competitors [1], [2].

According to Cuevas [3], Kautz [4] and Mathiassen [5], software process improvement is an obvious and logical way to address the increasing need to be competitive in the software industry. That is why software process improvement is the most critical and important effort that any organization should pursue, in order to increase their software development process capacity.

However, even when many organizations are motivated to improve their software processes, very few know how best to do so. Therefore, for most organizations software process improvement is still undisciplined, chaotic and completely unpredictable. So the process improvement initiatives are not successfully implemented [5] or have a limit success [6].

As a result, most improvement efforts fail, stakeholders feel frustrated, organizations are more convinced than ever that they must continue doing their work as before and the resistance to change in software process improvement increases [3]. The main problem is the difficulty that an organization faces when adapting the selected process improvement model to their current scenario [7], [8].

The goal of this paper is to present the analyses of a software process improvement experiences highlighting the lesson learned. The process improvement was done using a methodology that implements smooth and continuous process improvement, depending on the improvement pace accepted by the organization, to reduce the resistance to change named as MIGME-RRC by its Spanish acronym.

This paper is structured as follows: section two introduces to the multi-model methodology; section three presents the software process experience; section four analyzes the results of the software process experience; and section five presents the conclusions.

2 MIGME-RRC methodology

The methodology is a *methodology for a gradual process improvement implementation, focusing on diminishing change resistance* (MIGME-RRC by its Spanish acronym ***M****etodología para la* ***I****mplementación* ***G****radual de* ***ME****joras con enfoque en la* ***R****educción de la* ***R****esistencia al* ***C****ambio*).

This methodology enables a smooth and continuous improvement depending on the organization's business goals but allowing each unit to establish their own improvement implementation pace that prevent initial resistance to change in the organization. This methodology uses a bottom-up approach to software process improvement, which consists of identifying the internal best practices. Then, the external best practices proposed by most widespread models and standards through a multi-model environment, which complement the current practices and fit to the culture of the organization, are included.

Even when the description of the methodology is out of the scope of this paper, it is considered important to describe briefly how the methodology is composed in order to understand the experience presented below. Table 1 shows the four phases of the MIGME-RRC methodology [9],[10],[11].

Table 1. MIGME-RRC phases description

Phase	Description
Identify Internal Best Practices	analyzes how the organization works by identifying its best practices
Assess Organizational Performance	establishes the performance of its processes by comparing the business indicators achievement with the identified best practices
Analyze External Best Practices	analyzes the best practices of different standards and models in order to establish a multi-model environment as reference model
Implement Improvements	defines new processes and their implementation sequence depending on the internal and external best practices dependencies, their impact on achieving the business indicators and their facility of adoption

3 *The software process experience*

This paper shows an actual experience of a software process improvement that was conducted at everis consulting. This section sets first the context of the methodology application focused on the kind of organization and the application scope. Second, the activities carried out and the lessons learned are shown.

3.1 Everis consulting

everis (http://www.everis.com), is a multinational consulting firm with factories in Europe and Latin America where they develop and implement best practices to improve the performance of their factories. Since its creation in 1996, it has grown steadily in both invoicing and staff (in a constant way). In 2009, there were more than 7,000 employees, turnover was over €404m and, and over 1,000 projects were open every month. It is important to highlight that European and Latin America Offices were included in this study.

3.2 Application scope

The application scope was focused on everis project management because it is an activity with a broad impact on its business goals. So, processes related to project management should be appropriate and timely. Table 2 shows the application scope. It was set by taking into account two aspects: *process and staff*.

Table 2. Two aspects took into account for the application scope

Aspect	Description
Processes	Process related to project management process such as project planning, project monitoring and control, risk management, requirements management and supplier aggrement management,
Staff	• account managers, because they head one or more projects and are in charge of the projects, • projects managers and project leaders, because they work with the project management processes

3.3 Software process improvement experience

This section presents the experience and lessons learned to improve the project management process at everis. Next the activities carried out at everis according to MIGME-RRC methodology phases and the lessons learned of each phase are showed as follows: Table 3 shows the activities and lessons learned of phase Identify Internal Best Practices; Table 4 shows the activities and lessons learned of phase Assess Organizational Performance; Table 5 shows the activities and lessons learned of phase Analyze External Best Practices and finally Table 6 shows the activities and lessons learned of phase Implement Improvements.

Table 3. Activities carried out in phase Identify Internal Best Practices and lessons learned

Activities
•Account managers who have managed successful cases were interviewed •The information gathered from previous interviews were analyzed and approved by project managers and project leaders •The activities of all the approved diagrams were analyzed, in order to get a common set of activities. This common set is called *generic practices* •The organization's process documentation were analyzed in order to identify the practices contained in it •The generic practices diagrams and thepractices contained in organization's processes documentation were mapped in order to identify the organization's internal best practices •The best practices were approved at three levels: account managers, project managers and project leaders
Lesson learned
• Involve senior management in the selection of suppliers of project management success cases allows to get an appropriate senior management commitment since the beginning of the process improvement • Use open questions as interviews guide gives freedom to each interviewed when defining what they do, allowing a better tacit knowledge extraction • Perform generic practices validation at three levels has allowed to get stakeholders commitment and to assure that all stakeholders are speaking the same language, in the process context, through diferrents organization offices

• Integrate the source suppliers and the validation group assure that processes show the way the organization works

Table 4. Activities carried out in phase Assess Organizational Performance and lessons learned

Activities
• A coverage analysis using identified best practices, bussines goal key indicators and bussiness goal were done • The information of the business indicators established value for the fiscal year and the values measured over the 2007-2008 period as actual values were collected • A matrix of business indicators and internal best practices were made, filled and analyzed • The business indicators were prioritized by senior managers. The indicators selected were: (1) management rules; (2) project planning documentation; and start-up minutes
Lesson learned
• Use internal audits carried out by the organization as source have allowed to establish a real process performance • Map internal best practices with business indicators have allowed the identification of process strengths and process improvements opportunities • Show a state of current and desire process performance report to stakeholders has allowed to get the stakeholders commitment

Table 5. Activities carried out in phase Analyze External Best Practices and lessons learned

Activities
• The models and standards: CMMI-DEV v1.2, PMBOOK, PRINCE2, TSP, COBIT, ISO9001, and ISO/IEC 15504 were choosed • CMMI-DEV v1.2 model were choosed as the reference model • Most of the prioritized business indicators were related to project management. So, the project planning and project monitoring and control process were choosed because it is considered critical to successful project management. Besides, the level of mapping at specific practice level were establised • A template which includes inputs, subpractices, tools and techniques, work products, and informative components were established • The similarities among standards and models were identified and refined in order to stablish the multimodel-environment
Lesson learned
• Chosee CMMI-DEV v1.2 as reference model in order to establish the multi-model environment has allow to analyse different models and standards in order to establish a multi-model environment • Select standards and models based on the way the organization works such as CMMI-DEV v1.2, PMBOOK, PRINCE2, TSP, COBIT, ISO9001, and ISO/IEC 15504 has allowed to establish an adequate multi-model environment • Establish a multi-model environment at a specific practices level has enable different offices to select those practices that best tie with the way they work

Table 6. Activities carried out in phase Implement Improvements and lessons learned

Activities
• The main resistance and risk factors associated with the improvement were analyzed. On the one hand the resistance factors were: overload feeling, staff habits, lack of line manager`s commitment and any type of reward related to improvement effort. On the other hand ths risk factors were: inadequate training material, lack of adequate communication, lack of support, lack of improvement monitoting nas lack of staff involvement • The external best practices of a multi-model environment based on the current organizational culture and process were selected • The improved processes (that contain the best practices identified and a set of external practices of a multi-model environment to be incorporated) were grouped into a project management method, because the organization needed to develop a project management method as a part of its Corporate Methods methodology (COM). • The similarities among standards and models were identified and refined in order to stablish the multimodel-environment • everis' quality and methodology group validated and approved the method. Then pilot projects were performed To reduce risks when implementing the method we selected those projects with specific features: size: medium (duration time not greater than 3 months; staff: 4-7 people; budget: 100,000-15,000€; and project manager profile: junior project leader • The launch of the improvement processes was done through everis' intranet. The intranet allows the improvement process to be available for everis project managers
Lesson learned
• Analyze the main resistance and risk factors associated with improvement has allowed to take corrective actions to diminish them • Have a multi-model environment allows to select just those external practices that best fit with the way organization works • Carry out both analyses the dependences between internal and external practices and the facility of adoption, has allowed the selection of those practices that make more efficient process

• Keep a continuous monitoring and support throughout all implementation of process improvement process as well as an adequate communication strategy has allowed to diminish the change resistance of stakeholders • Select as pilot projects those projects that represent a low risk to the organization and that are formed by staff with early adopters profile has allowed to get a better feedback and therefore, a better analysis of the project improvement

4 Results of the software process experience

4.1 Analyzed data source

When everis introduce a new method, one of the main business goals is to increase the process adherence. The analysis was done using the everis internal audits as a source data. These audits allow to identify the noncompliance associated to key indicators related to everis business goals.

It is important to highlight that the noncompliance issues are problems identified in evaluations that reflect a lack of adherence to applicable standards, process descriptions, or procedures [12]. So, the analysis of the new process noncompliance associated with the indicator related to business goals allows to get information about the use of the new processes.

The noncompliance of key indicators in everis internal audits has been identified and analyzed through three fiscal years as Table 6 shows. The data has been collected at three times: before, during and after the project management improvement.

Table 7. Analyzed period

Fiscal year	Date	Period
FY'07	April 2007- March 2008	Before the software process improvement application
FY'08	April 2008- March 2009	During the software process improvement application
FY'09	April 2009- October 2010	After the software process improvement application

4.2 Control charts

A control chart is an analytic technique included in the group of techniques adapted from the field of mathematical statistics used for activities such as characterizing process performance, understanding process variation, and predicting outcomes [12].

Control chart allows to understand the process variation and to identify whether processes remain "in control" or stable over time, because the information provided by it is useful to obtain visual information of processes changes [13].

This chart shows simple statistical related to the process such as average and range: 1) *average*: the average of all of the data and 2) *range*: the difference between the Upper Control Limit, (UCL) and the Lower Control Limit, (LCL) [14].

Using Control charts as tool to analyze the noncompliance of process use at everis in order to establish the improvement results has allowed: 1) to understand the process variation; 2) analyze data to find patters, 3) monitoring the process performance during a specific period to detect change signals; and 4) communicate how a process has performed in a specific period of time.

4.3 Analysis of the process use

In order to measure the effects of the process improvement, three measures were defined: *process use, process useful* and *process performance* [10], [11]. This section shows the analysis of process use results as an experience of using MIGME-RRC methodology at everis.

Target the study on *process use* allowed to establish the process performance, because process use includes to analyze the degree of performance of best practices contained on the new processes. Process use analysis is focused on the three indicators that were selected according to the priorization done in the second phase of MIGME-RRC methodology (see Table 3).

The analysis of these indicators (*management rules, project planning and start of minutes*) was done at a corporative level and at office level. It is important to highlight that Europe and Latin America Offices were included. The name of the analyzed offices has changed for a confidentiality agreement done with everis offices are refereed as A, B, C and D. However, next a briefly description of the offices is presented: The office A is a european large office. The offices B, C and D are medium office, B is in Europe and, C and D are in Latin America. All of them offer services, which provide solutions to large companies of any sector based on three pillars: innovation, methodologies and efficiency through the use of specialist. The specialist use specific knowledge for each project and productivity in order to optimize the time and cost results.

The analysis of the three process indicators were collected at three data periods: 2007 (FY'07), 2008 (FY'08) and 2009 (FY'09). In order to get a better analysis, the data were divided in two groups as follows: group I (FY'07 and FY'08) and group II (FY'09). Below the analysis of each indicator by office is shown.

4.3.1 Management rules indicator

Figures 1, 2, 3 and 4 show the control chart for the management rules indicator of offices A, B, C and D respectively. In general, the controls chart show an improvement because of the average decreases from group I to group II. Besides, there is a shift of the upper limits from group I to group II, because of all them decrease. Moreover, there is a better control of new processes because the values are closer to the average.

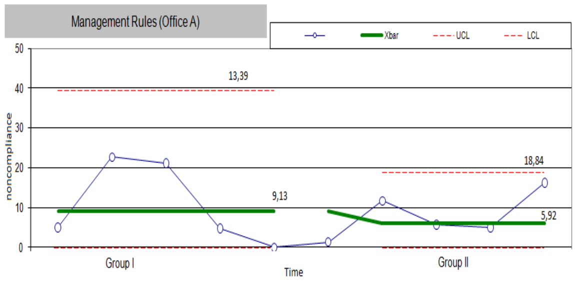


Fig. 1. Control chart of the management rules indicator office A

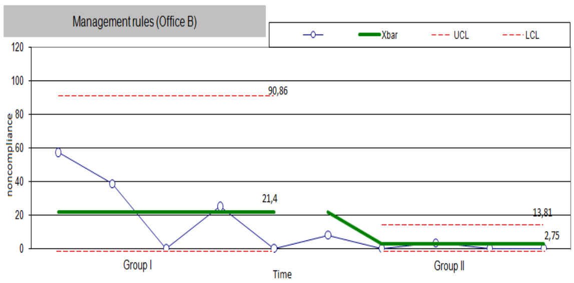


Fig. 2. Control chart of the management rules indicator office B

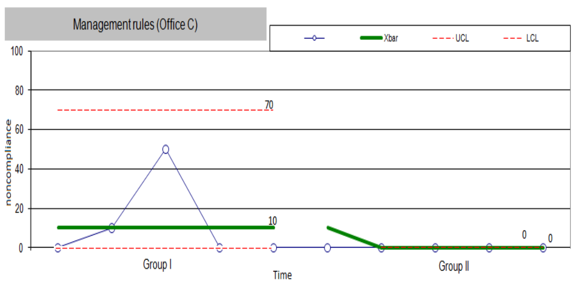


Fig. 3. Control chart of the management rules indicator office C

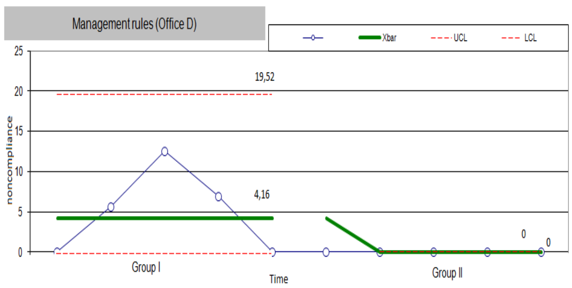


Fig. 4. Control chart of the management rules indicator office D

As figures 1, 2, 3, and 4 show when everis introduced the new project management method the degree of performance of best practices increase, therefore, it has a better business goals achievement because the noncompliances decrease. Then, it was increased the process adherence.

4.3.2 Project management indicator

Figures 5, 6, 7 and 8 show the control chart for the project management indicator of offices A, B, C and D respectively. In general, the controls chart show an improvement because of the average decreases from group I to group II. Besides, there is a shift of the upper limits from group I to group II, because most of them decrease. Moreover, there is a better control of new processes because the values are closer to the average.

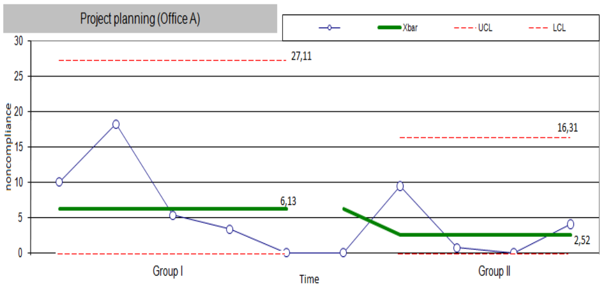


Fig. 5. Control chart of the project management indicator office A

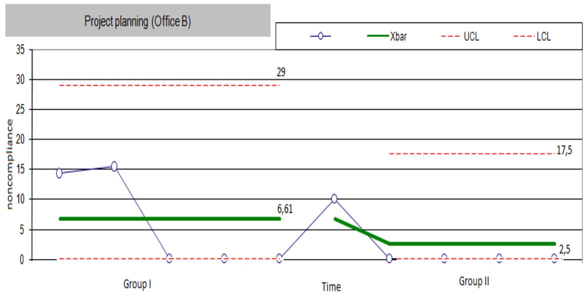


Fig. 6. Control chart of the project management indicator office B

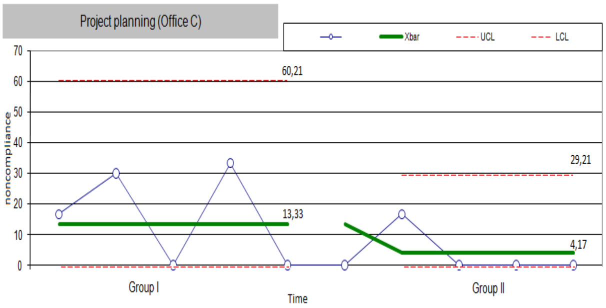


Fig. 7. Control chart of the project management indicator office C

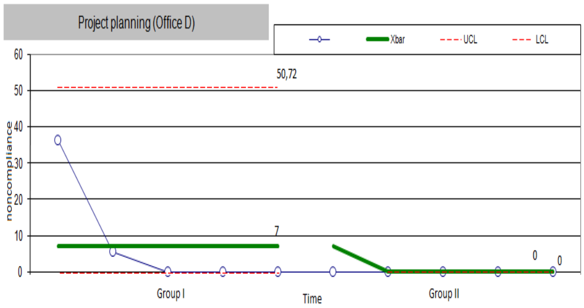


Fig. 8. Control chart of the project management indicator office D

As figures 5, 6, 7, and 8 show when everis introduced the new project management method the degree of performance of best practices increase, therefore, it has a better business goals achievement because the noncompliances decrease. Then, it was increased the process adherence.

4.3.3 Indicator start of minutes

Figures 9, 10, 11 and 12 show the control chart for the start of minutes indicator of office A, B, C and D respectively. The control charts show an improvement because of the average decreases from group I to group II. Besides, there is a shift of the upper limits from group Ito group II. Moreover, there is a better control of new processes because the values are closer to the average.

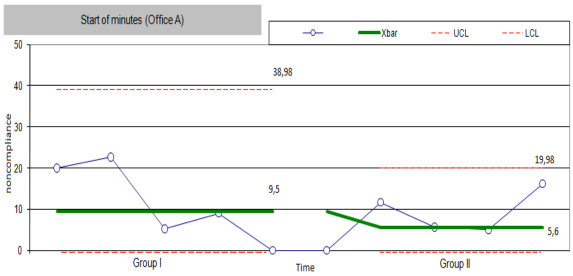


Fig. 9. Control chart of the start of minutes indicator office A

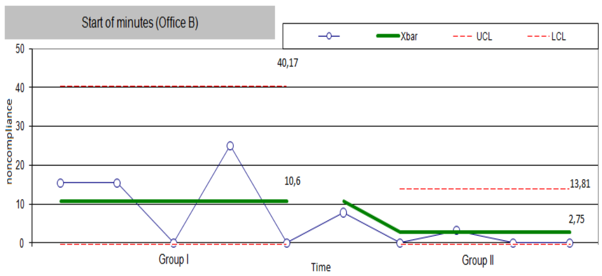


Fig. 10. Control chart of the start of minutes indicator office B

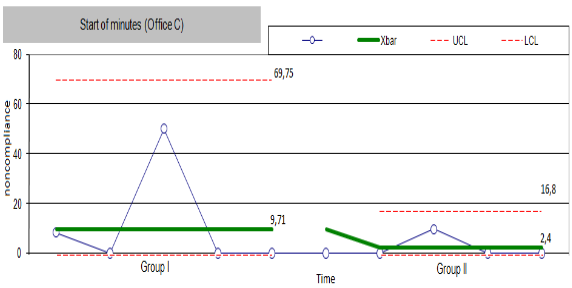


Fig. 11. Control chart of the start of minutes indicator office

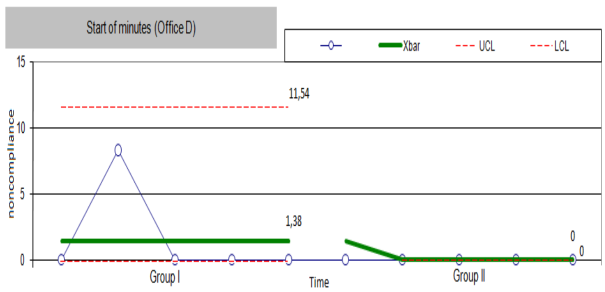


Fig. 12. Control chart of the start of minutes indicator office D

C

As figures 9, 10, 11, and 12 show when everis introduced the new project management method the degree of performance of best practices increase, therefore, it has a better business goals achievement because the noncompliances decrease. Then, it was increased the process adherence.

4.3.4 Summary of the offices level analysis

The analysis of the three indicators (*project rules, project management and management rules*) through everis offices (*A,B,C and D*) has shown a decrease in the average of nonconformance related to the new processes.

However, the nonconformance average decrease in a different way among offices because: 1) the quantity of managers considered as innovators and first adapters is not the same; 2) each office has its own capacity of change adoption and 3) each office has its own needs of practices that makes its processes more efficient.

5 Conclusions

Make a software process improvement establishing and using a multi-model environment as reference model has allowed the adoption of external practices which best fit with the way the organization works. Therefore, it is possible to get more efficient processes because those practices have a better acceptation.

As a result the staff's resistance to change, when implementing the improvement, was prevented, because they perceive the practices contained in the new processes as an evolution of the way they work. It is reflected by the project management method acceptation.

Particularly, the experience of using MIGME-RRC methodology for improving the project management processes at everis, has allowed that each office could establish its own software process improvement implementation pace, depending on the capacity of change adoption of each office. This was confirmed by the analysis of indicators related to everis business goals. Besides, a better adherence to new processes reflects that a gradual and continuous improvement has been achieved.

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

6 Literature

1. Mathiassen, L., Ngwenyama, O. K., AAEN, I. 2005. Managing change in software process improvement. IEEE Software, doi: 10.1109/MS.2005.159

2. CMMI appraisal program. 2010. CMMI for development SCAMPI Class A Appraisal Results 2009 End-Year Update

3. Cuevas, G., de Amescua, A., San Feliu, T., Arcilla, M., Cerrada, J. A., Calvo-Manzano J. A. and García, M. “Gestión del Proceso Software”. Universitaria Ramón Areces (Ed.). Mejora del Proceso Software, pp. 406-447, (2002)

4. Kautz, K., Levine, L., Hefley, B., Johansen, J., Kristensen, C., Nielsen, P. “Networked Technologies-The Role of Networks in the Diffusion and Adoption of Software Process Improvement (SPI) Approaches”. Software Engineering Institute (SEI). http://www.sei.cmu.edu/library/assets/sw-approaches.pdf. (2004)

5. Mathiassen, L., Ngwenyama, O. K., Aaen, I. “Managing change in software process improvement”. IEEE Software. doi: 10.1109/MS.2005.159, (2005)

6. Diane L., Gibson D., Goldenson, R., Kost K. Performance Results of CMMI ® -Based Process Improvement. August 2006 TECHNICAL REPORT CMU/SEI-2006-TR-004 ESC-TR-2006-004 (2004)

7. Conradi, H., Fuggetta, A. Improving software process improvement. IEEE Software. Vol 19. Issue 4. pp. 92–99. doi: 10.1109/MS.2002.1020295.(2002)

8. Potter, N., Sakry, M. “Making Process Improvement Work”. Addison-Wesley (Ed.). Developing a Plan. pp. 1-49. (2006)

9. Morgan, P. “Process Improvement- Is it a lottery?”. Software development magazine: UML, Agile, programming, software testing, quality assurance and project management. http://www.methodsandtools.com/archive/archive.php?id=52. (2007)

10. Calvo-Manzano Jose A., Cuevas Gonzalo, Muñoz Mirna, San Feliu Tomás. Metodología para la Implementación Gradual de Mejoras, con Enfoque en la Reducción de la Resistencia al Cambio. Proceedings 4ª Conferencia Ibérica de Sistemas y Tecnologías de la Información (CISTI 2009). (2009)

11. Calvo-Manzano, J. A., Cuevas, G., Gomez, G., Mejia, J., Muñoz, M., San Feliu, T., Sanchez, A.: Methodology for Process Improvement through Basic Components and Focusing on the Resistance to Change. EuroSPI 2009 Industrial Proceedings. ISBN 978-87-7398-151-1. (2009)

12. Calvo-Manzano, J. A., Cuevas, G., Gómez, G., Mejia, J., Muñoz, M., San Feliu, T. Methodology for process improvement through basic components and focusing on the resistance to change. Journal of Software Maintenance and Evolution: Research and Practice. (2010)

13. Chrissis M., Konrad, M. & Shrum, S.: CMMI Second Edition: Guidelines for Process Integration and Product Improvement. (eds.) Addison Wesley, pp.3-4, 31-42, 401-425 (2007)

14. iSixSigma. iSix Sigma Quality Resources for Achieving Six Sigma Results. www.isixsigma.com (2010)

7 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 PhD in Computer Science, a Master in Computer Science (2003-2005) and Computer Science Engineer (1996-2000) from the Institute Technological of Orizaba University of México. His current research interest consists on outsourcing process improvement. He has collaborated with an everis Foundation in research projects.

Muñoz Mirna

She is a PhD in Computer Science, a Master in Computer Science (2002-2004) and Computer Science Engineer (1996-2000) from the Institute Technological of Orizaba University of México. Her research field is Software Engineering. Her current research interest consists of the development of a methodology whose goal is to incorporate elemental process improvement components. She has collaborated with everis consultants in research projects.

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.

Modeling Software Development Practices using Reusable Project Patterns: A Case Study

Diego Martín, Javier García, Julián Urbano, Antonio Amescua

Abstract

Software patterns are a proven solution for information management that allows us to formalize the necessary knowledge to solve recurring problems. In process engineering they are useful too, as they can help formalizing knowledge about development methodologies, reference frameworks, best practices and process within an organization. One of the main problems for an effective application of process patterns in the software industry is the difficulty of formalizing the knowledge about the development process, specially the one coming from experience or particular of the company.

In this paper we discuss the main success factors and problems for software process engineers to be able to model the best practices of software engineering by modeling process patterns. Also, we propose and use a model called RPP (Reusable Process Pattern), containing the main information elements we can find in any development process.

The authors carried out a case study in the University Carlos III of Madrid, where senior Computer Science students used RPP. With this study we show that the quality of the pattern formalization is correlated with the quality of the bibliographical sources to extract information from, the experience on specific software engineering methods and the experience in formalizing knowledge. We also analyze the usefulness of RPP and the modeling easiness of each of the elements that compose RPP.

Keywords

Software Process Technology, Knowledge Management, Process Patterns, Experience Reuse, Patterns Modeling.

1 Introduction

Projects to improve the Software Development Process are very expensive, due to the great human effort and time necessary for their completion [1]. In these projects there are two main activities: the definition of the process and its deployment [2]. This paper is focused on the process definition.

In order to reduce the complexity related to Software Process Improvement, there is a research area centered on the introduction of efficient practices in a software organization by means of process patterns [3]. Process patterns is a problem-solution pair where the problem is the use of a software development efficient practice and the solution is a set of activities, templates and guidelines that help software engineers to apply efficiently the mentioned practice [4],[5],[6],[7].

As process patterns are considered useful knowledge management mechanisms in the software engineering area [8], but there is an absence of research studies discussing the process and the problems to elaborate new process patterns. In order to discuss this problem, this research work tries to identify the key factors to elicit the knowledge to be considered in the definition of process patterns to in software engineering.

In the scope of this research work, the authors consider a specific type of process pattern called RPP (Reusable Project Pattern) [9]. It was designed to facilitate the formalization of contents and to help in the adaptation of the patterns to be as effective as possible. Moreover, it offers an information retrieval system to help indexing and searching for relevant information from queries in natural language, structured information and metadata.

We focus on two main aspects to study RPP: effectiveness and usefulness. Effectiveness is considered as the capacity to formalize contents and the quality of the formalization of the RPPs produced. Usefulness is understood as subjective assessment of the RPP pattern to formalize knowledge regarding methodologies, best practices and reference frameworks.

This case study has been carried out with senior Computer Science students from the University Carlos III of Madrid. Twelve students participated playing the role of software engineers, each modeling three process patterns based on the best practices proposed by eXtreme Programming [10], SCRUM [11], Rational Unified Process [12] and Craig Larman [13].

The rest of the paper is organized as follows. In section 2 we outline the literature concerning the use of patterns. Next, we show the pattern model used and its information elements, and then discuss its advantages. Section 4 presents the research methodology we followed and Section 5 summarizes the results. The paper finishes with a section containing the conclusions gathered in this study.

2 Related Work

Within Information Science, we explore the knowledge management area, as one of its goals is to endure the activities that support the improvement of software processes, providing a positive interaction that both the organization and the improvement programs can benefit from [14]. Many authors have investigated the stages involved in knowledge management [15]. These stages are: data, information, knowledge and wisdom. In knowledge management there are several models that need to be mentioned. Bots and Bruijin [16] define a model for knowledge management, but it is actually a mere knowledge process chain that offers some added value. Massey et al. [17] offer a model based on understanding the organization, the users, and how the information flows. This study shows the success that can be achieved with the right knowledge management. Burke and Howard [15] describe a model whose success rests on the effectiveness of knowledge management in terms of the capacity of its infrastructure and the capacity of the knowledge process. Jennex and Olfman [18] offer a model showing that the effectiveness of the organization is improved with the development of an effective knowledge management system. Maier [19] describes a model very similar to [18], but with several modifications in certain dimensions to be considered that must be taken into account when designing the knowledge management system.

In 1994, Victor Basili [20] proposed a set of experience repositories to improve the flow of information and its reuse, and coined the concept “Experience Factory” for those organizations whose learning is based on knowledge repositories. Ever since, many works have focused on how to implement these knowledge repositories [21]. To be able to implement this concept, several technologies have been used, such as EPG, wikis, PAL, semantic webs, etc.

In 1998, Kellner et al. [22] proposed the EPG (Electronic Process Guides), defined as documents oriented to the workflow, structured to perform a process. Chau and Maurer [23] proposed “Wikis in Software Engineering” to represent experience repositories using Wiki technology [24], but they were not introduced until 2005. Then, PAL (Process Asset Library) is offered as a well indexed and organized repository, with process assets easily accessible by people requiring the process information. Then, Antunes et al. [25] proposed the use of semantic webs to describe a SRS (Semantic Reuse System) to acquire, manage and reuse the knowledge regarding the software development process.

There are many definitions for pattern in the Software Engineering literature, but the first one was given as an architectural concept in 1979 by architect Christopher Alexander [26]. There is not one single pattern model [27], as it ultimately depends on the author who creates them. The most extended and well known models are: the Alexander model [3], that presents a narrative style of the problem intended to be solved, the context in which to apply the patterns and the proposed solution; the GoF model [28] is proposed as a template with elements such as pattern name and classification, intent, also known as, motivation, and applicability; the canonical model or POSA model [29] can be used to describe patterns, and it offers the same sections as the Alexander format, though with more detail and precision.

Six types of pattern have been identified in Software Engineering. Process patterns describe an approximation successfully proven in several steps of the software development process using the object oriented paradigm [4]. Software Process Improvement patterns attempt to improve the quality by applying best practices [5]. Management and control patterns propose a language to properly apply the process into the SCM (Software Configuration and Management) [6]. Organizational patterns describe a management technique or a potential organization structure [30]. Collaborative patterns describe a set of ten patterns to help in the interaction of people's work [7]. Implementation patterns, which can be further categorized as: reference architectures, created for a particular domain of interest; architecture patterns, to represent a fundamental structural scheme of an organization in software systems; analysis patterns, a set of concepts to represent a common construct in business modeling [8]; and design patterns, to define, motivate and systematically explain a general design to address a recurring problem in designing object oriented systems [28].

In all these works we can observe a variety of types and pattern models, classified according to their goal. However, it is very difficult to find bibliographic references with serious studies on process patterns designed to help people create patterns to capture as much information as possible and hence to increase the quality of the patterns generated. Moreover, it is very difficult to find studies on the utility of a pattern from the point of view of the people who use it to model information, the information elements found the most relevant and the ones that tend to be the most difficult to model. We present a model oriented to the reuse of development processes (RPP), to aid in modeling and reusing the software development processes.

3 The RPP Model

In this paper we analyze what factors have an effect on the quality of software development process patterns at the time of their creation. To carry on this study we have decided to adopt the RPP model to formalize the patterns. RPP is a formalization model aimed at reusing the software development process, integrating indexing and knowledge searching capabilities provided by Lucene [31], Lemur [32] or RSHP [33], where natural language can be used to issue queries without the need to impose an specific formal language.

The RPP model is defined as a problem-solution pair. In the problem side we specify the types of software development projects the pattern is advised for, and in the solution side we define a set of information elements such as activities, workflows, product flows, best practices, etc. proposed by the software development methodologies, reference frameworks and best practices, all of which help solving the problem described in the RPP. To organize the information in the RPP we have designed a data model, see Figure 1.

In the problem definition side we have the following information elements: a textual description of the methodology, reference framework, best practices, etc.; a set of metadata for quantitative classification of the projects; a set of requirements we should be able to satisfy to apply the solution given in the pattern; and a set of risks we would assume if applying the solution.

In the solution side we have the following information elements: a WBS (Work Breakdown Structure) with a tree structure to organize the methodology activities; a workflow indicating the recommended sequence to perform the WBS activities; a product flow indicating how the products flow between activities; and a set of "to-does" with recommendations based on the best practices and lessons learned, though not modeled by the workflow.

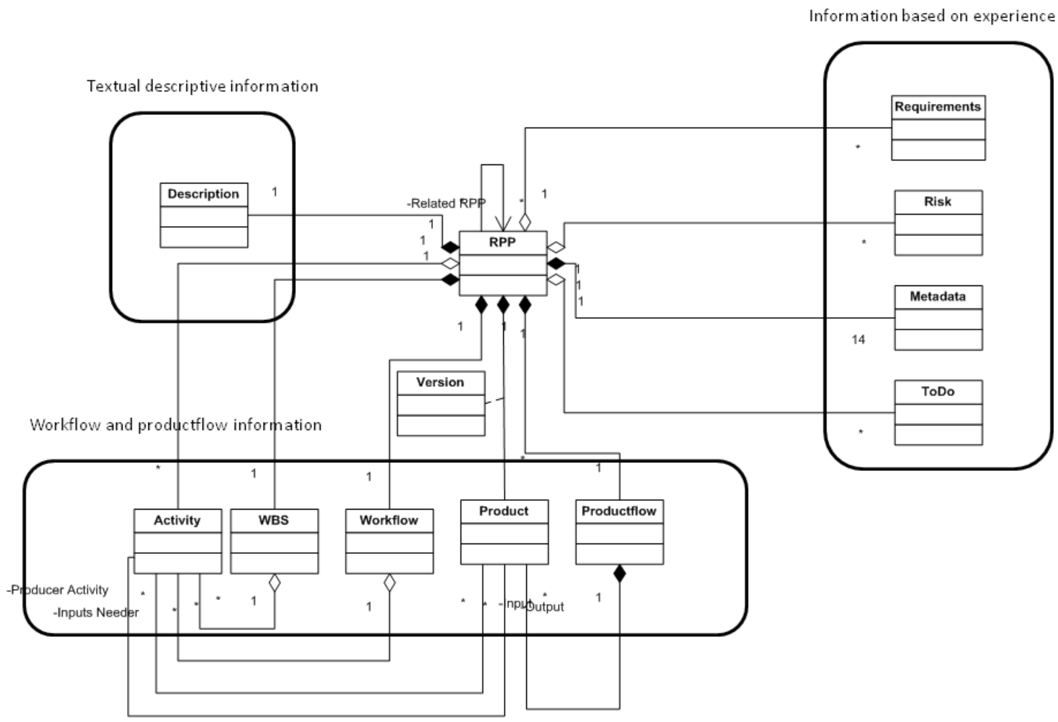


Figure 1: The RPP Model.

We have developed a collaborative tool called sdpReuser for the management and execution of the RPP patterns. This tool offers three types of functionality, see Figure 2. First, some functionality is oriented to the process managers, allowing for the creation, categorization and edition of RPPs. Second, it offers software engineers the possibility to implement a methodology to guide in the execution of the patterns they decided to implement. Third, it permits indexing and searching of RPPs, transparent to any user, and helping in the query specification with natural language text, query by example, workflow graphics, WBSs, metadata breakdown, files making up the products or a mixture of them.

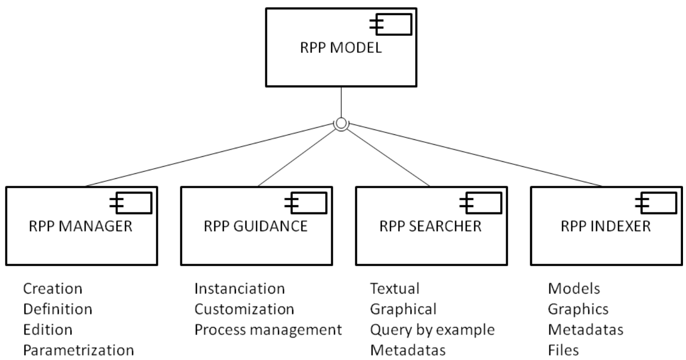


Figure 2: RPP architecture and functionality.

4 Validation Description

4.1 Objectives

The objective of this research work is to analyze what factors that influence in the knowledge elicitation and formalization activities required to define Reusable Project Patterns, analyzing the factors that influence on the quality of the project patterns elaborated and the usefulness of these artifacts to model the knowledge on effective practices required in software engineering projects.

From the effectiveness standpoint, we study the quality of the patterns elaborated by the subjects, analyzing what factors determine a correct modeling of each of the information elements a pattern must include. From the usefulness standpoint, we study the subjective assessment, by software engineering practitioners, of the project pattern techniques to identify and formalize best practices, as well as difficulties found in their application. We carry out not only a global study on the utility of RPP, but also a study on the subjective assessment of the utility of each of the elements in the project patterns.

4.2 Context

We have chosen four agile methodologies to model: XP, SCRUM, RUP and Craig Larman. We followed an orthogonal fractional design, where each of the 12 subjects modeled 3 of the 4 methodologies, summing up to 36 models. Nine different practitioners modeled each methodology. We chose senior students because their education and training is similar to industry professionals who could be responsible of formalizing RPPs. This is a common profile in innovative IT companies with little background on software development processes and information modeling.

4.3 Activities

As seen in Figure 3, case study is divided in three phases. The training phase is divided in two parts: first, an RPP expert trained the subjects on the model and during several lectures; second, the subjects had to gather information on each of the methodologies they had to model, and they could ask the experts for help despite this was a self-taught process. During the modeling phase, each subject had to model the three methodologies assigned aided by an RPP pattern modeling software. In the evaluation phase, the 12 participants had to fulfill a survey on the use of RPP where they were asked about the formalization of information using RPP.

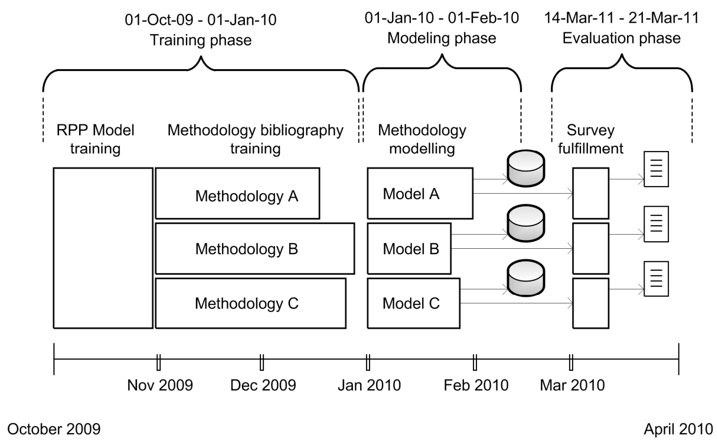


Figure 3: Study planning.

At the beginning of the case study all subjects were given the same guidelines to model the methodologies and all of them received the same training courses. Moreover, all of them had similar expertise.

4.4 Data Collection and Analysis

In the training phase, the expert kept track of the number of times the subjects asked for help in comprehending both the RPP model and each of the methodologies. By the end of the modeling phase the subjects handed in their models, from which we obtained a set of RPP and software modeling statistics. In the evaluation phase we distributed a survey for each model implemented and the set of bibliographic sources used in each case.

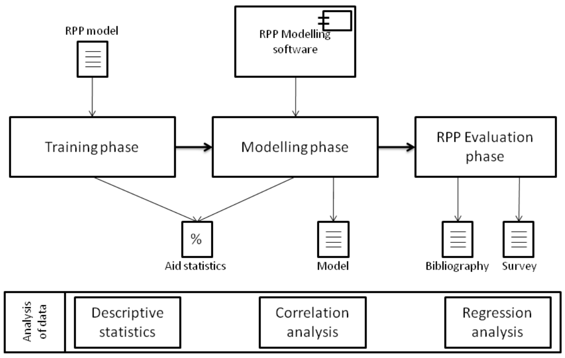


Figure 4: Inputs and outputs of the activities studied.

The pattern models generated by the subjects in the modeling phase were evaluated by two experts. Each expert assigned a quality score to each model, from which an average model quality score was assigned. In the last phase we asked the subjects for the set of bibliographic sources used for each methodology they modeled. Each bibliographic set was evaluated by an expert to assess its quality. By the end of this phase the subjects fulfilled a survey for each methodology they modeled, asking for the utility, relevance, satisfaction, easiness and frustration in the application of RPP to model the methodologies and each of its elements.

5 Results

5.1 Effectiveness

Two experts evaluated every pattern modeled, assessing the quality of the RPP elements following the criteria shown in Table 1. The overall model quality is the aggregate of the scores assigned for each of these criteria. To check for significant differences between the scores of the two experts, we calculated their agreement using Cohen's Kappa (squared weights). The agreement score is κ=0.6, showing a moderately high agreement.

RPP Element	Analyzed Questions	Weight
Description	Is the description accurate?	8
	Is the description size correct? (No more than 250 characters)	2
Workflow	Are the activities modeled as such? (Activity conceptual identification)	6
	Are the activities from the original methodology modeled?	7
	Do the activities have the correct level of abstraction?	2
	Is the workflow sequence correct?	15
	Is the workflow semantics correctly used?	10
Productflow	Are the products properly identified as such?	8
	Are all the products properly identified?	7
	Are the product dependencies properly identified?	5
To-does	Are the "to-does" set and "not to-does" set properly identified?	10
	Have they been mixed up with other elements of the model?	5
Metadata	Were any metadata identified?	3
	Are they significant?	2
Requirements	Were any requirements identified?	3
	Are they significant?	2
Risks	Were any risks identified?	3
	Are they significant?	2

Table 1: Questions analyzed in model quality.

Table 2 shows a summary of the average effectiveness per methodology, in a range from 1 to 5.

	SCRUM	RUP	XP	Craig Larman	Average
Quality of bibliographic sources	3.61	3.61	2.67	2.22	3.03
Help requests for methodology	1.33	1.33	1.67	1.56	1.47
Help requests for RPP	0.56	0.56	1.11	0.83	0.76
Model quality	3.53	3.51	3.43	3.11	3.39

Table 2: Effectiveness summary for each methodology.

The quality analysis of the bibliographic sources has been carried out by an expert, adding one point to the score for each RPP element contained in source. This score was normalized from 1 to 5 for the scale to be consistent with the rest of the variables. The overall quality of a methodology's bibliography is calculated as the median score of its set of bibliographic sources. Help requests for RPP and help requests for methodology measure the number of times each subject asked for help to understand the RPP model or the particular methodology. These scores were gathered by the experts who provided advice to the subjects, and it is again normalized from 1 to 5.

As shown in Table 2, the average model quality is 3.39 over 5. The minimum value observed was 2.23 for a subject modeling Craig Larman, while the maximum observed was 4.43 for a different subject modeling RUP. Overall, 75% of the observations were between 3 and 4.

Nonetheless, the quality obtained across RPP elements is very diverse. Figure 5 shows the distributions of quality scores for each RPP element. We can differentiate three groups: description, workflow and product flow have quite high scores; metadata, requirements and risks have quite low scores; and the set of to-does is the mid-levels, corresponding to recommendations on the effective practices application.

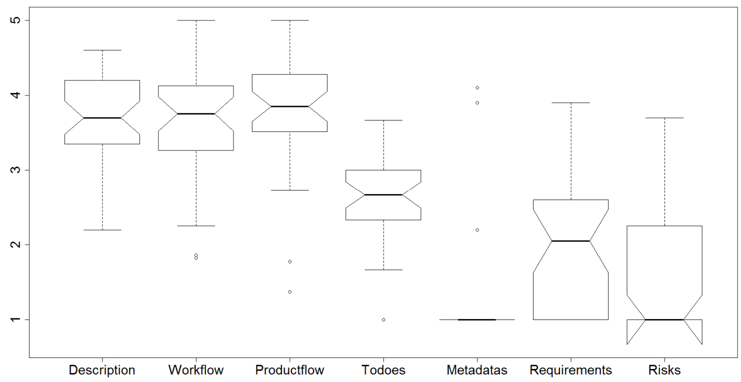


Figure 5: Distributions of RPP elements quality.

A multiple regression was found (R^2=0.833) among the variables: quality of the references supporting the knowledge elicitation processes (B), experience on the application of effective practices modelled (EM), and the explicit knowledge on the formalisms used to model the effective practices (ER). The equation is as follows:

$$\textit{Pattern Quality} = 2.4313 + 0.288 \cdot B + 0.1326 \cdot EM + 0.1001 \cdot ER$$

The first predictor is the quality of the references used to elicit the knowledge required for the project pattern that is being modeled. The quality of the references depends on the inclusion of information on the formal elements of the patterns (activities, workflow, products and product flow).

The second predictor of project patterns quality is related to the involvement of practitioners having the required experience on the lessons learnt on the application of the effective practices to be included on the project pattern. Normally, this experience related issues cannot be elicited from public references because are part of the tacit knowledge that the software engineers have in a specific business domain.

The last predictor of the pattern quality is the knowledge on the formalisms used to model the effective practices. Normally, practitioners elaborating project and process patterns do not have previous training on the use of these formalisms. In the case study performed, the quality of the project pattern was influenced by the number of question-answer sessions carried out with each practitioner.

5.2 Usefulness

In the survey, subjects were asked about their perceived overall usefulness of RPP for each methodology. The evaluation was very positive: 15 (47%) of the observations had the maximum score, with an average of 3.92 and very similar scores across methodologies. We studied the usefulness with more detail, asking for the perceived utility, relevance, satisfaction, easiness and frustration when modeling each of the elements in RPP. Table 3 shows the scores averaged per methodology and overall. Frustration scores are reversed so that larger values mean more usefulness.

	SCRUM	RUP	XP	Craig Larman	Average
Utility	3.78	4.00	4.22	3.67	3.92
Relevance	3.36	3.41	3.42	3.18	3.34
Satisfaction	3.48	3.72	3.68	3.43	3.58
Easiness	2.23	2.51	2.46	2.37	2.39
Frustration	2.86	3.28	2.94	2.64	2.93

Table 3: RPP usefulness summary for each methodology and overall.

Differences can be found among RPP elements, see Figure 6. The most troublesome elements are clearly the metadata, because they need knowledge and experience to be successfully modeled. Both workflow and product flow are the next most difficult elements to model, although they are quite well evaluated in terms of utility because they contribute a lot of information to model a methodology.

Comparing the easiness and frustration scores, we can see they are very similar for descriptions, workflows and product flows. However, the todo-es seem to be very frustrating yet easy to model and apparently useful. This is probably caused by the difficulty in finding this kind of information in the literature, although they seem to be easy to model once found. Metadata, requirements and risks are very difficult to model and little frustrating. In addition, their utility scores are quite low, and so we can conclude that they are hard to find in the literature and yet they do not turn out frustrating because they are not perceived as useful in the first place.

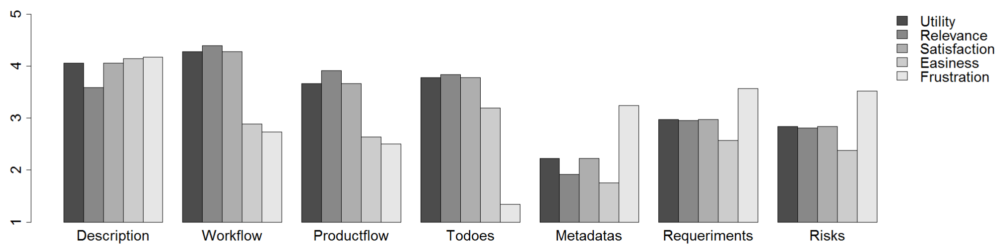


Figure 6: Utility, relevance, satisfaction, easiness and frustration for each RPP element.

6 *Conclusions*

Patterns are very useful for knowledge management, and they can be applied in virtually any area where information needs to be formalized. Process engineering is one such area where patterns are not only useful, but also necessary to deploy the software development process in an effective and efficient manner. In this paper we have focused in a previous stage, where these patterns are defined and formalized.

We present the RPP model to formalize process patterns. It allows to model the most commonly used information elements in development methodologies, reference frameworks and best practices. Moreover, RPP is supported by a framework that helps in the reuse of development processes and the collaboration among users of RPP.

We have carried out a case study to analyze what factors have an effect on the effectiveness of the process pattern formalization, as well as the usefulness of the pattern itself and each of the elements it contains. We can conclude that the effectiveness of the formalization depends on the quality of the bibliographical sources, the experience of the subjects and the model chosen to formalize patterns.

A high-quality bibliographic set is the one that addresses some of the elements proposed by RPP. These are the elements that we may find in most development methodologies, reference framework and best practices. As to the experience of the subjects, it is very important to have training mechanisms to provide the tacit knowledge that an expert has, because this information is hardly formalize in the literature and, when found, it is of rather low quality. The model to formalize patterns is a key factor, as it must facilitate information exchange and collaboration among experts. In this paper we show that RPP is a useful model to elicit knowledge.

7 Literature

[1] J. Garcia, Y. Rimawi, M. Sánchez, and A. Amescua, "RAMALA: A knowledge base for software process improvement," *Lecture Notes in Computer Science*, vol. 3792, 2005, pp. 106-117.

[2] M.I. Sanchez-Segura, F. Medina-Dominguez, A. De Amescua, and A. Mora-Soto, "Improving the efficiency of use of software engineering practices using product patterns," *Information Sciences*, vol. 180, 2010, pp. 2721-2742.

[3] C. Alexander, S. Ishikawa, M. Silverstein, M. Jacobson, I. Fiksdahl-King, and S. Angel, *A pattern language*, Oxford Univ. Pr., 1978.

[4] S.W. Ambler, *More process patterns: delivering large-scale systems using object technology*, Cambridge Univ Pr, 1999.

[5] B. Appleton, "Patterns for conducting process improvement," *Proceedings of PLoP*, Citeseer, 1997.

[6] S.P. Berczuk and B. Appleton, *Software configuration management patterns: effective teamwork, practical integration*, Addison-Wesley Professional, 2003.

[7] I. Sommerville, *Software Engineering (7th Edition)*, Addison Wesley, 2004.

[8] M. Fowler, *Analysis Patterns: reusable object models*, Addison-Wesley, 1997.

[9] D. Martín, J. García, A. Amescua, and J. Llorens, "Reusable Project Patterns to enhance Software Process Improvement," *European Systems and Software Process Improvement and Innovation Conference (EuroSPI'2007)*, 2007.

[10] K. Beck, *Extreme programming explained*, Addison-Wesley Professional, 2001.

[11] K. Schwaber and M. Beedle, "Agile software development with Scrum," 2001.

[12] P. Kroll and P. Kruchten, *The rational unified process made easy: a practitioner's guide to the RUP*, Addison-Wesley Professional, 2003.

[13] C. Larman, *Agile and iterative development: a manager's guide*, Prentice Hall, 2004.

[14] T. Dingsøyr, F.O. Bjørnson, and F. Shull, "What Do We Know about Knowledge Management? Practical Implications for Software Engineering," *IEEE Software*, vol. 26, May. 2009, pp. 100-103.

[15] G.D. Burke and W.H. Howard, "Knowledge Management and Process Improvement: A Union of Two Disciplines.," *CrossTalk - The Journal of Defense Software Engineering*, vol. 18, 2005.

[16] P.W.G. Bots and H. de Bruijn, *Effective knowledge management in professional organizations: going by the rules*, IEEE Comput. Soc, 2002.

[17] A. Massey, M. Montoya-Weiss, and T. O'Driscoll, "Knowledge Management in Pursuit of Performance: Insights from Nortel Networks," *Management Information Systems Quarterly*, vol. 26, 2002, pp. 269-289.

[18] M.E. Jennex and L. Olfman, "Modeling knowledge management success," *Proceedings of the Conference on Information Science and Technology Management*, 2004.

[19] R. Maier, *Knowledge management systems: Information and communication technologies for knowledge management.*, Springer-Verlag, 2007.

[20] H. Basili, V. And Caldiera, G. And Rombach, "Experience factory. Encyclopedia of Software Engineering. JJ Marciniak," *John Wiley & Sons*, vol. 1, 1994, p. 469--476.

[21] A. Amescua, L. Bermón, J. García, and M.I. Sánchez-Segura, "Knowledge repository to improve agile development processes learning," *Software, IET*, vol. 4, 2010, pp. 434-444.

[22] M.I. Kellner, U. Becker-Kornstaedt, W.E. Riddle, J. Tomal, and M. Verlage, “Process guides: effective guidance for process participants,” *Proceedings of the Fifth International Conference on the Software Process*, 1998, pp. 11-25.

[23] T. Chau and F. Maurer, “A case study of wiki-based experience repository at a medium-sized software company,” *International Conference On Knowledge Capture*, 2005, p. 185--186.

[24] W. Cunningham and B. Leuf, *The Wiki Way: Quick Collaboration on the Web*, Addison-Wesley Professional, 2001.

[25] B. Antunes, N. Seco, and P. Gomes, “Knowledge management using semantic web technologies: an application in software development,” *Proceedings of the 4th international conference on Knowledge capture*, ACM, 2007, pp. 187 - 188.

[26] C. Alexander, *The timeless way of building*, Oxford University Press, USA, 1979.

[27] M. Fowler, “Patterns,” *IEEE Software*, vol. 20, 2003, p. 56--57.

[28] E. Gamma, R. Johnson, R. Helm, and J. Vlissides, “Design patterns: Elements of reusable object-oriented software,” *Addison-Wesley*, vol. 1, 1995, pp. 1-2.

[29] F. Buschmann, K. Henney, and D.C. Schmidt, *Pattern-oriented software architecture: On patterns and pattern languages*, John Wiley & Sons Inc, 2007.

[30] J.L. Isla-Montes and F.L. Gutiérrez-Vela, “Diseño en base a patrones . Aplicación a sistemas,” 2004.

[31] A. Jakarta, “Apache Lucene-a high-performance, full-featured text search engine library,” 2004.

[32] J. Allan, J. Callan, K. Collins-Thompson, B Croft, F. Feng, D. Fisher, “The lemur toolkit for language modeling and information retrieval,” *Retrieved January*, vol. 8, 2003.

[33] J. Llorens, J. Morato, and G. Genova, “RSHP: an information representation model based on relationships,” *Soft computing in software engineering*, 2004, p. 221.

8 Author CVs

Diego Martín
He is a GTA in the University Carlos III of Madrid and PhD student in Computer Science. His main research subjects are: Software process improvement, Development Process Reutilization.

Javier García
He 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.

Julián Urbano
He is a GTA and PhD student in the University Carlos III of Madrid. His research areas are Evaluation in Information Retrieval, Music Information Retrieval, Crowdsourcing and Research Methods.

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.

Toward Lean Development in Formally Specified Software Processes

María Cecilia Bastarrica[1], Julio Ariel Hurtado Alegría[1,2], and Alexandre Bergel[1]

[1] Computer Science Department, Universidad de Chile
Blanco Encalada 2120, Santiago, Chile
[2] IDIS Research Group, University of Cauca
Street 5 # 4–70, Popayán, Colombia
{cecilia,jhurtado,abergel}@dcc.uchile.cl

Abstract

Formally specifying the software development process has been the way followed by several companies for making development more predictable. However this formality has frequently introduced bureaucracy into the process. Lean software development is an agile practice that promotes developing only those work products that are required, i.e., no waste should be included in the process. In this paper we present an automatic means of detecting and localizing the presence of certain type of waste in software processes that are formally specified using SPEM 2.0. We show our findings by analyzing the Scrum process model and the software development process model of a medium size software development company in Chile.

Keywords

Lean software development, software process improvement, software process model analysis

1 Introduction

Lean software development is an agile practice that promotes quality and productivity by focusing on core issues and not investing effort executing non essential tasks and building non required work products [16] [16]. Software companies tend to define and formalize their development processes in an effort to make them more predictable. This formalization is a hard task and implies a huge investment, so it is natural to try to get the highest return of investment out of it. Trying to cover all possible cases, it is not rare to introduce unnecessary tasks and work products as part of the formalized process, and this may build up waste into the process. It is not easy to identify the existence of waste, and even harder to localize it within the process. A typical case of waste is developing work products that nobody needs, i.e., that are neither deliverables nor required for executing any task within the process. We will focus on this kind of waste.

We have developed *AVISPA*, a tool for visual analysis of software processes. It is able to identify a series of error patterns that we have found to be frequent in practice [5] [5]. In this paper we extend *AVISPA* so that it is able to also identify certain type of waste –useless work products – in software processes formally specified in Eclipse Process Framework.

We apply the extended tool in two quite diverse scenarios: the Scrum process specification publicly available from the EPF Community, and the software development process of a medium size software company in Chile. In the former case, no waste of the type we are looking for (useless work products) has been found as expected, provided that Scrum is an agile method. In the latter case, we were able to identify and localize several waste elements, and all of them are opportunities for software process improvement.

The rest of the paper is organized as follows. In Section 2 a precise statement of the problem being addressed is detailed. The *AVISPA* tool and the mechanics of waste detection are described in Section 3. Section 4 shows the application of our tool for localizing waste in the two aforementioned software process models. Related work is discussed in Section 5. Finally, some conclusions and future work are described in Section 6.

2 Problem Statement: Localizing Waste in Formal Processes

Lean software development implies the application of seven principles [15] [15]: eliminate waste, build quality in, create knowledge, defer commitment, deliver fast, respect people, and optimize the whole. One of the most important of these principles is eliminating waste. But it is not necessarily clear the form that waste may take within a software process, and even less clear how it could be identified, let alone automatically localized. In this paper we will not necessarily focus on agile processes. Nevertheless, eliminating waste not only applies to this kind of processes; it may even be more relevant in the context of non agile processes as will be apparent from the experimentation presented in Section 4.2.

Formal processes unambiguously specify who does what and when, and which work products are built/modified as a result. SPEM 2.0 [13] is the OMG standard notation for specifying software processes, and the Eclipse Process Framework1 is a platform that allows the specification of SPEM 2.0 processes. In a lean software development process there should be no waste, therefore any work product should be either a deliverable or an input for some other task within the process itself, and it should be specified accordingly. SPEM 2.0 provides primitives for specifying that a work product is a deliverable. Therefore, if we find a work product that is neither specified as a deliverable nor as an input for any other task, then we are in one of the following scenarios, all of them problematic:

1. the work product is actually necessary for performing some task, but we have forgotten to specify it, so we have found an underspecification;

[1] EPF: http://www.eclipse.org/epf/

2. the work product is actually a deliverable, but it was not defined as so, so we have found another form of underspecification;
3. the work product is not really necessary, so we have found waste in the process: a useless work product.

3 Automatic Waste Detection

This section describes the whole procedure for automatically localizing the useless work product kind of waste in formal processes specified in EPF. First, we describe *AVISPA*, a tool for analyzing formal software processes. Afterwards, we describe the extension we propose to add to *AVISPA* so that it could also be applied for localizing waste.

3.1 Software Process Blueprints and AVISPA

Software processes may be composed by several hundreds of elements of diverse kinds. This issue makes sometimes difficult to analyze the quality of a process just through inspections. In [6] [6], we have proposed Software Process Blueprints that are partial views of the software process that allow the process engineer to visually analyze its quality. Each of these views focuses on one essential SPEM modeling element: role, task and work product, and thus we have a Role Blueprint, a Task Bluprint and a Work Product Blueprint. All of them are graphs formed by nodes and edges/arcs.

In the Role Blueprint nodes represent roles whose size corresponds to the number of tasks in which the role is involved. Also, an edge between two nodes represents the existence of collaboration between the two roles to perform a task. Therefore nodes that are too big may reveal overloaded roles, and disconnected nodes show roles that do not collaborate.

In the Task Blueprint nodes represent tasks, whose height is the number of input work products and whose width is the number of output work products for that task. An arc from one task to another, represents precedence, i.e., an output work product of the former task is an input work product for the latter task. In this way, very wide nodes suggest tasks without a clear goal, i.e., whose purpose is to produce a variety of work products.

In the Work Product Blueprint nodes are work products, where their height represents the number of tasks that require the work product as an input, and their width is the number of tasks that create/modify it. In this case a node that is too high reveals that certain work product is required by several tasks, and it therefore may become a bottleneck. Figure 1 depicts the Work Product Blueprint of the software process of a medium size company; there we can clearly see that the node corresponding to System Requirements Specification is much higher than most of the others suggesting that many tasks require it.

Also the existence of disconnected subgraphs in any blueprint reveals a misspecification in the process. However, in Process Model Blueprints, nodes that are much larger than others could suggest anomalies, but it is the responsibility of the process engineer to determine if they are actual errors, improvement opportunities, or if on the contrary they are defined that way on purpose. Moreover, it is not clear how big could be considered too big.

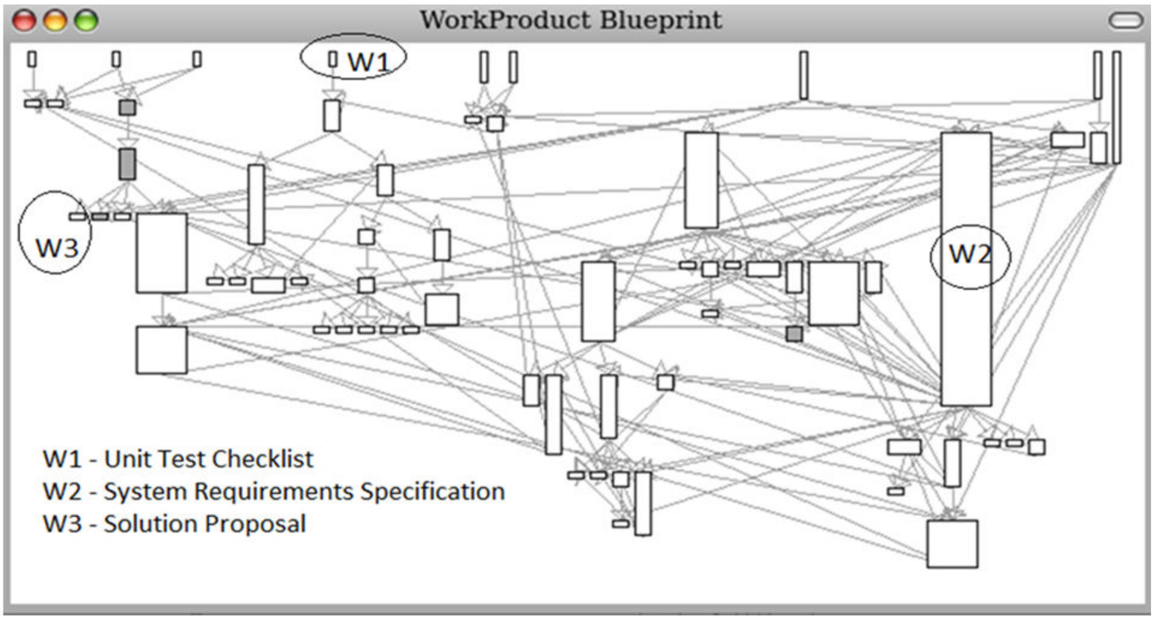


Figure 1: Work Product Blueprint of a Chilean software company

In [5] [5], we proposed *AVISPA*, a tool based on blueprints that automatically identifies and localizes a series of error patterns[2]. Table 1Table 1 describes those error patterns already identified. In *AVISPA* error patterns are highlighted in color, so that it is evident when there is a possible error. For this purpose some assumptions have been made such as defining that an element that is more than one standard deviation larger than the mean is considered too big and suggests the presence on an error. This assumption has worked fine in practice so far. For example, in Figure 2Figure 2 we show the Work Product Blueprint for the Scrum process; *AVISPA* has highlighted the Project Backlog as a work product that is too demanded, and as such it may be a bottleneck in the process as a whole. A thorough analysis of the Scrum process using *AVISPA* can be found in [4] [4].

Table 1: Error patterns identified by *AVISPA*

Error pattern	Description	Localization	Identification
No guidance associated	An element with no guidance associated	any blueprint	A completely white node.
Overloaded role	A role involved in too many tasks	Role Blueprint	Nodes over one standard deviation larger than the mean
Isolated role	A role that does not collaborate	Role Blueprint	A node that is not connected with an edge
Multiple purpose task	Tasks with too many output work products	Task Blueprint	Nodes that are more than one standard wider than the mean
Demanded work products	Work products required for too many tasks	Work Product Blueprint	Nodes more than one standard deviation higher than the mean
Independent subprojects	Independent subgraphs	Task Blueprint or Work Product Blueprint	Subgraphs that are not connected with edges

[2] AVISPA (Analysis and Visualization for Software Process Assessment): http://www.moosetechnology.org/tools/ProcessModel. AVISPA is freely available under the MIT license.

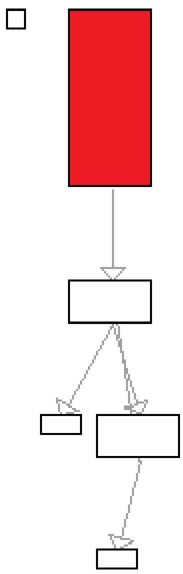


Figure 2: Identifying work products that are too demanded

3.2 Localizing Waste in Work Product Blueprints

Deliverables are those work products that need to be delivered to the customer as part of the final product. For example, a user requirements document and the source code are typical deliverable work products. In SPEM 2.0, some work products can be defined as deliverables so they could be easily identified.

As part of the software development process, not only deliverable work products are produced. There are other intermediate work products that are needed mainly for coordinating successive tasks probably performed by different people. For example, the test set is an output work product of the Design Test Set task and an input of the Execute Test Set task, but it is not necessarily a deliverable work product. However, if there are work products that are neither deliverables nor input for any other task within the process, they are a kind of waste we do not want to develop if we intend to have a lean process.

In the Work Product Blueprint, an arc connecting nodes represents precedence between work products. If there is a WP_a that precedes WP_b in the graph, that means that there is a task such that WP_a is its input and WP_b is its output. In this way, all leaves in the graph, i.e., nodes with no successor, should represent deliverable work products. In this paper *AVISPA* is extended so that it highlights in blue all those leaves that are not defined as deliverables. The process engineer then needs to analyze all highlighted nodes so that he/she could determine if each of the highlighted work products is actually required as an input of another task, and thus it is not a leaf, if it should have been defined as a deliverable and thus it should not have been highlighted, or if it is actually waste in the process and it is an improvement opportunity.

4 Application to Two Diverse Processes

In this section we apply the extended *AVISPA* for localizing waste in two software development processes. We apply our tool in two dramatically different scenarios. First we focus on the publicly available Scrum process model specification that can be found at the EPF Community web site[3]. A priori, Scrum, being an agile method, is expected to show no waste in its specification. Then we will proceed to analyze the software development process of a Chilean medium size software company. In this latter case we will see that looking for waste in real world software processes is not only much harder,

[3] Scrum: http://www.eclipse.org/epf/downloads/scrum/scrum downloads.php

but much more useful when identified provided that the size and complexity of the process model makes it almost impossible to analyze it manually.

4.1 Scrum

Scrum is an agile process used to rapidly develop software. It has been defined by Jeff Sutherland and more formally elaborated by Ken Schwaber [17] [17]. Scrum stresses management values and practices, and it does not include practices for technical parts (requirements, design, and implementation); this is why it is usually used in combination with another agile method such as Extreme Programming.

The application of Scrum enforces a few simple rules that have the potential to make a team self-organize into a process that can achieve 5 to 10 times the productivity of a waterfall-based process. However, most Scrum teams never achieve this goal [18] [18]. According to Sutherland, teams face difficulties to organize work in order to deliver working software at the end of each sprint. Moreover, they also experience trouble working with a Product Owner to get the backlog in a ready state before bringing it into a sprint. Also, organizing into a hyper-productive state during a sprint remains a challenging issue. Our findings analyzing the Scrum process model with *AVISPA* [5] are consistent with these ideas.

We claim that it may be the case that the publicly available Scrum process model may be misspecified, and thus people adopting it as it is may be using an inherently suboptimal process. We apply the extended *AVISPA* to the EPF community Scrum process model in order to look for waste and/or the other kinds of misspecification detailed in Section 2, from the point of view of the work products. Figure 3Figure 3 shows the results.

The Potentially Shippable Product Increment (A) has been highlighted. This work product needs to be an input to the integration task, but the public Scrum process model does not specify this fact, so (A) is an underspecification. The Release Burndown Chart (B) and Sprint Burndown Chart (C) are clearly necessary for executing the development tasks, but the model does not specify these dependencies either. They are also underspecifications. Therefore, the extended *AVISPA* is able to identify this kind of underspecifications even in a very small software process. But also from this analysis, we can see that no false positives are identified: all highlighted elements correspond to errors in the process model specification. Moreover, analyzing each highlighted element, we can confirm that there is no waste of the kind useless work product in the Scrum process model, as expected for an agile method.

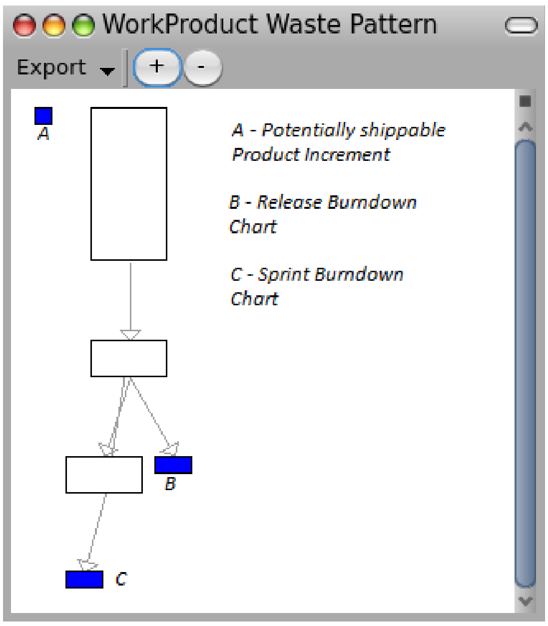


Figure 3: Work products that are potential waste in Scrum

4.2 Development Process of a Software Development Company

DTS is a Chilean software company that works in solutions for military and civil technology. It has around 250 employees, including engineers, certified technicians, operation workers and managers. Particularly, the Self-Service Systems Engineering Area in DTS (SSSEA-DTS) started to define its software process model in 2008, using the Rational Unified Process as a reference. In SSSEA-DTS software process improvement has been oriented toward recovering the software process currently applied in the organization, in order to formalize it, analyze it, and improve it if found necessary. SSSEA-DTS's process model is composed by 66 work products, 9 roles and 57 tasks. This model has been defined with a total effort of 0.5 person-months during 12 months.

Figure 4 shows the Work Product Blueprint in which blue nodes (dark colour on a black and white printout) identify potential waste work products or underspecifications. The tool highlights 22 problematic work products (33% of the whole), which include non-defined deliverables, underspecified task inputs and useless work products, i.e., actual waste.

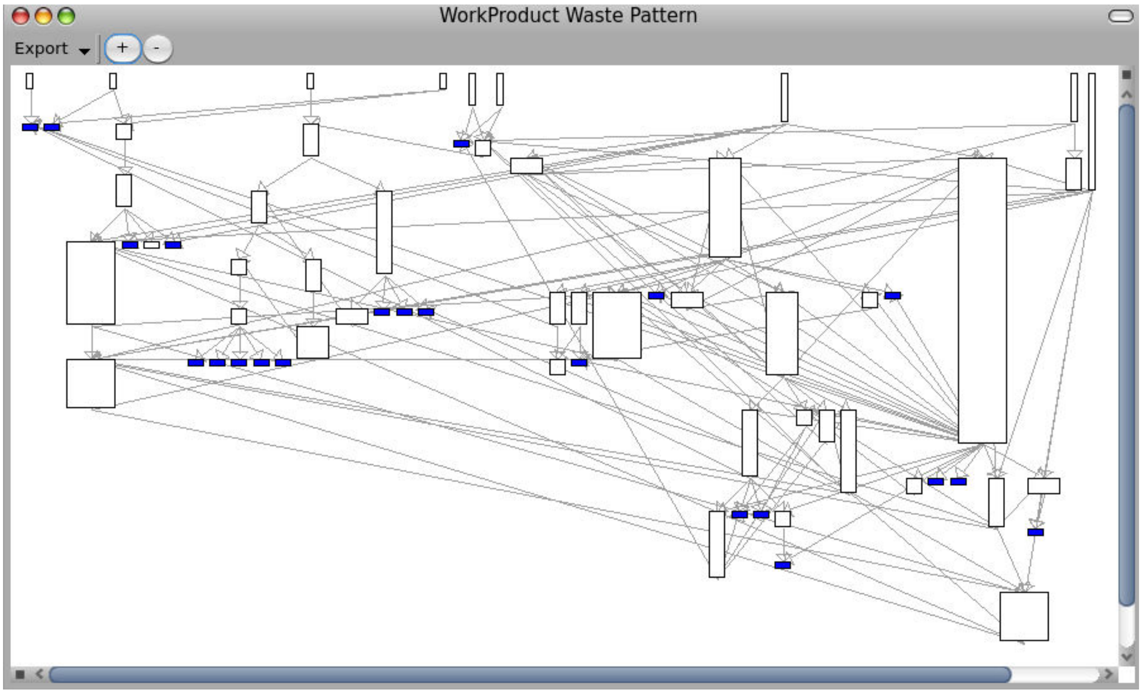


Figure 4: Work products that are potential waste in DTS

In order to validate our findings and assess the relevance of the waste work products, we confronted our blueprint with the SSSEA-DTS process engineer. From the 22 highlighted work products found, 3 have been confirmed to be non specified deliverables, 16 underspecified task inputs, and 3 are indeed waste work products.

In particular, these 3 work products correspond to evidence required for bureaucratic issues, i.e., evidence about the approval of other work products. For example, the Requirements Approval work product is the evidence of requirements acceptance by different stakeholders, with respect of the Requirements work product. This evidence could have been registered in each respective work product (for example as a field) instead of a new work product. But, according to the expert's opinion, and based on the implementation of software processes during the past five years, about 12 of the underspecified work products could have also been integrated as part of other work products, reducing bureaucratic work significantly. For example, activity specifications, operational state specifications, communication protocol specifications, and requirements observations from stakeholders can be all considered part of the system requirements work product, decreasing the effort to design, maintain, control and configure work documents. As a summary, the practical effectiveness of the tool has been confirmed with DTS by identifying 22 problematic work products: 13.6% percent of deliverables that had not been identified, 31.9% underspecified tasks (where the work product should have been specified as a task input), 13.6% waste and 40.9% could be improved (refactoring and integration).

The actual findings not only allowed the process engineer in DTS to improve their software process specification, but also allowed them to gain more confidence about the quality of their process.

5 Related Work

Software process improvement through the Lean Measurement (SPI-LEAM) method is proposed by Petersen et al [14] [14]. This method describes a way to implement lean principles through measurement in order to initiate software process improvement. The method uses collected data from projects executed to evaluate performance and quality aspects, particularly identifying causes of waste. Mujtaba et al [11] [11] propose a case study to identify waste in a software process by using value stream maps (VSM). The empirical data is collected using document analysis, extraction of phase times from a requirement tracking tool and interviews. It is used to construct a value stream map that shows the present state of the process. Static validation showed that the VSM methodology is useful for identifying waste and to propose measures to avoid it. Middleton et al. [9] [9] developed a study case where a company is followed using lean practices for two years. One of their most relevant findings was that the company had many non-value tasks. Data collected at the company showed an increment of 25% in productivity, schedule delay was reduced to 4 weeks from several months or years, and the time for defect fixing was reduced by 65% - 80%. In our approach, the waste error pattern is applied on the software development model, allowing part of the waste to be identified before the process model is actually enacted, whereas the previous approaches are based on data collected from the process applied in specific projects. So, the waste error pattern could be complementarily used with SPI-LEAM or VSM methodologies before a new process model is tested, as a form of static verification mechanism.

Visualization is regularly employed to identify deficiencies and errors in application source code. Polymetric views is a lightweight visualization representation, originally designed to analyze software source code. Polymetric views were first employed for reverse engineering [8] [8], code comprehension [1] and characterization purposes [2] [2]. Even though the application range of polymetric views has greatly expanded over the last few years [3] [3][10] [10], all these views make use of pattern recognition to visually identify abnormal situations. The blueprints and error patterns applied in this paper are no exceptions. However, as far as we are aware of, our work is the first usage of polymetric views to identify anomalies in software processes.

Knab et al. [7] [7] proposed a set of generic visual process patterns. With these patterns, the authors analyze effort estimation, length, and sequences of problem resolution activities. Based on the information obtained from issue tracking databases, the visual representation of a problem is classified as overestimated, underestimated and perfectly estimated. Our blueprints have a focus different from effort estimation. Instead of estimating the result of an effort already realized, we provide an indication and recommendation about how to prevent waste.

6 Conclusions

We have proposed an extension to *AVISPA* so that it is able to localize potential waste in software process models specified with EPF. We focus on the kind of waste represented by work products that are developed although they are neither necessary nor useful. These elements are all those leaves in a Work Product Blueprint that are not marked as deliverables. However, we have found that most elements that satisfy these conditions are due to incompleteness in the specification: they are either deliverables that are not defined as so, or they are not leaves because they should have been specified as input of a task within the process. In both cases colored elements highlight errors in the process specification, and thus they are also improvement opportunities.

Our method has proved to be effective in both scenarios applied and presented in the paper. In the case of Scrum, we corroborated that the process model has no waste in the form of useless work products as expected from an agile method, but it was still useful for identifying misspecifications in the publicly available process model. In the case of the process of the medium size software company, the tool resulted highly useful for identifying all kinds of possible errors. In this case we were able to identify actual waste. This fact has been validated with the process engineer at the company and they agreed on the recommendations. They are currently in the process of restructuring their development model taking our findings into account.

AVISPA, along with this new extension, is only able to identify as waste some work products when they are not specified as deliverables and no task defines them as input either. However, we recognize the existence of other kinds of waste that we are not yet taken into account. If a work product is defined as an input for a task then it will be assumed as useful, even if the task does not use it at all. This analysis is of finer grain and it would require the analysis of the definition of each task steps and activities. This kind of waste cannot be automatically localized for processes specified using EPF since tasks are the finer formal elements and their internal descriptions are only informal. However, the possibility of identifying the existence of the waste pattern increases the power of the other problematic patterns. Specifically, the process model efficiency could be analyzed by identifying unnecessary work products (waste patterns) and bottleneck risks (demanded work products [5] [5]). These issues are part of our ongoing work in developing *AVISPA*.

Acknowledgments

The work of María Cecilia Bastarrica and Julio Ariel Hurtado Alegría has been partly funded by project Fondef D09I1171 of Conicyt, Chile. The work of Julio Ariel Hurtado Alegría has been also partially funded by NIC Chile.

7 Literature

[1] Stéphane Ducasse and Michele Lanza. The Class Blueprint: Visually Supporting the Understanding of Classes. Transactions on Software Engineering (TSE), 31(1):75–90, January 2005.

[2] Tudor Gîrba and Michele Lanza. Visualizing and Characterizing the Evolution of Class Hierarchies. In WOOR 2004 (5th ECOOP Workshop on Object-Oriented Reengineering), 2004.

[3] Verónica Uquillas Gómez, Stéphane Ducasse, and Theo D'Hondt. Visually Supporting Source Code Changes Integration: The Torch Dashboard. In Working Conference on Reverse Engineering, pages 55–64, Los Alamitos, CA, USA, 2010. IEEE Computer Society.

[4] Julio A. Hurtado, María Cecilia Bastarrica, and Alexandre Bergel. Analyzing the Scrum Process Model with AVISPA. In XXIX International Conference of the SCCC, pp. 60 – 65, Antofagasta, Chile, November 2010, IEEE Computer Society.

[5] Julio A. Hurtado, Maríıa Cecilia Bastarrica, and Alexandre Bergel. Analyzing Software Process Models with AVISPA. Accepted for publication in the International Conference on Software and Systems Processes, ICSSP'2011, Hawaii, USA, May 2011.

[6] Julio A. Hurtado, Alejandro Lagos, Alexandre Bergel, and María Cecilia Bastarrica. Software Process Model Blueprints. In Münch et al. [12], pages 285–296.

[7] Patrick Knab, Martin Pinzger, and Harald C. Gall. Visual Patterns in Issue Tracking Data. In Münch et al. [12], pages 222–233.

[8] Michele Lanza and Stéphane Ducasse. Polymetric Views—A Lightweight Visual Approach to Reverse Engineering. Transactions on Software Engineering (TSE), 29(9):782–795, September 2003.

[9] Peter Middleton, Amy Flaxel, and Ammon Cookson. Lean Software Management Case Study: Timberline Inc. In Proceedings of the 6th International Conference on Extreme Programming and Agile Processes in Software Engineering, XP 2005, volume 3556 of LNCS, pages 1–9, Sheffield, UK, June 2005. Springer.

[10] Sébastien Mosser, Alexandre Bergel, and Mireille Blay-Fornarino. Visualizing and Assessing a Compositional Approach of Business Process Design. In Proceedings of 9th International Conference on Software Composition (SC'10), pages 90–105. LNCS Springer Verlag, July 2010.

[11] Shahid Mujtaba, Robert Feldt, and Kai Petersen. Waste and Lead Time Reduction in a Software Product Customization Process with Value Stream Maps. In Proceedings of the 21st Australian Software Engineering Conference, ASWEC'10, pages 139–148. IEEE Computer Society, 2010.

[12] Jürgen Münch, Ye Yang, and Wilhelm Schäfer, editors. New Modeling Concepts for Today's Software Processes, International Conference on Software Process, ICSP 2010, Paderborn, Germany, volume 6195 of LNCS. Springer, July 2010.

[13] OMG. Software Process Engineering Metamodel SPEM 2.0 OMG. Technical Report ptc/08-04-01, Object Managemente Group, 2008.

[14] Kai Petersen and Claes Wohlin. Software Process Improvement through the Lean Measurement (SPI-LEAM) Method. Journal of Systems and Software, 83:1275–1287, July 2010.

[15] Mary Poppendieck. Lean Software Development. In 29th International Conference on Software Engineering, ICSE Companion Volume, pages 165–166, Minneapolis, MN, USA, May 2007, IEEE Computer Society.

[16] Mary Poppendieck and Tom Poppendieck. Lean Software Development: An Agile Toolkit. Addison-Wesley Professional, May 2003.

[17] Ken Schwaber and Mike Beedle. Agile Software Development with Scrum. Prentice Hall PTR, Upper Saddle River, NJ, USA, 1st edition, 2001.

[18] Jeff Sutherland, Scott Downey, and Bjorn Granvik. Shock Therapy: A Bootstrap for Hyper-Productive Scrum. In Yael Dubinsky, Tore Dyb°a, Steve Adolph, and Ahmed Samy Sidky, editors, AGILE, pages 69–73. IEEE Computer Society, 2009.

8 Author CVs

María Cecilia Bastarrica

María Cecilia Bastarrica is an Assistant Professor at the Computer Science Department, at the Universidad de Chile. She coordinates the MaTE group (Model and Transformation Engineering) since 2007. She received her PhD. in Computer Science and Engineering from the University of Connecticut in 2000, a Master of Science from the Pontificia Universidad Católica de Chile in 1994, and a Bachelor in Engineering from the Catholic University of Uruguay in 1991. Her main research topics are software engineering, software architecture, model-driven engineering, and software product lines. Lately, her work has focused on applying using MDE techniques formodeling software processes.

Julio Ariel Hurtado Alegría

Julio Ariel Hurtado is Asociated Professor at the Systems Department at the Universidad del Cauca. He participates at the MaTE group (Model and Transformation Engineering) since 2007 and at the IDIS (Software Engineering Research) group since 2005. He received an Electrical and Telecomunication Engineer degreee from Universidad del Cauca – Colombia in 1997. He is a PhD(c) in Computer Science from the Universidad de Chile. His main research topics are software engineering, model-driven engineering, and software process lines. His doctoral thesis focuses on applying MDE techniques and software process lines for modeling software processes.

Alexandre Bergel

Alexandre Bergel is Assistant Professor at the University of Chile. He obtained his PhD in 2005 from the University of Berne, Switzerland, under the supervision of O. Nierstrasz and S. Ducasse. After his PhD, A. Bergel made a first postdoc at Trinity College Dublin, Ireland, and a second one at the Hasso-Plattner Institute, Germany. A. Bergel is the author of over 50 articles, and has intensively published in international and peer review scientific forums, including the most competitive conferences and journals in the field of software engineering. A. Bergel and his collaborators carry out research in diverse aspects of software engineering, software quality, static and dynamic analysis.

Suitable Practices for an Agile Systems Engineering Approach

Ernst Stelzmann
Graz University of Technology, Austria

Markus Kohlbacher
CAMPUS 02 University of Applied Sciences, Austria

Abstract

In today's fast changing world agility gets more and more important also for the development of embedded systems. Holistic agile software development methods like Scrum or XP are often not suitable when hardware is part of the system. But these methods provide a lot of single agile practices that can also be used separately. Application of these practices for developing systems consisting of software and hardware is a topic of research nowadays. The aim is to achieve a higher degree of agility within Systems Engineering. This paper creates a decision supporting tool for choosing agile software development practices that suits best to Systems Engineering matters. Therefore the underlying functionality and benefit of several practices are analyzed. It is shown which factors should be considered to use appropriate practices regarding context. A quantitative survey shows which agile practices have already been applied successfully within Systems Engineering. And a qualitative survey indicates which practices have the best potential to be used successfully in future agile Systems Engineering applications. The result of this paper is a matrix that helps deciding for suitable practices to make Systems Engineering more agile.

Keywords

Agile Systems Engineering, agility, agile methods, survey, success, context, decision support

1 Introduction

Agility is defined as “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” [1]. In the domain of software engineering some so-called “agile methods” like Scrum [2], Extreme Programming [3], Lean Software Development [4], Crystal [5] and Feature Driven Development [6] promise the achievement of agility.

In the domain of Systems Engineering there is also a discussion about how to achieve more agility [7], [8], [9], [10], [11], [12], [13]. This discussion is not only based on academic research. There are also initiatives for experience exchange among companies, like the SoQrates Agile Working Group. The whole discussion is strongly affected by ideas from agile software development. Most contributions are trying to transfer ideas and practices from agile software development to Systems Engineering [7], [10], [11], [12], [13]. But it is not the only question if these methods will be successful in Systems Engineering. There was already a lot of controversy about the superiority of agile methods over traditional “plan-driven” methods in software development [14]. This controversy concluded in defining different “home grounds” for agile methods opposite to plan-driven methods [14]. So agile methods are only believed to be successful in a specific context. The context in which agile software development methods usually prosper was further specified by Kruchten [15]. In contrast to software development in Systems Engineering also the development of hardware has to be considered. Therefore Stelzmann analyzed the differences for the right context of agility when systems are developed that substantially consist of hardware and software [16].

This paper combines all these factors that have to be considered, when a company tries to increase agility of its Systems Engineering processes, in a decision support tool. It shows which practices promise to successfully enhance agility in the Systems Engineering domain. But it also deepens the understanding of the right context for these practices. On the one hand it points out contextual requirements for the practices. On the other hand the specific purpose of each practice is analyzed. That is necessary because agility requires a lot of sub-functions to be fulfilled, as can be seen in its definition. And the single agile practices are often only providing one of the sub-functions. The result of this paper is a decision support tool for choosing appropriate practices for an agile Systems Engineering approach. This tool is a matrix that allows choosing specific agile practices to enhance agility of Systems Engineering processes. It will consider the specific demand of agility, the specific functionality that is needed, preconditions that must be fulfilled and anticipated success of particular practices. At first this paper shows how the content of the matrix was elaborated.

2 Context for Agile Practices

A long time before the context for agile software development was explicitly researched; Cockburn noticed that different methods are needed for different contexts. Within his family of Crystal methods he primarily uses the factors size and criticality to choose the right process [5]. Size is defined by the number of people involved in the project. Criticality by the risk a faulty product means for its environment. A smaller size of the project and a lower criticality means that a more lightweight process is recommended. And that should provide more agility.

Boehm and Turner [14] are using 5 factors to find the “homeground” for agile methods in contrast to traditional plan-driven methods. These factors are:

- Personnel: High skilled developers are needed throughout an agile project.
- Dynamism: Dynamic environments are home ground for agile methods.
- Culture: People who feel comfortable and empowered by having many degrees of freedom embrace agile methods.
- Size: Agile methods work best with small to medium sized teams.
- Criticality: Agile methods are untested on safety-critical products and they use only very little documentation. Therefore they are only recommended for products with low criticality.

Kruchten looks at two different levels for classifying the context. On the organizational level he uses [15] the contextual factors:

- Business domain: Different industries have different requirements (e.g. safety) for software development.
- Number of instances: Is the software being built for one customer or for many?
- Maturity of organization: How mature are processes and people?
- Level of innovation: How innovative is the organization?
- Culture: National and corporate culture may affect development processes.

On the project-level he uses [15]:

- Size: The overall size of the system is the main factor. It influences team size. Less than 15 team members is a good size for agile methods.
- Stable architecture: Can the architecture de facto be frozen at the beginning of the project? Regarding to Kruchten agile Methods should be used for stable architectures.
- Business model: Internal system, commercial product, open-source, etc.? Agile methods work best for in-house development.
- Team distribution: Co-location of team members is important for agile methods.
- Rate of change: How stable are requirements? Medium to high rates of change are predestinated for agile methods.
- Age of system: New system or evolution of old system? Agile methods work best with greenfield development.
- Criticality: What are the risks of a system failure? Agile Methods should not be used if life is endangered.
- Governance: How does the commissioning of the projects work? Agile methods need simple rules.

Kruchten states that agile methods may also succeed outside the defined “sweet spot” but this might be more challenging [15].

Stelzmann analyzed the context for Agile Systems Engineering and found one additional factor that he called “nature of system”. This factor concerns the major difference between software development and Systems Engineering where also hardware is being developed. The nature of software allows development in small cyclic steps which is the main principle of all agile methods [2], [3], [5], [6], [9], [12], [13]. For systems consisting also of hardware it has to be asked if:

- systems (hardware components) can be manufactured in short periods of time.
- prototyping can be done quickly and cheaply.
- testing can be done quickly and cheaply.
- implementing changes can be done quickly and cheaply.

Stelzmann also proposed that the factors should be treated differently. He recommends dividing them into factors for demand and factors for feasibility of agile methods [16]. The factors for feasibility are such that can only handicap agile development like criticality does. But if criticality does not exist that does not mean that agile methods are needed and should be applied. On the other hand factors for demand do only show the need for agility. For example if rate of change is high, agile methods are needed. But if rate of change is low, agile methods could still be applied. And some factors are also affecting both feasibility and demand. They could be analyzed more in detail by using both dimensions [16].

Now this paper further analyzes the demand for agility regarding different functionality of agile practices. The definition of agility [1] is very broad. And single agile practices cover different agile goals like speed, neutrality against change, learning attitude and so on. The decision support tool being generated in this paper takes care of this. The functions that are needed to achieve agility are (extracted from the definition in [1] and applied to Systems Engineering):

- to identify change (gather information),
- to handle changes within development process,
- to provide flexibility (in products, processes, organization …) to be able to react on changes,
- to increase speed of development,
- efficiency,
- effectiveness and
- to improve learning within the project and the whole company.

These functions can concretely be delivered by single practices. And different types of change can be handled. On the one hand there are changes outside development [11]. For example change in market demand or commodity prices. On the other hand there is change within development [11]. For example if new information gathered during development leads to changed requirements. The decision support tool will distinguish for which type of change the agile practices are beneficial.

3 Success of Agile Practices within Systems Engineering

Knowing about the benefits of an agile practice and about possible handicaps for its application is not everything. When deciding for practices to improve agility of Systems Engineering processes the anticipated success is also of interest. A survey about successful application of agile practices was already published by Maierhofer et al. [13]. But in that paper agile practices were combined to a few agile constructs. For this paper the data was reused to examine the success of all 18 single agile practices that were represented in the questionnaire. The results are presented in the decision support matrix in the column called “evidence for success of application within Systems Engineering”. There is evidence for successful application of almost no agile practice in the development of hardware together with software. But this result does not mean that agile practices cannot be applied to Systems Engineering. It just says that there is no evidence yet. Since agile practices are not well established in Systems Engineering yet, this is no surprise. Also questionnaire design did not care about the context in which agile practices were used [13]. After more agile practices have been established in systems development more research will resolve these issues and provide better information to choose successful agile practices. Nevertheless results of a qualitative survey about potential success for applying agile practices in future Systems Engineering projects are already available.

Research methodology was to interview 20 experts (project, product and quality managers) from companies within German speaking area. The companies were of different size (small to large) and operating in automotive, logistics, machine building, electronics and railroad industry. Some of them had experience with agile methods but only for developing software. The interviewees without knowledge about agility got the practices explained before. Then the practices were evaluated by the interviewees. Their potential to successfully enhance agility of Systems Engineering was evaluated as well as possible difficulties resp. handicaps.

The results of this survey are shown in the decision support matrix in the last two columns. “0” in the anticipated success column means that interviewees did not think that applying this practice is beneficial. ”+” means that they think it is to some degree beneficial. And “++” means that they expect this practice to be very beneficial for providing agility within Systems Engineering. Within the anticipated difficulties column the scale is used in a negative way.

4 Conclusion

The definition of agility is very broad. If agility is the aim that has to be achieved there are many sub-aims like speed, neutrality to change, learning attitude and so on. Therefore different agile practices support different sub-aims of agility. If a company wants to improve its Systems Engineering process regarding agility it could face different challenges. The decision support matrix presented in this paper helps companies to choose suitable practices for the challenges they are facing. It helps to find practices that provide the specific part of agile functionality that is missing in the company. The matrix also reveals which type of changes the practices are aiming at. Also preconditions for the practices and their probability of being successful are shown by the matrix.

The decision support matrix in Figure 1 is the final outcome of this paper. The practices surveyed by Maierhofer et al. [13] were complemented by practices that were not surveyed but were identified as important and widely used agile practices in [2] – [6].

This matrix can be used in different ways: If a specific practice should be discussed its intrinsic functionality is shown. If its functionality has something to do with change it is also revealed which type of change can be handled. Also preconditions are shown as well as a preliminary statement for its anticipated success.

Another way to use this matrix is to search for practices that provide a specific functionality. For example if flexibility within development process is an issue then it is possible to quickly search for practices that are concerning this. If sensing change is the concern then other practices are recommended.

Type of change is also distinguished. It is shown which practices can handle change from outside development (change of market demand, laws, commodity prices) and which can handle changes within development (requirements were unclear at the beginning, change of requirements due to new information gathered during development).

It is also possible to quickly eliminate practices from the selection process if preconditions are not given.

In future research more agile practices will be included in the matrix. Also the information on probability of success of the single practices will be improved in further research. But therefore it is necessary that agile practices are more frequently used for development of products consisting of hardware and software. Only then it is possible to gather good quantitative data on that matter.

So the matrix in its current state should already present a useful decision support tool for agile Systems Engineering practices. Its contents will be improved in future research.

		Type of change that can be handled		Functionality (agile benefit)				Preconditions						Evaluation		
		change in environment (not affected by development)	change directly affected by new information gathered during development	information gathering (observe and recongnize change)	handling of changes and consideration in development process	enhancement of flexibility in the area of	additional (supporting) functionality	no criticality regarding safety or securety	customer willing to be integrated in development	nature of system allows development in small steps	small team size (e.g. <15)	high skilled team members	culture fits to practice	evidence for success of application within Systems Engineering	anticipated success for future application dependent to right context	anticipated difficulties for future application
	practices concerning customer collaboration															
1	focus on customer's benefit		x	x			effectiveness		x			x	x		+	-
2	collaboration over contract negotiation	x	x		x	order processing			x				x		+	--
3	customer integration to get feedback		x	x					x	x			x	0	++	-
4	prioritization of requirements						effectiveness		x					0	++	0
	practices concerning organization															
5	small team size (partitioning of large teams)						communication							0	+	0
6	co-location of team members			x			communication				x		x	0	+	-
	practices concerning employees															
7	high-skilled team members	x	x	x	x	organization	efficiency					/		0	++	0
8	motivation	x	x	x	x		efficiency							0	+	0
9	teamwork		x	x			communication						x	0	+	0
10	direct and easy communication		x	x			efficiency				x		x	0	++	-
11	tacit knowledge		x			developm. process		x			x	x	x		+	--
12	positive attidude regarding change	x	x									x	x	+	+	0
	practices concerning development process															
13	adaptable processes („process teamownership“)	x	x			developm. process		x			x	x	x	0	+	--
14	development in small steps	x	x	x	x	developm. process	effectiveness	x		x			x	0	++	--
15	continuous integration	x	x	x	x					x					+	-
16	test-driven development			x			efficiency			x					+	0
17	continuous planning (instead of up-front)	x	x		x	developm. process	efficiency						x	0	+	-
18	daily meetings						communication				x		x		+	0
19	essential documentation only					developm. process	effect./effic.	x			x	x	x	+	+	-
	practices concerning product (system)															
20	adaptable architecture	x	x			product								0	++	-
21	simplicity	x	x			product	effect./effic.							0	++	0
22	modularity	x	x			product								0	+	-
23	reuse of modules						effic. (speed)							--	+	-
24	software first (or at least simultaneously)	x	x			product								0	+	0

Figure 1: Decision support matrix for selection of agile practices

5 Literature

1. A. Qumer and B. Henderson-Sellers, "Crystallization of agility back to basics", ICSOFT 2006 (2) pp. 121-126, 2006
2. K. Schwaber, Agile Project Management with SCRUM, Microsoft Press, 2004
3. K. Beck, Extreme Programming Explained: Embrace Change, Addison-Wesley Professional, October 1999
4. M. Poppendieck, T. Poppendieck: Lean Software Development – An agile Toolkit, Addison-Wesley, Boston, MA 2003
5. A. Cockburn, Crystal Clear: A Human-Powered Methodology for Small Teams, Addison-Wesley Professional, 2004
6. S. Palmer, J. Felsing, A practical guide to Feature Driven Development, Prentice Hall PTR, 2002
7. R. Turner, "Towards agile systems engineering processes", Cross Talk, The Journal of Defense Software Engineering, April 2007
8. R. Dove, Response Ability, John Wiley & Sons, New York, 2001
9. R. Haberfellner, O. de Weck, "Agile SYSTEMS ENGINEERING versus AGILE SYSTEMS engineering", 15th Annual International Symposium Proceedings, INCOSE, Rochester, NY, 2003
10. M. Wilson, H. Mooz, "Agile Systems Engineering for Rapid Project Solution Development", 13th Annual International Symposium Proceedings, INCOSE, 2003
11. E. Stelzmann, G. Spork, Ch. Kreiner, R. Messnarz, F. König, "Practical Use of Agile Methods within Systems Engineering", International Spice Days, Stuttgart, June 21-23 2010
12. E. Stelzmann, Ch. Kreiner, G. Spork, R. Messnarz, F. Koenig, "Agility Meets Systems Engineering: A Catalogue of Success Factors from Industry Practice", A. Riel et al. (Eds.): EuroSPI 2010, CCIS 99, pp. 245-256, Springer-Verlag Berlin Heidelberg 2010
13. S. Maierhofer, E. Stelzmann, M. Kohlbacher, B. Fellner, "Requirement Changes and Project Success: The Moderating Effects of Agile Approaches in System Engineering Projects", A. Riel et al. (Eds.): EuroSPI 2010, CCIS 99, pp. 245-256, Springer-Verlag Berlin Heidelberg 2010.
14. B. Boehm, R. Turner, Balancing Agility and Discipline: A Guide for the Perplexed, Addison-Wesley, Boston, MA 2003
15. P. Kruchten, "Contextualizing Agile Software Development," Industrial Proceedings 17th EuroSPI² Conference, Grenoble Institute of Technology, France, pp. 6.1 - 6.12, 1.-3. September 2010
16. E. Stelzmann, "Contextualizing Agile Systems Engineering", IEEE International Systems Conference Proceedings, Montreal, Quebec, Canada, April 4-7 2011

6 Author CVs

Ernst Stelzmann

Ernst Stelzmann is currently University Assistant at the Institute of General Management and Organization at Graz, University of Technology. He finished his studies of Industrial Engineering and Management and received his Diplomingenieur degree at Graz University of Technology 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”.

Markus Kohlbacher

Markus Kohlbacher works as a project manager for a European retailing company and as a research scientist at University of Applied Sciences *CAMPUS* 02. He received his PhD in Industrial Management and his Diplomingenieur degree in Telematics from Graz University of Technology. His main research interests include strategic management, organizational excellence, and quantitative research methods.

An approach to improving agility and discipline of software development with Scrum and CMMI

Katarzyna Bulska[1], Jakub Miler[1]
[1] Gdansk University of Technology, Narutowicza 11/12,
80-233 Gdansk, Poland
{Katarzyna.Bulska, Jakub.Miler}@eti.pg.gda.pl

Abstract

The paper reports on two case studies of combining the Scrum methodology with the CMMI maturity model to improve together agility and discipline of software development. First, we propose the CMMI-Scrum reference model, which maps Scrum practices onto 123 practices of CMMI Staged level 2 and 3. Then, for 60% of CMMI practices, which are insufficiently covered by Scrum we extend Scrum with new practices that improve discipline while maintaining agility. The practices for the improvement of an actual software development process are selected based on a questionnaire with 25 single-choice questions. The processes of two IT companies were analyzed with the proposed questionnaire and reference model, where on average 72% of the suggested practices were confirmed, 24,5% were mismatched and 3,5% were rejected.

Keywords

CMMI, Scrum, agile, discipline, process improvement

1 Introduction

Thus far, many large companies engaged in long-term and complex IT projects have improved their processes towards the disciplined development methods, which allowed them to control the process and ultimately enabled the introduction of maturity control models such as Capability Maturity Model Integration [4]. In recent years the agile development methodologies such as Scrum [9] stemming from the Agile Manifesto [1] have drawn more and more attention though they successfully improved the processes of smaller IT companies.

Some models combining Scrum and CMMI [5,7] as well as successful applications of both CMMI and Scrum have already been reported [2, 6, 8, 10]. However, the models insufficiently address compliance of these approaches, while the applications did not follow any defined improvement process, but extensively used costly top consultants in agile methodologies. The improvement of agility and discipline requires in-depth knowledge of both the CMMI model and Scrum. Therefore, the challenge that arises is to elaborate an approach for small and medium sized IT companies that would allow them to identify potential improvement opportunities of combined Scrum and CMMI on their own, support them in the appropriate choice of practices, as well as maintain the already achieved level of maturity or agility.

The paper presents two case studies in Polish IT companies of an approach to process improvement combining the agility of Scrum and the maturity of the CMMI model. The approach comprises:

- CMMI-Scrum reference model which maps the Scrum practices onto the practices of CMMI and extends Scrum to balance agility and discipline;
- the model application process supported with a questionnaire and a software tool.

2 CMMI-Scrum Reference Model

While creating the model for best practices of CMMI in conjunction with the practices of Scrum methodology it is necessary to determine what are the relationships between these practices in the first place. For this purpose we propose a reference model mapping Specific Practices of the 2nd and the 3rd level of maturity in the CMMI Staged representation onto the activities described by the Scrum methodology.

In the proposed reference model, for each of the selected practices from CMMI areas, based on in-depth analysis of the practices, the literature as well as our experiences with various development processes, we determined to what extent and how this practice is exercisable in the Scrum methodology. For those of CMMI practices that are implemented partially or not at all by the Scrum methodology, the necessary extensions of Scrum were specified in the model. They include the necessary modifications to the Scrum practices and they also contain some new practices. During the design of these extensions it was crucial to preserve the agility of Scrum while providing for more maturity and discipline. Therefore, in many cases the fact that the CMMI does not specifically define how to maintain the documentation for the artifacts of the practices [7] was beneficial. It enabled a convenient, agile-friendly approach to the documentation using images, video, notes, etc. In other cases, some new tasks were conveniently placed inside or along with the already existing practices and tasks of Scrum methodology.

Table 1 shows an example of the Monitor Project Risks CMMI practice from the the area of Project Monitoring and Control, which is only partially implemented by Scrum and needs some agile extensions to fulfil the requirements of the maturity model in the Scrum development process. In the column "CMMI Practice" is given a brief description of the CMMI practice, in the column "Scrum Practices" adequate Scrum practices are given., while the "Additional Practices" column defines the extensions to Scrum to achieve conformity with the CMMI model if they are needed.

Table 1. CMMI SP 1.3 Monitor Project Risks and its reference to Scrum

CMMI Practice	Scrum Practices	Additional Practices
The purpose of this practice is the continuous supervision and inspection of previously identified sources of risk.	The risks are written on the whiteboard, flip charts or in the form of a list of difficulties and obstacles and are monitored by the Scrum Master. Therefore they are monitored on an informal basis [7].	During the planning stage of the project the project risks (the list of hazards) should be identified using techniques such as brainstorming and identified hazards should be prominently displayed. During the planning phase and subsequent Sprint Reviews this list should be reviewed and possibly modified or supplemented. Additionally, during the daily meetings a member of the team who reported the impediment should report whether further difficulty arises and has it already been resolved. After the meeting he should write down in one sentence the solution on a relevant table.

Full CMMI-Scrum reference model covers 123 practices of CMMI Level 2 and 3 excluding the Organizational Process Definition, Organizational Process Focus and Organizational Training of CMMI level 3. 40% of them can be directly implemented with Scrum, 24% are partially implemented, while 36% are not implemented by Scrum The practices of the second and third group were covered by appropriate extensions to Scrum.

3 Model Application Process

The proposed process of agility and discipline improvement using the proposed CMMI-Scrum reference model consists of 6 steps:

1. Select a pilot project, in which a new approach combining Scrum practices and CMMI model will be implemented. Analyze the selected project to provide a preliminary description, which will include information such as the size of the team, the product characteristics etc..
2. Identify a list of the major problems that occurred earlier in the projects in the company to highlight weak points that require special attention.
3. Select appropriate practices from the CMMI-Scrum reference model according to the analysis in steps 1 and 2. In case of absence of adequate information, complete the analysis and return to the step 3.
4. Analyze selected practices with the list of problems identified in step 1 and 2 in mind and decide whether to implement all recommended practices and additional areas.
5. Implement the set of practices (with possible extensions arising from the specific character of the company) in the pilot project.
6. If successful implement the practices in other projects. Otherwise go to step 2 and continue with analyses and selection of practices or go to step 1 and select a different pilot project.

Naturally, the key point of the process is step 3. To support the selection of appropriate practices from the CMMI-Scrum reference model we developed a questionnaire. The questionnaire consists of 25 single-choice questions. To each answer, a set of suggested practices from the reference model is assigned. In some responses next to the name of a CMMI-Scrum model area there are added numbers of goals or specific practices. This information indicates that while implementation of the area is suggested to solve the problem, the highlighted practice or goal is particularly important. For some answers, supplementary activities are added which are not directly taken from the CMMI-Scrum reference model but are often necessary for the successful introduction of the Scrum and CMMI practices (e.g. professional training). Table 2 shows a sample question and answers from the questionnaire as well as suggested practices of CMMI-Scrum model and supplementary activities.

Table 2. An example of the question and its answers

Question		
Does the team have sufficient knowledge of the technologies in which the product will be built?		
Answers	CMMI-Scrum practices	Supplementary activities
Yes, the team will be composed mostly of employees with a very good knowledge of the technology used in the project.		
Rather yes, the majority of the members of the team has a sufficient knowledge of the technology used, some training would be adequate.	• Verification	
Probably not, most of the staff involved has only a basic knowledge of the technology used.	• Verification SG 2 • Product Integration SG 2 SP 2.2 • Technical Solution SG 2 SP 2.1, 2.2, 2.3	A professional training on the technology should be included in the Product Backlog.

If a user answers “Probably not” to the question presented in Table 2,the CMMI-Scrum areas and practices indicated in the table next to this answer are added to the list of suggested practices, As a result of the questionnaire a list of most frequently suggested areas and practices is presented along with a list of all the appropriate supplementary activities.

To additionally support the application of the proposed reference model, an internet-based software tool called MatureScrum [3] was developed. The tool allows to browse the entire reference model with a hierarchical table of contents as well as implements the questionnaire and the algorithms for selection and suggestion of practices based on the answers

4 Case studies

The goal of the case studies was to assess the suitability of the practices suggested by the approach to the process improvement. Due to limited resources and timeframe of the study, the validation was limited to the assessment of suggestions only, and the practices were not actually implemented to observe the real improvements.

The case studies have been carried out with the participation of two Polish SME IT companies Young Digital Planet and MOL which are potential users of the approach. The software development processes of these companies were analyzed by the authors of this paper and some improvement practices were suggested based on the CMMI-Scrum reference model and the results of the questionnaire. Then, the key managers of both companies assessed each suggestion of a practice as:

- *Interested in* - the company regards this practice as promising and would be interested in the implementation of the practice;
- *Already implemented* - the practice has already been implemented in the company and works well;
- *It did not work* – the company has already tried to implement the practice, but the results were unsatisfactory and it was abandoned;
- *Not applicable* - the practice does not apply to the specific character of the company or it cannot be implemented for reasons specific to the company.

As a positive assessment of a practice suggestion we consider both "Interested in", and "Already implemented". The former means that the practice can be implemented in the company and it is considered potentially beneficial, while the latter indicates an already implemented and beneficial practice. "It did not work" is regarded as a negative assessment of the practice suggestion indicating false suggestion and demonstrated inadequacy of the practice. The "Not applicable" assessment is treated as neutral because it means that a suggestion is potentially correct but is surplus in a particular study as it is not possible to implement the practice within the company and to determine its usefulness. This assessment may also indicate that the questionnaire is not sufficiently detailed or that the answers were incorrect due to insufficient data.

4.1 The Young Digital Planet case study

Young Digital Planet (YDP) provides advanced information technology solutions mainly in the field of education in Polish and foreign markets. The validation of the approach was limited only to the Department of Software Development which employs about 40 people. The company is not certified for CMMI and develops software in its proprietary disciplined methodology, which roughly matches the CMMI level 2. The management of the department intended to introduce new disciplined practices as well as some agility.

The authors of the paper visited the company and examined the development processes and tools. Based on discussions with managers and employees a list of main concerns and expected advises was prepared. In majority they refered to poor quality of requirements analysis, problems with testing as well as organising team meetings and providing project documentation on a satisfying level.

Because the company had not identified a pilot project we omitted step 1 of the model application process and examined a typical project of the company. Then, using the questionnaire described in Section 3 we analysed the needs of the department.As a result, these areas of CMMI-Scrum reference model were marked as recommended: Configuration Management, Technical Solution (SP 2.1 - 2.4), Project Monitoring and Control (SP 1.3, 2.3), Project Planning (SP 1.2, 2.2), Measurement and Analysis, Verification (SP 2.1 - 2.3), Process and Product Quality Assurance, Requirements Management (SP 1.4) and Requirements Development (SP 3.1, 3.5). In brackets the indexes are given for those Specific Practices which were highlighted by the questionnaire as crucial for the improvement. 3 areas of the reference model were suggested by the questionnaire as optional: Integrated Project Management (SG 1), Validation (SP 1.3) and Product Integration (SP 2.2). Some supplementary activities were recommended as well. Based on these results in step 4 of the model application process we selected a list of 20 practices most suited to the specific character of the company. The list of practices was sent for assessment to the department's managers.

In total, 65% of practices have been evaluated positively, while 35% of them were classified as “Not applicable” mainly due to suggestions going beyond the scope of the department. For example, project managers and product managers are usually employed by the company's business department. Thus the practices concerning the project manager have been dismissed as outside of the scope of the software development department. Similarly, the practices concerning the organisation of meetings (Daily Scrums, Sprint Reviews, etc.) and methods of identifying risks and impediments in the project were also assessed as “Not applicable”. In turn, the practices regarding failure analysis and code testing were recognized as interesting, in particular the ones regarding their measurements and metrics. What is important is the lack of practices assessed as "It did not work", which would have showed false suggestions of the model. Figure 1 presents summary of the assessment by YDP. So far, steps 5 and 6 of the model application process have not been implemented.

4.2 The MOL case study

The MOL company is currently the largest provider of software for libraries in Poland. It is a typical SME with 7 programmers employed in the software development department. The development processes are more agile than in YDP, however do not match directly the Scrum methodology. Currently, they are planning to implement the 2nd level of CMMI. The person responsible for the validation from MOL was the owner of the company. He assessed both the questionnaire and the set of recommended practices obtained using the MatureScrum tool.

The questionnaire has been regarded as relevant to the company’s situation. Most of the suggested practices have a reference to projects in the company. 107 practices were suggested on the basis of the questionnaire, which covered 75% of the problems and expectations identified in the analysis phase within the company (step 2 of the model application process). The detailed assessment of the suggested practices resulted in 79% of practices being evaluated positively (“Interested in” or “Already implemented”), 14% of them classified as “Not applicable”, and 7% of practices evaluated negatively as already have not worked in the company due to insufficient cooperation with customers and lack of economic benefits (e.g. pair programming). The summary of the assessment of the suggested practices by MOL are presented in Figure 1. So far, no feedback concerning implementation of the hybrid approach in the pilot project has been provided.

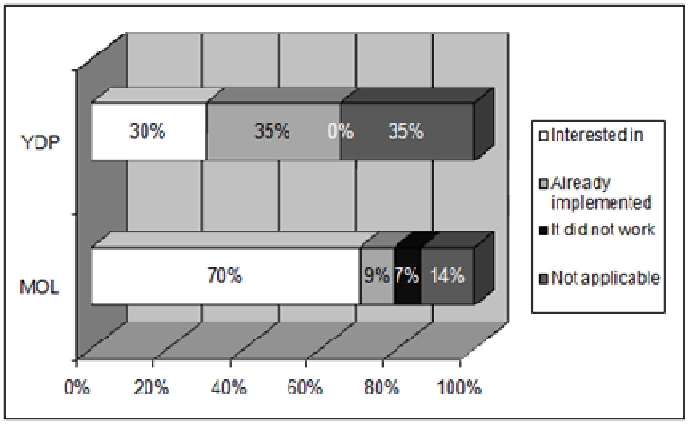


Figure 1: The results of the validation in Young Digital Planet and MOL.

5 Conclusions

The paper presented two case studies of the application of the proposed CMMI-Scrum reference model. The proposed reference model focuses on the characteristics of the project and the product without referring directly to the company’s structure. This follows from the nature of Scrum, which focuses on the process of software development, without referring to inner functions of the organization. It should be noted, however, that this area is very important for the functioning of the development process and during the application of the proposed approach it should also be analyzed by the organization.

To support the practical application of the proposed CMMI-Scrum reference model we elaborated a 6 step improvement process, which involves the analysis of the improvement needs, the selection of potentially beneficial practices from the model with a questionnaire of 25 single-choice questions, as well as the evaluation and implementation of these practices. The improvement process respects existing CMMI or Scrum experience and adds the complementary value only in necessary areas. The use of our questionnaire is additionally supported with a proprietary software tool MatureScrum.

This approach was validated in two case studies involving two representative small and medium sized Polish IT companies. The companies followed their proprietary methodologies with elements of CMMI and Scrum. We applied our approach to systematically identify the improvement potential for their processes with the suggestions from our CMMI-Scrum reference model. On average 72% of the suggested practices was regarded as already or potentially beneficial, while only 3,5% were incorrect. Some 24,5% of suggestions were evaluated as not applicable for organisational or economic reasons. The collected data shows that the approach to a large extent provides valuable suggestions for successful combining agility of Scrum practices with CMMI’s discipline. Naturally, final confirmation of its effectiveness would require the actual implementation of the suggested practices and the measurement of the added value.

Comprehensive definition of our CMMI-Scrum reference model, its application process and the MatureScrum tool as well as the full results of the case studies are presented in [3].

6 Literature

1. Agile manifesto, http://www.agilemanifesto.org
2. Babuscio, J.: How the FBI Learned to Catch Bad Guys One Iteration at a Time. In: Dubinsky J., Dyba T., Adolph S., Sidky A. (eds.) Agile Conference 2009, pp. 96--100. IEEE Computer Society, Los Alamitos (2009)
3. Bulska, K.: Integracja zwinnych metodyk wytwarzania oprogramowania z metodami kontroli dojrzałości procesu – wspomaganie wyboru odpowiednich praktyk. (Integration of the agile software development methodologies with maturity models – good practices assistant), MSc thesis, Department of Software Engineering, Faculty of Electronics, Telecomunications and Informatics, Gdansk University of Technology, Gdansk (2010)
4. Capability Maturity Model Integration, http://www.sei.cmu.edu/cmmi
5. Diaz J., Garbajosa J., Calvo-Manzano J. A.: Mapping CMMI Level 2 to Scrum Practices: An Experience Report, In: O'Connor, R., Baddoo, N., Cuadrado Gallego, J., Rejas Muslera, R., Smolander, K., Messnarz, R. (eds.) EuroSPI 2009. Communications in Computer and Information Science, vol. 42, pp. 93--104. Springer, Heidelberg (2009)
6. Glazer, H., Dalton J., Anderson D., Konrad M., Shrum S.: CMMI or Agile: Why Not Embrace Both!. Technical Note CMU/SEI-2008-TN-003, Software Engineering Institute (2008)
7. Marçal, A. C., de Freitas B. C., Furtado Soares F. S., Furtado M. S., Maciel T. M., Belchior A. D.: Blending Scrum practices and CMMI project management process areas. Innovations in Systems and Software Engineering, vol. 4, no. 1, pp. 17--29. (2008)
8. Potter, N., i M. Sakry.: Implementing Scrum (Agile) and CMMI together. Process Group. (2009), http://www.itmpi.org/assets/base/images/itmpi/Potter-ScrumCMMI.pdf
9. Schwaber, K.: ScrumGuide. Scrum Alliance (2009)
10. Sutherland J., Jakobsen C., Johnson K..: Scrum and CMMI Level 5: The Magic Potion for Code Warriors. In: Eckstein J., Maurer F., Davies R., Melnik G., Pollice G. (eds.) Agile Conference 2007, pp. 272--278. IEEE Computer Society, Los Alamitos (2007)

7 Author CVs

Katarzyna Bulska

She is a PhD student at the Department of Software Engineering, Gdansk University of Technology. Her main research interest concerns combining agile practices with more disciplined approach. More recently she has been working under a supervision of prof. Janusz Górski (Head of the Department of Software Engineering, Gdansk University of Technology) on applying agile practices to the development of safety-critical systems, particularly adapted to medical and healthcare domain. In 2010 she defended the Master's thesis "Integration of the agile software development methodologies with maturity models – good practices assistant".

Jakub Miler

He is Associate Professor at the Department of Software Engineering, Gdansk University of Technology and a member of the Information Assurance Group. In 2005, he received his PhD in the field of software risk management. He took part in several research and development projects including Integrated and STREP projects of 5th and 6th EU Framework Programme. Currently, he is a key member of the NOR-STA EU-funded project. He has about 10 years experience in risk management, software engineering and software project management. His present research interests include combining agile and disciplined methodologies of software development and management as well as issues of trust and usability of software systems.

Can you Outsource Innovation?

Kasper Selch[1], Jan Pries-Heje[2], Ada Scupola[2,] Jørn Johansen[3]
[1] Danske Bank, Denmark
[2] Roskilde University, Denmark
[3] DELTA, Denmark

Abstract of Source IT Book

The book seeks to enhance our understanding of the interrelation between sourcing and innovation in information systems development. An understanding of the interrelation between sourcing and innovation is important in its own right, but we also aim to use this understanding to provide instruments for balancing sourcing and innovation in ways that match the needs of development projects. The book addresses two questions:

- What are the preconditions for optimal sourcing in relation to innovation capability?
- How can an organization be most innovative, while making optimal use of sourcing?

Succeeding in either sourcing or innovation is far from easy, and succeeding in both at the same time is a real challenge. Yet, practitioners at many managerial and decision-making levels are faced with demands to devise innovative solutions in projects that involve various sourcing arrangements. These practitioners need guidance on how to make strategic, tactical, and operational sourcing decisions. At the same time, researchers are in need of concepts, case descriptions, and tool assessments that enable a better understanding of the challenges involved in successfully balancing sourcing and innovation. Apart from being useful to research and practice, this book provides cases and tools that can be used in education. With respect to education, we consider the book particularly relevant to advanced programmes in information systems development, software process improvement, and project management.

Reference

Hertzum M., Jørgensen C. (eds.) (2011): Source IT. Balancing Sourcing and Innovation in Information Systems Development.

The SourceIT book is part of the conference handout.

Is minimizing interaction a solution to cultural and maturity inequality in offshore outsourcing?

Morten Hertzum[1], Jan Pries-Heje[1]
[1] Roskilde University, Denmark

Abstract of Source IT Book

The book seeks to enhance our understanding of the interrelation between sourcing and innovation in information systems development. An understanding of the interrelation between sourcing and innovation is important in its own right, but we also aim to use this understanding to provide instruments for balancing sourcing and innovation in ways that match the needs of development projects. The book addresses two questions:

- What are the preconditions for optimal sourcing in relation to innovation capability?
- How can an organization be most innovative, while making optimal use of sourcing?

Succeeding in either sourcing or innovation is far from easy, and succeeding in both at the same time is a real challenge. Yet, practitioners at many managerial and decision-making levels are faced with demands to devise innovative solutions in projects that involve various sourcing arrangements. These practitioners need guidance on how to make strategic, tactical, and operational sourcing decisions. At the same time, researchers are in need of concepts, case descriptions, and tool assessments that enable a better understanding of the challenges involved in successfully balancing sourcing and innovation. Apart from being useful to research and practice, this book provides cases and tools that can be used in education. With respect to education, we consider the book particularly relevant to advanced programmes in information systems development, software process improvement, and project management.

Reference

Hertzum M., Jørgensen C. (eds.) (2011): Source IT. Balancing Sourcing and Innovation in Information Systems Development.

The SourceIT book is part of the conference handout.

Knowledge transfer in outsourcing: From theory to tooling

Sabine Madsen[1], Keld Bødker[1], Thomas Tøth[2]
[1] Roskilde University, Denmark
[2] Danske Bank, Denmark

Abstract of Source IT Book

The book seeks to enhance our understanding of the interrelation between sourcing and innovation in information systems development. An understanding of the interrelation between sourcing and innovation is important in its own right, but we also aim to use this understanding to provide instruments for balancing sourcing and innovation in ways that match the needs of development projects. The book addresses two questions:

- What are the preconditions for optimal sourcing in relation to innovation capability?
- How can an organization be most innovative, while making optimal use of sourcing?

Succeeding in either sourcing or innovation is far from easy, and succeeding in both at the same time is a real challenge. Yet, practitioners at many managerial and decision-making levels are faced with demands to devise innovative solutions in projects that involve various sourcing arrangements. These practitioners need guidance on how to make strategic, tactical, and operational sourcing decisions. At the same time, researchers are in need of concepts, case descriptions, and tool assessments that enable a better understanding of the challenges involved in successfully balancing sourcing and innovation. Apart from being useful to research and practice, this book provides cases and tools that can be used in education. With respect to education, we consider the book particularly relevant to advanced programmes in information systems development, software process improvement, and project management.

Reference

Hertzum M., Jørgensen C. (eds.) (2011): Source IT. Balancing Sourcing and Innovation in Information Systems Development.

The SourceIT book is part of the conference handout.

Innovation and Partner Sourcing from a Vendor Perspective

Anders Barlach[1], *Jesper Simonsen*[2]
[1] *CSC Scandihealth, Denmark*
[2] *Roskilde University, Denmark*

Abstract of Source IT Book

The book seeks to enhance our understanding of the interrelation between sourcing and innovation in information systems development. An understanding of the interrelation between sourcing and innovation is important in its own right, but we also aim to use this understanding to provide instruments for balancing sourcing and innovation in ways that match the needs of development projects. The book addresses two questions:

- What are the preconditions for optimal sourcing in relation to innovation capability?
- How can an organization be most innovative, while making optimal use of sourcing?

Succeeding in either sourcing or innovation is far from easy, and succeeding in both at the same time is a real challenge. Yet, practitioners at many managerial and decision-making levels are faced with demands to devise innovative solutions in projects that involve various sourcing arrangements. These practitioners need guidance on how to make strategic, tactical, and operational sourcing decisions. At the same time, researchers are in need of concepts, case descriptions, and tool assessments that enable a better understanding of the challenges involved in successfully balancing sourcing and innovation. Apart from being useful to research and practice, this book provides cases and tools that can be used in education. With respect to education, we consider the book particularly relevant to advanced programmes in information systems development, software process improvement, and project management.

Reference

Hertzum M., Jørgensen C. (eds.) (2011): Source IT. Balancing Sourcing and Innovation in Information Systems Development.

The SourceIT book is part of the conference handout.

The importance of being trained

Dr. Anne Kramer

Abstract

The success of software projects is driven by the magical triangle between time, budget and quality. While short term thinking focuses on time and budget, it is quality that has the most important impact on long term customer satisfaction. Thus, providing good quality software is crucial for success.

Today, we are all aware that quality is influenced by several factors. It depends on the entire software development process and related activities (e.g. requirements management, configuration management, design controls etc.). However, quality cannot be forced from outside. It is well known, but often forgotten, that it is the team and its relation to the project manager that decides on success or failure of a project.

Improving the skills of your project managers you will improve the quality of your products. The limiting factor usually is not related to technical skills that can be trained quite easily. In fact, project management is all about communication and leadership competencies which are far more difficult (but not impossible) to train. In this paper we report on our experiences with this kind of training. We point out what was most helpful and where the major difficulties lay.

Keywords

project management, soft skills, training, communication

1 Introduction

End of last century the Gallup Organization undertook extensive studies on "what the world's greatest managers do differently" [Gallup99]. Gallup surveyed over a million employees from various companies, industries and countries. Among other questions they tried to find out what talented employees need from their workplace.

The result was surprising. In the introduction of their book "First, break all the rules" the authors state that "the talented employee may join the company because of its charismatic leader, its generous benefits, and its world-class training program, but how long that employee stays and how productive he is while he is there is determined by his relationship with his immediate supervisor." [Gallup99]

This illustrates the crucial rule of project managers and the impact of the relationship between the individual team members and the project manager on the performance – and thus on quality. Consequently, if you want to improve efficiency and quality of your development teams, you should start improving the efficiency of the project managers.

2 The challenge

Trainings usually focus on technical skills. You learn about processes and methods (such as work breakdown structure or estimating techniques). Only a tiny part is dedicated to soft skills. Of course, processes and methods are important for project success, but they do not improve the "relationship" of the employee with "his immediate supervisor". For this soft skills are required – and should be trained.

The relationship between employees and their supervisor is closely related to the idea of leadership. People will follow those they admire, those that communicate a vision and those they feel attached to. In the world of SW development we tend to forget this aspect. We are used to base everything on logic. Emotions and conflicts are taboo. We are conditioned to keep them low.

Relationship cannot be explained purely by logic. It is something very personal, since it concerns individuals. One of the results what the world's greatest managers do differently is that they are not "just", which means that they do not treat their employees equally. Instead, they give different degree of autonomy and even different kind of benefits, depending on what is most motivating for the employee. This implies that the manager has to understand the individuals, their personality, their wishes and needs. Tom DeMarco goes even further in "The Deadline" when he states that "you have to like them" [DeMarco97].

Can this be trained? Surely, a certain disposition of the trainee is mandatory. What can be definitely trained are communication skills and the understanding of schemes and mechanisms. You can sensitize team leaders and project managers to the human factor in project management. Of course, this is not easy. Like skiing, it becomes more difficult with growing age.

Also, a lot of practice is required to internalize new behavior. The more you practice, the higher is the probability that you react correctly under pressure – which unfortunately has become rather the rule than an exception in project management nowadays. Obviously, we practice during daily work, but this is not necessarily the ideal solution. Difficult situations should not be exercised with real employees in critical projects. Instead, the training provides an ideal environment.

3 The training course

In our company, we recently performed a systematic training of 14 junior and senior project managers. The training consisted in 11 parts of one and half days each. The entire training took more than one year. In addition, three workshops were conducted in the beginning, at half-time and at the end.

The topics were all more or less related to communication and motivation. The following list is only an excerpt of the curriculum:

- roles and responsibilities
- communication and related competencies
- well defined goals
- delegation
- feedback and I-statements
- praise and acknowledgement
- team building
- moderation of meetings
- managing conflicts
- lateral management

Some topics (e.g. I-statements) were integrated in several modules so that we had the opportunity to practice a lot.

One aspect that was intrinsically covered by the entire training was employee motivation. Motivated people need transparency (what am I doing and what for?), the possibility of self-determination along with adequate work conditions, individual praise, development opportunities and a good personal relationship with their supervisor.

The difficulty with training soft skills is that you cannot learn them just by listening to someone. You have to do it yourself. You have to experience how it feels if something goes wrong. For this, roles plays are an ideal training method. Let us give an example: If your supervisor criticizes your work in the beginning of a discussion, it takes a long time until you are again able to listen to all the positive points that were said afterwards. In a role play, it is very instructive for both sides when one of them says: "I did not hear that" and everybody else confirms "He said it". After this salutary experience, you are really sensitized to the problem!

Apart from role plays we often worked in small groups. As we were a rather heterogeneous group regarding the level of managerial responsibility, this training method gave us the opportunity to get insight into each other's view. Sometimes, the groups had different exercises to solve, each of them approaching a topic from a different angle.

4 Experiences

The beginnings were difficult. The group was quite heterogeneous, some trainees reporting directly to others. At first, this created some irritations on both sides, but after some time people relaxed. Even more: the group grew together and a strong feeling of mutual respect has been developed. Today, we have a common understanding of what leadership means for us (and our company). Also, we developed some kind of shared responsibility for team management. As an example, we now give each other more feedback than before (also across divisions and hierarchic levels). This is probably one of the most important results that have been obtained by the training.

Depending on their current position and project, the trainees were more or less able to apply the lessons learned during daily work. The increased perception of communication problems proved to be most helpful. Just to give some examples:

1. People that are angry or preoccupied will not even hear what you are saying unless you given them time. Therefore, you have to check the other's mind-set to be sure that you can be heard.

2. Your vis-à-vis is not able to read your mind. Explain your actions and decisions. Transparency

is extremely important for the acceptance of what has been said!

3. Omissions are omnipresent. Think of it… What do you expect from your employee (or your supervisor) without having it said explicitly? Maybe, you take it for granted that the source code delivery includes design documentation – but what about the others?

As a curious side effect, the increased perception can also create an increased irritability. For example, it is exasperating to be confronted with unclear goals. However, thanks to the training we realize the problem and are able to react – by asking questions.

Training soft skills requires an attitude change of the trainees. Obviously, this is not easy. If you succeed, you may encounter another difficulty. Your changed behaviour is surprising for those who have not participated in the training. People tend to be suspicious about your new attitude. Again, transparency is everything.

For the trainer, teaching soft skills to people from IT – all with technical background and quite matter-to-fact – is a particular challenge. In the beginning, there was a considerable reluctance to enter the game. Also, it was difficult to reconcile daily project work with the rather time intensive training.

5 Conclusion

Project management has direct impact on what could be called "intrinsic quality" of a project. The key activities are communication and motivation. If these are well done, your employees will do good work. Therefore, it is as important to train the related soft skills as it is to train project management processes and techniques.

Of course, all this is not new. In theory we all know about the do's and don'ts of project management. If not, read Tom DeMarco [DeMarco97]. We also know that employee motivation is the most important task of project management. So, why do we have to talk about it? The fact is that even if we know the theory, practice locks different. A common misunderstanding regarding employee motivation is that you cannot motivate someone. Instead, people must be self-motivated. With this mentality we tend to forget how easy it is to de-motivate people.

In our company, we performed a systematic training of junior and senior project managers: We obtained excellent results. Besides individual progress we obtained a common understanding of leadership and build a strong project manager team.

6 Literature

[DeMarco97] "The Deadline: A Novel About Project Management", Tom DeMarco, Dorset House Publishing Co, Inc., 1997

[Gallup99] "First, break all the rules: what the world's greatest managers do differently", Marcus Buckingham and Curt Coffman, The Gallup Organization, Simon and Schuster, New York 1999

7 Author CVs

Dr. Anne Kramer

Anne Kramer works as project manager and senior consultant at sepp.med gmbh, a service provider specialized in IT solutions with integrated quality assurance in complex, safety-relevant domains.

Anne holds a diploma in Physics and received her PhD at the Université Joseph Fourier in Grenoble (France). Her first contact with quality assurance dates back to 1996, when she started her career as a software developer for smart card testing tools at Schlumberger Systems in Paris. Two years later she became project manager for point of sales terminals.

In 2001 she joined sepp.med where she specialized in quality assurance methods and IT development processes for the medical and pharmaceutical domain. She regularly contributes to magazines and conferences as author or speaker. Anne holds a certificate as Certified Professional for Requirements Engineering (Foundation Level).

Design of a competence Model for Global Software Development Teams

Javier Saldaña-Ramos, Ana Sanz-Esteban, Javier García-Guzmán, Antonio Amescua-Seco
Computer Science Department. Carlos III University of Madrid
Avda. de la Universidad, 30, 28911 Leganés (Madrid)
jsaldana@inf.uc3m.es, asanz@inf.uc3m.es, jgarciag@inf.uc3m.es,amescua@inf.uc3m.es

Abstract

Nowadays, it is common to develop software development projects collaboratively among team members or organizations in different geographical locations. These teams, known as global software development teams, allow organizations to save costs as well as have available highly qualified personnel. This kind of team is different from traditional teams, so it is necessary for team members to develop other essential competences to work efficiently in a global context.

Unfortunately, there is no well-defined competence model that allows organizations to assess personnel competences and establish the relevant training program that allows them to work efficiently in such teams. This work defines and implements, in four global software development teams, a competence model specifically designed to address challenges that people face when they work in a global software development team. This competence model has been defined considering tasks a global software development team have to carry out, bodies of knowledge and existing competence models for the software engineering profiles and the authors' experience working in such teams. As a result of the implementation process, it was confirmed that the competence model is a key factor for human capital improvement. When personnel have these competences, team and individual efficiency and product quality increase, and delays in delivering products decrease.

Keywords

Competences, competence model, global software development, GSD, global software development teams, Global IT Professionals

1 Introduction

Globalization has resulted in a substantial increase in outsourcing in different industrial and service sectors (Braun 2007). Outsourcing of software activities (development, test, maintenance, programming, and incidence management) is essential to maintaining the required levels of competitiveness and productivity in large software engineering projects (Herbsleb and Moitra 2001; Sengupta et al. 2006). This requires the creation of multidisciplinary teams composed of people working as a single team in the same software development project but in different locations; this is called Global Software Development (GSD) team.

Many organizations have implemented Global Software Development and have found that GSD teams are highly complex (Herbsleb and Moitra 2001). This complexity arises from four main challenges they face. First, lack of common understanding of team goals makes team members feel isolated, so they are reluctant to collaborate and share knowledge and work together (Striukova and Rayna 2008). Second, difficulties in communication among different team members who are geographically distributed (Fuller et al. 2007). Third, differences among individual management mechanisms and the associated skills create problems and bottlenecks in project execution (IPRC 2007). Fourth, ineffective management of shared knowledge among team members causes duplication, inconsistencies and lack of knowledge of project assets (Rosen et al. 2007).

Competences are relevant to increasing the human capital of an organization. If the staff is well-qualified, they will do their tasks efficiently (Teodorescu 2006). According to Sagi-Vela (2004), a competence is: "*the knowledge, skills and attitudes that applying to carry out certain responsibility or professional contribution ensure their success*". The competence level of an organization's staff is the most relevant factor for its success, as without the suitable competences the personnel is one of the major risks of a software project (Boehm 1991). Individual skills constitute a decisive success factor and communication, motivation and teamwork are essential competences (Isaac et al. 2006). Although most of the required competences are similar to those in traditional software development, there are others which have to be developed by the members of GSD teams in a specific way (Hawthone and Perry 2005;Petkovic et al. 2006).

One requirement to reduce the GSD problems mentioned above, and which ensures the success of software projects developed in global contexts, is having highly-skilled staff to work in a GSD team (Garcia 2008), that is, team members should acquire the appropriate competences to work in a GSD project. Therefore, a competence model for GSD teams should be used. A competence model is a catalogue in which both general and technical competences needed to perform a professional role are identified and defined, including the level required for each one (Urquiza 2009). It is used as a key mechanism to improve human capital and has an important influence on business processes (Sagi-Vela).

The most widespread bodies of knowledge do not take into account the special features of GSD teams (Honing and Prasad 2007) nor do education programmes (Petkovic at al. 2006). As a consequence, software engineers do not have the appropriate competences to work in a global context. They need special training courses to do so. The authors have not found in the literature a well-defined competence model which includes the appropriate competences for GSD teams that allow to train personnel efficiently. Therefore, it is necessary to define a competence model, specifically addressed to these teams, that allows organizations to have available highly-skilled personnel.

This paper defines a competence model which determines the competences required for staff who work in a global software development context in order to manage efficiently GSD teams and ensure project success. The proposed competence model was implemented in four global software development teams.

This paper is organized as follows. Section 2 justifies the need for a competence model for GSD teams. Section 3 describes the process followed to define this model. Section 4 presents a brief description of the proposed competence model. Section 5 describes the implementation process and analyzes and discusses the main results. Finally, section 6 presents the conclusions drawn from this work.

2 A Need for a Competence Model for GSD Teams

Global software development teams are teams especially defined to work in geographically distributed projects and involve multiple participants from different countries, so different cultures and languages co-exist (Cohen & Gibson 2003). There are several strategies to configure this kind of team; the strategy adopted depends on the interdependence that exists among team members. In some projects, project management, requirement specification and architectural design are carried out in one location and the software development or modification of software components by software factories are set up in separate geographical locations (Edwards and Sridhar 2003; Smite 2004). Other projects require the creation of several coordinated multidisciplinary teams, bringing together highly-skilled individuals working in geographically dispersed locations.

The specific features that differentiate GSD from traditional teams are the following: members working on interdependent tasks across different geographical locations (Hyrkkänen et al. 2007), and the relevant knowledge for each task is usually distributed among different sites (Striukova and Rayna 2008). In addition, team members have different mother tongues and cultures that contribute to making communication difficult (Kankanhally et al. 2007) and communication activities rely strongly on technologies (Petkovic at al. 2006). Finally, team work dynamics and the different work location dynamics need to be combined because of the time difference, making agenda management more difficult (Lee-Kelley and Sankey 2007). As a result, staff must be properly trained, developing the appropriate competences to carry out the project tasks in a GSD team.

Software engineers acquire competences which are not sufficient to develop their tasks in a global context; there are new ones which are not included, although they should be, in computer science studies (Hawthorne and Perry 2005). An ACM study on software outsourcing and globalization recommends that computer engineers must be better prepared, and it mentions specifically the need for training in soft-skills, teamwork, communication, process organization and issues related to globalization (Petkovic et al. 2006). In these contexts, competences that individuals need to achieve their goals have become more complex, requiring a mastery of certain skills that are more closely defined (Rychen and Salganik 2003).

Competence models are used as a basis for management systems by competences, which are key mechanisms to improve human capital in an organization and influence business processes (Sagi-Vela 2004). Draganidis and Mentzas (2006) identify the sequence of activities associated with the key processes for management by competences: first, obtain the Competence Map (Competence Model) and the required levels for each role; second, carry out employee diagnosis and gap analysis; third, develop competences based on previous results and needs identified; and fourth, monitor progress of competences development continuously.

According to this proposal, the first step is to look for a competence model for GSD teams or to define a new one if none exists. The authors have not found any well-defined competence model which includes the set of appropriate competences in the literature. Some authors, such as Wongmingji (2005) or Sampaio and Moniz (2009), have made several proposals about what competences should be included for work in GSD teams. Nevertheless, these proposals do not fulfil requirements completely. Competences should be related to specific practices which support: a) explicit team management practices because this kind of team involves people from different countries with different cultures, so it is necessary to define explicitly and publish these practices; b) shared knowledge management practices; c) practices for work processes integration d) adaptive technology-based collaboration patterns practices since communication and collaboration rely strongly on technology. Considering there is no published competence model specifically addressed to GSD teams and the need for a well-defined competence model for GSD teams, this research work defines one and implements it in several GSD teams.

3 Competence Model Definition Process

There are different approaches to elaborating a competence model. Rothwell and Kazanas (1993) summarize them in three ways: direct use of an existing model, adaptation of an existing model according to specific features of the organization, or establishment of the organisation's own model.

When an organisation is going to create its own model, several methodologies can be used:

- Job analysis: It is the most widespread methodology and it meets an "internal" type of model. This methodology identifies competences from job features in terms of mission in the organization, operative objectives, assessment criteria, organizational situation, activity areas and autonomy level, and job responsibility (Pereda and Berrocal 2001).
- Collection and synthesis of statistical information: This approach is done through interviews or surveys, and/or experts filling in complex forms in each matter or activity area in order to synthesize answers and results into the model (Dawes and Helbig 2006).
- Mixed model: In addition to the two previous types of methodologies, some mixed approaches have emerged recently. They combine both internal and external issues of the organization. One example of this kind is the POCCI (Process Oriented Core Competency Identification) Model (Yang et al. 2006). It identifies the core competences of an organization based on external factors of the organization's environment, then it creates strategic guidelines associated with the initial analysis.

In business scenarios, the generation of a competence model aims to identify competences (and competences levels) that really have an impact on results, and typically a methodology based on job analysis is applied (Draganidis and Mentzas 2006). The competence model developed in this work has been defined following this methodology, considering the main type of tasks to be performed in a GSD team.

In order **to carry out the job analysis, the authors used a lifecycle model for managing GSD teams, VTManager** (Garcia-Guzmán et al. 2010), because it provides a set of tasks that a GSD team has to carry out in order to develop software development projects in a global context, identifying what knowledge, skills and attitudes are necessary in each situation **as well as the roles defined for GSD teams** (Saldaña, 2010). Figure 1 shows this lifecycle.

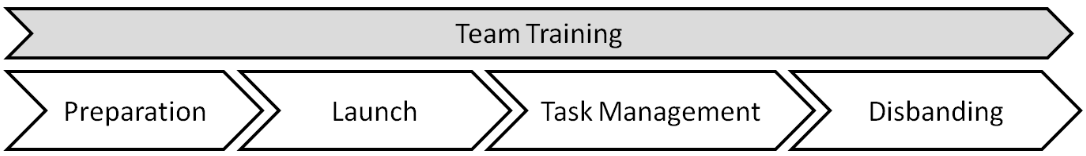


Figure 1: Lifecycle model to Global Software Development teams.

- Preparation: Determine team composition and goals.
- Launch: Establish an action plan that allows team members to achieve the defined goals.
- Task Management: Develop the planned tasks and track the project progress.
- Disbanding: Decide how to disband the global software development team as well as how to re-integrate team members.
- Team Training: Define and carry out a training program to enable team members to work in a global context.

Moreover, the authors used other existing competence models for computer science profiles and this information provides the main bodies of knowledge as a basis to design the competence model. This new model includes competences and the competence level for four professional roles that are involved in a global software development team. These roles are:

- Organizational unit manager (OUM): His main responsibilities are managing resources, both material and human, and actively participating in team-building activities. He provides information on needs and validates the results, and also participates in project monitoring and tracking.
- Project manager (PM): He performs team coordination and management tasks. He is responsible for project management and takes part in project monitoring and tracking. Further, he participates in the result evaluation.
- Team leader (L): His main task is to coordinate the GSD team and to achieve both team and project objectives. He supports and encourages team members and coordinates the training activities. He is a liaison between the project manager and software engineers and provides support to the project manager.

- Software engineer (SE): This role is responsible for carrying out the technical project tasks, specifically addressed to developing and validating the software product in a global software development context.

The competence model definition process, which is presented following the SPEM meta-model (OMG 2008, consists of three main phases (Figure 2).

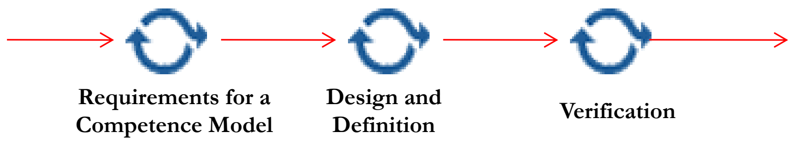


Figure 2: Competence model definition process

First, requirements for competence model were established. For this purpose the GSD literature that shows the need for a competence model for GSD teams was analysed (Figure 3), for example Honig and Prasad (2007), Petkovic et al. (2006), Hawthorne and Perry (2005).

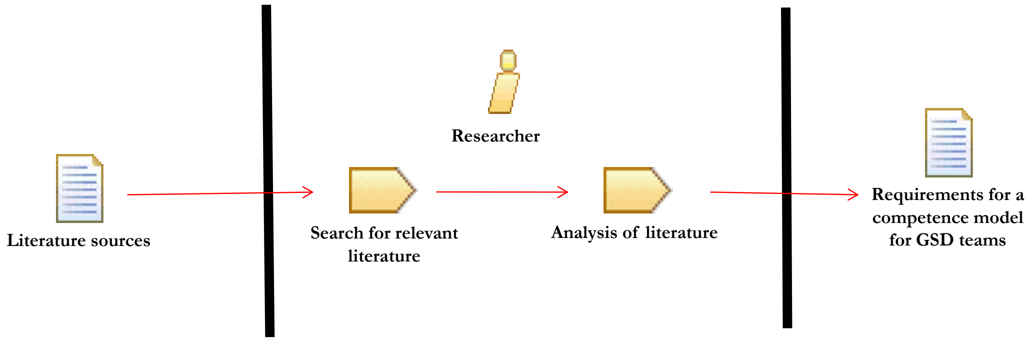


Figure 3: Requirements for a competence model for GSD teams

Second, the competence model was designed and defined (Figure 5).

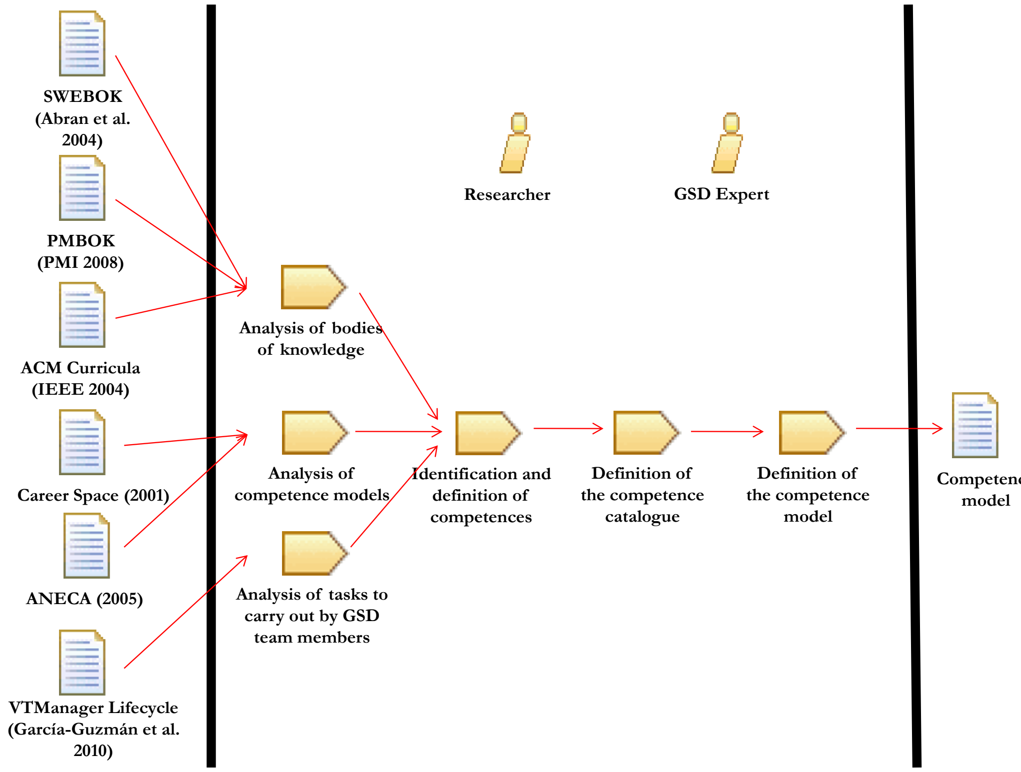


Figure 4: Definition and design of the competence model

The main bodies of knowledge used to develop the computer science studies and existing competence models for computer science profiles were analysed as well as tasks to be carried out by GSD team members based on the VTManager lifecycle. Considering this information and the authors' expertise in GSD teams, the competence catalogue was defined. First, competences that are the same for traditional teams were identified and defined, and then those required specifically for GSD teams. Finally, the competence model was defined, establishing for each competence the level required for each role.

The last phase consisted of verifying the competence model and determining whether it was appropriate for GSD team members (Figure 5). For this purpose, the competence model was implemented in four Global Software development teams that took part in the C@R project. After that, and resulting from this process, it was adjusted according to the results obtained. Section 5 shows this process in more detail.

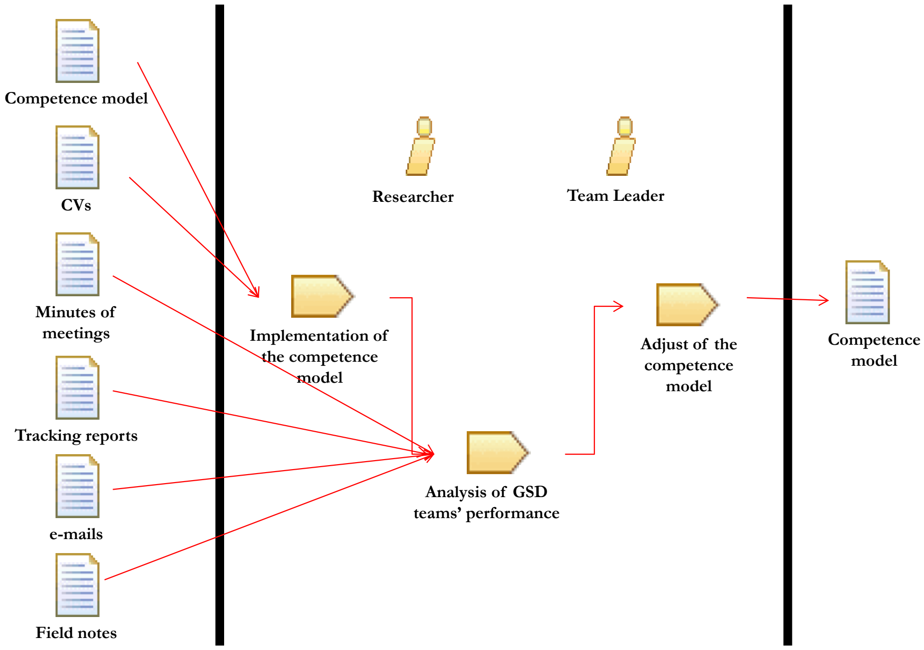


Figure 5: Verification of the competence model

4 Competences for GSD Teams

The detailed competence model developed in this work is available at http://seldata.sel.inf.uc3m.es/docs/CompetenceModel/competencemodelforGSD.pdf, and it includes both technical and general competences required for working in a GSD team (45 technical and 27 general competences**) as well as the competence level required for each role** (organizational unit responsible, project manager, team leader and software engineer).

Due to the length of the paper, **the authors have only presented a reduced version of the competence model and included only those competences that are new, have either never been identified in the literature or have been redefined substantially to suit the features of GSD teams.** The remaining competences have been modified slightly according to the specific characteristics of the distributed environments. Table 1 gives a brief summary of these.

Technical Competences		General Competences	
1	Management Meeting.	16	Interpersonal conflict resolution adapted to distributed environments.
2	Competence identification.	17	Positive attitude and motivation capacity.
3	CV interpretation.	18	Self-learning capacity.
4	Management of synchronous and asynchronous communication in global contexts.	19	Interpersonal relationship ability.

Technical Competences		General Competences	
5	Requirement identification.	20	Ability to use communication and information technologies.
6	Requirement estimation and prioritizing.	21	Ability to work in a global setting
7	Requirement management.	22	Ability to communicate orally and in writing in English.
8	Collect, analyse and interpret information.	23	Organization and planning capacity.
9	Technical problem solution.	24	Initiative and leadership.
10	Share knowledge management.	25	Decision taking
11	Brainstorming session management		
12	Establish rules to work with shared data.		
13	Analyses of synergies from information gathered from social networks.		
14	Online training creation and management.		
15	Use of advanced techniques for distributed communication.		

Table 1: Technical and general competences for GSD teams.

Moreover, the authors have identified the competence level, technical and general competences that the main roles who are involved in a global software development should have (see Figure 6).

Tecnical competence level				
TC	ROLES			
	OUM	PM	L	SE
1	H	VH	M	L
2	VH	VH	M	L
3	VH	VH	L	L
4	M	H	VH	H
5	M	VH	VH	H
6	M	H	VH	M
7	M	H	VH	H
8	VH	VH	H	L
9	M	H	H	VH
10	H	H	VH	VH
11	H	VH	M	L
12	H	VH	M	L
13	M	H	VH	L
14	L	M	H	L
15	M	H	VH	H

General competence levels				
GC	ROLES			
	OUM	PM	L	SE
16	VH	VH	H	M
17	H	H	VH	M
18	M	H	H	VH
19	H	VH	VH	H
20	H	H	H	VH
21	H	H	VH	VH
22	VH	VH	VH	H
23	VH	VH	H	M
24	H	H	VH	L
25	VH	H	M	L

Figure 6: Competence levels required for each role.

The competence levels shown in Figure 6 have the following meaning.

- Low (L): Team member does not have this competence.
- Medium (M): Team member is still learning. The competence has been partially acquired.
- High (H): Team member is autonomous. The competence is totally acquired.
- Very High (VH): Team member masters the competence and practices it in a stable way.

5 Implementation

The main objective of the implementation is to verify that the competences identified are appropriate for GSD teams and, if necessary, to modify the model according to the implementation process results. Further, it allows to check if the team works more efficiently and problems among members decrease when team members have the competences associated with their roles. If this is the case, GSD team performance will increase and the competence model has been developed appropriately.

The *Software Engineering Lab* (SEL) of Carlos III University of Madrid participated in a European Project "C@R - Collaboration at Rural" (IST-2006-034921) (http://www.c-rural.eu/), to implement this model which was developed in a global setting. This project lasted 36 months and the consortium was composed of 30 international partners from 15 different countries. Its main goal was to develop collaborative software tools to overcome some of existing barriers to accessing knowledge society in rural settings. The competence model was implemented in four of the teams that participated in the project.

5.1 Description of the Implementation

The authors carried out a multiple case study (Yin 2008) with one unit of analysis, competences, to implement the competence model in the four teams. Several techniques were used to ensure the validity of data. In particular, the authors used multiple data collection methods such as project documents, minutes of meetings, interviews or emails; and different techniques to analyse and interpret data: multiple triangulation, prolonged participation and periodic observation as well as logical chain of evidence and explanation building.

The case study developed consisted of three main phases:

- <u>Case study definition and design</u>: In this phase the authors analysed the environmental features and selected a multiple case study as the best option to implement the competence model.

- <u>Preparation, collection and analysis of evidence:</u> In order to implement the competence model a plan was made based on the length of the project (from 2006 to 2009), and types of evidences were established: documents (CVs, minutes of meetings, tracking reports and e-mails), records of participant observation (field notes) and interviews, when evidences were going to be collected, and the strategy to corroborate results was defined.

  As regards collection of information, multiple triangulation, prolonged and persistent participation techniques were used to ensure that data were objective and representative. Finally, the data collected were analysed and the results corroborated periodically by interviewing the team leaders. This way the competence model could be improved and refined.

- <u>Conclusions and reporting:</u> In this phase the conclusions extracted from data were analysed and the results reported, thus obtaining the competence model presented in this work.

As a result of the case study the competence model was refined as shown in section 5.2.

5.2 Analysis of Results and Discussion

According to evidences gathered from CVs of team members as well as data obtained from staff tracking reports and interviews with team leaders, it was possible to determine that team members have the necessary qualification and a great experience working successfully in global software development teams.

From this, the competences identified were checked, reviewed and modified to achieve all requirements. In some competences the changes were related to the competence level for each role, the main results are:

- *"Management of synchronous and asynchronous communication in global contexts"*. Firstly, a high level had been assigned to the team leader in this competence. However, findings showed that the team leader plays a key role in communication management. He must ensure that communication is fluent and select the best technology to do so. Therefore, the required level was increased from high to very high.

- "*Requirement identification".* At first, a high level had been assigned to the project manager in this competence. However, findings showed that he plays a key role in tasks related to requirement identification because of his responsibility for establishing the team goal. So, the required level was increased from high to very high.

- "*Requirement management".* Initially, a medium level had been assigned to the software engineer in this competence. However, findings showed that he plays a key role in tasks related to architecture analyses included in task management phase. To do so, the software engineer frequently needs to work with functional and non-functional requirements and he sometimes makes changes to requirements. The required level was increased from medium to high.

- *"Ability to use the communication and information technologies"*. Initially, a medium level had been assigned to the software engineer in this competence. However, findings showed that collaboration and communication are important elements in his work, so software engineers need to know and use communication and information technologies. The authors considered it necessary to increase the required level from medium to very high. In addition, a high level had been assigned to the team leader in this competence. However, findings showed that he plays a key role when the intra-team rules are defined, and in controlling the team work. These tasks required good use of communication and information technologies, so the required level was increased from high to very high.

In other cases, it was necessary to combine two competences. For instance, at first there were two competences related to language, one called "*knowing about English*" and "*ability in oral and written communication*"; team leaders indicated that due to the nature of multicultural teams these competences were actually only one: "*ability to communicate in oral and written in English*". Another example is the general competence "*Initiative and leadership*" which was initially two competences.

Finally, some competences, such as "*objectivity and diplomacy",* were eliminated because team leaders considered them unnecessary or irrelevant to GSD teams. The findings showed that it was really a general competence which was very difficult to identify based on the curriculum and personal interviews. Also, it did not have significant repercussions on selecting the team members.

After reviews and modifications, the authors determined that competences described in the competence model developed in this work were appropriate for GSD teams. Moreover, it was possible to verify the importance of knowing individual competences to assign the suitable person for each task, so that each member developed the activities based on their knowledge and skills. This fact improves the efficiency and performance of each member because they are more motivated and conflicts among team members decrease. At the end of the project, the team leaders and project team stressed the importance of a well-defined competence model, such as the one described in this work, to assign personnel to tasks which best fit them. A suitable assignation increases motivation and improves performance. As a result, products have more quality and the delivery period decreases. Being well-prepared to develop a specific task not only increases someone's motivation but also their confidence , and consequently, team members are more willing to share knowledge, and communication and collaboration improve. Competences related to TICs and self-learning reduce communication difficulties and ineffective information management. The final conclusion obtained from these results is that a well-defined competence model for GSD projects improves project performance and increases the quality of the product.

6 Conclusions and future lines

The main objective of this work was to define and implement a competence model for GSD teams. This competence model contains both general and technical competences required to work efficiently in a GSD team, as well as the competence level required for each role. Roles have been identified from the main professional profiles that appear in a global software development team.

The competence model was developed considering the needs (requirements) found in the literature and because there is no competence model specifically addressed to these teams. For this purpose, the authors performed a job analysis following the VTManager lifecycle, analysed the main bodies of knowledge used to create education programmes in computer science and considered their experiences working in GSD teams in order to identify the competences required and the competence levels for each role.

Once this model was defined, it was implemented in four global software development teams to determine whether the competences and competence levels were appropriate. As a result of this implementation, the competence model was refined, eliminating or modifying some competences and adjusting competence levels in several cases. Team leaders and members interviewed corroborated that the competence model was appropriate for GSD teams and that when team members have these competences the efficiency of team members increases and the quality of products improves.

Further, the competence model developed in this work is the first step to implementing a management system by competences, which is being used successfully in other types of organizations to improve their human capital and, consequently, their business processes and results. Future lines of work will be addressed to implement this competence model in other GSD teams and develop specific courses to acquire the competences required by GSD team members according to their individual profile or role aligned with the VET programmes.

7 Literature

ABRAN, A., MOORE, J.W., BOURQUE, P., DUPUIS, R., and TRIPP, L.L. (2004). *SWEBOK: Guide to the Software Engineering Body of Knowledge: 2004 version*. IEEE Computer Society, Los Alamitos, CA, Tokyo

AGENCIA NACIONAL DE EVALUACIÓN DE LA CALIDAD Y ACREDITACIÓN (ANECA). (2005). *Libro Blanco, Ingeniería Informática*. Madrid, 2005.

BOEHM, B.W. (1991). Software risk management: principles and practices. *IEEE software*, 8(1), 32-41.

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, Los Alamitos, CA, USA.

CAREER SPACE. (2001). Generic ICT Skills Profiles: Future Skills for Tomorrow's World. CEDEFOP, Thessaloniki, Greek.

COHEN, S. G., and GIBSON, C. B. (2003). Putting the Team Back in Virtual Teams. In *Proceeedings of the 18th Annual Conference of the Society for Industrial/Organizational Psychology*, Orlando, Florida.

DAWES, S., and HELBIG,N. (2006). Building government IT workforce capacity: a competency framework. In *Proceedings of the 8th Annual International Digital Government Research Conference*, Center for Technology in Government, Albany, New York, USA.

DRAGANIDIS, F. and MENTZAS, G. (2006). Competency based management: a review of systems and approaches. *Information Management & Computer Security*, 14(1), 51-64.

EDWARDS, H.K., and 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*, Big Island, Hawaii. IEEE Computer Society Press, LA, California, USA.

FULLER, M., HARDIN, A., and DAVISON, R. (2007). Efficacy in Technology-Mediated Distributed Teams. *Journal of Management Information Systems*, 23(3), 209-235.

GARCÍA, G. (2008). Equipos de Trabajo Virtuales: Un nuevo reto de liderazgo. *Factor Humano en el S. XXI*, 65, 1-2.

GARCÍA-GUZMÁN, J., SALDAÑA-RAMOS, J., AMESCUA-SECO, A., and SANZ-ESTEBAN, A. (2010). How to get mature global virtual teams: a framework to improve team process management in distributed software teams. Software Quality Journal, 18(4), 409-435.

HAWTHORNE, M. J., and PERRY, D.E. (2005). Software engineering education in the era of outsourcing, distributed development, and open source software: challenges and opportunities. In *Proceedings of the 27th International Conference on Software Engineering*, (pp. 166-185), St. Louis, Missouri, USA. Springer.

HERBSLEB, J.D., and MOITRA, D. (2001). Global Software Development. *IEEE software* 18(2), 16-20.

HONIG, W.L., and PRASAD, T. (2007). A classroom outsourcing experience for software engineering learning. *In Proceedings of the 12th annual SIGCSE conference on Innovation and technology in computer science education*, (pp. 181-185), Dundee, Scotland. ACM, New York.

HYRKKÄNEN, U., PUTKONEN, A., and VARTIAINEN, M. (2007). Complexity and workload factors in virtual work environments of mobile work. *Ergonomics and Health Aspects*, 4566, 85-94.

IEEE COMPUTER SOCIETY/ACM JOINT TASK FORCE ON COMPUTING CURRICULA (IEEE). (2004). *Software Engineering 2004: Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering*. Retrieved 08/10/2009 from: http://sites.computer.org/ccse/SE2004Volume.pdf.

INTERNATIONAL PROCESS RESEARCH CONSORTITUM (IPRC). (2006). *A Process Research Framework*. Pittsburgh :Software Engineering Institute.

ISSAC, G., RAJENDRAN, C., and ANANTHARAMAN, R.N. (2006). An instrument for the measurement of customer perceptions of quality management in the software industry: An empirical study in India*. Software Quality Journal*. 14(4), 291-308.

KANKANHALLI, A., TAN, B., and WEI, W. (2007). Conflict and Performance in Global Virtual Teams. *Journal of Management Information Systems*, 23(3), 237-274.

LEE-KELLEY, L., and SANKEY, T. (2007). Global virtual teams for value creation and project success: A case study. International Journal of Project Management, 26(1), 51-62.
O.M.G. (2008). Software Process Engineering Metamodel Specification. Version 2.0, formal/2008-04-01. Retrieved 21/02/2011 from: http://www.omg.org/docs/formal/08-04-01.pdf.
PEREDA, S., and BERROCAL, F. (2001). *Técnicas de Gestión de Recursos Humanos por Competencias*. Centro de Estudios Ramón Areces, Madrid, Spain.
PETKOVIC, D., TODTENHOEFER, R., and 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), San Diego, CA, USA. IEEE Computer Society Press.
PROJECT MANAGEMENT INSTITUTE (PMI). (2008). *PMBOK: A Guide to the Project Management Body of Knowledge (Fourth edition)*. Project Management Institute, Newtown Square, Pennsylvania, USA.
RYCHEN, D.S. and SALGANIK, L.H. (2003). *Key competencies for a successful life and a well-functioning society*. Hogrefe Publishing, cambridge, USA.
ROSEN, B., FURST, S., and BLACKBURN, R. (2007). Overcoming Barriers to Knowledge Sharing in Virtual Teams. *Organizational Dynamics*, 36(3), 259–273.
SAGI-VELA, L. (2004). *Gestión por Competencias*. ESIC, Madrid, Spain.
SALDAÑA, J. (2010). *VTManager: Un Marco Metodológico para la Mejora en la Gestión en los Equipos de Desarrollo Software Global*. Thesis, Carlos III University, Leganés, Spain.
SAMPAIO, J. and MONIZ, A. (2009). *Assessing Human and Technological Dimensions in Virtual Team's Operational Competences*. Retrieved 25/06/2010 from: http://mpra.ub.uni-muenchen.de/18325/.
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. (2004). Global Software Development Project Management-Distance Overcoming. In *Proceedings 11th European conference on Software process improvement,* (pp. 23-33), Trondheim, Norway. Springer-Verlag.
STRIUKOVA, L., and RAYNA, T. (2008). The role of social capital in virtual teams and organizations: corporate value creation. *International Journal of Networking and Virtual Organizations*, 5(1), 103-119.
TEODORESCU, T. (2006). Competence versus competency: What is the difference?. *Performance Improvement*, 45(10), 27-30.
URQUIZA, A. (2009). Creación de un Marco de Competencias para la Evaluación del Rendimiento de los Gestores de Sistemas de Información en las Grandes Organizaciones. Thesis, University of Alcalá, Alcalá de Henares, Spain.
WONG-MINGJI, D.J. (2005). *Leadership Competencies for Managing Global Virtual Teams*. Retrieved 08/07/2010 from: http://encyclopedia.jrank.org/articles/pages/6654/Leadership-Competencies-for-Managing-Global-Virtual-Teams.html.
YANG, B.C., WU, B.E., SHU, P.G., and YANG, M.H. (2006). On establishing the core competency identifying model: A value-activity and process oriented approach. *Industrial Management & Data Systems*, 106(1), 60-80.
YIN, R. (2008). *Case study research: design and methods. 4th Edition.* Sage Publications, Inc, London, United Kingdom.

8 Acknowledgments

This research is partially supported by the Integrated Research Project C@R-Collaboration at Rural (IST-2006 034921) funded by the European Commission. The research work for this paper has been also funded by TRAGSA under the UC3M/2008/00259/001 and UC3M/2007/04245/001 projects.

9 Author CVs

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 received his PhD in computer science from the Carlos III University of Madrid.

Javier is 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.

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 lecturer 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. Her research is also related to human resource management, she is developing a competence system to apply in software testing area.

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 published several research articles, book chapters and conference papers in these areas.

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.

Presentation of ISO 26000 and SOCIRES

Beatrice Bachmann[1], Dr. Romana Vajde Horvat[2]
[1] SIBAC gmbH, Germany
[2] proHUMAN Ltd, Slovenia

Abstract

In this paper we present the International Standard ISO 26000 which was developed to give guidance on Social Responsibility in companies and organisations worldwide. The importance and its potential implementation in communication methods especially for quality management is emphasized. It is to be expected that more and more aspects of Social Responsibility (SR) will become necessary - or some of them even obligatory - for organizations. This will lead also to the need for Social Responsibility competency and SR skilled employees in organizations. We introduce SOCIRES, the Social Responsibilities Training and Certification Schema for the Social Responsibility Manager (SRM) on ISO 26000 related content. Results of the project will be used for training and certification of ECQA Certified Social Responsibility Managers.

Keywords

Certification, Communication, Cooperation, ECQA - European Certificartion and Qualification Associtiion, International Standard ISO 26000, Quality Management, Skills, SOCIRES, Social Responsibility, Social Responsibility Manager, Training

1 Introduction

Every organization somehow influences the society. This includes influence on health, well-being and life in general of employees and their relatives, influence on environment, cooperating organizations and various stakeholders, including customers or consumers, workers and their trade unions, members, the community, nongovernmental organizations etc.

The topic of social responsibility proved to be very important within last two years when the financial crises stroke the market. It quickly become evident how many individuals are directly or indirectly influenced by a certain company. Let's take an example where a large textile company with 3000 employees would be closed. Consequently, numerous of subcontractors and partners will have to close as well. Due to lower amount of the money available for spending in the region, other companies offering various services and products will have difficulties in finding customers. Closing one large company might influence so much that the whole region might enter the crisis. Similar scenarios are currently happening in majority of countries all over the world.

Within governmental programs it is very often planned that some measures have to be taken to prevent similar scenarios in the future. Preventive actions should also include the understanding of social responsibility of a certain organization and to understand what potential measures might be efficient to optimize the balance between influence, internal interests of organization and external interests of external partners. It is very likely that some aspects of Social Responsibility (SR) will become obligatory for organizations, like it happened for environmental management (ISO 14000). This will lead to increase of need for SR skilled employees in organizations.

In addition the governmental bodies will need to know the SR influence of certain organization in local environment. This awareness will lead to increase of need for SR skilled employees in governmental and other important decision-making bodies.

One of the tools for improvement of the status is the ISO 26000 is an ISO International Standard which will be presented further in this paper.

To help dealing with the need for improvement of knowledge, related to the SR topic on the market, the SOCIRES - Social Responsibilities Training and Certification Schema **for the Social Responsibility Managers** has been started under LLP Programme. The project will be presented in the second part of the paper.

2 Guidance on social responsibility - ISO 26000

An excellent explanation about the intent and content of ISO 26000 is found for example in www.iso.org. It basically explains the following:

„ISO 26000 is an ISO International Standard giving guidance on Social Responsibility. It is intended for use by organizations of all types, in both public and private sectors, in developed and developing countries, as well as in economies in transition. It will assist them in their efforts to operate in the socially responsible manner that society increasingly demands.

ISO 26000 contains voluntary guidance, not requirements, and therefore is not for use as a certification standard like ISO 9001:2008 and ISO 14001:2004.

ISO 26000 is not a management system standard but a guidance. It is not intended or appropriate for certification purposes or regulatory or contractual use as such. NO certification against the standard exist, there for any offer to certify, or claims to be certified, to ISO 26000 would be a misrepresentation of the intent and purpose of ISO 26000." [1]

ISO 26000 provides guidance on:

- Concepts, terms and definitions related to social responsibility
- Background, trends and characteristics of social responsibility
- Principles and practices relating to social responsibility
- Core subjects and issues related to social responsibility
- Integrating, implementing and promoting socially responsible behaviour throughout the organization and, through its policies and practices, within its sphere of influence Identifying and engaging with stakeholders
- Communicating commitments, performance and other information related to social responsibility.

By applying these seven core elements of ISO 26000, an organisation will enhance in the field of social, environmental, legal, cultural, political and organizational diversity, as well as in differences in economic conditions.

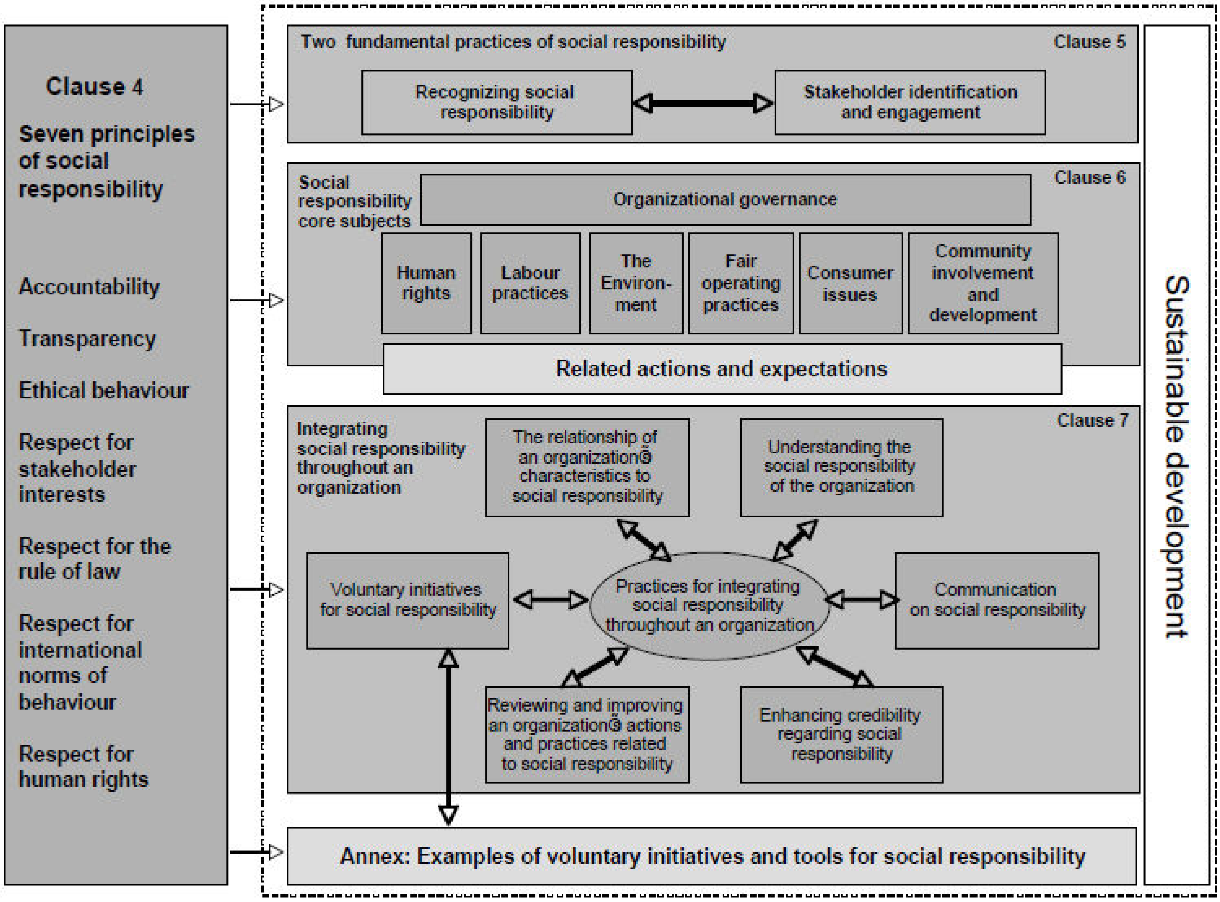


Figure 1: Elements of ISO 26000

2.1 Content of ISO 26000

As shown above ISO 26000 addresses seven core subjects of social responsibility defined in the standard. These seven core subjects together with their sub items can be found in *Entwurf Önorm ISO 26000:2009*. They are presented in Table 1 on next page.

***Table 1.* Core subjects and issues of Social Responsibility defined in ISO 26000**

Core subjects and issues	***Addressed in sub-clauses***
Core subject: Organizational governance	6.2
Core subject: Human rights	6.3
Issue 1: Due diligence	6.3.3
Issue 2: Human rights risk situations	6.3.4
Issue 3: Avoidance of complicity	6.3.5
Issue 4: Resolving grievances	6.3.6
Issue 5: Discrimination and vulnerable groups	6.3.7
Issue 6: Civil and political rights	6.3.8
Issue 7: Economic, social and cultural rights	6.3.9
Issue 8: Fundamental principles and rights at work	6.3.10
Core subject: Labour practices	6.4
Issue 1: Employment and employment relationships	6.4.3
Issue 2: Conditions of work and social protection	6.4.4
Issue 3: Social dialogue	6.4.5
Issue 4: Health and safety at work	6.4.6
Issue 5: Human development and training in the workplace	6.4.7
Core subject: The environment	6.5
Issue 1: Prevention of pollution	6.5.3
Issue 2: Sustainable resource use	6.5.4
Issue 3: Climate change mitigation and adaptation	6.5.5
Issue 4: Protection of the environment, biodiversity and restoration of natural habitats	6.5.6
Core subject: Fair operating practices	6.6
Issue 1: Anti-corruption	6.6.3
Issue 2: Responsible political involvement	6.6.4
Issue 3: Fair competition	6.6.5
Issue 4: Promoting social responsibility in the value chain	6.6.6
Issue 5: Respect for property rights	6.6.7
Core subject: Consumer issues	6.7
Issue 1: Fair marketing, factual and unbiased information and fair contractual practices	6.7.3
Issue 2: Protecting consumers' health and safety	6.7.4
Issue 3: Sustainable consumption	6.7.5
Issue 4: Consumer service, support, and complaint and dispute resolution	6.7.6
Issue 5: Consumer data protection and privacy	6.7.7
Issue 6: Access to essential services	6.7.8
Issue 7: Education and awareness	6.7.9
Core subject: Community involvement and development	6.8
Issue 1: Community involvement	6.8.3
Issue 2: Education and culture	6.8.4
Issue 3: Employment creation and skills development	6.8.5
Issue 4: Technology development and access	6.8.6
Issue 5: Wealth and income creation	6.8.7
Issue 6: Health	6.8.8
Issue 7: Social investment	6.8.9

2.2 The Role of Social Responsible Communication

Especially innovative projects, e.g. in the automotive or electronics sector, have reached an enormous complexity. This complexity is not only reflected in the number of modules or parts to be integrated on a technical scale, but also in the complexity of distributed teams in the collaboration to reach the project goals. Often these teams are formed across different organisations or even across different states. This, however, creates new challenges for the technical developers, innovation managers, managers in general, quality managers etc. It can even be said that every individual in the workplace is affected by working across department's organisations or even foreign organizations. The same complexity equally applies ISO to research organisations and in public service organisations.

To achieve the best solution in such cooperation, the quality management principles might be used. An organization, complient with requirements for quality, has to [5]:

1. determine the processes needed for the quality management system and their application throughout the organization
2. determine the sequence and interaction of these processes,
3. determine criteria and methods needed to ensure that both the operation and control of these processes are effective,
4. ensure the availability of resources and information necessary to support the operation and monitoring of these processes,
5. monitor, measure where applicable and analyse these processes, and
6. implement actions necessary to achieve planned results and continual improvement of these processes.

These principles are even more important within diverse teams described above.

The role of communication in such environments is still underestimated even though everybody knows that it is important and tries to do his/her best. In development or quality management processes large groups of experts have to be coordinated: developers of the different parts of a product, experts with a diversity of expert knowledge, staff from different areas which might not be directly involved in the project team (e.g. marketing, sales, accounting) etc. To achieve the project goals all these people have to cooperate and communicate and various communication tools and methods might be used to support them. Often members of working-groups have to admit, that a meeting or a process did not achieve the best possible results.

The quality principles provide a firm ground for cooperation. A further level of efficiency in project teams can be achieved applying the principles of social responsibility such as:

- Promoting social responsibility for development processes
- Fair treatment, involving and informing of every process member and concerning community
- Sustainable resource-orientated team work on long term
- Social responsible conflict solving communication
- Confidential and fair behavior of leaders and members towards each other and their community
- Intercultural fair behaviour
- Respect for property rights, data protection and privacy

To apply a **Social Responsible Communication** it is important to consider:

- How can Social Responsible Communication help us to check up and improve our quality in processes?

- How can you use the whole potential and ideas of every member of the process by acting social responsible?
- How can you assure the information flow by using the entry of Social Responsible Communication?
- How could Social Responsible Communication help us to avoid or reduce interrupting factors of communication and information flow in:
- Leading/moderation (e.g. lack of involvement, lack of credit, demotivation, unfair treating of members)
- Matter of fact level (e.g. misunderstanding by using terms with different meanings in different domains, not given information)
- Emotional level (e.g. actual or old trouble in between single project partners or departments, lack of confidence)
- Acting level (e.g. disrespect of environment, property rights, intercultural differences)
- Which new communication tools could you use to achieve higher quality in processes?

Team members should be trained to analyze and afterwards implement the adequate tools and methods:

- Analyzing moderation / leading
- Analyzing group finding processes
- Analyzing group communication processes
- Moderational trainings
- Member trainings
- Training in social responsible behaviour and conflict management
- Live-training in the meeting
- Meeting-supervision

3 SOCIRES Project

SOCIRES - Social Responsibilities Training and Certification Schema for the Social Responsibility Manager (SRM) on ISO 26000 is a Leonardo da Vinci Lifelong Program project, started under Slovenian Institute for Standardization in Ljubljana, at 1.1.2011. It will end at 31.12.2012. Other partners involved are: proHUMAN Ltd. (Slovenia), Gustav-Siewerth-Akademie (Germany), SIBAC GmbH (Germany), University of Alcala (Spain), International I.S.C.N. (International Software Consulting Network) GesmbH (Austria), University of Thessaloniki (Greece) and Dennis Gabor College (Hungary).

SOCIRES will contribute to the raising of the awareness of all types of organizations about the importance and implementation of social responsibility. In addition it will contribute to the increase of SR competence level on labor market. A set of skills required to analyze and to define social influences and responsibilities of an organization as well as skills needed for definition of procedures which assure the optimal ratio between social responsibilities and economic interests of the organization will be defined within the project. The training and certification schema will be developed and accredited within European Certification and Qualification Association (ECQA, www.ecqa.org). Following will be developed/ implemented within the project:

- ***Training Material*** - modular: Units /Elements/Performance Criteria; supported with exercises, aimed for usage for VET trainings or distant learning;

- ***Pool of exam questions*** - unified, supported by ECQA electronic exam system;
- ***Learning Portal*** - providing training material and exercises;
- ***Trials and Exploitations of training materials and the training system*** will be performed for 260 participants.
- ***Train the Trainer*** (2 trainers per partner will be trained and certified for trainings after the end of project)
- ***Integration of training schema into ECQA Schema for Certification***. Two levels of certificates will be issued to trainees. “ECQA Participant Certificate” will be issued to those who will attend the trainings and “ECQA Certified Social Responsibility Manager” certificate will be issued to those who will successfully pass the exam.

Results of the project will be used for training and certification of ECQA Certified Social Responsibility Managers on the market. It is to be expected, that such specialists will be needed by:

- companies / organizations – to improve own Social Responsibility awareness and status
- governmental and other important decision making bodies - to monitor the Social Responsibility influence and status of certain organizations in the local environment
- non-governmental bodies and organizations aiming to improve Social Responsibility status in society – to manage own activities more efficiently

The SOCIRES training and certification schema will be unified all over Europe. Within each country the same procedures will be implemented and the same exam pool for verification of knowledge will be used. The knowledge required to become "ECQA Certified Social Responsibility Manager" will be publicly available through marketing and dissemination material. Recognition of qualifications will be higher, since ECQA works as a European association.

Social Responsibility Manager (SRM) will obtain knowledge on:

- core principles of social responsibilities (as defined in ISO 26000),
- other models and related knowledge (like Ethics in Business)
- identification of organization's Social Responsibility stakeholders,
- methods for analyzing the alignment of business processes to Social Responsibility requirements,
- definition of Social Responsibility processes in organisations and improvement of business processes,
- Social Responsibility related monitoring and relevant models for reviewing Social Responsibility processes
- continuous improvement of Social Responsibility within the company, etc.

Training participants will be invited from different types of organizations or directly from labor market. To enhance the interest on SR topic even further, the so called "job role committee (JRC)" will be established for the long term maintenance and updating of training material. JRC also approves new training organizations and Trainers. Consortium members will automatically become members of JRC as developers. New members of Job Role Committee will be invited to further improve and update the training schema as well. Broader community will be regularly informed about project results during the project duration.

4 Literature

[1] ISO 26000 – Social responsibility, ISO 26000 project overview, http://www.iso.org/iso/iso_catalogue/management_and_leadership_standards/social_responsibility/sr_iso26000_overview.htm, Accessed: 18.3.2011

[2] ISO 26000:2010(E), Guidance on social responsibility, 2010

[3] ISO, International Standards for Business, Government and Society www.iso.org.

[4] Entwurf Önorm ISO 26000:2009

[5] Quality management systems – Requirements (ISO 9001:2008)

5 Author CVs

Beatrice Bachmann M.A.

Since Beatrice Bachmann has studied communication sciences at the Ludwig Maximilian University in Munich, she has worked in several positions of social public institutions and associations dealing with communication and social work. In 1993 she started her own business working as trainer for parents with serious conflicts with their children. From 1998 till 2004 she worked as a communication-trainer of trainers in social work in and outside of Germany. In 2004 she founded CEF e.V., an organisation together with Dr. Volker Ovi Bachmann to help couples in conflict situations highly effective by new developed communication methods. In 2008 she founded SIBAC gmbH together with Dr. Volker Ovi Bachmann and brings her social expert knowledge about communication into industry. Since 2000 she published many articles in relevant professional journals and in the Internet (www.online-familienhandbuch.de/bachmann) as well as 9 books dealing with conflict management and education. Furthermore she developed new approaches to strategies for resolving conflicts in quality management.

Dr. Romana Vajde Horvat

Dr. Romana Vajde Horvat has over 20 years experience in education and industry in the field of informatics and entrepreneurship. She has worked for 19 years as a researcher, teaching assistant and assistant professor at University of Maribor in Slovenia where she lectured topics related to process improvement in information technology, communication, project management and quality management. She has cooperated within numerous local projects with industry as well as in 8 European projects so far. She has various management experiences, since she was a manager of Technology transfer center for 5 years, she is a member of management board of Slovenian Institute for Standardization and she is manager of proHUMAN for three years. She is a vice president of ECQA responsible for processes and she is a job role committee member for several professions within ECQA. She is a project manager for SOCIRES project.

Understanding the Relation of SPI and SR: A proposed Mapping of the SPI Manifesto to ISO 26000:2010

Sonja Koinig[1], Miguel-Angel Sicilia[2], Richard Messnarz[1], Elena García-Barriocanal[2], Miguel Garre-Rubio[2], Kerstin Siakas[3], Adrienne Clarke[1]

[1] ISCN GesmbH, Austria / ISCN Ltd, Ireland
{skoinig, rmess, aclarke}@iscn.com
[2] University of Alcalá, Computer Science Department, Spain
{elena.garciab, miguel.garre, msicilia}@uah.es
[3] Alexander Technological Educational Institution of Thessaloniki, Department of Informatics, Greece
{siaka}@it.teithe.gr

Abstract

Software Process Improvement (SPI) activities aim at driving change in IT development towards increased quality and responsiveness levels. The SPI Manifesto describes the key values and principles for a successful implementation of SPI. About two thirds of its principles relate to human, social and organizational aspects and one third to technical aspects. This raises the question on how these aspects interact with the view of business aims which are broadly known as “social responsibility”. The recent ISO 26000:2010 standard is aimed at providing guidance on social responsibility (SR), describing the subjects and issues an organization has to consider when implementing social responsibility. In this paper we describe how the values and principles of the SPI Manifesto interact with the social responsibility issues described in the ISO 26000. The result of this mapping sets the basis for an implementation guidance, and illustrates how social and people factors can largely interact with the success of SPI.

Keywords

Corporate social responsibility, SPI (System, Services, Software Process Improvement), SPI Manifesto, ISO 26000

1 Introduction

In September 2009 a group of experts in Software Process Improvement (SPI) from all over the world gathered in connection with the EuroSPI Conference for a workshop at University of Alcalá in Spain. The mission of EuroSPI[1] is to develop an experience and knowledge exchange platform for Europe where SPI practices can be discussed and exchanged and knowledge can be gathered and shared.

At the workshop 15 experts presented their 'wisdom' grounded in many years of process improvement experience. Based on the presentations, 30 workshop participants brainstormed core values and principles specifically for process improvement. Later, affinity analysis and group thinking resulted in the SPI Manifesto that was published in early 2010. The manifesto includes ten principles organized around three values as key success factors for successful innovation and improvement in organizations (Pries-Heje & Johansen 2010). Figure 1 and 2 show excerpts of the SPI manifesto with the values and principles, respectively.

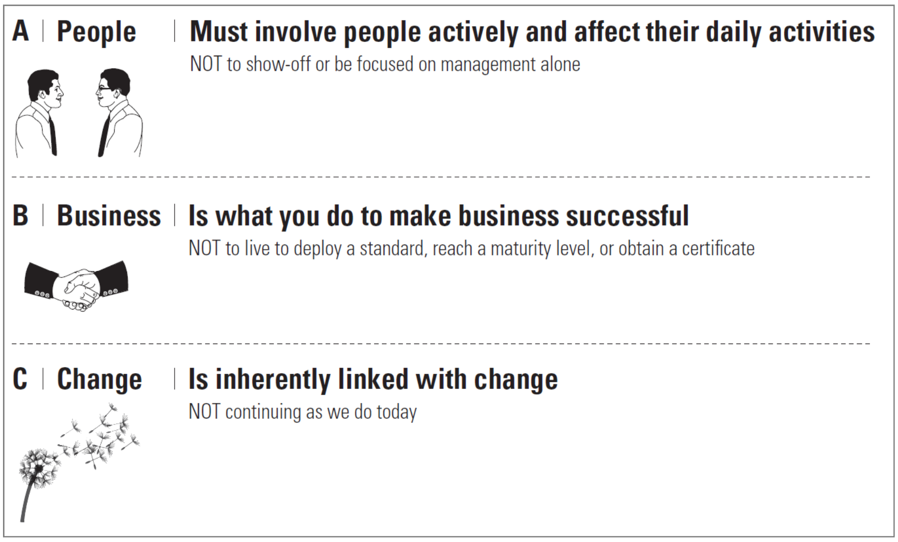


Figure 1: SPI Manifesto: three values

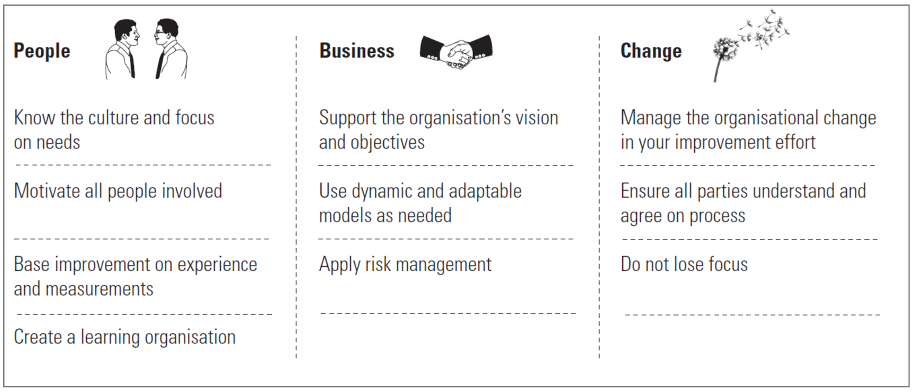


Figure 2: SPI Manifesto: ten principles

As can be appreciated in Figures 1 and 2, some of the principles considered have to do with people, cultural and social issues. For example, the consideration of cultural issues is clearly linked to fair labour practices that are typically in focus of SR management. In other direction, the SPI Manifesto is based on the alignment of processes with organizational vision and objectives. In organizations that have SR issues as an integral part of their mission and objectives, it becomes clear that SR issues should be considered as targets for process improvement, but this obvious link has still not being explored explicitly. From an empirical perspective, it has been found elsewhere that continuous improvement activities can be a key source of environmental improvement (Hanna, Newman and

[1] www.eurospi.net

Johnson, 2000). Here we attempt to explore the interactions of improvement principles and social responsibility aspects in general, using two established practice guides for a conceptual matching.

The rest of this paper is structured as follows. Section 2 gives background on social responsibility and presents some evidence on the concerns of IT companies with SR issues. Then, Section 3 described a mapping of the SPI manifesto to the ISO 26000 standard. Finally, conclusions and outlook are provided in Section 4.

2 Background on Social Responsibility in IT companies

The field of Corporate Social Responsibility (CSR) has grown significantly in the last decade, both in importance and in diffusion. Proof of this is the publication of the ISO 26000 guidance, and the European project called “SOCIRES”. The project ‘Social Responsibility Training and Certification Schema (SOCIRES)’ started in January 2011 within the framework of the European Certification and Qualification Association[2] and with funding from the EU Lifelong Learning Programme. The project aims at developing a training and certification scheme for CSR managers. The ISO 26000:2010 standard was launched in November 2010 and aims to provide guidance to both business and public sector organizations regarding social responsibility (SR). The new standard contains consensual definitions on what SR means and the SR issues that organizations need to address. It is expected to add value to existing initiatives regarding social responsibility by providing harmonized and globally relevant guidance based on international consensus among expert representatives of main stakeholder groups. In this section, we present an analysis on how ISO 26000 issues are considered in the Web pages of a selection of large IT companies. This analysis reveals significant differences in how SR concerns are communicated externally, but also points out that SR is currently being considered as an important objective especially in some areas.

2.1 Analysis of consumer electronics companies

Organizations repeatedly encounter demands from different stakeholders to allocate resources for CSR and to engage in CSR actions. Simultaneously growing expectations on organizations regarding CSR action and increasing demands for transparency push organizations to measure, report, and continuously improve their environmental, social, and economic performance (Tsoutsoura, 2004), which entails also an appropriate external communication. In general, the way SR is communicated varies across countries and companies. For example, Hartman, Rubin and Dhanda (2007) found differences in SR reports between US and European companies, including different uses of terminology and different justification focuses.

As Web pages are nowadays a mainstream means of communication, some studies have departed from corporate Webs to gain insight on communication practices. Concretely, Esrock and Leichty (1998) studied how large corporate entities made use of the Web to present them as socially responsible. They found that most sites presented items addressing community involvement, environmental concerns, and education. Maignan and Ralston (2002) studied corporate Web pages in France, the Netherlands, the U.K., and the U.S., concretely, the nature of CSR principles, processes, and stakeholder issues. The results showed that businesses in the four countries do not display the same eagerness to appear as socially responsible and employ diverse means to convey social responsibility images.

Pollach (2003) investigated how six companies communicate their ethical stance on their Web sites. The author used linguistic analysis of the company Web pages based on a functional approach of discourse analysis focusing on the ideational, the interpersonal, and the textual function of discourse. The findings propose that the six companies belong to three different ethical paradigms regarding typicality and systematic variety. Although the investigated company web sites adopted different approaches to corporate ethics, their communicative strategies turned out to be quite similar regarding content, self-reference, audience address, persuasive appeals, and message organization. Other studies have addressed the rhetoric of the justification of CSR in corporate Web pages, notably Coupland (2005) in a study monitoring four corporate Web pages found a number of legitimating

[2] http://www.ecqa.org/index.php?id=37

patterns by analysing linguistic expression. Here we present an analysis on how a set of IT companies communicate the different elements of the ISO 26000 standard through their Web pages.

2.2 Analyzing Web pages

The sample selected here are the companies participating in the "Guide to Greener Electronics" (GGE) elaborated by Greenpeace[3]. Currently the guide ranks the 18 top manufacturers of personal computers, mobile phones, TVs and game consoles in the extent to which they comply with the following aspects:

1. Clean up their products by eliminating hazardous substances.
2. Take back and recycle their products responsibly once they become obsolete.
3. Reduce the climate impacts of their operations and products.

To illustrate how leading and successful organizations publicly support the social responsibility on a corporate level, a study of communication practices in corporate web pages was conducted. The procedure for doing that was to access the web pages of each of the companies, then looking for the links in that page pointing to SR or environmental issues. Once the pages are identified, they were read by two SR experts looking for each of the SR issues identified in the ISO 26000.

Table 1 provides the details of all the core issues in the ISO 26000 and which of them appear in the Web pages of each of the companies. The analysis of Table 1 shows that some core issues are less represented in the Web pages analyzed. Notably, organizational governance issues are not made explicit, with the exception of Toshiba that mentions its adherence to the ONU Global Compact policy initiative. Other core issues as "Due diligence" were not identified explicitly. This can be attributed to these issues being more related to generic intention to behave rather than concrete measurable issues (as may be the reduction of carbon footprint). In contrast, there are issues that are more commonly covered. For example, "employment creation and skills development" is covered by all but one of the pages. In the environmental area, "climate change" is covered in 15 out of 18 of the pages, sustainable resource use in 14, pollution in 13 and protection of the environment only in 9. Of the companies covering fewer topics in the environment area, only Acer is scoring above 4 in the last edition of the GGE.

In what concerns environmental issues there is no clear pattern on how the topics are covered in the Web pages. For example, Nintendo is covering all the important topics, even though they rank poorly in the GGE. In general, no direct strong relation between Web page coverage and performance can be found. Another interesting finding is that there is no relation between the coverage of SR topics in general and performance in green computing as reflected in the GGE and the coverage of other core areas. For example, Nokia is covering poorly the area of "fair operating practices".

[3] A complete description of the criteria used to elaborate the Guide can be found at Greenpeace pages: http://www.greenpeace.org

	Acer	Apple	Dell	Fujitsu-Siem	HP	Lenovo	LGE	Microsoft	Motorola	Nintendo	Nokia	Panasonic	Philips	Samsung	Sharp	Sony	Sony-Ericc.	Toshiba
6.2 Organizational governance																		⊙
6.3 Human rights		⊙																
Due diligence																		
Human rights risk situations				⊙	⊙	⊙		⊙	⊙	⊙	⊙	⊙	⊙	⊙	⊙	⊙	⊙	⊙
Avoidance of complicity				⊙	⊙													
Resolving grievances																		
Discrimination and vulnerable groups	⊙	⊙	⊙	⊙	⊙		⊙	⊙	⊙	⊙	⊙	⊙	⊙	⊙	⊙	⊙		
Civil and political rights			⊙	⊙	⊙													
Economic, social and cultural rights			⊙	⊙														
Fundamental principles and rights at work	⊙	⊙	⊙	⊙	⊙		⊙	⊙	⊙	⊙	⊙	⊙	⊙		⊙	⊙		
6.4 Labour practices			⊙		⊙													
Employment and employment relationships	⊙	⊙	⊙	⊙	⊙		⊙	⊙	⊙	⊙	⊙	⊙	⊙	⊙	⊙	⊙		
Conditions of work and social protection	⊙		⊙		⊙	⊙	⊙	⊙	⊙	⊙	⊙	⊙			⊙	⊙		
Social dialogue				⊙					⊙			⊙			⊙			
Health and safety at work		⊙		⊙	⊙	⊙		⊙	⊙	⊙	⊙	⊙	⊙	⊙	⊙	⊙		⊙
Human development and training in the workplace	⊙		⊙	⊙	⊙		⊙	⊙	⊙	⊙	⊙	⊙		⊙	⊙	⊙		⊙
6.5 The environment																		
Prevention of pollution	⊙	⊙	⊙	⊙	⊙	⊙		⊙	⊙	⊙	⊙	⊙		⊙	⊙	⊙	⊙	⊙
Sustainable resource use		⊙	⊙	⊙	⊙	⊙		⊙	⊙	⊙	⊙	⊙	⊙	⊙	⊙	⊙	⊙	
Climate change mitigation and adaptation		⊙	⊙	⊙	⊙	⊙	⊙	⊙	⊙	⊙	⊙	⊙	⊙	⊙	⊙	⊙	⊙	⊙
Protection of the environment, …		⊙	⊙	⊙		⊙			⊙	⊙	⊙	⊙		⊙	⊙	⊙		
6.6 Fair operating practices																		
Anti-corruption				⊙	⊙	⊙		⊙	⊙					⊙	⊙	⊙		⊙
Responsible political involvement								⊙	⊙									⊙
Fair competition									⊙					⊙				⊙
Promoting social responsibility in the value chain		⊙	⊙	⊙	⊙	⊙		⊙	⊙	⊙	⊙	⊙	⊙	⊙	⊙	⊙	⊙	
Respect for property rights				⊙		⊙		⊙		⊙				⊙				⊙
6.7 Consumer issues																		
Fair marketing, …					⊙							⊙						
Protecting consumers' health and safety						⊙				⊙	⊙	⊙		⊙	⊙	⊙	⊙	⊙
Sustainable consumption		⊙	⊙	⊙	⊙		⊙	⊙	⊙	⊙	⊙		⊙		⊙	⊙	⊙	⊙
Consumer service, support, …				⊙						⊙				⊙	⊙	⊙		
Consumer data protection and privacy				⊙	⊙				⊙	⊙		⊙		⊙	⊙	⊙		⊙
Access to essential services			⊙															

Education and awareness										⊙	⊙	⊙		⊙		⊙	⊙	
6.8 Community involvement and development																		
Community involvement			⊙	⊙	⊙	⊙		⊙	⊙	⊙	⊙			⊙	⊙	⊙	⊙	⊙
Education and culture			⊙	⊙	⊙	⊙		⊙	⊙		⊙	⊙	⊙	⊙	⊙	⊙	⊙	⊙
Employment creation and skills development		⊙	⊙	⊙	⊙	⊙	⊙	⊙	⊙	⊙	⊙	⊙	⊙	⊙	⊙	⊙	⊙	⊙
Technology development and access			⊙	⊙	⊙			⊙	⊙	⊙	⊙		⊙			⊙	⊙	
Wealth and income creation					⊙			⊙	⊙		⊙			⊙	⊙			
Health			⊙		⊙					⊙								
Social investment			⊙	⊙		⊙			⊙	⊙	⊙	⊙		⊙	⊙	⊙	⊙	⊙

Table 1: Summary of explicit occurrences of topics related to each ISO 26000 issue

3 Mapping of the SPI Manifesto to SR issues of the ISO 26000 Standard

The following cause illustrates the mapping of the values and principles of the SPI Manifesto on the different issues and chapters of the ISO 26000 Standard. The mapping method is based on the following 4 steps and the results are illustrated in below mentioned tables:

1. Proposing main assumptions why social responsibility potentially supports specific principles and values of the SPI Manifesto;
2. Searching for core subjects and issues of the ISO 26000 Standard (definite SR implementation demands), which support the previously defined main assumptions;[4]
3. Searching for general ISO 26000 chapters supporting these main assumptions, like “Recognizing of social responsibility” or “Communication on social responsibility”;
4. Summarizing the coverage of ISO 26000 chapters and how the standard needs to be interpreted to become a valuable source for SPI driven industry in Europe.

Table 2 illustrates main assumptions regarding the connection of SR issues in general and the values and principles of the SPI Manifesto. For the analysis the following question had priority: How do SR issues basically support the principles and values of the SPI Manifesto?

Table 3 depicts the mapping of the SPI Manifesto principles of the value “SPI must involve people actively and affect their daily activities” onto the core subjects and issues of the ISO 26000 Standard. The results of this mapping are:

The ISO 26000 core subjects “Organizational governance”, “Human rights” and “Labour practices” (addressed by core issues in Table 3) and the clause “Recognizing social responsibility and engaging stakeholders” of the ISO standard are related to these SPI Manifesto principles.

With reference to the chapter “Recognizing social responsibility and engaging stakeholders” (ISO 26000: Cause 5) the first 4 principles of the SPI Manifesto shall help to promote social responsibility in an organization. An aspect, which is not included in the SPI Manifesto, is that these 4 principles only concern the relationship between organization and stakeholders[5]. The relationship between organization and society and stakeholder and society are not implied.

[4] An overview of the ISO 26000 core subjects and issues can be found in table 1.

[5] Stakeholders are „*individual or group that has an interest in any decision or activity of an organization*“ (ISO 26000 Standard 2010: 4), like employees, customers, consumers, suppliers etc.

Values of SPI Manifesto	Principles of SPI Manifesto	Discussion of relationship between SR issues and values/principles of SPI Manifesto
Value A "People": SPI must involve people actively and affect their daily activities	Principle 1: Know the culture and focus on requirements	**Main Assumption:** If the organisational culture is not taken into consideration and well known when implementing or developing SPI, it is nearly impossible to support social principles for individuals of an organization or rather to support social responsibility in an organization.
	Principle 2: Motivate all people involved	**Main Assumption:** If there is no open social space for all people involved in an organization, they will not be motivated to contribute and acquire their knowledge, and if there is no possibility for knowledge sharing and extension, the motivation will also decrease.
	Principle 3: Base improvement on experience and measurements	**Main Assumption:** SPI is not only a task for external experts (e.g. consultants) and the management, this process should also include experiences and requirements from all people involved in an organization (employees, customers, consumers etc.).
	Principle 4: Create a learning organization	**Main Assumption:** A social space should allow organization´s people to continuously learn and adapt themselves to altered structures and to new situations of an organization on the market. Continuous change of the organization means consequently continuous change of the work environment for involved people. All organization´s people should be part of the problem solving, experts and the management should not be the only involved parties.
Value B "Business": SPI is what you do to make business successful	Principle 5: Support the organisations vision and objectives	**Main Assumption:** Individuals of an organization shall fully understand and support organizations vision and objectives. A strategy shall be agreed with the management, which supports both, business success and social rights for people/social responsibility in an organization in the long term.
	Principle 6: Use dynamic and adaptable models as needed	
	Principle 7: Apply risk management	
Value C "Change": SPI is inherently linked with change	Principle 8: Manage the organisational change in your improvement effort	**Main Assumption:** Changes shall be planned and managed in a way that all objectives and social impacts of an organization are always visible (e.g. through a web page) and verifiable.
	Principle 9: Assure all parties understand and agree on process	**Main Assumption:** The meaning of the structural changes have to be communicated, approved by all people involved and shall consider social impacts of an organization.
	Principle 10: Do not lose focus	**Main Assumption:** Once a change agreed, both the involved people and the management have to consequently stick to the agreements.

Table 2: SPI Manifesto – SR principle Mapping Assumptions

Values of SPI Manifesto	Principles of SPI Manifesto	ISO 26000 Core subjects and issues (Table 1)
Value A "People": SPI must involve people actively and affect their daily activities	Principle 1: Know the culture and focus on requirements	Assignment to: Core subject: **Organizational Governance** Principle 1 - 4 of the SPI Manifesto can be assigned to the core subject "organizational governance" because all these principles affect directly or indirectly the decision making within an organization, support an organization to take responsbility for the impacts of its decision and activities and promote to integrate SR throughout the organization. The principles are instructions to how an organization shall act and a possiblity of increasing the organization´s ability to behave in a socially responsible manner. Core subject: **Human rights**, Issue: **Economic, social and cultural rights** The different economic, social and culturual rights of people involved in an organizations have to be respected (important for personal and organizational development) and are crucial parts of the organizational culture.
	Principle 2: Motivate all people involved	Assignment to: Core subject: **Organizational Governance** (see Principle 1) and Core subject: **Labour practices**, Issue: **Human development and training in the workplace** People can be motivated by expanding human capabilities and functioning. This shall include the opportunities for being creative and productive, for enjoying self-respect and a sense of belonging to a community or to the organization. Different trainings and courses can faciliate the motivation, the readniss to learn and extend the people´s knowledge which results in new (innovative) ideas. This faciliates a learning organization and correlates strong with the principle 3 (base improvement and experience and measurements) and principle 4 (create a learning organization)
	Principle 3: Base improvement on experience and measurements	Assignment to: Core subject: **Organizational Governance** (see Principle 1) and Core subject: Labour practices, Issue: **Human development and training in the workplace** (see Principle 2)
	Principle 4: Create a learning organization	Assignment to: Core subject: **Organizational Governance** (see Principle 1) and Core subject: Labour practices, Issue: **Human development and training in the workplace** (see Principle 2)

Table 3: Mapping of the SPI Manifesto value "SPI must involve people actively and affect their daily activities" onto the ISO 26000 Standard

Table 4 illustrates the mapping of core subjects and issues of SR (according to ISO 26000) onto the SPI Manifesto principles of the value "SPI is what you do to make business successful". The ISO 26000 subjects "Organizational governance" and "Human rights (addressed by core issues in Table 4) and the clause "Practices for integrating SR throughout an organization" cohere with the 3 principles of the SPI Manifesto value. With respect to implementing SR principles in an organization as a strategy (ISO 26000: Cause 7.4) these principles of the SPI Manifesto support the integration of SR principles in an organization. The three-step model “unfreeze - move - freeze”, described in principle 8 in the SPI Manifesto, is particularly suited to really change the behavior of people involved in an organization (regarding social responsibility etc.).

Values of SPI Manifesto	Principles of SPI Manifesto	ISO 26000 Core subjects and issues (Table 1)
Value B "Business": SPI is what you do to make business successful	Principle 5: Support the organizations vision and objectives	Assignment to: Core subject: **Organizational Governance** Organizational governance is the system through which an organization implements decisions and pursues objectives. The ISO standard requires a balanced approach between reaching organizational objectives as social principles as well. Principle 5 - 7 of the SPI Manifesto can be assigned to the core subject "organizational governance" because all these principles affect directly or indirectly the decision making within an organization, support an organization to take responsbility for the impacts of its decision and activities and promote to integrate SR throughout the organization. The principles are instructions to how an organization shall act and a possiblity of increasing the organization´s ability to behave in a socially responsible manner.
	Principle 6: Use dynamic and adaptable models as needed	Assignment to: Core subject: **Organizational Governance** (see Principle 5)
	Principle 7: Apply risk management	Assignment to: Core subject: **Organizational Governance** (see Principle 5) and Core subject: Human rights, Issue: **Human rights risk situations** This kind of risk management is only a small part of the entire risk management of an organization but an organization shall also take care of human rights risk situations. Examples are - a culture of corruption or - complex value chains which involve work performed on an informal basis without legal protection etc.

Table 4: Mapping of the SPI Manifesto value “SPI is what you do to make business successful” onto the ISO 26000 Standard

Table 5 depicts the mapping of core subjects of SR (according to ISO 26000) onto the SPI Manifesto principles of the value “SPI is inherently linked with change”.

The core subject “Organizational governance” of the ISO Standard (addressed by core issues in Table 5) and the clause “Communication on social responsibility are related to these 3 principles. With reference to this ISO 26000 clause (ISO 26000: Cause 7.5) the principles of the SPI Manifesto value “SPI is what you do to make business successful” support the communication of social responsibility and therefore improvement in an organization.

Values of SPI Manifesto	Principles of SPI Manifesto	ISO 26000 Core subjects and issues (Table 1)
Value C "Change": SPI is inherently linked with change	Principle 8: Manage the organisational change in your improvement effort	Assignment to: Core subject: **Organizational Governance** Organizational governance is the system through which an organization implements decisions and pursues objectives. The ISO standard requires a balanced approach between reaching organizational objectives as social principles as well. Principle 8 - 10 of the SPI Manifesto can be assigned to the core subject "organizational governance" because all these principles affect directly or indirectly the decision making within an organization, support an organization to take responsbility for the impacts of its decision and activities and promote to integrate SR throughout the organization. The principles are instructions to how an organization shall act and a possiblity of increasing the organization´s ability to behave in a socially responsible manner.
	Principle 9: Assure all parties understand and agree on process	Assignment to: Core subject: **Organizational Governance** Governance defines processes, structures and objectives for the entire organization. Governance implements control mechanisms for meeting processes and goals. SPI (Process Improvement!) is exactly focussing on these processes and objectives and ISO 26000 mentions here actions such as - develop strategies together, - use incentives to win staff to support processes and improvement, and - balance the requirements of the organisation and the stakeholders (the staff is also part of the stakeholders).
	Principle 10: Do not lose focus	Core subject: **Organizational Governance** Governance in ISO 26000 especially mentions the activity "demonstrate leadership and commitment" to the governance and social aspects, which is directly related to the SPI statement "do not lose focus".

Table 5: Mapping the SPI Manifesto Value "SPI is inherently linked with change" onto the ISO 26000 Standard

4 Conclusion and outlook

To successfully implement SR in organizations, a synergy between organizational strategy and supporting the social rights of individuals at the same time is required, thus supporting SR issues as an integral part of an organization's mission and objectives. SR issues are actually considered important concerns in the objectives of IT organizations, even though there are significant differences in how they address different SR topics, as has been revealed by the study reported in this paper. The relation of SPI to SR concerns are apparent in some areas as the consideration of cultural issues in change processes, but a detailed examination of the interaction between SPI and SR is required for a full understanding of these. This paper has reported a first mapping between SPI and SR based on the SPI Manifesto and the recent ISO 26000 standard. Interaction in both directions has been found, which suggests that SR concerns should be considered an integral part of SPI.

Future work should continue in the direction of exploring the relationships of process improvement and SR implementation, both from a conceptual and an empirical perspective.

Acknowledgements

The work presented in this paper has been partially supported by the activities in the project "Social Responsibilities Training and Certification Schema" (SOCIRES)[6], number 510098-LLP-1-2010-1-SI-LEONARDO-LMP, funded by the EC LLP under the Leonardo da Vinci programme with support from the European Commission. This publication reflects only the views of the authors, and the Commission cannot be held responsible for any use, which may be made of the information contained therein.

References

1. Carroll, A. B. (1979). A Three-Dimensional Model of Corporate Performance. *Academy of Management Review*, 4(4), pp. 497-505
2. Coupland, C.: Corporate Social Responsibility as Argument on the Web. Journal of Business Ethics (2005) 62: 355–366
3. Crooks B., Harvey. B., Limbuwala Z., Newcombe L., Ross M., Staples G., Bayley, P. (2009). Raising Awareness of Green IT- the BCS way British Computer Society, *International conference on Software Quality Management (SQM, 2009)*, 6th - 8th of April, 2009, Southampton Solent University, Hampshire, UK,
4. Esrock, S.L., Leichty, G.B. (1998) Social responsibility and corporate web pages: Self-presentation or agenda-setting?. *Public Relations Review*, 24(3), pp. 305-319
5. Hanna, M.D., Newman, W.R. and Johnson, P. (2000) Linking operational and environmental improvement through employee involvement. International Journal of Operations & Production Management 20(2), pp. 148-165.
6. Hartman, L., Rubin, R. and Dhanda, K.: The Communication of Corporate Social Responsibility: United States and European Union Multinational Corporations. Journal of Business Ethics 74(4), 373-389, 2007
7. ISO (2010): International Standard ISO 26000. Guidance on social responsibility. First edition 2010-11-01. Reference number ISO 26000:2010(E)
8. McWilliams, A. and Siegel, D. 2001. Corporate Social Responsibility: A Theory of the Firm Perspective. *Academy of Management Review*, 26(1), pp. 117-127
9. Maignan, I., Ralston, D. (2002) Corporate Social Responsibility in Europe and the U.S.: Insights from Businesses self-presentations. *Journal of International Business Studies*, 33(3), pp. 497-514
10. Oketch, M.J. (2004) The corporate stake in social cohesion. *Peabody Journal of Education*, 8(4), pp. 30-52
11. Orlitzky M., Schmidt, F. L., Rynes S. L. (2003). Corporate Social and Financial Performance: A Meta-analysis, *Organization Studies*, 24(3), pp. 403–441
12. Pollach I., (2003), Communicating Corporate Ethics on the World Wide Web, A Discourse Analysis of Selected Company Web Sites, *Business Society,* 42 (2), pp. 277-287
13. Pries-Heje, J., Johansen J. (2010): SPI Manifesto. Version A.1.2.2010. Available at: http://www.iscn.com/Images/SPI_Manifesto_A.1.2.2010.pdf
14. Rushton K., 2002. Business Ethics: a Sustainable Approach, *Business Ethics: A European Review,* 11 (2), pp. 137–139

[6] http://www.adam-europe.eu/adam/project/view.htm?prj=6730

From Specific Project Challenges to Improvement of Mjølner's Software Process

Jens Bæk Jørgensen, Mikkel Yde Kjær, Inge Mølgaard, Søren Snehøj Nielsen, Niels Mark Rubin
Mjølner Informatics A/S
Finlandsgade 10
DK-8200 Aarhus N
{jbj, myk, imo, ssn, nmr}@mjolner.dk

Abstract

In this paper, we present a real-life software development project; analysis, design and implementation of a new home control in a joint effort between a large industrial company and the software company Mjølner Informatics A/S. We discuss the main challenges the project have faced and are facing, we reflect upon how to address the challenges and we consider how the specific project can provide input to Mjølner's general SPI program. The main challenges include getting in line with many diverse stakeholders, different approaches to the design process, dealing with requirements, and accommodating agile software development within the constraints imposed by traditional (non-software) industrial practice.

Keywords

Software process improvement, case study, user experience, stakeholder management, requirements, software development, agile process

1 Introduction

This paper is about a project that develops a new "home control" – a remote control device that has a touch user interface and is able to control several devices in a home. The project is carried out in a joint effort between a large industrial company ("the customer" in the rest of the paper) and Mjølner Informatics A/S (for commercial confidentiality reasons, the identity of the industrial company cannot be published).

Mjølner Informatics A/S (Mjølner) is a Danish software company, founded in 1988. Mjølner develops custom-made software for Danish and international customers, both in the private and the public sector. Mjølner has expertise in development of a broad range of system types. Mjølner has a User Experience Centre (UX Centre) dedicated to ensure that all solutions have a high degree of user-friendliness. One role of the User Experience Centre is to ensure that the users' needs are supported by the systems that are developed. Mjølner's trademark is to combine a high professional competence with effective and flexible project management and cooperation with the customers. The emphasis is on knowledge sharing, innovation and partnership.

In early 2010, the customer contracted Mjølner to develop the user interface of the new home control. At the time of writing this paper, the project has lasted about a year, and is planned to continue until the autumn of 2011.

This paper will present the main challenges encountered until now, and describe how these challenges have been addressed or have been decided to be addressed. The primary purpose is to put spotlight on areas where this particular project has inspired improvements of Mjølner's software development process [Kjær & Jørgensen 2010], based on the special needs in this project. In the end, future projects must benefit from the experiences gained in the particular project.

A short introduction to the project will be presented, and then four examples of main challenges where an adjustment of the process was needed will be given. Subsequently a list of possible improvements to Mjølner's software development process, derived from the findings in the paper, is presented.

2 The Project and its Main Challenges

The goal of the project is to design and implement a new version of a home control. The customer has developed a similar home control in-house a number of years ago, and this is in wide-spread use world-wide. Many users, however, find the current home control too difficult to use. The customer believes that it has the right functionality, but it is too difficult to find the functionality. The purpose of the project is to analyze the needs of the users, design a new graphical user interface that is in aligned with the users' needs, and implement the new graphical user interface.

The project started in February 2010 at a kick-off workshop with approximately 20 stakeholders. In March and April 2010 Mjølner's User Experience specialists conducted field studies, user tests, focus groups, and other user research activities in two of the main markets of the customer.

All the research information gathered in the user research activities was scrutinized, discussed with the customer and consolidated. Among the most important results from this process was the definition of personas [Nielsen 2004] that represent key user profiles, whose wishes should have proper priority in the design of the new user interface. Along with the personas a list of the main user scenarios for the home control was created in order to focus the following prototypes on the main usage and user needs. The analysis work was guided by Mjølner's software development process, which contains a user experience "tool box" that was applied for user research.

In August 2010, the preliminary results were presented to the top management of the customer. The top management wanted to approve fundamental decisions about the home control, which is of strategic importance. At this meeting fundamental decisions were taken, involving design and distribution of the home control, and it was decided that the top management were to be involved in evaluating project milestones.

Afterwards Mjølner started to write a requirement specification. Though the new home control should have the same functionality as the existing control, no specifications existed and it was experienced that different stakeholders had different requirements. The requirements specification work obviously benefitted from the user research that had already been carried out. This work was expanded and supplemented. In particular, the business goals were clarified; the business goals are statements of the customer's ultimate purpose with the project, often in the form of increased revenue, higher attractiveness and bigger market share, which was also the case here. The business goals were coupled with the scenarios – many of which already existed – that describe how the business goals can be reached. Use cases decomposed the scenarios and it ended up with verifiable requirements, as shown in Figure 1.

Figure 1 - Elicitation of requirements.

Respecting the fact that requirements should be elicited and categorized on four levels as in this approach is an essential ingredient in Mjølner's software development process. It is a well-known approach similar to, e.g., Lauesen's categorization of requirements in goal-level, domain-level, product-level, and design-level requirements [Lauesen 2002].

In the autumn, two competing interactive and graphical prototypes were developed and tested with end-users and evaluated by the stakeholders including the top management. In December 2010 key decisions about the hardware, the purchase process, and graphical user interface were taken, which meant that the design process could now focus on more detailed design decisions. In January 2011 the software development started in close cooperation with a team at the customer.

In this paper the four main challenges that have been faced in the project, will be elaborated:

- A large and diverse set of stakeholders
- The customer was used to a functionality-based process
- Many requirements are implicit and not well-agreed upon
- A high degree of parallelization

Obviously, it is not the first time that we meet these and similar challenges, and, in general, we do have methods to deal with them. However, in the specific project we are discussing in this paper, we have seen these challenges materializing more strongly than often, and this is what have inspired us to pursue even better methods to address such challenges more effectively.

3 A Large and Diverse set of Stakeholders

As the project progressed Mjølner gradually realized that the participants from the kick-off meeting were not the only stakeholders and decision makers in the project. Some of the other stakeholders include communication and brand managers, hardware specialists, various middle managers, translation and instruction manual representatives, and not the least top management.

The large and diverse set of stakeholders often had conflicting perspectives on the design and functionality requirements. E.g. at a certain point in time it was learned that the sales and marketing representatives felt that their wishes did not get sufficient attention since the home control appeared too technical and not sufficiently attractive to the market. Also the brand and communications managers, who ensure the brand consistency, were involved very late in the process after several design iterations. The involvement of the top management obviously is a prime example of key stakeholders that make decisions that changes the foundation of the project. In general, the involvement of different stakeholders have changed and expanded the project a number of times. This of course meant changing a lot of the produced work products.

There are several reasons why relevant stakeholders were not always involved sufficiently early. One of them was because the product owners of the company had more focus on the entire functionality requirements than the overall appearance and brand factor. A second reason was a geographical distance and professional difference between the project manager of the customer and the product owners of the customer.

A software company must be aware not only to focus too much on the technical product at hand, how to develop it, and how it fits the users' needs. This is the case because important information might be missed from important persons at the customer. It is especially true when the customer is a huge organization with a strategically important product with many stakeholders. In this case a close cooperation with the entire stakeholder gallery is of utmost importance.

To get closer cooperation between the stakeholders and the development team at Mjølner, a number of full day workshops were arranged. At these workshops relevant stakeholders were gathered and encouraged to express their viewpoints and ultimately make joint decisions. These frequent, physical meetings improved the stakeholder management.

In general terms, the observation is that in a project of a huge organizational complexity it is important on a continuous basis to take all important stakeholders into account and ensure cooperation between these stakeholders and the project team. Mjølner's software development process – including project management process – does offer support for these activities, but will benefit from even better tools and methods to support the identification and involvement of many diverse stakeholders.

4 The Customer was Used to a Functionality-based Process

When Mjølner's UX-team began the design process, the purpose was to meet all the functional requirements of the home control and make them fit into a small touch interface in a user friendly design. The users were a diverse group with many different products, needs and readiness towards technology. The analysis showed that the vast majority of the users only utilized a small part of the current home control since the user interface was too complicated and they did not know the full potential of the home control.

After several workshops and different design proposals it became evident that the product owners from the customer and the UX-team at Mjølner had a very different approach to the optimal design process. The two different views can be classified as:

1. The functional-based approach: Focus the design concept on the total set of detailed requirements.
2. The UX-driven approach: Focus the design concept and the interfaces to the main usage scenarios.

The first approach suggests that the user interface design is based on the entire set of functional requirements, meaning that the interface is created with the full suite of features in mind and prioritizing the advanced users as much as the main users.

In contrast, the UX-driven approach suggests that a simple user interface can only be the result of focusing on the mainstream usage [Colborne 2011]. This means that each element in an interface must be evaluated against the main use scenarios and possibly discarded when it is not a need of the main user. Creating a user friendly interface is about considering each dialogue so it does not contain elements that are rarely used [Nielsen 1994]. This approach is based on terms such as user friendliness and aesthetic experiences and it recognizes that the user's emotions – both immediate and obtained through use - are important factors of the user's dialogue with a product [Hallnäs & Redström 2002].

This meant that every time that the UX-team tried to prioritize the scenarios and user interfaces to the main users the product owners were concerned that the small and technical details were forgotten. The customer frequently brought up these issues during the discussions of the main usage scenarios. This was a challenge since the main purpose was to make the interface user friendly and these discussions were a symptom of a difference between how the product owners of the customer wanted the home control and the findings from the user research.

A combination of the two approaches was needed to get the process going and make sure everybody was committed. To ensure a focus on including all functionality requirements, a parallel process of writing down all the requirements was started – see next section for more details about this. This assured the product owners of the customer that no requirements were forgotten. At the same time the workshops were focused on a design for the main personas and usage scenarios, and the requirement specification helped drive this process.

As another mean to combine these two different approaches, Mjølner used the personas actively in all workshops and discussions throughout the entire development process. The personas were also included in all the use cases in the requirement specification as well as in a priority-matrix of all the functionality mapped with each persona. This priority-matrix aided the discussion of what functionality was relevant to which users and consequently, what functionality should be easily available and what should be tucked away in a lower level of the interface. This persona priority-matrix was furthermore used in the iterative development plan in order to develop the most essential functions first.

In general terms, the observation is that when engaging in a UX-driven design process with very functionality-minded product owners, care should be taken to combine the two approaches in order to make workshops more effective and focused. In addition, the documentation of all use cases and the persona priority-matrix made the hierarchy between personas and their needs more clear to the customer. This meant that the design and development process could be focused in accordance to the mainstream user and usage, and the customer could still evaluate that all important functionality would be available in the end.

5 Many Requirements Were Implicit and not Well-agreed Upon

During the early stages of the project in the spring of 2010, the main activities were user research activities, including identification of personas and main usage scenarios. It was assumed that the existing functionality should be kept and the task was to define a new interaction structure and presentation layout.

As the work progressed, it became clear that the basic product requirements were neither clear nor stable. Many requirements were implicit and not well-agreed upon. To address this problem, the writing of a requirement specification was started. With reference to Figure 1, the work until now had focused on business goals and, in particular, scenarios. The requirement specification should consist of system requirements and include use cases and verifiable requirements.

The goal was that all expectations to the product should be stated in a way that gave the UX-team the freedom to choose different user interface designs, without a new negotiation of the requirements.

In general, the requirement specification describes the functionality by scenarios and use cases [Cockburn 2000]. Each scenario is an imagination of how a user would solve everyday tasks with the home control. In this approach, a scenario consists of a number of use cases (we are aware that in the literature, it is sometimes vice versa). Every use case describes an interaction with the product by an “actor”. Each actor has different skills and represents various personas. An essential characteristic of a use case is to avoid any use or reference to user interface controls like buttons, windows or text fields. By doing so, the user interface design may change after the requirements specification has been stabilized and agreed upon. To ensure that a number of fundamental properties are stated, every use case has a number of one-liner verifiable requirement statements. A requirement statement can be verified with a “yes” or “no”. These one-liner statements can later be used as testable statements in test specification.

During the specification phase, there were many meetings and coordination activities between the requirement stakeholders on the customer side and Mjølner's project team. The pace of the work on user interface design was reduced, because the requirement stakeholders were uncertain that the user interface design covered all thinkable scenarios. When the first draft of the requirement specification was ready, it was difficult for the stakeholders to adapt this abstract and general kind of view of the home control behavior. After an introduction period, the requirement stakeholders became accustomed to reading it and the use case approach made a lot of things clearer for both parties.

The cooperation about the requirement specification was a positive and fruitful process and the further writing and maintenance of the requirement specification has become a joint project between the requirement stakeholders on the customer side and Mjølner's project team. The user interface design specification, which could now easily include examples of design and behavior, was finished afterwards, based on clear fundamental requirements. The two documents described jointly the expectations to the home control.

The observation, more generally, is that when engaging in a UX-driven design process, a more technical process of identifying and writing down the entire set of requirements should be done in parallel. The purpose is to get a solid foundation for agreement about the entire functionality and to ensure the customer that no requirements are forgotten even though the main user scenarios are in focus.

6 A High Degree of Parallelization

Mjølner's software development process is an iterative, agile approach [Larman 2004]. Overlaps between and iteration of analysis, design and implementation are a part of the process. For Mjølner's projects it is common that the user interface is not completely designed before the implementation phase is initiated. Of course, if the interface could be designed fully before development it would be a more simple matter to convert the complete set of fully designed graphical user interfaces into an actual implementation. In practice, however, the project in question has seen several reasons why such a sequential approach will prove impossible:

- Project deadlines dictate a high degree of parallelization. There are even periodical milestone deliveries which include some amount of work from both UX-team and the software developers.
- The UX-team requires input from the developers in order to design the user interface. This could be performance data such as achievable frame rates or which animation effects can be implemented.
- The developers require input from the UX-team in order to create a solid architecture. Some knowledge about how the user interface would look like is necessary.
- The UX-team needs to elaborate on design details and continually evaluate the implementation, which means they need to continue to work on the project at least in some extent during the implementation phase.

So it was obvious that a parallel effort by the UX-team and developer teams was required. The software development started before the user interface was completely designed and finally approved. This meant that the software developers were faced with several challenges: Since the UX-team had focused on the key scenarios, a considerable part of the user interface was largely unknown. Still, the software developers needed to come up with a coherent software architecture that adequately supported the secondary scenarios and not-yet-designed user interfaces.

Arguably the most important aspect of solving inter-team dependencies is effective communication. To that end, a project office accommodating both the UX specialists and the project team was set up early in the project. Daily meetings with all project members attending is used, and everyone gets CC'ed on important information. This is the standard way to set up a project in Mjølner.

The software architecture was defined by an approach inspired by Presenter First [Alles et. al, 2006] which enabled the developers to concentrate on core functionality and defer user interface implementation until later. This was true in theory but in order for the developers to translate the use cases to code they needed to know what the user interface would look like. This entwined UX-team and developers further.

Being agile does not mean that all kinds of changes are always embraced. It is certainly preferable that some key decisions are made and remain unchanged from early in the project. Specifically in this project constraints have been seen, which are difficult to deal with in an agile fashion, and which are not addressed by the measures taken. A main problem has been that the target hardware platform is not yet available, so it is difficult for the developers to produce the input needed by the UX-team regarding performance and possible animation effects. Estimation of the software development efforts and estimation of platform memory usage – the latter being input for the hardware design team – are other examples.

It was a big challenge to solve of performance estimation needed by the UX-team. An overly optimistic estimation may yield disappointed customers when the product is finalized, whereas a too conservative approach could result in a product that was subpar. A performance study on a similar hardware platform was conducted, and this gave a rough performance indication. Since that particular platform was not optimized for this specific purpose, it was decided to adopt the performance numbers as worst-case. The developers were then able to do simple calculations on the effects requested by the UX-team and provide their best guess as to whether implementation was feasible. Through this input, constructive dialogue between UX-team and developers usually provided a compromise both could support.

The next step was to implement a PC-based simulator of the target system. Because the hardware was unavailable, the need for a simulator was rather obvious unless development was to halt completely, but there are additional derived benefits from having a simulator. It enables early evaluations of the user interface. The simulator is also used in milestone deliveries to the customer to document progress throughout the project. Since it can easily be emailed, it can even be used to provide context for text translators or education of support personnel.

In more general terms, the conclusion is that was necessary to move forward with the implementation, even though the target hardware was not available; this was an external constraint that could not be removed, but only accepted as a given fact. Introducing the simulator was the solution to deal with this constraint and it has proved very useful. Furthermore it was found that because UX needs input from developers and vice versa a very close collaboration between UX-team and developers is imperative for a project like this.

7 From Project Findings to Software Process Improvements

Four main challenges that the project have faced and are facing have been presented. To sum up, these challenges – and the key observation made about them - are:

- A large and diverse set of stakeholders: It is important for projects of this type to actively involve a broad range of stakeholders and put them together in workshop like environments to get good decisions fast.
- The customer was used to a functionality-based process: Theory might note a UX-driven design process as the best, but in cases where the customer is focused on functionality a mix of a UX-driven and a functionality-based approach makes collaboration easier.
- Many requirements are implicit and not well-agreed upon: Even though the customer has a functionality-based approach and the product developed is a new version of an old product, it is not certain that a clear picture of the requirements exists.
- A high degree of parallelization: Due to external constraints this project had to do things in parallel even though a more sequential approach would be more efficient. Building a simulator and using close cooperation and communication – also with the development team at the customer – have been essential.

These observations may obviously be applied to other similar projects as well. Therefore, each of the four observations should have an impact on Mjølner's software development process.
At the time of writing the observations reported in this paper have resulted in the following proposals for improving Mjølner's software development process:

1. The stakeholder management process should include best practises on how to get the customer involved actively and involve all relevant different stakeholders as early as possible.
2. Our UX design process should to a higher degree embrace the fact that different stakeholders at the customers have different opinions.
3. Our UX design and requirements processes should be updated with the knowledge gained in this project about combining UX-driven and a functionality-based approach.
4. Decoupling of functional requirements and the user interface design should be adopted as a best practise in our requirement process
5. Our project management process should to a higher degree emphasis the need to map out the need of parallel activities and the relations between these activities.

8 Conclusion

In this paper, we have discussed a particular project – development of a home control in a joint effort between a large industrial customer and Mjølner - and its challenges. Moreover, we have described preliminary considerations about how these findings can be generalized to have an impact on Mjølner’s software development process. Making the actual change to Mjølner's software development process is work that will be in progress in the near future.

We believe that reporting our experiences are of interest to other software companies as well. We think that the challenges discussed are common challenges in many software projects – especially for projects that have a high degree of user interface design. And the challenges are even more relevant when a software company is delivering to an industrial company, whose main business is not software, but something else.

We hope that the descriptions in this paper contribute to fruitful discussions in general about how to address problems of this type in software development projects. Moreover, we hope that discussions within the EuroSPI community may guide Mjølner to find ideas to improve the software development process so that it more generally deals with challenges like the ones we have discussed in this paper.

9 Literature

[Alles et al 2006]: M. Alles, D. Crosby, C. Erickson, B. Harleton, M. Marsiglia, G. Pattison, and C. Stienstra, *Presenter First: Organizing Complex GUI Applications for Test-Driven Development,* 2006, http://www.atomicobject.com/files/PresenterFirstAgile2006.pdf.

[Cockburn 2000]: A. Cockburn, Writing Effective Use Cases, Addison-Wesley, 2000.

[Colborne 2011]: G. Colborne, Simple and Usable. Web, mobile, and interaction design. New Riders, 2011.

[Hallnäs & Redström 2002]: L. Hallnäs & J. Redström, Abstract Information Appliances; Methodological Exercises in Conceptual Design of Computational Things. In Proceedings of DIS2002, London, ACM Press.

[Kjær & Jørgensen 2010]: M.Y. Kjær and J.B. Jørgensen, Mjølner's Software Process Improvement – a Discussion and Strengthening Using the SPI Manifesto, Proceedings of EuroSPI2010, Grenoble, France, Springer, 2010.

[Larman 2004]: C. Larman, Agile & Iterative Development, Addison-Wesley, 2004.

[Lauesen 2002]: S. Lauesen, Software Requirements – Styles and Techniques, Addison-Wesley, 2002.

[Nielsen 1994]: J. Nielsen, Ten Usability Heuristics, 1994, http://www.useit.com/papers/heuristic/heuristic_list.html

[Nielsen 2004]: L. Nielsen, Engaging Personas and Narrative Scenarios, PhD dissertation, Copenhagen Business School, 2004.

10 Author CVs

Jens Bæk Jørgensen

Jens Bæk Jørgensen works as senior project manager at Mjølner. Jens has a Master's Degree from 1991 and a PhD degree from 1997, both in computer science from the University of Aarhus. Previously, Jens has worked in the software industry for Meta Software Corporation in Cambridge, Massachusetts, and for Systematic Software Engineering in Aarhus, in addition to an employment as assistant and associate professor at the University of Aarhus. Jens has experience with a broad range of roles in software development projects – developer, analyst, architect, consultant, and project manager. Jens has been doing research on software engineering for many years, in particular on requirements engineering; he has authored approximately 40 peer-reviewed scientific papers (see http://www.daimi.au.dk/~jensbaek/), most of which are published in premier journals (e.g., IEEE Software) and well-respected conferences (e.g., International Conference on Requirements Engineering).

Mikkel Yde Kjær

Mikkel Yde Kjær works as SPI coordinator and line manager at Mjølner. Mikkel has a Master's Degree in computer science from the University of Aarhus from 1999. Previously, Mikkel has worked for Systematic Software Engineering in Aarhus, where he has been actively involved in getting the company certified as CMMI level 3, 4 and 5. Through the years he has been involved both as a project manager and by working 1½ years as Change Agent in the SPI department. Mikkel has been using Scrum and other agile methods since 2004 and has a special interest in Lean Software Development. Mikkel has a broad software experience – developer, project manager, consultant, change agent, teacher, and facilitator.

Inge Mølgaard

Inge works as UX (User experience) consultant at Mjølner. Inge has a Master's Degree in Information Science from the University of Aarhus from 2006. Previously she has worked as a UX-consultant for the digital agency Creuna and for the Alexandra Institute that ensures a closer cooperation between the world of research and private corporations and public institutions. She has been in charge of large-scale web- and intranet solutions as well as it-solutions integrated in libraries and museum throughout Denmark. As a UX consultant she is used to collaborate closely with customers in order to gain insight in the organization and business goals for the it-system and to manage processes that collect information and requirements from diverse groups of stakeholders.

Søren Snehøj Nielsen

Søren Snehøj Nielsen has a Master's Degree in Engineering from the University of Aarhus and works as a senior software developer at Mjølner. Søren has 7 years experience developing embedded software for a wide range of industrial systems. Over the years Søren has been active in many aspects of the Software Process Improvement activities at Mjølner, especially relating to architecture and project management.

Niels Mark Rubin

Niels Mark Rubin works as senior project manager at Mjølner. Niels has a Bachelor Degree in Electronic Engineering from Faculty of Engineering at University of Southern Denmark (former Odense Teknikum) from 1979. Working in the software area for over 30 years, Niels has experience with the many aspects and challenges of software development. He has been employed at Brüel & Kjær, Nærum for 10 years working with development of an OO operating system and applications - development of Element Management Systems at DSC Communications, Ballerup – developing mobile supervision and control solutions for industrial control systems at Resource Software Engineering, Aarhus - designing and managing complete flow control solutions for industrial turn-key systems at KJ-Industries, Aarhus and at Systematic Software Engineering, Aarhus. Niels has experience with a broad range of roles – developer, team-leader, product manager, architect, consultant, and project manager.

Improvement of a Release Management Process

Burcu Büyükkağnıcı, İlker Çelik, Bahadır Pekmen
INNOVA IT Solutions, Ankara-Turkey
{bbuyukkagnici; icelik; bpekmen}@innova.com.tr

Abstract

This paper presents a release management process improvement project where delivery processes of INNOVA TELCO OSS group has been re-planned and monitored according to customer needs and the problems recorded during release and deployment of the product. INNOVA is the first company in Turkey that has SPICE Level 2 certification according to ISO/IEC 15504 Part 7.

The customer needs, mentioned in the paper, are the key drivers and trigger the improvement project

In this process improvement work, we establish a release management environment that makes our software development environment measurable and visible. Different tools are used and interdisciplinary skills are orchestrated. An holistic approach for the overall project success is used during the project.

The authors outline the process improvement, knowledge management and infrastructure process implementations with tools during this project.

Keywords

ISO/IEC 15504, Software Process Improvement Project, Release Management, JIRA, SONAR, TELCO, Ant, Sharepoint, Bug

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.

INNOVA has different line of businesses and one of the core programs is Telecom Customer Management System (TMS) Solutions operating under TELCO OSS (Operations Support System) Line of Business. TMS has been providing services to Turk Telekom more than 10 years. These services include customer subscription services, campaign management, customer management system, online services, and trouble ticketing and customer fulfillment services. TMS is developing, installing core OSS functionality for Turk Telekom.

TMS has as both operational and project based activities. Both maintenance activities and new requirements are running at the same time. TMS has both web and desktop applications based on JAVA and PL/SQL. Since this product gives services to millions of Telekom customers, reliability, availability, security and performance requirements of this system are more delicate than many other systems. Therefore from the beginning, TMS has a stable and robust product architecture and also good configuration and release management infrastructure.

After monopoly of Turk Telekom has ended and new TELCO operators entered to the Telecom market, the company has to be more competitive. In order to attract potential customers and keep customers loyal, the company offers more campaigns, new product packages, new infrastructure. So the time to market delivery thresholds drop, business volume rises. This situation effects software development life cycle of TMS clearly. In order to be faster, the defect rates, development cycles and development costs should be decreased. These conflicting requirements make a process management and improvement program must. TMS has, therefore, initiated a software process improvement project mainly to achieve better product quality, greater schedule predictability, productivity and consequently increased competitiveness.

INNOVA had SPICE level 2 rating according to ISO 15504 Part 7. The improvement program basically re-defines and implements release management process, establishes a knowledge management infrastructure by orchestrating different development tools with each-other.

The process improvement methodology in INNOVA is based on IDEAL [1] and complied with process improvement process of SPICE. With this project, process improvement process, which is a Level 3 process, deployed to process asset library of INNOVA.

Following the famous quote of Watts Humphrey [2], "If you don't know where you are, a map won't help", so as a first step it is decided to make a problem definition by identifying current state of defect arrival rates and improvement opportunities. INNOVA decided to have an assessment to see its position in the process improvement journey.

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.[3]

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.[3]

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.[3]

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.

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].

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. [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. Problem Definition
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 Problem Definition

3.2.1 Process Improvement Project Team

This process improvement project has concrete business drivers, the success and sustainability of a process improvement program are clearly related to management support and leadership. This process improvement project is initiated by TMS program manager and TELCO OSS director. Both managers have been working in the company for more than 10 years.

The core decision makers are comprised of program manager, project manager and a quality assurance specialist. The sponsor is the TELCO OSS director.

TMS has nearly 30 employees consisting developers and test & business specialists. All of them participate to the project. We do not experience resistance towards the project, this is because leadership of the sponsor and program manager are accepted and respected by the project team.

3.2.2 The Problem Statement

The need for the release management process has been originated by customer needs. The customer wants to operate faster and be first to the market in many business areas to be more competitive. This need requires our products more defect free, reduction in rework times. The customer forces us to improve our defect arrival rates and other kinds of bugs that are not development related however integration, installation and configuration related.

From the point view of INNOVA problem is to find a solution to meet the growing requirement rate of Turk Telekom. The problems to be solved are listed as follows:

- Number of bugs discovered by internal and external tests are high.;
- The effort to correct defects are high,
- The defects found in production environment are high.
- Release notes are created manually by team members just before product release, so some fixes related to that release can be missing.
- “Dependency hell“ [4] problem exists because of manual gathering of jar files.

Following this list, alternative solutions are defined.

3.2.3 The As-Is Development Environment

TMS has monthly planned main production releases, usually after each production release within 5 days there is a bug fix release. For each main deployment, the issues are composed of bugs, new feature requests, improvements.

Issue management including task management is done with JIRA 3.13 [5]. The integration and build process are done by Cruise Control and Ant. The development is done with JAVA and PL/SQL.

3.2.4 The Solution Set

Alternative solutions are listed and prioritized according to emergency and practicality of them. The following are solution alternatives:

- To establish a release management environment that we can measure current defect rates per release basis.
- To implement checklists to detect potential bugs before code is deployed to test.
- To define frequently made especially coding mistake and establish an environment to communicate best coding practices and other assets.
- To standardize and implement new test strategies to detect more bugs before code is deployed to production.
- To automate regression tests
- To establish a performance indicator database to store results for defects rates of each release.
- To automate release notes
- To automate dependency management process.

3.3 Planning the Project

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

The Project Management Plan consisted of:

- Problem analysis and definition,
- Stakeholder identification,
- Gathering requirements from the stakeholders,
- Definition and scheduling improvement activities,
- Definition of success criteria,
- Assessment of improvement activities
- Project Evaluation

The plan is published to all project team members in document management system.

3.4 Implementation of SPI

3.4.1 SPI Project Establishment

SPI Project is also managed as a separate project. A project definition is done in JIRA and the work products of this project are kept in document management system of TMS. Tasks and requirements of this project are assigned and controlled as issues in JIRA.

3.4.2 Collecting Defects Found Before and After Customer Delivery

Issue and task management has been done using JIRA by TMS since 2007; however issues were not managed via version basis.

This step is required since it is necessary to know what the current state of the defect arrival rate in order to improve the situation. After this step is completed, it is intended to monitor these rates to determine whether proposed improvement ideas are actually working.

Release management needs for TMS were determined and documented in TMS Release Management Requirements Specification by program manager, project manager, and JIRA administration team.

The process for release management and needed customizations are documented in this SRS.

- For every monthly release, the related project manager creates new version descriptions in the project.
- After each deploy, if necessary bug fix release version definitions are done immediately.
- Relating an issue with affected version and component are made compulsory while creating an issue.
- Domain names are used to identify the issues that are created by customer or INNOVA employee.
- Several rules about versioning in JIRA are documented and published to TMS employees.
- Communications between employees are done through Sharepoint document management system and also e-mail groups.
- After this system starts to operate, the issues categorized as task, bug, improvement, new feature, are collected per version as found before and after product delivery.
- Some of the versioning rules are automated, some of them are not. Therefore, data validity is also a concern and should be controlled. A quality assurance engineer is responsible for checking whether issue entries are done right or not.
- Rules are updated according to the feedbacks of TMS team members.

3.4.3 Collecting Information about Re-Opened Bugs

Re-opened bug rates can be the indicator of solution quality. There are known issues which are result of the current infrastructure of TMS solution, so there are only workarounds to solve the problems. However, in order to differentiate between the known issues and inferior solutions, reopened bug information is collected from JIRA while collecting defect per version metrics.

3.4.4 Assessment of Product Testability and Quality

Mainly cyclometric complexity and LOC metrics are used to determine product testability. Also to see the product design quality and avoid coding mistake, three static code analyzers for JAVA (Check style, Findbugs and PMD) are used. An open source tool that is orchestrating all these tools called SONAR[6] is used to present these metrics. The important thing here is to monitor these metrics for the newly added LOCs, since TMS codebase has been working for 10 years.

3.4.5 Establishment of the Metrics Analysis Database and Sharing Results with Team

The important thing with the metrics is to understand the current situation and trends as we are implementing the improvement suggestions. SONAR is a metrics database that can do trend analysis per build and show the progress on a timeline. Also it allows defining and storing manually collected metrics. It is also a dashboard that shows many product quality indicators in one screen. The obvious solution is to store all the metrics in one place, SONAR. However, SONAR is working synchronous with continuous integration tool and takes version numbers from it; for manual metrics we are working with JIRA version numbers and currently there is no mapping between continuous integration tool version numbers and JIRA release numbers. Therefore, manual metrics are stored in document management system under version control in Sharepoint.

3.4.6 Getting Out of Dependency Hell

Dependency hell is a colloquial term for the frustration of some software users who have installed software packages which have dependencies on specific versions of other software packages Dependency hell problem is solved quite easily by introducing IVY to the release process. Ant. IVY is a tool for managing (recording, tracking, resolving and reporting) project dependencies. It is also open source and can be integrated with Ant which is already a tool buing used for build process.

3.4.7 Release Notes Automation

Release notes define basically what has been fixed, developed, and improved in that release. Before release notes automation, team members manually enter that information to an excel sheet. Although this information can be extracted from JIRA, since both some issues have specific configuration changes ,database development need and also there are different kind of issues such as tasks, sub tasks, development bugs that should not be included in release notes. To overcome these problems, we have modified screens in JIRA, added configuration changes and database development information into the screen that developer resolves issue. A checkbox that indicates whether the issue should be included to release notes document or not is added also to the resolution screen. We have developed a php program that gets this release notes information and publish this information to the integration server dashboard. In this way, release notes are generated with each build real-time.

3.4.8 Deployment of Improvement Ideas

Process improvement activities are mainly manual and leader driven process that is supported with a workflow based tool. After observing current rate of reopened bugs, cyclometric complexity and static code analysis results, each development team leader start to watch SONAR dashboard for weekly builds and communicate the results with the team. A component called infrastructure improvement suggestion is created under TMS project in JIRA, and SONAR findings to be improved are assigned to the developers. Each team member also has access to SONAR server and has freedom to improve his own coding style according to results without waiting for team leader's assignment.

Time reduction for non value added activities is an important thing to achieve our goals, so automating functional tests is a must. Test specialists automate some of the functional tests. The as-is test methodology is re-defined with test specialists and communicated through team.

Deployment process is also a time consuming process because of known issue dependency hell. IVY is introduced to our build process to manage dependency of files and avoid missing jars.

3.4.9 Knowledge Management Infrastructure Establishment

Knowledge is an intangible asset that is stored and forgotten in e-mails, people's minds, should be managed as a tangible asset. It means that deprecated data should be immediately archived.

Project team sites are created for TMS and its subprojects. Sharepoint 2010 is used as a content management system. Information policies are defined in the tool to archive the out-dated contents etc. Blogs and wikis are created to share knowledge. Both team blogs and personal blogs are created, e-mails are kept in the tool also not to forget and for new comers.

A system admin is assigned for administrative reasons. Knowledge management procedure is documented and released to project team.

The results of this improvement program have been published using the project site.

4 Assessment and Analysis

We need to test whether the improvement ideas are improving the current situation and solving the listed problems. Some improvement ideas have immediate effects, but some of them need time to be assessed correctly.

- Apache IVY solves the problems originating from missing jar files very concretely. The bug fix releases because of wrong deployments ended since IVY introduction.
- Release notes automation ended customer complaints regarding in which release the issues customer entered is solved.
- Static code analysis shows clearly the coding practices that are needed to be improved. Each team leader raises the awareness in his/her team to SONAR results. Necessary refactoring activities are implemented.

In order to assess whether the overall bug rates are improved significantly or the number of bugs that are found by INNOVA increases relative to the customer, we need time to gather sufficient and enough metrics.

Since INNOVA is a SPICE level-2 company, we also assess ourselves informally according to related SPICE processes.

Table 1: Process List

Process Name	Level	Process Purpose [7]	INNOVA Outcomes
Knowledge Management	3	The purpose of the Knowledge management process is to ensure that individual knowledge, information and skills are collected, shared, reused and improved	• Sharepoint 2010 as CMS. • JIRA as issue tracker and management dash-

		throughout the organization.	board
Infrastructure	3	The purpose of the Infrastructure process is to maintain a stable and reliable infrastructure that is needed to support the performance of any other process.	• Integrated application lifecycle management environment. • JIRA + Hudson + Ant + IVY + Sonar+ SP 2010 • Definition of roles and responsibilities
Product Release	2	The purpose of Product release process is to control the availability of a product to the intended customer.	• Release notes are automated. • Versioning and dependency management practices are improved. • Customer interaction about release packages are managed through JIRA, and e-mail groups.
Process Improvement Process	3	The purpose of the Process improvement process is to continually improve the organization's effectiveness and efficiency through the processes used and maintained aligned with the business need.	• Improvement workflows are implemented on workflow of JIRA • Process improvement process is documented and published via document management portal. • A managerial board is established to support improvement activities. • Improvement activities are managed and monitored as a project, success criteria are documented and measured.
Measurement	3	The purpose of the Measurement process is to collect and analyze data relating to the products developed and processes implemented within the organization and its projects, to support effective management of the processes and to objectively demonstrate the quality of	• Data collection and analysis mechanism is defined. • Measurement activities supporting project goals are defined.

		the products.	• Data validity mechanisms are defined. • Data inventory and versioning mechanisms are established. • Needed infrastructure tools are acquired.

4.1 Future Work

The results of the project and achievements will be observed and measured in the near future. The metrics database, analysis and reporting activities should be enhanced. Next 12 months will pass with data collection per release until the data is enough and valid to end with decisions regarding process capability and control limits. Statistical background of project team members should be enriched.

After project results are observed, these will be shared with the rest of the organization. We expect that if the results are good, other teams will involve those kinds of activities voluntarily.

Metrics for architecture quality evaluation will be used more systematically and methodologies will be developed to use metrics for product quality purposes.

Although informal assessments are done against SPICE level 3 projects, in the end formal assessments will be done for level 3 activities.

4.2 Lessons Learned

As the project progresses, we observe the followings are crucial for project success:

- Process improvement activities need executive commitment and awareness of the teams participating in projects. [3]If executives support and track the results of the project and communicate their expectations with the team, then team members would like not only to involve but also give feedback and become active members of it.

- Leadership is important to sustain the process improvement project. In order to be respected by the team a leader should have knowledge, technical background, open to communication and take action.

- Holistic approach, having short term, achievable goals and having clear project success criteria are important for project members to stay focused.

- are important

- Customer needs drives the attention of executives in order to start and monitor a process improvement project

- Integrated development environment and developer friendly tools are important aspects for process improvement success and speed of this success.

- It is impossible, if the project is not accepted by team members, for a process improvement to be successful.

- Models such as SPICE, CMMI and etc. provide reusable knowledge and best practices that are acceptable by common sense and they are good recipes to start what you should do when you come across a problem.

5 Conclusion

While operating in a competitive environment, it is impossible not to change. Process improvement activities including not only software but systems and business as a whole will continue to happen as we are operating.

Release management is at the center of the application development life cycle. Different processes or systems and people with different roles interact while implementing this process. No matter how separate processes or teams are successful, if integration between them cannot be coordinated, success of the project will be very doubtful.

As a team we have a holistic view for process improvement activities. We emphasize to improve a process area that will have the maximum positive and visible effect on overall project success.

When process improvement activities are beneficial to stakeholders, solve frequent problems, and the results are objective, participation of the team increase.

In order to have positive results, improvement team should consist of problem owners, has authority to make and implement decisions, and has related technical skills and holistic view. Also improvement team should be very agile, has short term achievable and visible goals.

So far, what we have done is to make our working environment less chaotic and become more eligible what we are achieving with current development and management processes. We have metrics that show defect rates which means we know our process capability, after this point we will find ways to improve these rates, and continue the next phase of the process improvement project.

6 Literature

1. McFeeley, B., IDEAL: A User's Guide for Software Process Improvement, Handbook CMU/SEI-96-HB-001, Carnegie Mellon University, (1996)
2. Humphrey, W. Managing the Software Process, Addison-Wesley, (1989)
3. Büyükkağnıcı B, Yazıcı E, Tüfekçi Ö.,The First Achievement of SPICE Level 2: a Case Study from Turkey, EuroSPI 2010
4. http://en.wikipedia.org/wiki/Dependency_hell#cite_note-Dependency_Hell-0, Retrieved 2011.03.02
5. http://www.atlassian.com/software/jira/, Retrieved 2011.03.02
6. http://www.sonarsource.org/ , Retrieved 2011.03.02
7. ISO/IEC 15504- Process Assessment Model

7 Author's CV

Bahadır Pekmen

Mr. Pekmen was born in 1975. He was graduated from Computer Engineering Department of Hacettepe University in 1997. He has been working in INNOVA for 11 years as program manager of TELCO Projects.

Burcu Büyükkağnıcı

Miss Büyükkağnıcı was born in 1982. She was 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.

İlker Çelik

Mr. Çelik was born in1980. He was graduated from Computer Engineering Department of Ege University in 2002. He joined INNOVA IT Solutions in 2008; he is the project manager of Turk Telekom Trouble Ticket and Service Fulfilment Solution.

Engineering Object Change Management Process Observation in Distributed Automation Systems Projects

Dietmar Winkler, Thomas Moser, Richard Mordinyi, Wikan Danar Sunindyo, Stefan Biffl

Christian Doppler Laboratory for Software Engineering Integration for Flexible Automation Systems (CDL-Flex), Institute of Software Technology and Interactive Systems, Vienna University of Technology, Austria

{firstname.lastname}@tuwien.ac.at

Abstract

Development processes for automation systems, such as power plants and industrial production plants, involve several engineering domains, e.g., mechanical, electrical, and software engineering, as well as artifacts based on heterogeneous tools and data models. Observations in automation systems development highlighted challenges (a) in the collaboration of engineering domain experts and (b) in process observation for project monitoring and control across engineering disciplines and tools. However, efficient data exchange is a pre-condition for process observation and control and enables efficient collaboration within systems development projects. Based on the Engineering Service Bus (EngSB) – a middleware platform for supporting collaboration across tools and domain borders – this paper presents the implementation of a change management process in an industry showcase at a system integrator in a power plant engineering domain. First results showed increased flexibility, improved collaboration capabilities, and the ability to measure process metrics across discipline borders, which could not easily be measured in common automation systems development processes.

Keywords

Automation Systems Development, Engineering Service Bus, Process and Project Observation, Engineering Process Improvement.

1 Introduction

Automation Systems Engineering (ASE), e.g., the engineering of production automation systems or power plants, includes a wide range of heterogeneous disciplines, such as mechanical engineering (e.g., physical layout), electrical engineering (e.g., circuit diagrams), and software engineering (e.g., UML diagrams, function plans, and software code) [1]. Normally, different disciplines apply individual engineering processes, methods, and tools with specific data models, addressing individual needs of involved engineers.

Observations at our industry partner – a power plant systems integrator – showed that traditional ASE processes follow a basic sequential process structure with distributed parallel activities in specific phases and suffer from a lack of systematic feedback to earlier steps, inefficient change management and synchronization mechanisms of disciplines, and low engineering process automation across domain boundaries [7]. Specific tools and data models typically address only the needs of one individual discipline and hinder efficient collaboration and interaction between disciplines. Because of this lack of collaboration, change management becomes even more difficult, leading to development delays and risks for system operation.

For instance, changing hardware components (e.g., hardware sensors) might require changes in software components (e.g., caused by changed value ranges, data types (e.g., analogue / digital sensor), or number of connection points). Thus, key questions in context of change management are (a) how changes can be handled more efficient and (b) how relevant change requests can be passed to involved engineers. From project management perspective, loosely coupled processes hinder a comprehensive observation of current project states and make project control more difficult [1]. Thus, there is a need for flexible and comprehensive engineering process support across disciplines to enable collaboration and interaction between disciplines, tools, and data models. In context of process observation, key questions include (a) how to model and evaluate engineering processes in heterogeneous environments and (b) how to derive project metrics based on these observations.

The Engineering Service Bus (EngSB) – a middleware platform for supporting collaboration across disciplines and domain borders – bridges the gap between heterogeneous disciplines by providing semantic integration of data models [3][7][8] based on the technical integration of domain-specific tools [1][2]. An integrated view on heterogeneous engineering environments enables (a) comprehensive process support across disciplines, (b) efficient change management, and (c) process and project observation. Defined processes intertwined with tools and data models can enable process automation, observation, and improvement based on process measurement.

This paper presents (a) the EngSB concept and (b) the implementation of a flexible and efficient change management process, and (c) demonstrates the ability for process monitoring based on real-world project data from a power plant systems integrator. The results support researchers in systematically analyzing and improving engineering processes in a distributed and heterogeneous environment and practitioners in controlling distributed engineering projects. The remainder of this paper is structured as follows: Section 2 presents related work on automations systems engineering, engineering processes, and the Engineering Service Bus (EngSB) approach as a foundation for automation systems development processes. Section 3 identifies the research issues. Section 4 describes the solution approach and section 5 presents a snapshot of a power plant engineering project as a prototype implementation for process observation. Finally, section 6 concludes and identifies future work.

2 Related Work

This section summarizes background information of automations systems engineering, engineering processes, and the basic concepts of the Engineering Service Bus approach.

2.1 Automation Systems Engineering

Automation systems (AS), such as complex industrial automation plants for manufacturing [3] or power plants [10] depend on distributed software to control systems behavior. In automation systems engineering (ASE) software engineering depends on specification data and plans from a wide range of engineering aspects in the overall engineering process, e.g., physical plant design, mechanical and electrical engineering artifacts, and process planning. This expert knowledge is embodied in domain-specific standards, terminologies, people, processes, methods, models, and tools. The weak technical integration of tools within an engineering environment and the weak semantic integration of the expert knowledge across domain boundaries of engineering aspects make late changes in the engineering process inefficient, error-prone, and risky.

Additional risks come from the complexity and size of plants and type of projects [10], e.g., large and customized power plants designed according to individual customer needs (i.e., individual customer-specific solutions) and modernization projects of existing power plants (i.e., maintenance and extensions). Furthermore, the technical plan documentation often differs from the real solution (at the plant site) because of last-minute on-site modifications (e.g., during construction and/or commissioning) with limited documentation of changes. Consequently, there is a lack of feedback of the "as-built" documentation (at the plant site) to the engineering documentation (at the engineering site). Thus, changes are time consuming (research of as-built documentation), risky (testing of changes) and error-prone because of a highly manual activity, e.g., synchronizing engineering documents across disciplines.

Assuming that technical and semantic gaps between different engineering teams lead to a lack of quality assurance (QA) of artifacts and inefficient change management approaches across engineering domains [9], a major challenge is to bridge the gap between heterogeneous disciplines on a technical and semantic level to enable efficient change management and data collection for project monitoring and control during development, commissioning, and maintenance.

2.2 Engineering Processes

Traditional and sequential software and systems development processes, e.g., provided by the waterfall model or the V-model process approach [5][11], hinder efficient changes, especially late in the project. Additional requirements based on tight interaction with hardware components require modified engineering processes, e.g., GAMP 5 [5][6]. Nevertheless, interaction of heterogeneous disciplines with respect to (late) changing requirements and/or identified defects is still challenging. Figure 1a presents a basic sequential engineering process in ASE development projects. Late changes (1) across disciplines and tool borders make projects more risky and error prone if not considered by related disciplines. For instance, a missing alarm indicator of a critical measure in the control centre (e.g., oil pressure alarm indicator), identified during test/commissioning, might affect software engineers (alarm indicator not/wrongly implemented), electrical engineers (sensor not wired) or process engineers (oil pressure sensor not planned) and might lead to project delays and quality problems.

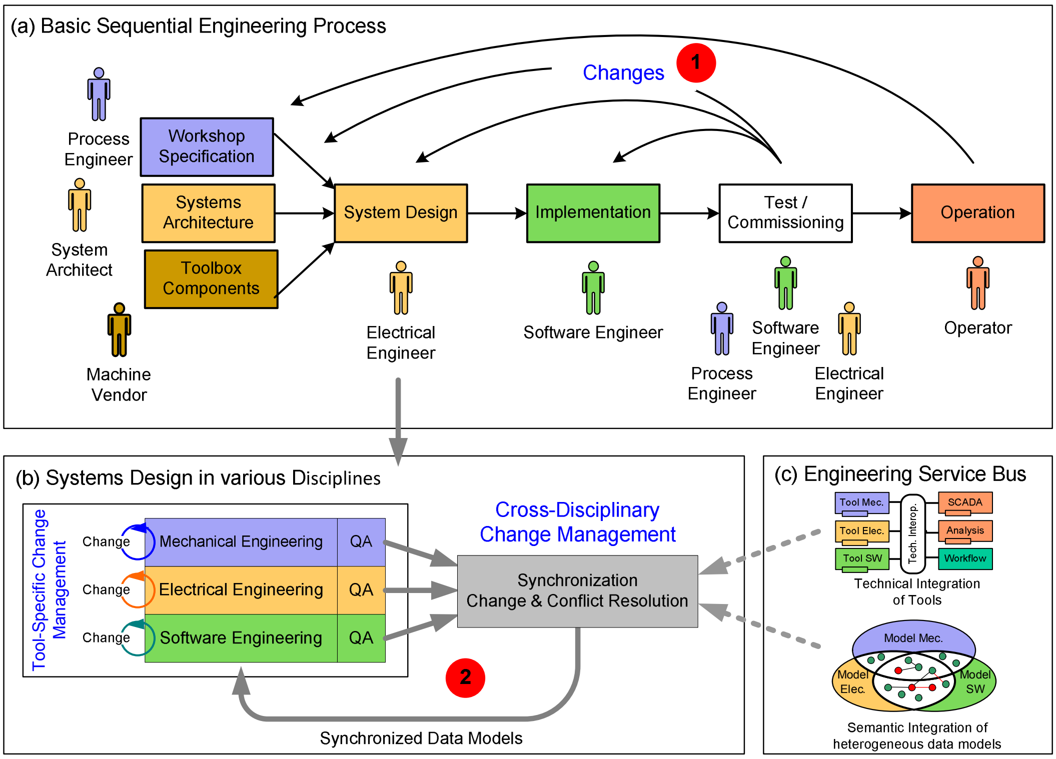


Figure 1. Challenges in ASE Development Processes

Our observations at our industry partner revealed that individual phases are not strictly sequential but individual engineers work both distributed and concurrently. Figure 1b illustrates a concurrent development approach and the need for data synchronization of individual engineers. This synchronization step is an important activity to keep the overall project consistent. Nevertheless, domain-specific tools, processes, methods, and data models hinder efficient point-to-point data exchange. Our observation was that this synchronization step is done rather infrequently (e.g., in a few months intervals), because the synchronization process turned out to be a time consuming and error prone manual activity. Thus, a major challenge is to synchronize data models across disciplines and tool borders more frequently (2). Automating this synchronization step will enable easy and more frequent synchronization activities, decrease effort and cost, and increase product quality. Note that Figure 1b focus on the systems design phase as an example; similar development practices apply to all phases of ASE development.

Distributed, concurrently, and less transparent workflows and processes hinder efficient and ongoing process monitoring and control; project managers are hardly able to assess the current project state frequently but have to capture related data from heterogeneous sources on request. Thus, measurement data on critical changes is missing and hinders appropriate project planning, monitoring, and control. Automation supported capturing of process events and data can enable efficient project monitoring and increase project transparency as a foundation for project management.

Technical and semantic integration of tools and data models across disciplines can enable (a) frequent synchronization and data exchange, (b) supports efficient change management, and (c) provides a foundation for process monitoring and control. Figure 1c illustrates the basic contribution of the EngSB approach for technical integration of tools and semantic integration of data models to enable automation supported change management and project observation.

2.3 Engineering Service Bus

The Engineering Service Bus (EngSB) [1][2], based on the Enterprise Service Bus Concept [4], provides the foundation for flexible technical integration of heterogeneous engineering environments and semantic integration [8] of related tool-specific data models derived from individual disciplines. In context of efficient data exchange and synchronization across disciplines, a key challenge is how to map individual data models to identify changes and conflicts. Mapping of individual data models are represented by white areas in Figure 1c, where two or more disciplines have to exchange data. Note that data exchange can be limited to common data (i.e., a subset of available data) without considering domain-specific data (not relevant for other disciplines).

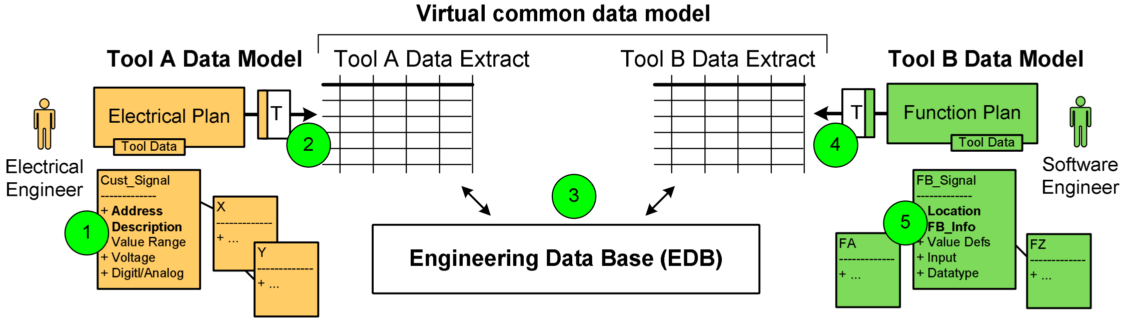


Figure 2. Schematic Overview on the Virtual Common Data Model (VCDM)

This Virtual Common Data Model (VCDM) is embedded within a semantic integration concept [8] and represents the foundation for efficient data exchange. Common concepts, i.e., signals or engineering objects used in all related disciplines, are the foundation for the VCDM to link individual heterogeneous data models.

Figure 2 presents the basic concept of VCDM involving two domain-specific roles, i.e., the electrical and the software engineer. Individual disciplines use domain-specific data and common data for synchronization purposes (1) and (5). The Engineering Data Base (EDB) offers the knowledge for mapping individual domain-specific data models and common data models (3). Domain-specific data models are checked-in from related tools, transformed to the VCDM (2) and stored in the EDB (3). Consequently, data from the EDB are checked-out, and transformed to the target format, e.g., the software related data model (4) and used in the domain-specific data model of the software engineer (5).

3 Research Issues

Based on the observed engineering processes and the need for interaction, collaboration, and synchronization of engineering artifacts across disciplines, we identified two major research issues: (a) how to design a change management process for supporting systems engineering projects in heterogeneous engineering environments (including notification of changes to propagate changes to involved stakeholders) and (b) how to measure and analyze engineering processes for process evaluation (process verification and validation) and project monitoring and control (project observation).

RI.1) Change management process approach. Strong collaboration and interaction of the involved stakeholders is necessary to support efficient change management processes including interaction support of various disciplines. Thus, we address two steps: (a) definition of a change management process and (b) evaluation of the process for verification and validation purposes. Process evaluation focuses on investigating whether the process behaves like expected.

RI.2) Change management data collection and analysis. Project measurement, analysis, and control are key activities of project managers. Our observation showed shortcomings regarding data collection and evaluation in heterogeneous engineering environments. The second research issue focuses on the analysis of process events and data with respect to identifying the number of changes and individual change types.

4 Solution Approach

This section presents (a) “Signal Engineering” as a common concepts in the automation systems domain, (b) a proposed change management process for signal changes at a power plant systems integrator, and (c) concepts for process evaluation (verification and validation) and analysis (as basis for project observation).

4.1 Signal Engineering

Collaboration between heterogeneous disciplines and tools requires common concepts for mapping individual models and activities. Our observation at the power plant systems integrator showed that signals [12] are common concepts in this domain that link information across different engineering disciplines. Signals include process interfaces (e.g., wiring and piping), electrical signals (e.g., voltage levels), and software I/O variables. Consequently, we use signals as a vehicle to link domain-specific data between different engineering disciplines and define the application field “Signal engineering” with focus on signal management facing the following important challenges: (a) make signal handling consistent, (b) integrate signals from heterogeneous data models/tools, and (c) manage versions of signal changes across engineering disciplines. Note that the identification of common concepts (signals in this paper) depend on the application domain and may differ. More general, the common concept can be described as “engineering object” and may include other building blocks of the automation systems, e.g., hardware components or software control units.

4.2 Change Management Process Design

System integrators have to synchronize several engineering data, i.e., signals, derived from heterogeneous sources, e.g., electrical, mechanical and software engineering, to a virtual common data model (VCDM) used by the EDB. Note that changes are defined within modified signal lists derived from individual tools and have to be synchronized with the current overall signal list in the EDB. Thus, change management refers to the merging process of signal lists with EDB data during synchronization. Figure 3 presents a basic signal check-in workflow at the synchronization step.

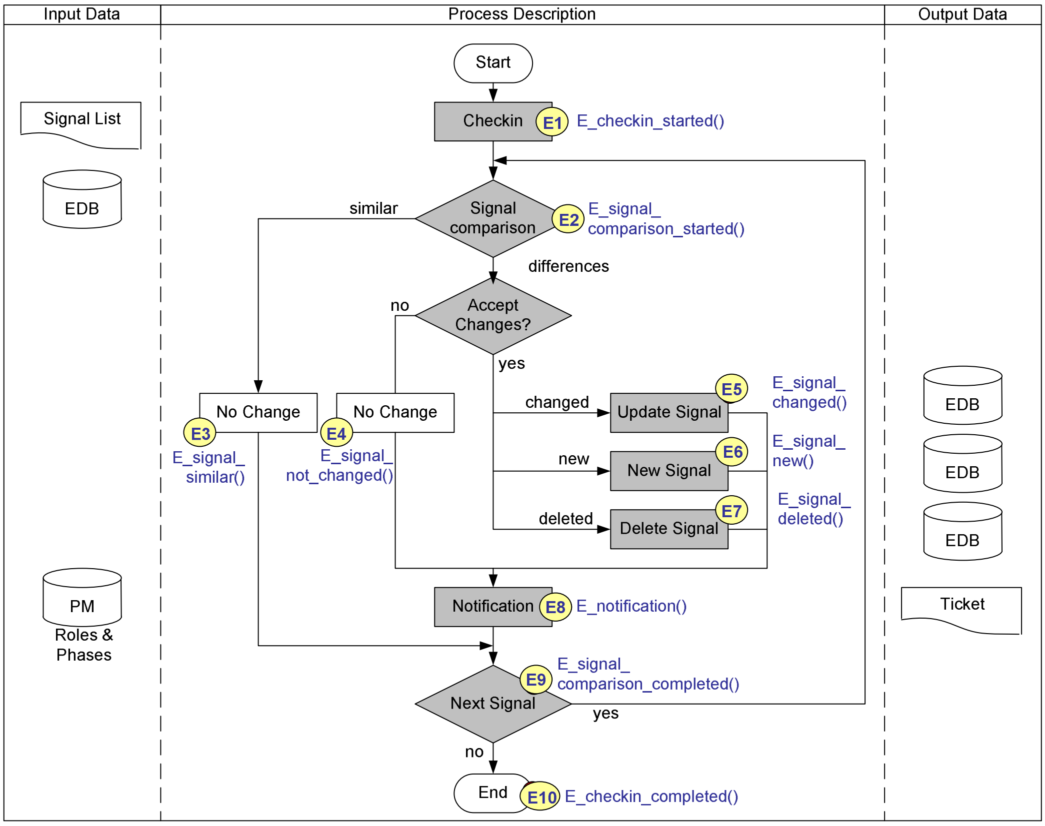


Figure 3. Signal Change Management Workflow

In addition to unchanged signal, changes can include (a) new signals, (b) removed signals, and (c) modified signals. Signal modifications result in a notification of involved stakeholders based on the project environment, e.g., involved stakeholders, related roles, and engineering process phase. Based on signal synchronization, individual engineers gain updated signals (prepared for specified tools and data models) for application within their individual tools. We introduced events (marked by circles in Figure 3) to (a) evaluate the change management process and (b) to measure project metrics based on changes. Table 1 summarizes the events used to observe and evaluate the proposed change management process.

Table 1. Change Management Process: Table of Events

Abbr.	Event Name & Description
E1 E10	*E_checkin_started (E1) and E_checkin_completed (E10)* represent one completed check-in sequence, i.e., a sequence of individual signals derived from a defined source.
E2 E9	*E_signal_comparison_started (E2) and E_signal_comparison_completed (E9)* focus on one signal within a check-in process in one signal list.
E3	Unchanged signals are reported by using the event *E_signal_similar (E3)*. This event is necessary to see whether the change management process works as expected.
E4 E5 E6 E7	Signal changes (i.e., deviations of EDB signal attributes and new signal list attributes) can be rejected *(E_signal_not_changed (E4))* or accepted. In case of accepting signal changes, three different events are required: (a) signal modified *(E_signal_changed (E5))*, (b) a new signal introduced *(E_signal_new (E6))*, or an existing signal should be removed from the EDB, i.e., missing entry in the signal list *(E_signal_deleted (E7))*.
E8	After change handling (accepted or rejected) a summarized notification *(E_notification (E8))* will be sent to all related stakeholder.

4.3 Process Evaluation and Project Metrics

Events are the foundation for process evaluation and project measurement. Derived event data from the change management process implementation show whether the process behaves like expected (process verification and validation) and what the bottlenecks of individual process steps are in terms of number of executed transitions and duration. In addition, process event data is the foundation for analyzing data for project planning and control. We applied ProM[1], a process mining workbench [13] for process evaluation, observation and analysis, i.e., defining the expected process model and evaluating the implemented process workflow based on captured events. Table 1 presents the event description of the implemented change management workflow presented in Figure 3.

Evaluating captured project event data enables *project observation and monitoring* from project management perspective. Based on discussions with our industry partners, important metrics focus on signal changes over project phases and time. A key assumption of our industry partner was that approximately 20% of signals are changed along the project progress. These assumptions are based on expert estimations. Missing process observation data and loosely coupled non-transparent processes of individual disciplines hinders measurement of the amount of changes in detail. Thus, we evaluated the captured events and derived a set of metrics based on the defined change management process.

Table 2: Project Metrics based on Captured Events

Number of ..	Metrics Description
Check-ins	Number of different signal lists from various sources.
Signals	Number of signals handled during an individual check-in.
Similar Signals	Number of unchanged signals (signal list compared to EDB signals).
Accepted changes	How many changes were accepted during an individual check-in? Accepted signals include (a) new signals, (b) deleted signals, and (c) modified signals.
Rejected changes	How many changes were rejected during check-in?

5 Prototype Implementation

This section presents the results of the prototype implementation of the change management process based on real world-data from a project at our industry partner, a power plant system integrator.

5.1 Study Material and Data Collection

We used three different signal lists from a real world project at our industry partner and captured occurring events, defined by the change management process. Table 3 presents an overview on used signals per signal list and captured events by the EngSB application. Note that the project is in a very early stage of development, i.e., in the systems design phase, where changes come up frequently. Also note that the signal change handling process does not include the overall project data but a subset of selected and defined components.

Table 3: Source Signal Data from our Industry Partner

	Phase 1.1	Phase 1.2	Phase 1.3
Number of signals	708	720	592
No of captured events	2,834	5,113	2,450

[1] http://prom.win.tue.nl/tools/prom6/

5.2 Process Evaluation

Process evaluation focuses on verification and validation of the implemented change management process. The basic process evaluation includes (a) inspection of event traces with ProM [13], (b) analysis of captured events per event type (see Table 4), and (c) consistency checks. In a first step, we analyzed the captured events after every phase, i.e., phase 1.1, 1.2, and 1.3, and compared the event traces with the expected traces, defined in the change management workflow (see Figure 3). The results showed that the implemented process behaves like expected, i.e., the process and the event capturing approach was implemented correctly. The second step includes a basic evaluation of captured events based on the ProM evaluation results. Table 4 presents the details of this analysis.

Table 4: Occurrences of Events based on ProM Data Analysis

Occurrence of events	Phase 1.1		Phase 1.2		Phase 1.3		Total	
	Abs.	Rel.	Abs.	Rel.	Abs.	Rel.	Abs.	Rel.
E1: E_checkin_started	1	<0.1%	1	<0.1%	1	<0.1%	3	<0.1%
E2: E_signal_comparison_ started	708	25.0%	1,300	25.4%	720	29.4%	2,728	26.3%
E3: E_signal_similar			89	1.7%	432	17.6%	521	5.0%
E4: E_signal_not_changed			20	0.4%	12	0.5%	32	0.3%
E5: E_signal_changed			19	0.4%	148	6.0%	167	1.6%
E6: E_signal_new	708	25.0%	592	11.6%			1,300	12.5%
E7: E_signal_deleted			580	11.4%	128	5.2%	708	6.8%
E8: E_notification	708	25.0%	1,211	23.7%	288	11.9%	2,207	21.2%
E9: E_signal_comparison_ completed	708	25.0%	1,300	25.4%	720	29.4%	2,728	26.3%
E10: E_checkin_completed	1	<0.1%	1	<0.1%	1	<0.1%	3	<0.1
Total	2,834	100%	5,113	100%	2,450	100%	10,397	100%

The third step includes a set of consistency checks to verify that the process was completely executed and that all signals have been processed. The consistency checks include the following metrics:

(a) *Number of signals:* According to the defined workflow, the number of signals (input data and EDB data) must be equal to the number of signal comparison events (E2 = E9).

(b) *Signals compared*: The number of compared signals summarizes similar signals, accepted and rejected changes.

(c) *Notification*: Signal changes (rejected or accepted signal changes) result in notifications to related stakeholders. Note that notification objects (e.g., engineering tickets) are summarized on component level to keep the number of notifications as small and focused as possible.

Based on ProM process analysis and analyzing the individual occurrences of events within the defined workflow, we reason on a well-designed and correctly implemented workflow.

5.3 Project Monitoring and Observation

The initial process evaluation confirmed the implementation of the proposed change management process and enables a more detailed analysis for project management purposes, e.g., measuring the number of signal changes per event type. Table 5 summarizes the derived metrics from analyzing captured events. The data sets have been derived from a very early phase of the development project, i.e., system design phase, which is an explanation of the high variability of signals (1,211 changes and 89 similar signals in phase 1.2; the number of changes decreases to 288 and the number of similar signals increases to 432 in phase 1.3).

Table 5. Change Management Metrics based on Signal Comparisons

	Phase 1.1		Phase 1.2		Phase 1.3		Total	
	No	%	No	%	No	%	No	%
Similar Signals	0	0%	89	6.9%	432	60%	521	19.1%
Accepted Changes	708	100%	1,191	91.6%	276	38.3%	2,175	79.7%
Rejected Changes	0	0%	20	1.5%	12	1.7%	32	1.2%
Signal Comparisons	708	100%	1,300	100%	720	100%	2,728	100%

Note that we do not consider multiple changes of the same signals. The metrics are based on signal comparison values, i.e., the number of signals in the EDB and the number of signals captured during check-in sequences. See Figure 4 and Figure 5 for bar charts of individual check-in sequences.

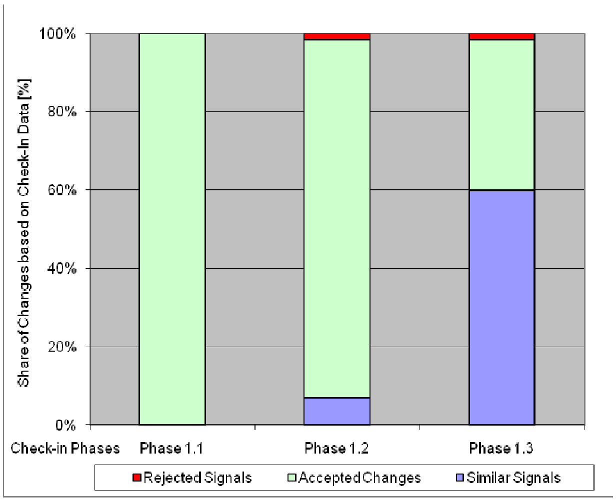


Figure 4. Accepted/Rejected Signals per Phase

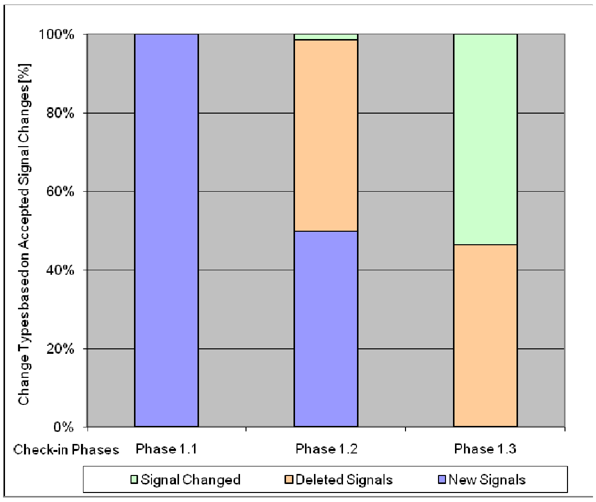


Figure 5. Change Type of Accepted Changes

Figure 4 presents the analysis results of accepted, rejected and unchanged (similar) signals per phase. The number of similar signals increases to 60% in phase 1.3 and the number of accepted changes decreases across the three check-in phases. Figure 5 and Table 6 present a more detailed view on signal changes based on accepted changes and on signal comparison activities. An interesting finding was that in phase 1.2 and phase 1.3 a high number of signals were removed, because sets of components have been replaced in early phases of development. Note that no new signal was introduced in phase 1.3.

Table 6. Signal Change Type of Accepted Signal Changes

	Phase 1.1		Phase 1.2		Phase 1.3		Total	
New Signals	708	100%	592	49.7%	0	0%	1,300	59.8%
Deleted Signals	0	0%	580	48.7%	128	46.4%	708	32.5%
Changed Signals	0	0%	19	1.6%	148	53.6%	167	7.7%
Accepted Changes	708	100%	1,191	100%	276	100%	2,175	100%

Based on the observed events and captured data after the third check-in (i.e., summarizing phase 1.1, 1.2, and 1.3) we observed an overall number of 2,728 signal comparisons and a number of 2,175 accepted changes (new, removed, and changed signals). A more detailed analysis of changes showed an overall change acceptance rate after phase 1.3 of 79.7%. Experts estimated approximately 20% of signal changes along the overall project course – this seems to be contradictory. Nevertheless, explanations for this deviation are: (a) we applied signal lists based on a very early project phase, i.e., the systems design phase, with incomplete and unstable requirements and a basic systems architecture; (b) the signal lists does not cover all components of the plant but is limited to a small subset of components, i.e., most critical components.

Nevertheless, we see the prototype evaluation as proof-of-concept of the proposed process, product, and project observation approach based on events that can support project managers in better understanding and analyzing the underlying processes and measuring generated products along the project course.

6 Conclusion and Future Work

Collaboration and interaction between different engineering fields are critical issues in automation systems engineering (ASE) because individual disciplines apply different tools and data models. This heterogeneity hinders efficient collaboration and interaction between various stakeholders, such as mechanical, electrical, and software engineers. Based on the EngSB, the Virtual Common Data Model (VCDM) enables efficient data exchange based on common concepts, e.g., signals or engineering objects, as foundation for change management and process observation.

RI.1) Change management process approach. Process evaluation is required for process verification and validation, i.e., whether the designed (and implemented) workflow behaves like expected. Collecting and analyzing event data with data mining and analysis tools (e.g., ProM) enables the investigation of processes and event traces for process and workflow verification and validation purposes. Applying process measurement within the signal change management process, the results showed that (a) the designed change management process is appropriate in context of automation systems engineering projects and (b) the presented event definition, collection, and evaluation is a valuable approach for process evaluation.

RI.2) Change management data collection and analysis. In addition, event data (captured during process execution) can enable project monitoring and control for project management purposes. Implicit data, such as the number of changes per phase (and/or per time interval), can be made explicitly for decision makers to get an overview on the overall project and to implement counter-measures in case of project plan deviations.

Future Work will include (a) the integration of additional workflows aligned with a more detailed engineering process to better understand and automate engineering processes in general, (b) refining the change management process with respect to identify the number of changes per signal, i.e., detecting multiple changes per signal, and (c) an ongoing observation of product metrics with respect to better understand automation systems projects based on real-world project observations. In addition, a more sophisticated workflow component will allow a more flexible engineering process definition and implementation, and thus would make a large step towards agile signal change management in the automation systems engineering domain. Furthermore, we will focus on the introduction and adaptation of agile processes from software engineering to the automation systems engineering domain.

Acknowledgements. This work has been supported by the Christian Doppler Forschungsgesellschaft and the BMWFJ, Austria. We want to thank the domain experts at our industry partners for providing their expertise in discussing the processes described in this paper.

7 Literature

[1] Biffl S., Schatten A.: “A Platform for Service-Oriented Integration of Software Engineering Environments”, Proc. of the 8th Conf. on New Trends in Software Methodologies, Tools and Techniques (SoMeT), pp.75-92, 2009.

[2] Biffl S., Schatten A., Zoitl A.: “Integration of Heterogeneous Engineering Environments for the Automation Systems Lifecycle”, Proc. of the 5th IEEE International Conference on Industrial Informatics (IndIn), pp. 576-581, 2009.

[3] Biffl, S., Sunindyo, W.D., and Moser, T.: “Bridging Semantic Gaps Between Stakeholders in the Production Automation Domain with Ontology Areas”, Proc. of the 21st Int. Conf on SE and Knowledge Engineering (SEKE), Boston, USA, pp. 233-239, 2009.

[4] Chappell D.: “Enterprise Service Bus”. O'Reilly Media, 2004.

[5] GAMP 5: Good Automated Manufacturing Practice. International Society for Pharmaceutical Engineering (ISPE), 2008.

[6] Medeia-Consortium: “Medeia: Requirements Analysis and Technology Review”, Medeia Consortium, 2008.

[7] Moser T., Biffl S.: “Semantic Tool Interoperability for Engineering Manufacturing Systems”, Proc. of the 15th IEEE International Conf. on Emerging Technologies and Factory Automation (ETFA), Bilbao, Spain, 2010.

[8] Moser T.: “Semantic Integration of Engineering Environments Using an Engineering Knowledge Base”. Vienna University of Technology, Vienna, Austria, 2009.

[9] Schäfer, W., Wehrheim, H.: “The Challenges of Building Advanced Mechatronic Systems”, Proc. of ICSE, Future of Software Engineering, Washington DC, pp. 72-84, 2007.

[10] Schindele, A.: „logi.CAD Anwendungen in Wasserkraftanlagen“, Presentation at logi.cals Powerdays, Monheim, Germany, 2009.

[11] Sommerville I: “Software Engineering”, 8th Edition, Addison Wesley, 2007.

[12] Sunindyo W.D., Moser T., Winkler D., Biffl S.: “A Process Model Discovery Approach for Enabling Model Interoperability in Signal Engineering”, 1st Workshop on Model Driven Interoperability (MDI), Oslo, Norway, 05.10.2010.

[13] van der Aalst, W. M. P., “Business Alignment: Using Process Mining as a Tool for Delta Analysis and Conformance Testing,” RE Journal, vol. 10, no. 3, pp. 198-211, 2005.

Author CVs

Dietmar Winkler

Dietmar Winkler received an MS in computer science from TU Vienna, Austria, in 2003. He worked as a guest researcher at the Czech Technical University, Department of Cybernetics in 2007 and received a PhD research scholarship at Fraunhofer IESE in Kaiserslautern, Germany, in 2008. Since 2010 he is researcher for quality management and software process improvement in the Christian Doppler Laboratory for “Software Engineering Integration for Flexible Automation Systems” (CDL-Flex). Moreover, he works as software engineering and process management consultant. His research interests include Software Engineering, Engineering Processes, Quality Management, and Empirical Software Engineering.

Thomas Moser

Thomas Moser is post-doctoral researcher of software and knowledge engineering at the Institute of Software Technology and Interactive Systems, TU Vienna, Austria. He received his MSc and PhD degrees in business informatics from TU Vienna in 2006 and 2010, respectively, and works as a post-doc researcher in the research area "Semantic Integration" in the Christian Doppler Laboratory “CDL-Flex” since 2010. His research areas include data modeling and semantic web technologies for modeling engineering knowledge and the connection of design-time and runtime engineering processes, as well as practical and business applications of semantic web technologies.

Richard Mordinyi

Richard Mordinyi received an MS and PhD in computer science from TU Vienna and works as a post-doctoral researcher in the research area “technical integration and security” in the

Christian Doppler Laboratory “CDL-Flex”. His main research interests are secure coordination strategies in complex distributed systems, agile software architectures, and model-driven configuration of technical platforms.

Wikan Danar Sunindyo

Wikan Danar Sunindyo received his master degree in computational logic at the Dresden University of Technology, Germany, and works as a PhD researcher at TU Vienna in the research area “Complex Systems” since 2008. His main research areas include process analysis, data warehousing and semantic web technologies to better integrate heterogeneous engineering environments.

Stefan Biffl

Stefan Biffl is an associate professor of software engineering at the Institute of Software Technology and Interactive Systems, TU Vienna. He received MS and PhD degrees in computer science from TU Vienna and an MS degree in social and economic sciences from the University of Vienna in 2001. He received an Erwin-Schrödinger research scholarship and spent one year as researcher at the Fraunhofer IESE, focusing on quality management and empirical software engineering. Also, in 2001 he received the Habilitation degree Venia Docendi in Praktische Informatik for his work on empirical software engineering in project management and quality management. In 2006 he worked as guest researcher at Czech Technical University, Department of Cybernetics. Since 2010 Stefan Biffl is the head of the Christian Doppler Laboratory “CDL-Flex”.

Enhancing Productivity Through Products And Knowledge Reuse

Maria-Isabel Sanchez-Segura, Fuensanta Medina-Domínguez, Arturo Mora-Soto, Antonio Amescua

Abstract

One of the symptoms of project failure is the low efficiency of use of software process assets and the quality of products developed. This paper describes a reuse-based solution to the existing problems and limitations of representing knowledge, experience of processes and products generated during the development of a software project. This solution has been successfully validated in real projects.

Keywords

Knowledge reuse, product patterns, software process improvement

1 Introduction

Software is present in almost every aspect of our daily lives, which makes software development very important. However, software projects do not obtain the desired results. A Standish report [1] lists several reasons why more than 66% of software projects fail. Of these, we have focused on; Insufficient use of software processes, insufficient use of software engineering techniques, existing problems and limitations to represent knowledge, experience of processes and products generated during a software project.

If organizations do not solve these problems, the required development time and budget of their software projects will continuously and unnecessarily increase.

The authors believe that, by solving the existing problems and limitations of representing knowledge and experience in software project development, the other two other reasons for failure will be solved, i.e. usability of software processes will improve and the use of software engineering techniques will increase. In this paper, we describe a practical solution to represent software processes and products knowledge to make software engineering best practices and processes more usable for software organizations. This improvement will be reflected in the efficiency of use of the software products developed, and achieved through the use and reuse of organization business assets, represented as “product patterns” (a concept defined by the author), thereby minimizing development time which does not affect the quality of the product developed.

In summary, we propose the concept of product pattern to achieve the following goals:

- Goal 1: Improve efficiency of use of Software processes.
- Goal 2: Improve Software products quality parameters.

As a starting point, we proposed the following hypothesis:

Using a mechanism for software processes reuse and knowledge encapsulation, it is possible to improve the efficiency of use of software processes, the quality of software products obtained and, as a result, reduces the rework time of software products.

There are several reuse solutions that allow knowledge representation [2, 3, 4], but none represents knowledge and experience of processes and project assets that can be used throughout the phases of the software development lifecycle. Patterns are an extended solution for reuse. However, as they do not share a common representational language, there is no interoperability among them. We proposed a new concept, *product patterns*, to promote reuse in the software engineering field. Product patterns are product-oriented so, on the one hand, they can be used for any methodology where the product, which represents each product pattern, is needed. On the other hand, they can be used in any phase during the software development lifecycle and share a common representational language.

In short, this is a practical solution for dissemination, use, and reuse of business process assets based on processes, projects and knowledge management to improve the efficiency of use of processes in projects, the quality of the products developed as well as the rework time during the software lifecycle.

This paper is structured as follows: Section 2 summarizes the most relevant works, Section 3 describes the solution proposed by the authors, Section 4 shows the validation, and finally Section 5 presents the conclusions.

2 Related Works

This work focuses on process assets reuse to improve the efficiency of software processes and the quality of the products developed. Thus, we focus on reuse (patterns) and software repositories because software reuse improves productivity (directly related to the efficiency of use of teams) and must be supported by software repositories.

Reuse identifies artifacts and mechanisms to reuse the knowledge of expert engineers, processes and software projects assets. The artifact needed has to provide a practical solution to a problem in a context and with a set of forces. The main characteristic is that it must be for reuse. This artifact exists in software engineering and is called pattern.

Patterns, which were defined by Alexander [5], originated in the field of architecture and were adopted by the software engineering discipline where they represent solutions to the problems that emerged in software project development. The key is that the pattern is a generalization or reuse abstraction that can be used as a starting point for future solutions. This is why patterns fit well into this work since they are oriented to facilitating the diffusion, use and reuse of process assets. Consequently, patterns can be used as artifacts to gather the knowledge of software engineering best practices as well as the tacit knowledge of the organizations.

In the last few years, the number of works, studies and researches into patterns has increased exponentially. Patterns are now being used for many solutions in software engineering. It must be stated that there is no unique and precise classification of patterns. A complete classification can be found at [6]. As a summary we can say that existing patterns: Reference architectural patterns [7], Architectural Pattern [7, 8], Analysis patterns [9], Design Patterns [10], Process patterns [11, 12], Software Process Improvement patterns [13], Software configuration management patterns [14], Organizational Patterns [15], have the same deficiencies:

- Lack of portability. These patterns can only be applied in specific phases of software development (except for process patterns where they can be used for the whole lifecycle) but they only work with the methodology, process etc, they were intended for.
- Lack of uniformity in the description of different kinds of patterns. In general, the problem with existing patterns is that they are written in different languages with different formats, so they cannot be combined to develop an entire project.
- Lack of implementation tools and feedback mechanisms. The main deficiency of these patterns is the absence of tools to support their implementation. Without tools, patterns cannot be applied efficiently and the experiences obtained from their application in projects cannot be gathered and reused. Without feedback mechanisms, the knowledge gathered in patterns cannot be transformed into innovation.

Product pattern is intended to gather knowledge about producing a software product that can be used for any process, methodology, etc. because it is a product and not a process or activity approach. As a result, it is more portable and interoperable than the existing patterns.

3 Collaborative Framework Definition

Our solution is based on process assets reuse, through product patterns, to improve the efficiency of use of software processes and the quality of the products developed.

In this section we describe the concept product pattern, which encapsulates the knowledge of software products to be reused; the feedback and creation of new patterns mechanisms; collaboration mechanisms; product patterns language; meta-model representation and the product pattern catalog.

3.1 Product patterns and feedback mechanisms

The concept “product pattern” is a new term that comes from the Alexandrian Patterns where a pattern can be described as “A recurring solution to a common problem in a given context and system of forces” [5].

We selected patterns mainly for two reasons:

1. We are using an approach based on the reuse of process assets and patterns that, from the inception, were intended for reuse.

2. The environment in which a software product is developed is similar to that of using a pattern. A software product is developed in a **context** to solve a **problem** and entailing a set of **forces**. The terms in bold are the key elements of a pattern description.

Product pattern is described in terms of a set of fields widely described in [16, 17], a set of developed product patterns can be reached at http://productpatterns.sel.inf.uc3m.es.
Not all the fields in the product pattern are used in the same way. Pattern fields, can be classified into three main sets: Name, context, problem and forces, are used to decide whether or not a pattern is suitable for a Project activity execution. Fields solution, entries, exits, and collaboration are used to develop a project activity. Fields Lessons Learned, Templates, Examples and information resources can be used as support for the execution of the pattern and also to gather the knowledge acquired in the execution of a Project activity.
When the product patterns are instantiated and executed in a software project, the knowledge acquired can provide feedback for the following fields of the product patterns: Lessons Learned, Templates and Examples.
However, if there are new heuristics, i.e. new ways to apply the original product pattern because a new restriction has arisen or the original context or problem has changed, these heuristics promote the creation of a new product pattern. So, a new product pattern is created when a new heuristic is added to the Forces, Context and Problem fields.

The feedback and creation of new patterns mechanisms are very innovative and are as important as the recovery and search mechanisms because the knowledge evolution chain is not complete if all four mechanisms are not taken into account. Thanks to the search and recovery mechanisms, knowledge can be reused; and thanks to the feedback and creation of new patterns mechanisms, knowledge can be transformed into innovation.

4 Using product patterns in projects: impact quantification

This experiment was designed to validate product patterns before creating or using them in a software application. If we had incorporated two variables, product patterns and the software tool, at the same time, we would not have been able to determine whether the benefits of using product patterns really comes from its definition and use in software projects or the software tool in which product patterns are incorporated. To validate this, ten software projects were developed. Six projects P1, P2, P3, P4, P5 y P6 were developed at Carlos III University in Spain and four, P7, P8, P9 y P10 at SIAS and Alonso Software companies in Mexico. On average, all the projects lasted six months and everyone involved in this validation had between 2 and 4 years' experience. Three software engineers with different roles (project manager, analyst, designer and programmer) were involved in each project. The projects were developed in two phases (Figure 1), where *SEn*, is the acronym for each software engineer that participated in the experiment:

Phase I: Ten projects were developed without product patterns. The stakeholders developed the project exactly as they had been working to date.

Phase II: The same ten projects were developed using the product patterns. In this phase, the project activities were developed using the product patterns catalog implemented as a wiki (http://productpatterns.sel.inf.uc3m.es). The project manager performed the Gantt chart of the project and he or she linked the most suitable product pattern included in the wiki to these activities based on the problem, forces and context of the activity. In this way, each role of the project knew what they had to do and how. In this phase, the development teams were different from those of phase I, so they did not have any previous experience of the project to be developed. Software engineers that participated on the development of the projects using product patterns were given a 3-hour training course on product patterns and the wiki usage.

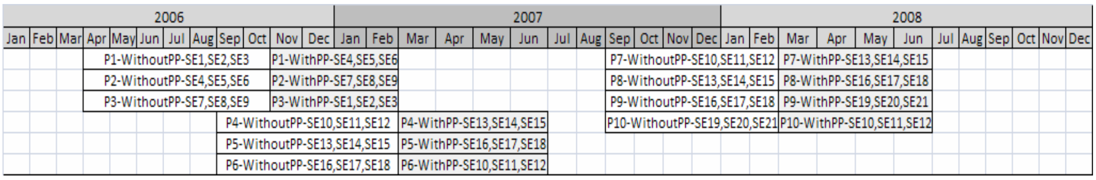


Figure 1: Phases of the project

The activities for this validation were: Use Cases diagram, Class Diagram, Sequence Diagram.

This experiment will validate the following goals:

- **Goal 1:** Improve efficiency of use of software models development.
- **Goal 2:** Improve a set of software models quality parameters.

In this section, sample 1 (10 projects, each having three software engineers) represents the projects which were developed without product patterns. And Sample 2 (10 projects, each having three software engineers) represents the projects which were developed using the product patterns catalog implemented as a wiki (http://productpatterns.sel.inf.uc3m.es).

In order to validate the impact of using product patterns wiki, the following parameters were analyzed: a) Time to develop each product analyzed, b) Volatility of requirements c) Classes and Methods not identified in the first version. Next, we focus on each parameter, analyzing the reason for its selection and its impact.

a) Efficiency of use of each model analyzed (Goal 1)

We compared the time the software engineers spent in developing the models: use cases diagrams, class diagrams and dynamic diagrams (sequence diagrams), recognized as the main development models is software development [8, 9, 18, 19].

We analyzed the models individually. An iterative development was chosen, so each project was developed in three iterations. A T-test and the representation of the Interval-plot were used.

a.1) Model I: Use Case Diagram: Efficiency of use

The time taken to develop use cases diagram for each iteration was stored and the analysis of the data obtained is shown in Figure 2. A T-test and the representation of the Interval-plot were used.

H0-ItX: the time spent on ItX use case diagram development for both samples is the same.

ItX WithoutPP: time taken (in hours) to perform use cases diagram in iteration X in projects where the product patterns were not used (sample 1).

ItX WithPP: time (in hours) to perform use cases diagram in iteration X in projects where the product where the product patterns were used (sampled 2).

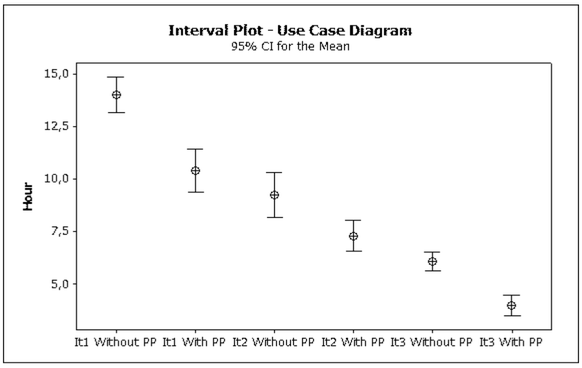


Figure 2: Interval Plot – Use Case Diagram development time

With p-values of 0,000 for It1; 0.004 for It2 and 0.000 for It3 lower than 0,05 we can reject H0-It1, H0-

It2 and H0-It3 respectively, so the use case diagram development time for both samples is different for each iteration. And as Figure 2 shows, we can affirm that the time taken to develop use cases diagram in projects where the product patterns were used is lower than in projects where they were not used.

It can be observed that the time spent in It1, It2 and It3 decreases from It1 to It3; the reason is two-fold:

- For the iterations in general, the uses cases were prioritized according to the criteria of allocating the most complex use cases in the first iterations, so the most complex functionalities are described in the first iterations.
- For iterations in projects where the product patterns were used, the time decreased in each iteration because software engineers learned about the product patterns each time they used them. This behavior was observed in class and sequence diagrams development time.

a.2) Model II: Class Diagram: Efficiency of use

The time taken to develop class diagram for each iteration was stored and the analysis of the data obtained is shown in Figure 3.

H0-ItX: the time spent on ItX classes diagram development for both samples is same.

ItX WithoutPP: time taken (in hours) to perform class diagram in iteration X in projects where the product patterns were not used (sample 1).

ItX WithPP: time taken (in hours) to perform class diagram in iteration X in a projects where the product patterns were used (sample 2).

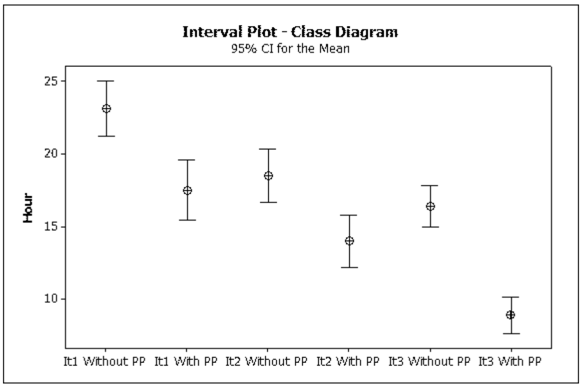


Figure 3: Interval Plot –Class Diagram

With p-values of 0,000 for It1; 0.001 for It2 and 0.000 for It3 lower than 0,05, we can reject H0-It1, H0-It2 and H0It3 respectively, so the class diagram development time for both samples is different. And as Figure 3 shows, we can affirm that the time taken to time to develop class diagrams in projects where the product patterns were used is lower than in projects where they were not used. The software engineers that participated in this experiment affirmed that one of the reasons for the improved productivity because the product patterns wiki provided them immediately with the information and knowledge needed to develop their models.

a.3) Model III: Sequence Diagram: Efficiency of use

The time taken to develop sequence diagram for each iteration was stored and the analysis of the data obtained is shown in Figure 4.

H0-ItX: the time spent on ItX sequence diagram development for both samples is same.

ItX WithoutPP: time taken (in hours) to perform sequence diagram in iteration 1 in projects where the product patterns were not used (sample 1).

ItX WithPP: time taken (in hours) to perform sequence diagrams in iteration 1 in project where the product patterns were used (sample 2)

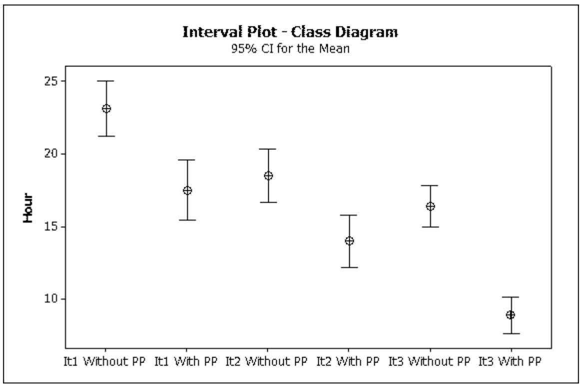


Figure 4: Interval Plot –Class Diagram

In iteration 1, the p-value is 0.064 so we cannot reject H0-It1. It means that in the first iteration the sequence diagram development time for both samples is similar; there is no difference between the development of the sequence diagram without using product pattern and with product pattern. The sequence diagram product pattern was improved after developing the software project's first iteration because the software engineers experienced some difficulties understanding this pattern specification.

In the second and third iterations, the p-values are lower than 0,05, we can reject H0-It2 and H0-It3 for these, so the sequence diagram development time for both samples is different. We can affirm that the time taken to time to develop sequence diagrams in projects where the product patterns were used is lower than in projects where they were not used, and the time even decreased from iteration two to three.

b) Non identified classes and methods in the class diagram in the first version of the class model (Goal 2)

Another parameter that we wanted to analyze was the number of non-identified classes and methods in the first version of the development of class diagrams and sequence diagrams. It is very important because, time is critical and the better the models developed, the better the paradigm performs. A T-test and the representation of the Interval-plot It were used.

b.1) Classes not identified in the class diagram and sequence diagrams in the first version of these models.

The number of non-identified classes for each iteration in the first version of the class diagram was stored and the analysis of the data obtained is shown in Figure 6.

H0-ItX: the number of non-identified classes in the first version of the It2 class diagram for both samples is the same.

ItX WithoutPP: number of non-identified classes in the first version of class model in iteration 1 in projects where the product patterns were not used (sample 1).

ItX WithPP: number of non-identified classes in the first version of class model in iteration 1 in projects where the product patterns were used (sample 2).

The p-value of iteration 1 is 0,235, bigger than 0,05, so we cannot reject H0-It1. This means that the number of non-identified classes in the first iteration of projects where product patterns were used is no different from the number of non-identified classes in the first iteration of the project where product patterns were not used. But in iterations 2 and 3, p-values are 0,002 and 0,000 respectively, so we can reject H0-It2 and H0-It3. This means that in these two iterations the number of non-identified classes is different in project where product patterns were used from those in projects were they were not used.

The difference between iteration 1 and the other two might be related to the fact that the product pattern sequence diagram had to be improved after the first iteration because of some problems with understandability; these were identified and addressed. So these corrections explain the above mentioned behavior for iterations 2 and 3.

b.2) Methods not identified in the class diagram and sequence diagrams in the first version of these models.

The number of methods non-identified for each iteration in the first version of the class diagram was stored and the analysis of the data obtained is shown in Figure 7.

H0-ItX: the number of non-identified methods in the first version of the It2 class diagram for both samples is the same.

ItX WithoutPP: number of non-identified methods in the first version of class model in iteration X in projects where the product patterns were not used (sample 1).

ItX WithPP: number of non-identified methods in the first version of class model in iteration X in projects where the product patterns were used (sample 2).

In the number of methods not identified something similar to what happened with classes not identified occurs. P-value for It1 is 0,097, greater than 0,05 so H0-It1 cannot be rejected, so the number of not identified methods in iteration 1 for projects were product patterns were used is not different from projects where they were not used.

P-value for It2 and It3 are 0.001, it is lower than 0,05 so we can reject H0-It2 and H0-It3. This means that in these two iterations the number of methods not identified is different in projects where product patterns were used from projects where they were not used. As can be seen in Figure 7, the number of methods not identified in the first version of class and sequence diagrams is lower when product patterns are used to develop these models.

The difference among iteration 1 and the other two, might be related to the fact that the product pattern sequence diagram had to be improved after the first iteration because of some problems with understandability; these were identified and addressed. So these corrections explain the above mentioned behavior for iterations2 and 3.

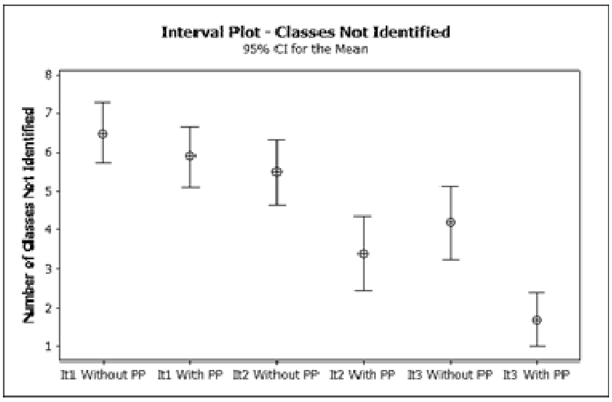


Figure 6: Interval Plot – Non Identified Classes

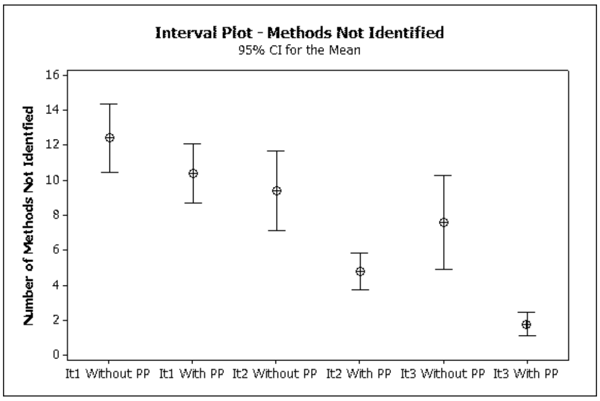


Figure 7: Interval Plot –Methods Not Identified

5 Conclusions and Future Research

Our solution is a product patterns language which has the following features:
Product patterns are portable, meaning they can be used in different methodologies. In this respect, we promoted the multi-model approach to software development, one of the fields in which software engineering must focus in the years to come.
We dealt with lack of uniformity in the description of different kinds of patterns. Product patterns have a formal description and can be used for any phase during the software lifecycle.
We implemented a wiki where product patterns are published and we defined the feedback mechanisms, so product patterns can be transformed from knowledge to innovation, an important asset for business.

After validating the solution described in this paper we can conclude that, by using patterns as a knowledge encapsulation artefact, the time required to perform an activity is reduced through the support of product patterns.
In relation to some aspects of quality analysed, we can conclude that class and sequence models are developed with fewer mistakes in the early stages of their development, which will impact in the number of reviews required for each model until the required quality is achieved.
After demonstrating the usefulness of product patterns in software projects development, we are focusing on the development of a collaborative tool that supports the creation and feedback mechanisms of product patterns.

6 Acknowledgements

This work has been partially funded by the Spanish Ministry of Science and Technology through the TIN2009-10700 project and the Spanish Ministry of Industry through projects PPT-430000-2008-54, PPT-430000-2009-0012 and PPT-430000-2009-0013.

7 Literature

1 Standish Group. *Third Quarter Research Report*. The Standish Group International, Inc., West Yarmouth, M A, (2004).

2 Moe NB, Dyba, T. The Use of an Electronic Process Guide in a Medium-sized Software Development Company. *Software Process Improvement and Practice,* 11: 21-34, (2006).

3 Antunes B, Seco N, Gomes P, 2007. Knowledge Management using Semantic Web Technologies: An Application in Software Development. *The K-CAP Proceeding of the Poster Session of the 4th International Conference on Knowledge Capture* (K-CAP 2007), ACM, Whistler, Canada, (2007).

4 Brunner J, Ma L, Wang C, Zhang L, Wolfson DC, Pan Y, Srinivas K. Explorations in the Use of Semantic Web Technologies for Product Information Management. *Proceedings of the 16th international conference on World Wide Web* (2007), Banff, Alberta, Canada, 747-756, (2007).

5 Alexander C. *The Timeless Way of Building*. Oxford: Oxford University Press, (1979).

6 Medina-Dominguez F, Sanchez-Segura M, Mora-Soto A, Amescua A. Patterns in the field of software engineering. *Encyclopedia of Information Science and Technology 2nd. Edition*. Idea Group Publishers, (2008).

7 Buschmann F, Meunier R, Rohnert H, Sommerlad P, Stal M. *Pattern Oriented Software Architecture. A system of Patterns*. New York: John Wiley & Sons, (1996).

8 Shaw M, Garlan D. Software Architecture: Perspectives on an Emerging Discipline. Prentice Hall, Upper Saddle River, (1996).

9 Fowler M. Analysis Patterns Reusable Object Models. Addison Wesley, (1997).

10 Gamma E, Helm R, Johnson R, Vlissides J. *Design Patterns: Elements of Reusable Object-Oriented Software.* Addison Wesley, (1995).

11 Ambler S. Process Patterns: Building large-scale systems using object technology. Cambridge University Press, (1998).

12 Garson E. A Whirlwind Introduction to Process Patterns. Dunstan Thomas Consulting, (2006).

13 Appleton B. Patterns for conducting process improvement. *Proceeding on conference Pattern Languages on Programs*. Monticello, Illinois, USA, (1997).

14 Berczuk S, Appleton B. Software configuration management. Effective teamwork. Practical integration. Addison-Wesley, (2003).

15 Ambler S. More Process Patterns: Delivering Large-Scale Systems Using Object Technology. Cambridge University Press, (1999).

16 Amescua A, Garcia J, Sanchez-Segura M, Medina-Dominguez F. A pattern-based solution to bridge the gap between theory and practice in using process models. *International Software Process Workshop and International Workshop on Software Process Simulation and Modeling*, SPW/ProSim 2006. Lecture Notes in Computer Science 3966 Springer, Shanghai, China. 97-104, (2006).

17 Medina-Dominguez F, Sanchez-Segura M, Mora-Soto A, Amescua A. Extending Microsoft Team Foundation Server Architecture to Support Collaborative Product Patterns. Proceeding International Conference on Software Process, ICSP 2007. Lecture Notes in Computer Science 4470 Springer 2007. Minneapolis, MN, USA, 1-11, (2007).

18 Purao S, Storey V, Han T. Improving Analysis Pattern Reuse in Conceptual Design: Augmenting Automated Processes with Supervised Learning. *Information Systems Research*; 14(3): 269-290 (2003).

19 Kodaganallur V, Shim S. Analysis patterns: A taxonomy and its implications. *Information System Management*; 23(3): 52-61, (2006).

8 Author CVs

Dr. Maria-Isabel Sanchez-Segura

Dr. Maria-Isabel Sanchez-Segura 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. Maria-Isabel holds a BSc in Computer Science (1997), an MSc in Software Engineering (1999), and a Ph.D. in Computer Science (2001) from the Universidad Politécnica 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. She is also the author of more than thirty papers presented at the most relevant conferences in the field of software process improvement, software engineering and Virtual Environments.
Maria-Isabel was one of the co-instructors, in conjunction with Carnegie Mellon and Politécnica of Madrid researchers, at tutorials held at the ACM CHI conference, Vienna April 2004 and IEEE ICSE Conference, Edinburgh May 2004. She is also a reviewer for some conferences such as ICSE2007, ICSE 2006, RE2006, SEPG2007, SEKE 2001-2006, etc., and she reviews papers for international journals such as IEEE software, Journal of Systems and Software, Multimedia tools and applications, etc. Maria-Isabel is a member of the program committee of many conferences related to software process in the field of software engineering.

Dr. Fuensanta Medina-Dominguez

Dr. Fuensanta Medina-Dominguez is assistant professor at Carlos III University of Madrid. She has been working in the software engineer field since 2000 and she is a faculty member of the Computer Science Department in the Carlos III Technical University of Madrid since 2004.
Her research interests include software engineering and software process improvement focusing on technology transfer in organizations through the use of new trends in computer supported collaborative work technology. Fuensanta Medina-Dominguez holds a BSc in Computer Science (2004) and a Ph.D. in Computer Science (2010) from the Universidad Carlos III of Madrid.

Arturo Mora-Soto

Arturo Mora-Soto is assistant professor at Carlos III University of Madrid and he belongs to the Software Engineering Lab. His research involves collaborative Web environments, new e-learning trends and Six Sigma applications in the software engineering realm especially in process improvement. He has been working as software engineer as well as a university teacher in Mexico since 2002. He holds a BSc in Computer Science from Celaya Institute of Technology (ITC), Mexico, an MSc in e-commerce as well as an MSc in Computer Science and Technology from Carlos III University of Madrid. Nowadays he is working in his doctoral thesis dealing with software process usability through innovative knowledge transference techniques and collaborative learning.

Dr. Antonio de Amescua

Dr. 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, and Software Project Management.

Rule-Based Requirements Management Methodology

Badari Kotejoshyer, Birendra Singh, Suhas Tittibha

Honeywell Technology Solutions
151/1, Doraisanipalya,
Bangalore – 560076, India
badari.kotejoshyer@honeywell.com, birendra.singh@honeywell.com, suhas.tittibha@honeywell.com

Abstract

This paper proposes a methodology to identify and prevent/control non-compliances and conflicts arising out of requirement attribute values during requirements management. It is intended to provide a cost effective approach for managing non-compliances in projects that employ the Requirements Based Engineering (RBE) approach. The method involves continuous checking and monitoring of requirement attribute values using a rule based engine concept. This methodology can serve as an enhanced feature for any commercial requirement management application software available today (such as IBM DOORS©, Seimens TcSE© etc.).

The methodology offers the following advantages:

1) Enables standardization in the development and management of requirement modules.
2) Decreases development and maintenance cost for requirement modules by reducing the effort needed for audits, compliance checks, rework etc.

Keywords:

Requirements Based Engineering, Requirements Management, Requirement attributes, Requirements V&V, System-of-systems

1 Introduction

Requirements management is a critical activity in the development of large systems, especially in the field of aerospace and automotive engineering. It involves defining requirement attributes based on the system verification and validation (V&V) plan and, building the traceability for requirement modules based on the system hierarchy. The system hierarchy determines the way in which requirements are to be flowed down to different modules (e.g. system to subsystem to components). Figure 1 shows a typical requirement module with attributes defined for requirements management following a V&V plan.

Commercially available software application packages like IBM DOORS©, Seimens TcSE© etc. are specialized in managing requirements for large systems (or System-of-Systems). The packages offer an efficient way to create requirements modules with attributes and build traceability between different modules. Requirement traceability is established by linking the modules together (or flowdown). The packages also offer several features to perform requirement analysis such as filters, traceability wizards and impact wizards.

- Commonly seen attributes with associated values across projects are:
  - a) Object Type – Used to identify if the object text is a **Requirement** or not
  - b) Requirement Class – Identifies whether the requirement is a **direct flow down** (flowed down from a higher module, or **decomposed** (broken down) or a **derived** (created new) kind
  - c) Requirement Type – Belongs to category such as **functional**, **safety** and **performance**
  - d) Validation Method – By means such as engineering judgment, similarity, analysis and lab testing
  - e) Requirement Acceptance Status – Accepted or Not accepted
  - f) Requirement Means of Compliance – By design review, simulation, lab test and inspection
  - g) Requirement Verification – Could be textual description for the evidence.

Object ID	Object Text	Object Type	Requirement Class	Requirement Type	Validation Method	Requirement Acceptance Status	Requirement Means of Compliance	Requirement Verification
1001	The weight of the component shall not be more than 10 lbs	Requirement	Direct Flowdown	Performance	Similarity	Accepted	Review	
1002	The component shall be capable of withstanding an impact of ------	Requirement	Direct Flowdown	Performance	Analysis	Not Accepted		
1004	Diagram of the component	-						
1005	The component shall provide an indication of its current status to the user	Requirement	Derived	Functional	Engineering Judgement	Accepted	Testing	

Fig. 1. A Typical Requirement Module with Attributes

2 Current Scenario

During the development of large systems, several requirement modules containing many requirement attributes along with the links have to be managed. The population of these attribute values and creation of links are performed based on the project's requirements V&V plan. Due to the presence of multiple teams working on different modules, issues and non-compliances would arise by not adhering to the V&V plan in attribute value population or linking. This situation is further complicated by teams located across different geographies. Organizations tackle the challenge using the corrective philosophy through periodic checks, engineering audits and design reviews. These activities require significant amount of effort and resources to perform the checks and implement the corrections.

Figure 2 shows the available methods that could be used in the current commercial requirements management software. IBM DOORS© is considered as an example. It can be noted that both these methods need resources for accomplishing the checks. Also, effort is needed in the correction of the attribute values and links (traceability). Usage of scripting language offered by the commercial software is critical to the accomplishment for checks and audits. However, it is heavily programming oriented, and would need software development and maintenance expertise.

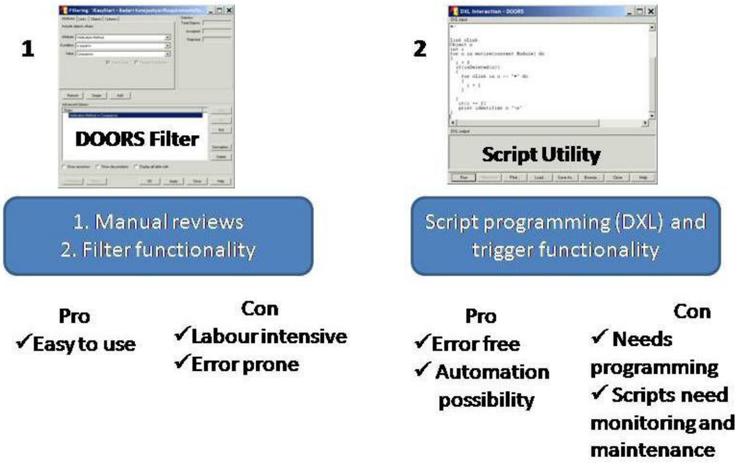


Fig. 2. Existing Methods for Checking Noncompliance

3 Rule-Based Requirements Management

The rule-based methodology identifies and manages the conflicts arising out of attribute values and links in a continuous manner. The attribute relationships defined for the project are captured as a "set of rules" in a rule based engine. Every requirement module in the project would be monitored by this rule based engine. The whole project would thereby be monitored for the rules defined.

The methodology can be better detailed with an example. The project group MYGROUP is developing an aircraft system based on certain requirements from the customer. The project MYGROUP splits the work into several work packages, which individually generate and link requirement modules as shown in Figure 3. Figure 3 is a fairly simplistic way described for the purpose of explanation. The RBE tool referred in the figure corresponds to requirements management application software. Every individual workpackage further flows down the requirements to different levels.

The requirement modules developed by the Workpackages need to adhere to the group's requirements management process. As an example, we define three attributes for a requirement statement:

1) Requirement type – Can have values as "Derived" or "Direct"
2) TBD Monitor – Can have values as "TBD" or "None"
3) Requirement status – Can have values as "Accepted" or "Open"

Figure 4 shows a sample module for System 1. Please note that the Object ID provides a unique tag to the requirement and the arrow head denotes linking to a higher level module.

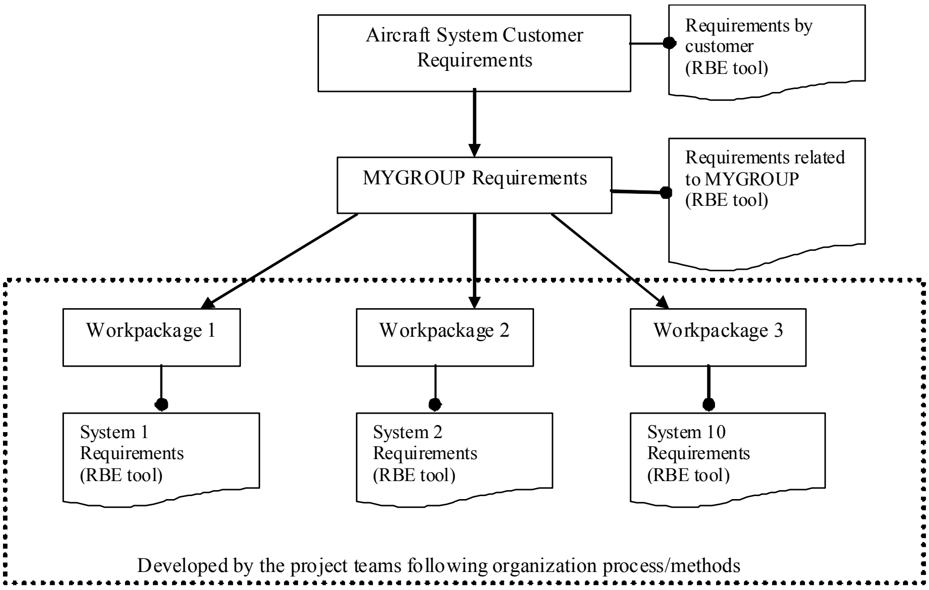


Fig. 3. Example – Simplified RBE Structure

Object ID	Requirement Statement		Reqt. Type	TBD Monitor	Reqt. Status
1	The system shall not weigh more than 14 pounds (TBD).		Direct	TBD	Accepted
2	The system shall have a fail safe mechanism and provide indication to the user of system failure.		Derived		Accepted

Fig. 4. Example – System 1 Requirement Module

A close review of the module reveals two non-compliances.

a) Object ID 1 contains TBD and the TBD Monitor is set. Hence the requirement status cannot be set to Accepted.

b) Object ID2 is a derived requirement and is linked to a higher level module. With regards to the V&V plan, a derived requirement must not have links to higher level modules (outlink).

It can be observed that such non-compliances can be identified using the existing methods described in Figure 2.

3.1. Incorporating Rule-Based Methodology

In this methodology every module would refer to a rule-based engine. The rule-based engine is developed by translating the requirement management process as a list of rules related to attribute values and links. For the example project, three rules can be created as:

Rule #1: If “Requirement Type” is “Derived”, then object should NOT have any “Outlink”

Rule #2: If “Requirement Statement” CONTAINS “TBD”, then “TBD Monitor” should CONTAIN value “TBD”

Rule #3: If “TBD Monitor” CONTAINS “TBD”, then “Requirement Status” should be set to “Open”

The rules described can be created and loaded in the rule-based engine. This rule engine is referred by every module continuously. When the engine is executed, it checks for rule conflicts, and provides an indication. It is flagged as shown in Figure 5 for easy identification next to the object ID. Upon moving the cursor (mouse-over)or by clicking on the symbols, the reason for conflict can be displayed. Additionally, it can also generate a report or provide indication in any useful way to catch attention to conflicts. Further some of the rules can prevent the user from setting attribute values that creates conflicts or non-compliances.

Object ID	Requirement Statement		Reqt. Type	TBD Monitor	Reqt. Status
1	The system shall not weigh more than 14 pounds (TBD).		Direct		Accepted
2	The system shall have a fail safe mechanism and provide indication to the user of system failure.		Derived		Accepted

Fig. 5. Flagged Noncompliances of System 1

3.2. Loading Rules into Rule-Based Engine

The rules described as an example in Section 3.1 can be created and loaded in the rule-based engine using a simple tool construct as described in Figure 5. Figure 5 is generalized for various sets of issues and non-compliances arising out during requirements management. It is not limited to just attribute values. Set of rules can be added as a Rule Set and saved for a project. Several different rules sets can be associated for a project and it would continuously monitor for all the requirement modules. Rules can be constructed for attribute values, traceability, attribute list, module views etc.

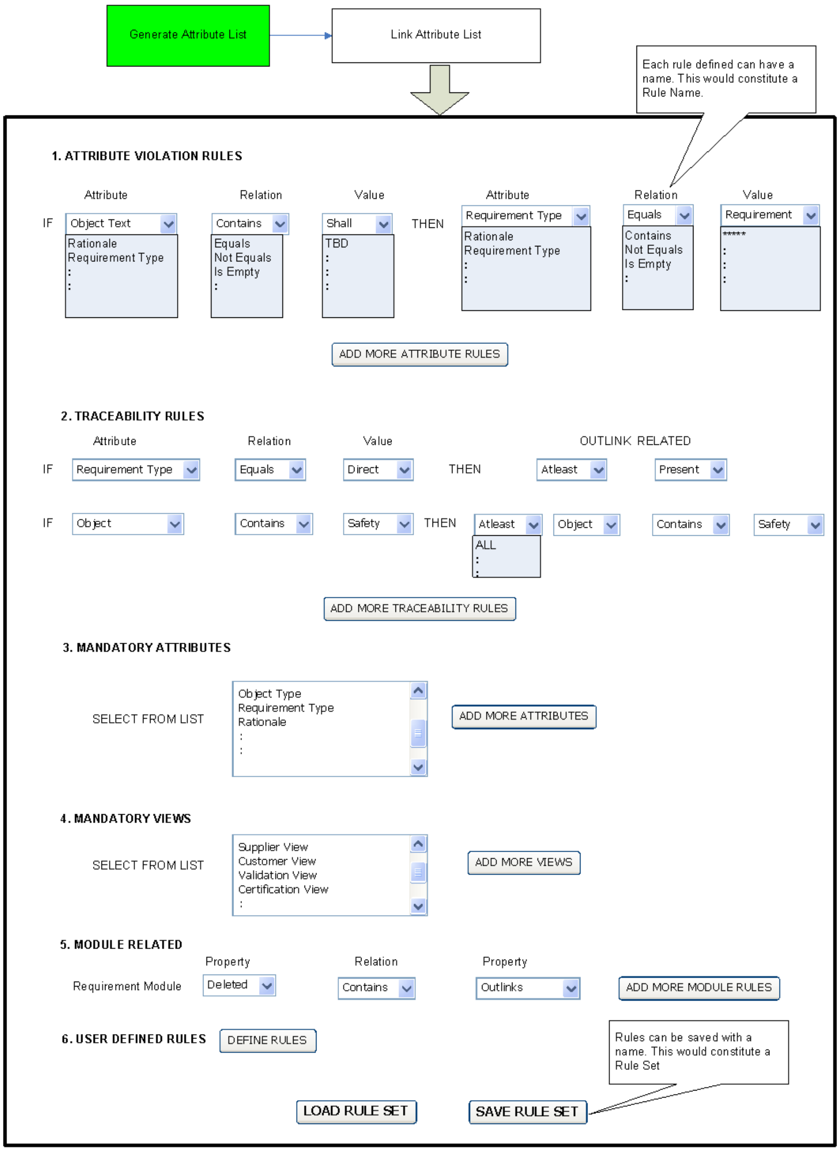


Fig. 6. Mechanism to Construct and Load Rules

3.3. Overall Implementation of Methodology

The flowchart shown in Figure 7 describes the implementation scheme for the rule-based approach. It is to be noted that the rule engine will be running as per the execution strategy described in Section 3.4.

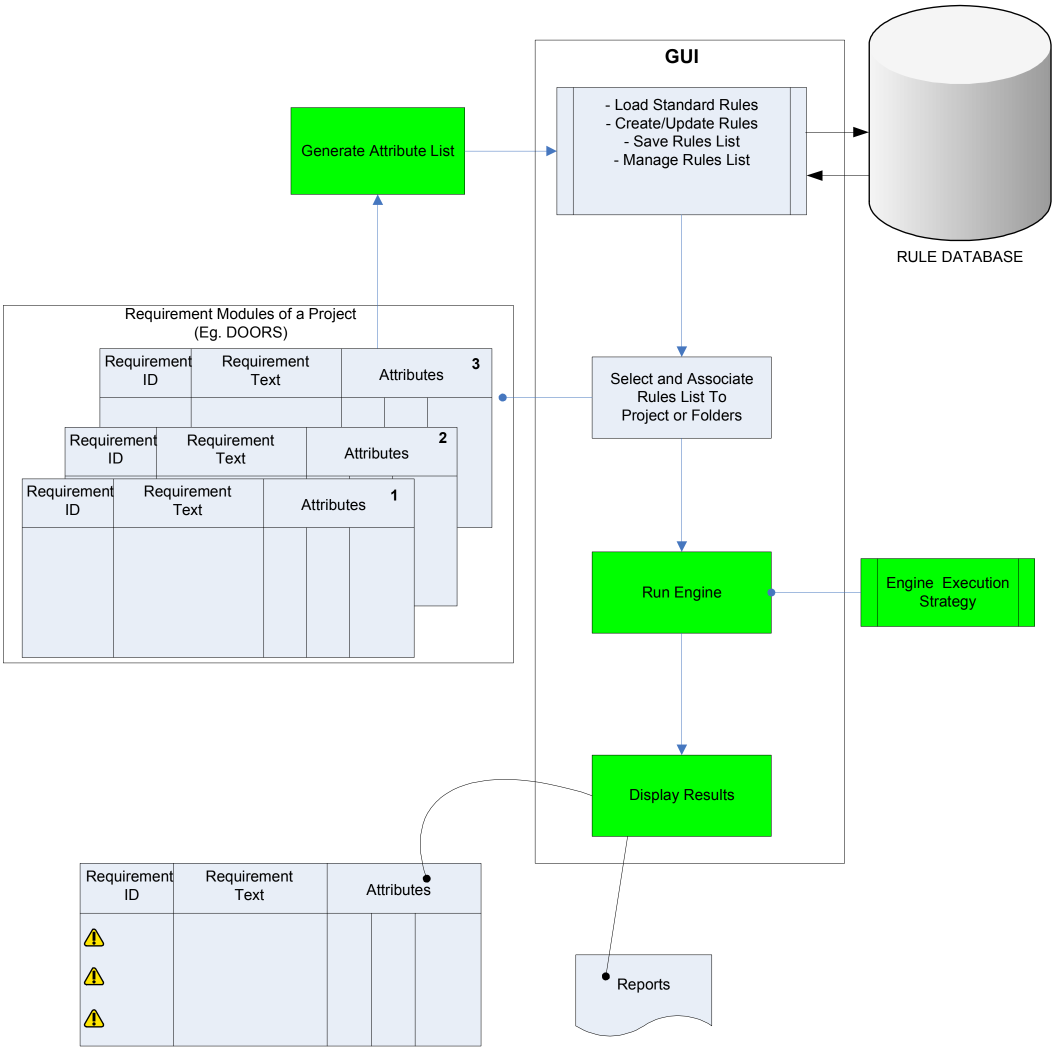


Fig. 7. Overall Implementation Steps

3.4. Rule-Engine Execution Strategy

The rule engine can be strategized to execute at important stages (or events). This would enable optimal usage and potential run problems that may adversely affect the execution time for the application.

The rule engine can be executed when:

- User requests the execution (On Demand)
- Module is saved (Event Based)
- Module is opened (Event Based)
- Important operations like Baselining, Analysis etc. are being executed.

The rule based engine is connected to a requirement project so that all the future modules created under the project are governed automatically by the rules.

4 Prototype Development

In order to demonstrate the feasibility of the proposed concept, a prototype is developed using the commercial requirements management package IBM DOORS©. It is accomplished utilizing the scripting language (DXL) and triggers functionality offered by the software. Please note that similar prototype can be developed for other requirements management packages also. The translation of this concept into a more effective and robust feature can be made possible by the commercial application software vendors following the implementation in Section 3.3.

Figure 8 shows the snapshot of a sample requirements module in DOORS. The DXL script with rules has been implemented using the triggers functionality. A violation in the rules will change the colour indicator from green to red and displayed as the column titled “RuleConflict”. The rule conflicts are checked on a continuous basis. The prototype demonstrates four rules and the violation indication can be seen in the column named “Rules_Violated”.

Rule1: If Object Type is marked as Requirement, Requirement Type cannot be empty.

Rule2: If Requirement Status is Accepted, Validation Method cannot be empty.

Rule 3: If TBD-TBC s not empty, Requirement Status cannot be accepted

Rule 4: If Requirement Type is Derived, outlinks should not be present.

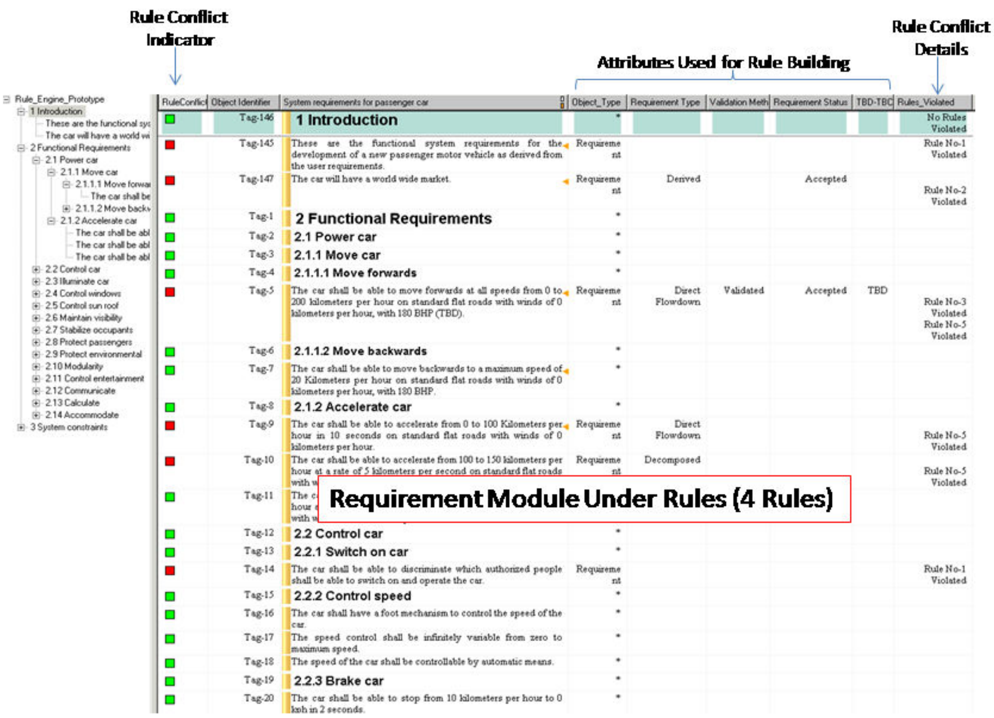


Fig. 8. Sample Requirement Module Under Rules

5 Cost and Effort Savings

The demonstrated concept provides a method to save the efforts and eventually save the costs involved due to review and rework of the requirement modules. The information provided in Table 1 considers real data from a live project executed. Specific details are omitted due the proprietary and confidentiality nature. The estimates presented are on a conservative level. It is to be noted that review and implementation hours take a significant time for adhering to the organization process on requirement change management (standard process management).

Table 1. Effort and Cost Saving Summary

Number of Requirements for a Module (Average)	930 requirement statements (system level) 285 requirement statements (component level)
Number of Requirement Attributes for a Module (Average)	20
Number of Violations Checked	22
Quality Checks Performed on a Single Module (Man-hours per month)	3 Hours
Review and Implementation of Changes (Man-hours per month)	16 Hours
Total Number of Modules Present	100 Requirement Modules
Approximate Effort Savings	1900 Hours Per Month
Approximate Cost Savings	1900 Hours x Resource Charge Rate

6 Conclusion

"Do it right the first time" works well with the aerospace industry and the proposed methodology promotes this philosophy. The methodology is preventive rather than being corrective by nature. The methodology also adds a productivity aspect to the project cost reduction as described in Section 5.

At present, large aerospace projects are being executed following the RBE process using commercial software packages. The tools currently do not provide a cost-effective way for checking non-compliances related to the requirement management process. The suggested methodology can:

i) Save development cost and improve the quality of requirement modules
ii) Ensure uniformity in adherence to the organization/group process and
iii) Improve schedule since there may not be elaborate audits or reviews for checking attribute related non-compliances.
iv) Serve as a V&V process ready reckoner by displaying warnings and suggestions

The solution provided is flexible, compatible and scalable from different perspectives like number of requirements, number of rules and relationships and non compliances. It could serve as a concept and guideline for the commercial software packages to provide enhanced features in requirements management area.

Coping with Multi-sourcing Decisions: A case study from Danske Bank

Jan Pries-Heje [1], Linda Olsen [2]
[1] Roskilde University, Denmark
[2] Danske Bank, Denmark

Abstract of Source IT Book

The book seeks to enhance our understanding of the interrelation between sourcing and innovation in information systems development. An understanding of the interrelation between sourcing and innovation is important in its own right, but we also aim to use this understanding to provide instruments for balancing sourcing and innovation in ways that match the needs of development projects. The book addresses two questions:

- What are the preconditions for optimal sourcing in relation to innovation capability?
- How can an organization be most innovative, while making optimal use of sourcing?

Succeeding in either sourcing or innovation is far from easy, and succeeding in both at the same time is a real challenge. Yet, practitioners at many managerial and decision-making levels are faced with demands to devise innovative solutions in projects that involve various sourcing arrangements. These practitioners need guidance on how to make strategic, tactical, and operational sourcing decisions. At the same time, researchers are in need of concepts, case descriptions, and tool assessments that enable a better understanding of the challenges involved in successfully balancing sourcing and innovation. Apart from being useful to research and practice, this book provides cases and tools that can be used in education. With respect to education, we consider the book particularly relevant to advanced programmes in information systems development, software process improvement, and project management.

Reference

Hertzum M., Jørgensen C. (eds.) (2011): Source IT. Balancing Sourcing and Innovation in Information Systems Development.

The SourceIT book is part of the conference handout.

Sourcing capability assessments

Morten Korsaa [1], Jørn Johansen [1]
[1] DELTA, Denmark

Abstract of Source IT Book

The book seeks to enhance our understanding of the interrelation between sourcing and innovation in information systems development. An understanding of the interrelation between sourcing and innovation is important in its own right, but we also aim to use this understanding to provide instruments for balancing sourcing and innovation in ways that match the needs of development projects. The book addresses two questions:

- What are the preconditions for optimal sourcing in relation to innovation capability?
- How can an organization be most innovative, while making optimal use of sourcing?

Succeeding in either sourcing or innovation is far from easy, and succeeding in both at the same time is a real challenge. Yet, practitioners at many managerial and decision-making levels are faced with demands to devise innovative solutions in projects that involve various sourcing arrangements. These practitioners need guidance on how to make strategic, tactical, and operational sourcing decisions. At the same time, researchers are in need of concepts, case descriptions, and tool assessments that enable a better understanding of the challenges involved in successfully balancing sourcing and innovation. Apart from being useful to research and practice, this book provides cases and tools that can be used in education. With respect to education, we consider the book particularly relevant to advanced programmes in information systems development, software process improvement, and project management.

Reference

Hertzum M., Jørgensen C. (eds.) (2011): Source IT. Balancing Sourcing and Innovation in Information Systems Development.

The SourceIT book is part of the conference handout.

The human being as one key element for holistic Software Process Improvement (SPI) -

The Human being is the most interlinked element in SPI

Alfons Unmuessig, Alfons.Unmuessig@t-online.de

Abstract

The paper aims to explain a new approach of software process improvements (SPI). The approach will not replace the existing methods, but will support them for SPI from another view. The approach is based on a comprehensive PhD paper about SPI and defect prevention from the author. Today there are various actions and constructive methods in software process improvements used. As there are a lot of different elements and subjects in the process of improvements involved - it is a complex process. The most involved elements and subjects are e.g. management, members of staff, work psychology, methods, organisations, customer, culture etc. The author's own experience and studies confirm that the human being is one of the most important elements in the process. The human being is much more involved in the process than considered in the daily work today. His work performance e.g. software process improvements depends on a lot of interlinked factors. This paper will use 12 important elements and their interlinked connection, direction and influence on other elements. The author selected these important elements from over 100 elements. A software tool is used to present and simulate the interrelationship to the other elements. The results can be used in all software process improvements / software development processes to support the existing SPI approaches and measures.

Keywords

Software process improvement, holistic, human being, interlinked elements, defect prevention, processes, methods, organisation, culture, SPI Manifesto 2010.

1 Introduction

Today's software process improvement techniques have made substantial progress over the last years in a contribution to better software quality, but this is not enough.

The key focus in the new approach is on Requirement Analysis and Specification of the software development process. Based on the results of the research of available sources and in the author's own experience, 50 to 70 % of all failures come about in these two first development phases (Masing 2007). At present, software is still prevalently compiled by humans. This means the key failure source for SW errors are human factors. The paper will present some interlinked aspects for Software Process Improvement (SPI).

2 Software Development Process

Process / Procedure Models

For the development of software, methods, process and procedure models are in use (see Fig. 1). The activities and results fall under different types of operations such as project management. Process models further also define the relevant roles, tools and methods (Wallmüller 2001). Figure 1 shows the SW Development process and one of the various process / procedure models.

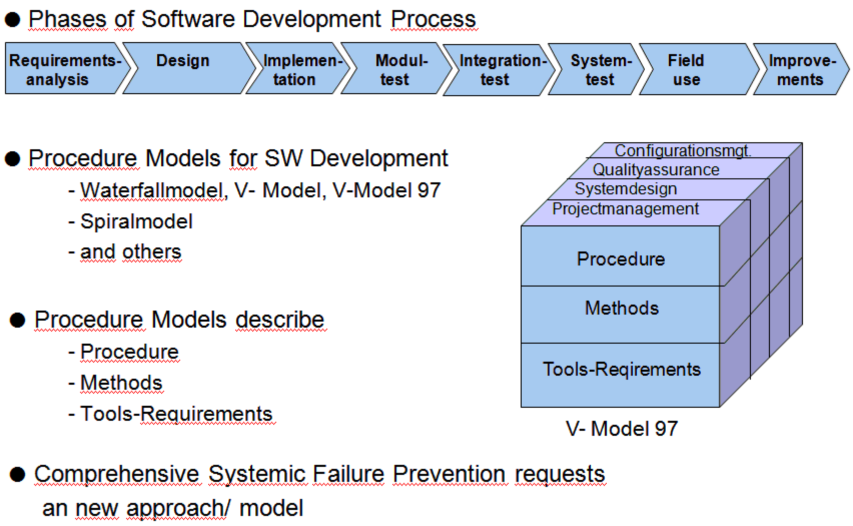


Figure. 1: Software Development Process
Source: Relating to [Hindel 2004, p. 18]

Process / procedure models outline the applicable organisational framework and the sequence within the software development process. As synonyms to procedure model according to (Masing 2007, p. 824), terms such as software lifecycle, phase model, project model or process model also apply, organisational framework includes the following aspects (Balzert 1998, p. 38):

- The partial actions and sequence thereof within the process sequence (project phases);
- Planned schedules;
- Required employee qualification;
- Applicable standards, references, directives, tools and techniques.

Different procedure models emerged out of the framework conditions. The differences between the procedure models mainly relate to the level of complexity, the number of phases and the sequence of steps within procedures. The complexity of an adequate procedure model depends on circumstances in each particular case.

3 Software Process Improvement

The following **analytical, constructive, organisational** and **psychological** actions and **software process improvement models** e.g. Quality and Capability Maturity Models e.g. CMM, are important Inputs for Software Process Improvement (Wallmüller 2001).

1. Analytical Quality Actions

Analytical quality actions are implemented e.g. in order to verify current quality levels. The examinations do not establish absolute certainty regarding conformity/non-conformity of a product. There are e.g. following analytical quality actions used: white box test, black box test and cause-effect analysis.

2. Constructive Quality Actions

Constructive quality actions mainly aim at defect prevention. For e.g. procedure/process models, methods, program language, quality gates, Risk management, Personal & Team Software process (PSP / TSP), FMEA & Reviews.

3. Organizational Quality Actions

Organizational actions include the design, implementation and maintenance of e.g. a corporate Quality Management System, Project Management System, team-building and others.

4. Psychological Prevention Actions/People Actions

Software development should not be considered a purely technical procedure only. As software development is carried by the human input (the development team), communication processes, leadership, quality culture and other variables this entails are key to the output quality.

Software Process Improvement Models

Process improvements may be implemented in a number of different ways such as under the classical PDCA (**P**lan, **D**o, **C**heck, **A**ct) cycle or by way of capability maturity models. An advantage to using capability maturity models is that they systematically impose successful practices on different models. Examples of capability maturity models include CMMI or SPICE (Hörmann, 2006, p. 5).

Application of the models and actions under the models promote process improvements and thus aid the failure prevention/minimisation effort. In order to be able to effect process improvements, first an inventory check is needed of the actual status. The inventory check takes the form of an assessment. Under the assessment, formal and actually practiced processes are examined separately and correlated with ideal values and criteria under the respective capability maturity model. The correlation output is a sc. action matrix that indicates the problem areas and processes found. In this way, process improvements may be initiated and implemented in an ideal manner. Fig. 2 shows the interdependencies between processes, process assessments, capability determination (assessment of capability maturity level) and process improvements on applying a capability maturity model such as SPICE or CMM(I).

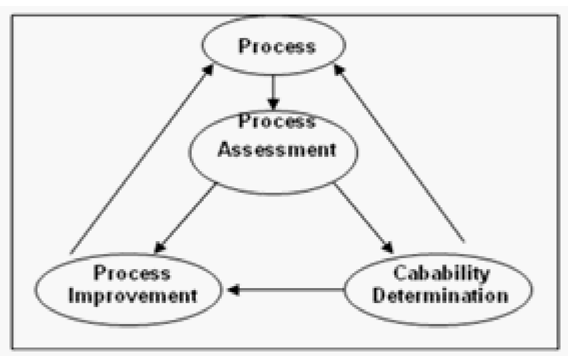


Figure 2: Software-Process Assessment with SPICE according to ISO 15504
Source: Petrasch 1998, p. 119

Additionally as output, the capability maturity level is determined. Most models differentiate between levels on a range from 1 to 5, 1 to 6 or 1 to 7. The higher the level within a model, the better the processes concerned are formally defined and practiced.

SPI are important for reducing failure and increasing quality. Late failure detection adds massively to costs. According to various analyses, failure elimination costs, as a variable, multiply by a factor of about five to ten for each phase in the sequence. This means that a failure generated in the initial specification phase that has been identified only as late as in the programming phase becomes n-times more expensive to eliminate by then. The cost growth factor between the initial specification phase and acceptance testing may reach as much as 100 (Balzert 1998, p. 288). The author´s own practical experience and analyses confirm this figure. Consequently, SPI should be pursued as the primary aim. The goal may be complied with much easier by practicing quality control in each phase of the process rather than at the very end of the development chain only (Balzert 1998, p. 287 ff.). Quality has to be generated within the process as opposed to "imposing it" on the product post process. A about 50% of failure occur in the initial design and specification phases (Balzert 1998, p. 289), it is particularly important to concentrate on SPI specifically in the initial phases.

The new approach for SPI in this paper will support the existing SPI activities including the following 10 statements from the SPI Manifesto in 2010 (SPI-Manifesto 2010). See Table 1.

People	**Business**	**Change**
Know the culture and focus on needs	Support the organisation's vision and objectives	Manage the organisational change in your improvement effort
Motivate all people involved	Use dynamic and adaptable models as needed	Ensure all parties understand and agree on process
Base improvement on experience and measurements	Apply risk management	Do not lose focus
Create a learning organisation		

Table 1: SPI-Manifesto 2010; (source: SPI-Manifesto 2010)

4 *New approach*

As SPI involves a lot of different areas/elements e.g. people, culture, organisation, methods, communication etc. it will be considered in this paper as complex network system.

Complexity according to Ulrich und Probst (1988, p. 57) is defined first by the structure, number and variety of and the interrelationships between the different elements in a system and second by variability in time. Grossmann (1992, p. 19) recognises four basic types of systems:

- **Simple systems** (with few components and interrelationships),
- **Intricate systems** (a variety of components and interrelationships, with mostly predetermined system behaviour);
- **Fairly complex systems** (few components and interrelationships which however exhibit a variety in potential behaviour modes and variable cause and effect chains);
- **Highly complex systems** (a variety of components, highly varied interrelationships, a large variety of potential behaviour modes with variable cause and effect chains between the components).

Systemic Perspective on Software Development and SPI

The software development process and SPI are complex network processes with a number of applicable determinants interactions and as such, are to be viewed in its entirety (Frick 1995, p. 1). For examination purposes concerning holistic systemic SPI in and for development process, the determinants or influencing factors need to be identified and analysed.

Method

The methods for dealing with complexity are subject to 26 criteria according to Grossmann (Grossmann 1992, p. 44). Some of the 26 criteria are (excerpt):

1. Providing for a large number of components;
2. Reflecting mutual relationships between the components;
3. Providing for variability in time;
4. Providing for changing cause and effect chains;

5.-26. Other criteria.

Model

There are various models for better dealing with complex systems. The models allow among other things simulations of the different forces and interactions at work in complex systems in order to identify approaches that best facilitate problem solutions (Grossmann 1992, p. 55). Models are a reflection of reality. In order to be able to effectively deal with complexity and hence facilitate comprehensive holistic SPI, a model has to include both the factual and behaviour-related dimensions. In comprehensive systemic examination, human-related and factual dimensions have to be explored as the determinants and interactions at work falling under both dimensions. Grossmann (1992, p. 59 ff.) examined nine methods/models for dealing with complexity and placed each within a square with factual and behaviour-related dimensions. The method/models for dealing with complex tasks have been conceived by e.g. the following authors:

1. System Dynamics (by Forrester);
2. Thinking in networks (Ulrich, Probst);
3. Sensitivity Model (by Vester).

A conclusion based on assessment/analysis of three of the total of nine concepts (Grossmann 1992, p. 188) is that “Thinking in networks” was selected. It appears to be based on current knowledge and by earning the best assessment grades as viable methods for the task of comprehensive SPI in/for the software development process. Thinking in network relies, according to Ulrich and Probst (1991, p. 25ff.), on seven basic principles. From the seven basic principles, Ulrich and Probst (1991, p. 114ff.) derived the following six steps within the problem solving process.

(1) Defining objectives and modelling the problem situation
(2) Analysing causal chains
(3) Exploring and interpreting possible trends in the future based on current status
(4) Exploring control options
(5) Planning strategies and actions
(6) Implementing the problem solution.

Various simulation and practical implementation (operationalisation) tools have been derived from the thinking in network method. The tools include: - **The GAMMA tool; - CONSIDEO MODELER;** and others.

The GAMMA tool has a long tradition with the Vester (2001) approach and the (Consideo Modeler) is a new approach but also based on Vester (2007). In this research the GAMMA tool was selected.

GAMMA tool

The GAMMA tool is a combined concept and tool for dealing with complex problems (Hub 2002, p. 29). The concept refers to the thinking in networks method. The six steps within the problem solving process and/or the six steps under the thinking in networks method translate in the GAMMA concept into the following phases:

- Problem identification;
- System modelling;
- System analysis;
- Definition of actions to take.

The new approach of the SPI is based on selection of 12 (Number 2 to 13 in Fig. 3) important influence factors. Number 1 is our goal (SPI). The whole range of the considered influential factors is over 100. The selection on the above mention factors was done via literature research, empirical researching, own experience and simulation with various numbers of factors. The simulation showed that the amount of factors is not as relevant, as key factors. In Fig. 3, the selected influence factors/elements are presented with the GAMMA tool.

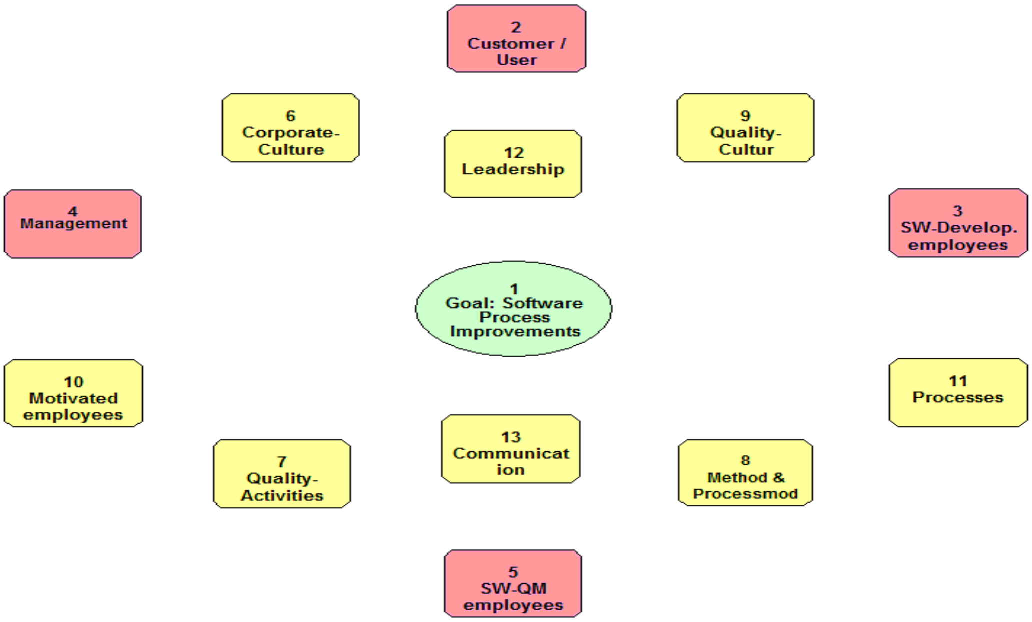


Figure 3: Elements of Software Process Improvement

The best results for selection of the elements and the interlinking of them will be reached by Group discussions.

Description of the elements

(1) Target Variable

The target variable is identifiable with the element "**Software Process Improvement" (SPI)** in the centre of Fig. 3

- ***Elements***

The following elements are extracts of the elements affect the target variable directly or indirectly.

(2) Customer/User
The customer/User are playing an important role in the SPI because they have to define the requirements of the SW. The requirements have to be written in an understandable way and structure for the SW-Development employees/organization. Especially in the definition phase of the SW it is important to integrate the customer in the SW-Development employees/organization for a better transformation process of the requirements.

(3) SW-Development Employees
The SW-Development Employees/organization is one of the central persons in the SPI. He has via processes, process-models and methods, the main tasks of the SW-Development. His qualifications have to be very high. Ongoing training in the SPI is mandatory for better SW and the motivation of the employees.

(4) Management
The management of the organization has the task to organise all the processes to reach the goal of the company. Management in the organisation has two main tasks:
Task No. 1 is to leading the employees, motivate them (see motivated people (10)) and to coordinate all tasks that the employees/subordinates can fulfil.
Task Nr. 2 is the "Process of the management". This means planning, organising, staffing and controlling an organisation. Organisation means a group of people or entities.

(5) SW-QM-employees
SPI needs a highly qualified person for the quality management tasks. Product quality starts with the requirements analysis and has to support all the SW-Development phases with quality tasks. The SW-QM employees should be integrated in the development process. Job rotation is a very good integration process.

(6) Corporate culture
Corporate Culture is the entirety of shared values, standards, stances and attitudes that shape the decisions, actions and behaviour of the members of the organisation (Gabler, a). Consequently, the attitudes towards successful failure essentially grow out of corporate culture. For further discourse, refer to Schein (2003).

(7) Quality-Activities
For SPI quality activities are necessary e.g. Reviews and FMEA. The application of constructive actions is the base for defect preventions. The lessons learned from the process will improve the SW-processes from project to project and contribute to the SPI.

(8) Methods and Process-model
SW-Development is based on using a method and process-model. There are various methods available but the methods have to be tailored to the SW task. to use CMMI or SPICE will be a very good base for SPI. The author has positive experience by using maturity models.

(9) Quality culture
For successful Software process improvement, the quality culture practiced by the firm is key, as the quality culture that grows out of corporate culture is an integral part of enterprise quality (Seghezzi 1996, p. 181). Managerial functions should be a role model in quality matters in their everyday stances and attitudes. Quality culture means e.g. customer focus; open for discussing

about failures which happen; willing to learn; quality focus as quality is everyone job (Mc Donald 2008).

(10) Motivated employees

Motivation is interpreted as the cause for a specific human behaviour (Strunz 2001, p. 49). As a result, the employee becomes directed towards a specific objective. Applied to the effort herein, the objective is "to avoid software failures". Available sources e.g. (MC Donald 2008) deal comprehensively with the topic herein.

(11) Processes

The processes hold together all the following critical dimensions (Chrissis 2003): People, procedures, methods, tools and equipment. It is very important that processes has a going to be improved to do things (Software) better. A focus on process improvement is necessary to manage the changing world and to be more competitive (Chrissis 2003).

(12) Leadership

Leadership is defined in expert sources in a number of different ways, some of which are: Being a role model (Seghezzi 2007, p. 79) and actively dealing with topics (in reference to Strunz 2001, p. 168); Directing staff towards a vision so that they respond in a motivated manner; Making staff understand the context by means of suitable communication (employee orientation).

(13) Communication

Within the comprehensive variety of communication, open communication that includes feedback promotes fairness, the will to listen and the readiness to report failures including software failures without fear. Open communication also promotes motivation. Communication should avoid isolated knowledge and should promote upward communication also (Mc Donald 2008).

The next step in network thinking is the interlinking of the elements.

Interlinked causal interdependencies

After the definition of the elements the elements will be interlinked. The interlinking of the elements is based on communication with various SW experts, literature research and from experience of the author. As there are more as 30 direct connection of elements, only a part of them will be explained here. In the following part some causal interdependencies will be explained e.g. the loop with the starting Nr. 4 → 10 → 3 → 8 → 1 → 10 → 6 → 4. Time delays in the network in figure 5 have a green colour.

(4) Management → (10) Motivated employees

One of the important management tasks is to motivate the employees. Motivate people is one of the basic elements for SPI.

(10) Motivated employees → (3) SW-Development Organization.

Motivated employees will drive the whole SW-Development Organization for SPI.

(3) SW-Development Organization → (8) Method& Process-model

The SW-Development Organization has to use a Method & Process-model to have success in SPI.

(8) Method & Process-model → (1) SPI

The use of a Method & Process-model has positive influence on the SPI.

(1) SPI → (10) Motivated employees

The success in SPI will additionally with a time delay, motivate the employees.

(10) Motivated employees → (6) Corporate Culture

In a long time horizon, motivated employees will slightly increase the corporate culture.

(6) Corporate Culture → (4) Management
The corporate culture will increase the management attitude to SPI.

(2) Customer/User → (4) Management
The customer has the "obligation" to write down understandable requirements for the supplier.

(12) Leadership → (3) SW-Development & (5) SW-QM employees
Leadership has a strong influence to the employees. Soft skills are the basics for motivation of employees.

(3) SW-Development & SW-QW employees → processes
Employees have created the processes and have to follow them for successful SPI.

Interlinked Causal interdependencies, their effect directions and effect intensities

Fig. 4 shows the interlinked elements, effect directions and effect intensities of each element. The interdependencies (arrows), effect intensities (thickness) and a plus or minus sign are indicated. A plus sign before the numerical intensity signifies in addition to the effect direction it promotive effect, a minus sign its inhibitive effect outlined with the GAMMA Software-Tool (Hub 2002).The blue colour of the interlinked line is a not delayed interaction and the green colour is a delayed interaction. This means that e.g. element No. 1 (SPI) will have a delayed influence the positive motivation of the employees. (Successful SPI will additionally motivate the employee).

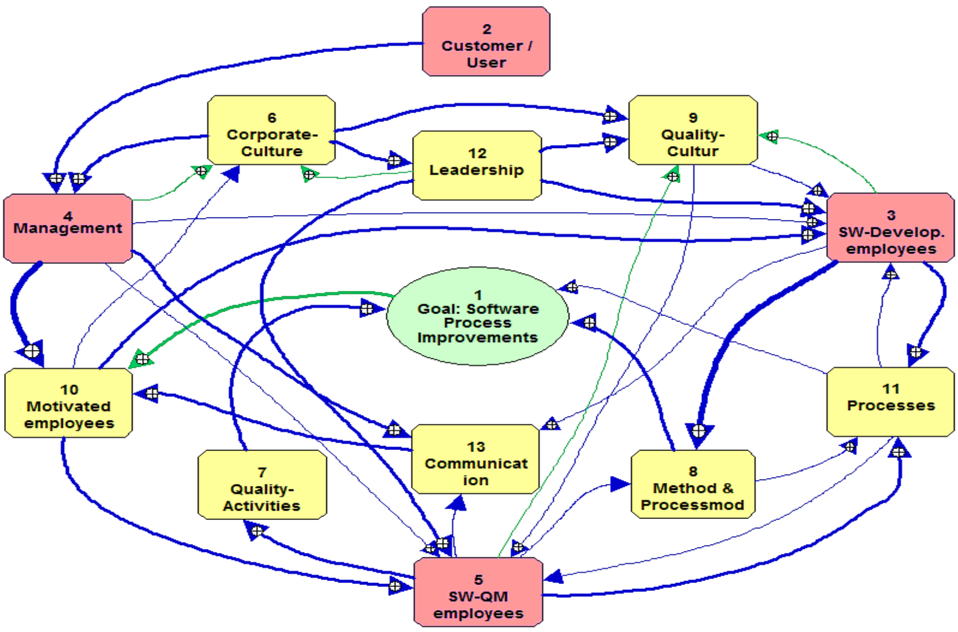


Figure 4: Interlinked Elements presented with GAMMA tool (source: the author)

In Fig. 5 the Cross-Impact-Matrix is shown for the identified network factors. The matrix was developed by Vester (2007) and also called "Vester-Papiercomputer". The matrix is used to reflect on all FROM / TO effects / relationships of the elements. The strength of the effect is defined by amount registered. It can take a value between 1 and 3 (even larger values for some effect relationships). The number 1 represents a weak relationship and 3 represents a strong relationship. An example effect is the vertical No. 4 (Management) on the horizontal No. 10 (Motivated employee) with the effect strength of 3.

Wirkung VON / AUF	1	2	3	4	5	6	7	8	9	10	11	12	13	Su.E
1. Goal: Software Process Improvements										2				2
2. Customer / User				2										2
3. SW-Develop. employees								3	1		2		1	7
4. Management			1		1	1				3			2	8
5. SW-QM employees							2		1		2		1	6
6. Corporate-Culture				2					2			2		6
7. Quality-Activities	2													2
8. Method Processmodel	2										1			3
9. Quality-Cultur			1		1									2
10. Motivated employees			2		2									4
11. Processes	1		1		1									3
12. Leadership			2		2	1			2					7
13. Communication										2				2
Summe Beeinflussung	5	0	7	4	7	2	2	3	6	7	5	2	4	

Figure 5: Cross-Impact-Matrix (Vester-Papiercomputer) of the Network

5 Results

The "Influence Matrix" shows the position of every individual element. Thereby, the role of every individual element in the system is clearly identified (Vester 2007, S. 194). Through this representation, the individual interlinking in the system can be determined.

The Matrix has four quadrants with the following impact:

Quadrant active (down right = red) means, that elements in this area influence other elements strongly, but they are not strongly influenced from others;

Quadrant critical (top right = yellow) means, that elements influence other elements strongly and are strongly influenced from others;

Quadrant passive (top left = blue): elements-influencing others only a little, but they are influenced from other elements strongly;

Quadrant buffer (down left = green): elements-influencing others only a little and they are influencing others also only a little.

The calculation of the position of the elements in the matrix is based on the active/passive (active= influencing, passive = influenced) amount of each element. Active amount is the calculation of the values in the line (active sum). Passive amount is the calculation of the values in the column (passive sum). The highest value is then 100% (see Fig. 5) e.g. the highest value of the active sum of elements is line Nr. 4 (management) with the value 8= 100%. This means that Nr. 4 is placed on the most right position in the matrix. For the y position Nr. 4 gets 50% of the passive sum maximum. The complete calculation of each elements place in the matrix is explained in Hub (2002). There are two opportunities to analyse the system as the influence matrix can calculated in two ways: 1. Direct influenced elements; 2. cumulated influenced elements.

In Fig. 6 the influencing (Einflussnahme) and influenced (Beeinflussung) elements are shown in the Matrix.

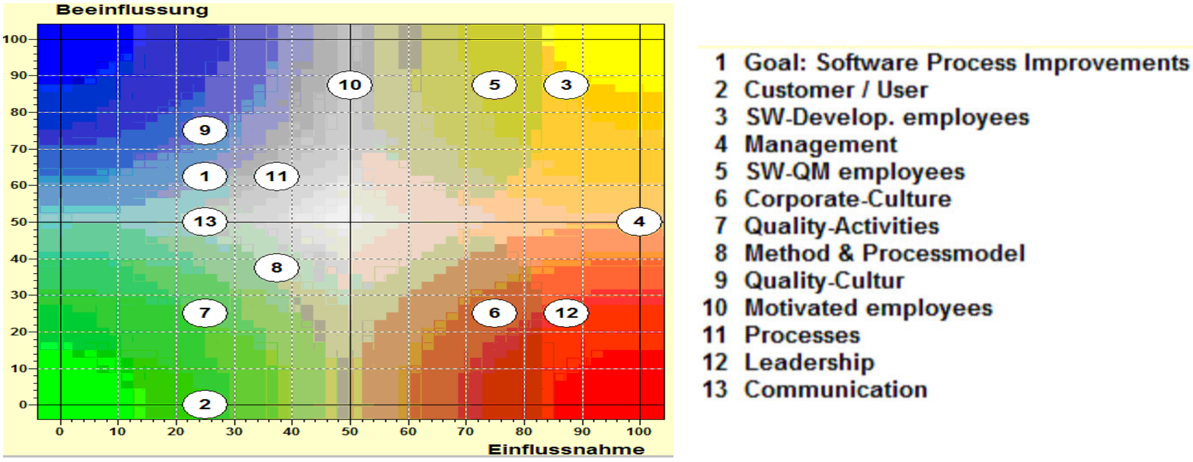


Figure 6: Results of the interlinked elements in the influence- matrix of influencing (Einflussnahme) and influenced (Beeinflussung) elements

In this paper the direct influenced elements are calculated in the influence-matrix.

We can see that 7 elements are in the left of the field (Nr. 2, 7, 8, 13, 1, 11 and 9). This means that they are not very strongly influencing the system.

On the right site there are 6 stronger elements (Nr. 6, 12, 4, 3, 5 and 10). Element No. 6 (Corporate culture) and 12 (Leadership) are very active elements for the target element No. 1 (**Successful Process Improvements=SPI).**

Element No. 1 (SPI) should be more influenced, as it is our goal to improve this

Element No. 4 (Management) has a strong influence in the system, but it is also influenced from other elements (by 50%).

Elements No. 3, No. 5 (and No. 10) are critical elements. This indicates that they can influence the system strongly but reinforcing loops in the system can have a negative influence.

Elements No. 1, 9 11 and 13 are passive elements in the system (less influence).

(Element No. 9 is strongly influenced from other elements and has low influence to the system).

Elements No. 2, 7, 8 are balancing the system as the influencing power is low and they are weakly influenced from others.

Further analyses of the interlinking possibilities in the networked system are:

1. Analyse causal chains
2. Feedback analysis
3. Time analysis
4. and others.

to 1. Analyse causal chains:

- How to widen the effects with the interlinking in the network?
- What kind of causal chains will achieve a determined influence level for an effect?
- What are the indirect effect relationships?

There are two possibilities for the functional chain analysis .

a) Which connection is going e.g. to the element No. 1(SPI) to influence it?
b) Which connection is starting from e.g. the element No. 3 (SW-Development employees) to influence
other elements.

In this example we decided to use No. a: Which connection is going to element No. 1 (SPI)

The functional chain analysis shows us in Fig. 7 that 3 elements directly influence the goal SPI.

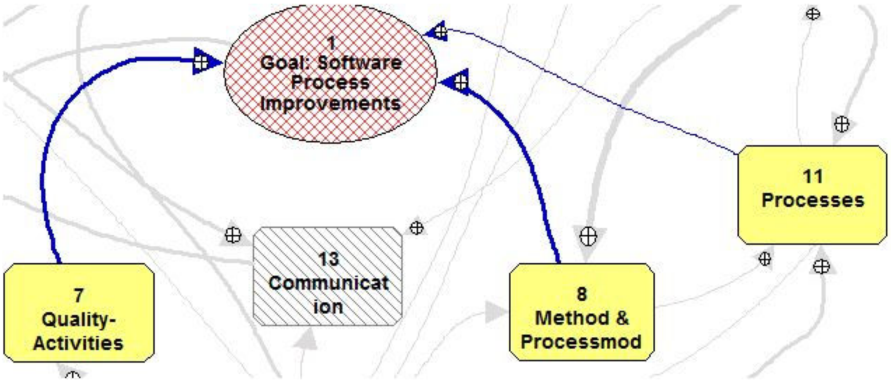


Figure 7: Analysing of functional chain

To go deeper to analyse the system for various measures it is possible to select e.g. which elements are influencing element No. 8 (Method & Process model).

to 2. Feedback Analysis

This analysis is carried to determine whether there are causal chains that have:
- Desired stabilisation
- Undesired strengthening
- Escalating effects arise.

Feedback analysis is not shown here.

to 3. Time Analysis
Time analysis will not be gone into here.

On the basis of these analyses, strategies / interlinking are derived in the network. This is to clarify which elements are and are not influenced. For which elements do interlinks have a special effect? The optimisation of the system can be made on the basis of different approaches, e.g. which elements influence the active ones in order to strengthen these (optimisation of the entire chain).

Definition of action to take / System intervention

After all of the various analysis, the decision is for the following system intervention: Enforce the following activities /connections to test how the system will react.

(10) Motivated employees → (5) SW-QM employees
(5) SW-QM employees → (7) Quality activities
(7) Quality activities → (1) SPI
(10) Motivated employees → (3) SW-Development employees
(3) SW-Development employees → (8) Methods and Process-model
(8) Methods and Process-model → (1) SPI
(9) Quality Culture → (5) SW-QM employees
(9) Quality Culture → (3) SW-Development employees.
In figure 8 the test results are visible. The number of passive and buffer elements decrease from 8 to 5.
The number of active and critical elements increased from 5 to 8.

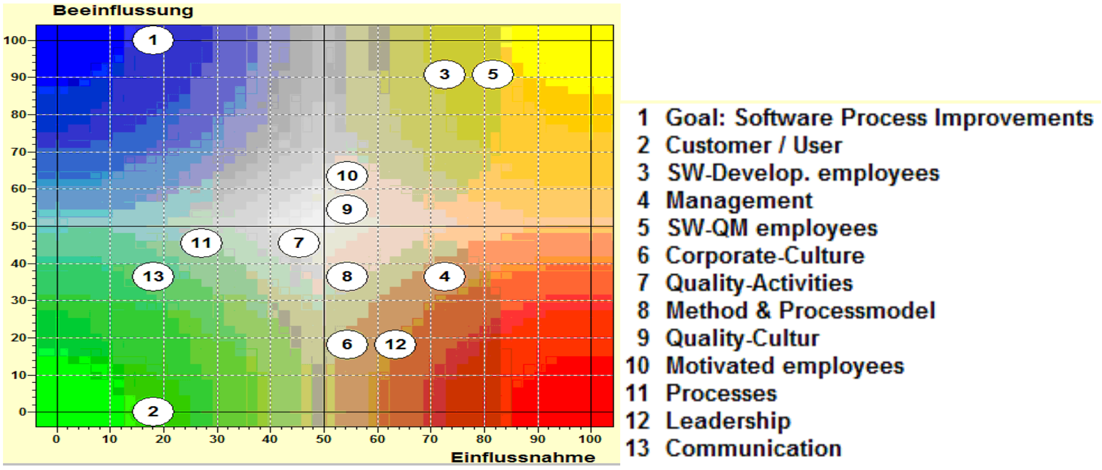


Figure 8: Results of the interlinked elements in the matrix influencing and influenced after System modification

The results show that the measures activate more elements. On the left side there are now 5 not so strong elements compared with 7 in figure No. 6. Nr. 1 (Software Process Improvements) is now a very strong influenced element. The most elements in the system "play" after the system intervention a more active role.

6 Conclusion

The results in both influence matrix in (figures 6 and 8) provide indications that the human being represented by management, leadership, employees and culture, are key elements for SPI. The most significant elements with strong positive effect in successful SPI are No. 4 (Management), No. 6 (Corporate culture) and 12 (Leadership). The No. 3 (Software Develop. employee) and 5 (SW-QM employee) are active but critical elements and need further consideration. The elements No.7 (Quality activities), No. 8 (Methods & Process-model), No. 11 (Processes) and No. 13 (Communication) could be enforced. The customer (2) has a low influence and is not influenced by any element. The SPI (1) is and has to be a passive element in the system as SPI is our goal and has to be influenced strongly by the elements.

One additional goal of this paper is to visualise different relationships of the elements in the SPI and to contribute to the system thinking in the SPI. The results of the network and influence matrix should not be interpreted very precisely, but it is an import end indication of the interlinking of the elements and its priority for SPI.

Generally, a note is due that any actions and approach expansions made need to be verified for their interlinking effects and balanced accordingly as the elements function in a matrix of mutual relationships and consequently also need assessment in a relationship matrix.

This new approach will not replace the existing SPI methods but support it from a holistic view.

It is worthwhile for further research on this system e.g. deeper analysis or the system expansion with additional elements.

7 Literature

[1] Balzert, H.F. (1998). Lehrbuch der Software-Technik; Software-Management, Software-Qualitätssicherung, Unternehmensmodellierung. Spektrum, Akademischer Verlag Heidelberg/Berlin

[2] Chrissis(2003). CMMI Guidelines for Process Integration and Product Improvement, Tenth printing, March 2006, Pearson Education, Inc. Boston, MA. USA

[3] CONSIDEOModeler (2009). Modellierungssoftware von Consideo GmbH, D-23562 Lübeck

[4] Grossmann, C. (1992). Komplexitätsbewältigung im Management; Anleitungen, integrierte Methodik und Anwendungsbeispiele; Verlag GCN, Winterthur

[5] Frick, A. (1995). Der Softwareentwicklungsprozess; ganzheitliche Sicht; Grundlagen zu Entwicklungs-Prozess-Modellen, Hanser Verlag München 1995

[6] Gabler_a. Gabler online Wirtschaftslexikon

[7] Hörmann, K./Dittmann, L./Hindel, B. (2006). SPICE in der Praxis; Interpretationshilfe für Anwender und Assessoren, basierend auf ISO/IEC 15504, dpunkt.Verlag, Heidelberg

[8] Hub, H. (2002) GAMMA Software, Praxisbeispiele zum ganzheitlich vernetzten Denken; DMG Verlag, Nürtingen 2002

[9] Masing, W. (2007). Handbuch Qualitätsmanagement; Herausgegeben von T. Pfeifer/R. Schmitt, Hanser Verlag München

[10] Mc. Donald, M./Musson, R./Smith, R. 2008. The Practical Guide to Defect Prevention. Techniques to Meet the Demand for More Reliable Software; Microsoft Press; Redmond, Wa, USA 2008

[11] Schein, E.H. (2003). Organisationskultur; The Ed Schein Corporate Culture Survival Guide; MIT/Cambridge USA; Edition HumanistischePsychologie, BergischGladbach

[12] Seghezzi, H. D. (1996). Integriertes Qualitätsmanagement; Das St. Galler Konzept München; Hanser Verlag München

[13] Seghezzi, H.D. (2007). Integriertes Qualitätsmanagement; Der St. Galler Ansatz; Hanser Verlag München

[14] SPI- Manifesto (2010). Software Process Improvement, eurospi.net 10.12.2010

[15] Strunz, H. / Dorsch, M. (2001). Management; Managementwissen für Studium und Praxis, Oldenburg Wissenschaftsverlag München

[16] Ulrich & Probst 1988 & 1991. Anleitung zum ganzheitlichen Denken und Handeln; Ein Brevier für Führungskräfte, Verlag Paul Haupt, Bern/Stuttgart 1988 und 1991

[17] Vester, F. (2007). Die Kunst vernetzt zu denken; Ideen und Werkzeuge für einen neuen Umgang mit Komplexität, dtv Verlag, München

[18] Vester, F. (2001). Die Kunst vernetzt zu denken; Ideen und Werkzeuge für einen neuen Umgang mit Komplexität, dtv Verlag, München

[19] Wallmüller, E. (2001). Software Qualitätsmanagement in der Praxis, Hanser Verlag München

[20] Forrester, J.W.: World Dynamics, Wright-Allen-Press Cambridge; USA, MA 1971

8 CV

Alfons Unmuessig

– **Study at HTWG* and Academy**

Computer Science and Business Administration

– **Work Experience**

- 2000-2009

CEO Siemens ElectroCom GmbH Berlin and

Vice President Quality Management global at Siemens AG Konstanz

- 1996-2000

Upper Management Functions at Siemens AG e.g. Business Unit Leader

- until 2000

Software Development, Software Department Manager

– **Lecturer part time**

- 2009-today

HTWG* Konstanz and DHBW** Ravensburg

*HTWG University of Applied Sciences

**Dual University

Towards SPICE for Nuclear (S4N) – On the integration of IEC 61508, IEC 60880 and SPICE

Risto Nevalainen, Finnish Software Measurement Association FiSMA
risto.nevalainen (at) fisma.fi
Timo Mäkinen, Tampere University of Technology, Pori
Timo Varkoi, Tampere University of Technology, Pori
timo.makinen / timo.varkoi (at) tut.fi

Abstract

ISO/IEC 15504 has been applied to several domains, where safety is one main characteristic for software and systems. Nuclear power industry is one of the most challenging domains, because it´s safety requirements are strictest. Nuclear power domain has also a long tradition in using its own standards, classifications and certification schemes.

This article discusses the SPICE for Nuclear (S4N) model and how it combines both SPICE related generic requirements with functional safety and nuclear specific elements.

Conformance with standards is not any absolute guarantee for safety. Safety can be achieved only by use of several different approaches, which all provide their own evidences and support for qualification and licensing.

This article is a part of the research project CORSICA in Finnish nuclear research program SAFIR 2014.

1 Current standards and models

Developing SPICE for Nuclear (S4N) does not happen in isolation. On the contrary, the main challenge is to combine several, quite separate and inclusive approaches together as one system of its own. The key word is integration. Nuclear power industry has used standards and models for centuries – being one of the pioneers in safety. Only few other domains are as mature and advanced: space, avionics and railways as examples.

In fact, current SPICE family (ISO/IEC 15504) has also its roots in safety. Part 5 uses software lifecycle model ISO/IEC 12207, and its main predecessor has been DoD 2167 Defense System Software Development standard. Defense systems are safety systems by default. Some domains have developed their own SPICE variants for safety-critical systems, for example Automotive SPICE and SPICE for SPACE (S4S). At least electro-medical industry is also developing its own model for assessing safety-critical systems.

In this article SPICE means current draft version of ISO/IEC 15504 Part 5:2011. It is an exemplar assessment model for software engineering life cycle processes. It is fully aligned with ISO/IEC 12207:2008, which is the process reference model containing also process categories for systems engineering, project management, organizational management and acquisition. Total number of processes in ISO/IEC 15504 Part 5 is 58. ISO/IEC 15504 Part 10 is safety extension, and it is also included fully in S4N model. Part 10 defines three processes for assessment in safety related development environments.

Main standard in functional safety is IEC 61508. It has seven parts, software requirements in Part 3. The safety concept is a bit different in this standard, using term “safety related”. In nuclear domain it would mean safety system in lower level safety class 3. So, in this article we use term safety-critical also when covering IEC 61508. This standard is essentially also a process standard, like ISO/IEC 12207. Main content in Part 3 is requirements for software engineering and management. Most interesting and useful content is in the annexes A – C of Part 3. It has numerous method tables, organized mainly by the SIL levels 1 – 4 (SIL = Safety Integrity Level). Current version of IEC 61508:2010 is still a bit old-fashioned in its methods, demonstrating also a more common aspect of safety standards and models. Better to be a bit conservative than too radical!

Nuclear domain standards have their roots in IEC 61508 [9]. The standard IEC 61513 highlights the need for complete and precise requirements, derived from the plant safety goals, as a pre-requisite for generation the comprehensive requirements for the total instrumentation and control (I&C) system architecture, and hence for the individual I&C systems important to safety. This standard is the highest I&C safety standard in the nuclear industry given requirements for I&C systems and providing also high level requirements for safety software. The standard can be considered as a section standard for IEC 61508 (parts 1 and 2) consisting appropriate clauses for Overall safety life cycle for I&C, System safety life cycle, Overall integration and commissioning, and Overall operation and maintenance [1].

IEC 61513 does not use the concept of SIL as a classification. Nuclear domain has its own safety classifications, and IEC 61513 covers safety classes 1, 2 and 3. Safety relevance is expressed by classifying functions in categories A, B, C and D.

Software standards in nuclear domain are IEC 60880 for safety-critical software (Class 1) and IEC 62138 for safety-related software (Class 2 and 3). They can be categorized as “second level” standards, referring directly to IEC 61513 but not directly to IEC 61508 Part 3. IEC 60880 tackles the issue of software aspects for I&C systems performing the category A functions. [10]

Comparison of IEC 60880 and IEC 61508 Part 3 is both easy and difficult. Mapping and comparison of standard clauses is quite straightforward. Difficulty to compare is in their different phrasing and level of abstraction [2]. Anyway, they are quite similar and comparable, with some need for interpretation. In general, IEC 60880 is more detailed and has also some nuclear specific additional requirements. Mapping of IEC 60880 with SIL levels in IEC 61508 Part 3 is meaningless, because it covers all SIL levels. For this reason our starting point in S4N is to cover also SIL4, which brings all methods in IEC 61508 Part 3 into S4N.

Safety-related software standard in nuclear domain is IEC 62138. This standard is not covered in this article, even though it is a part of the SPICE for Nuclear (S4N) concept. It has been discussed in some previous articles [3], [4].

2 S4N Framework

2.1 Requirements for model development

A key component of the S4N framework is a process assessment model (PAM) [5]. A PAM combines a process reference model (PRM) with a measurement framework. In this case, the PRM consists of a subset of processes from ISO/IEC 12207 (with possible extensions from other PRMs later on). The measurement framework declares e.g. capability levels and process attributes. In S4N the measurement framework is the one defined in ISO/IEC 15504-2. Altogether, we apply the requirements explicitly stated in ISO/IEC 15504-2 for a PAM: *A Process Assessment Model shall contain a definition of its purpose, scope and elements; its mapping to the Measurement Framework* *and specified Process Reference Model(s); and a mechanism for consistent expression of results [15504-2].* The requirements are grouped in table 1.

Class	Requirement
Scope	
1.	A PAM shall relate to at least one process from the specified PRM(s)
2.	A PAM shall address, for a given process, all, or a continuous subset, of the levels (starting at level 1) of the Measurement Framework for process capability for each of the processes within its scope
3.	A PAM shall declare its scope of coverage in the terms of: a) the selected PRM(s); b) the selected processes taken from the PRM(s); c) the capability levels selected from the Measurement Framework.
Indicators	
4.	A PAM shall be based on a set of indicators that explicitly addresses the purposes and outcomes of the processes in the selected PRM;
5.	The set of indicators demonstrates the achievement of the process attributes within the capability level scope of the PAM.
Mapping	
6.	A PAM shall provide an explicit mapping from the relevant elements of the model to the processes of the selected PRM and to the relevant process attributes of the Measurement Framework.
7.	The mapping shall be complete, clear and unambiguous.
8.	The mapping of the indicators within the PAM shall be to: a) the purposes and outcomes of the processes in the specified PRM; b) the process attributes (including all of the results of achievements listed for each process attribute) in the Measurement Framework.

Table 1: ISO/IEC 15504-2 requirements for a PAM

The development of the S4N framework produces a novel solution to a specific business need i.e. the need to assess processes that are to used to develop software with utmost safety requirements. Therefore, we apply design science approach (Hevner et al.) to the development of the S4N framework. [11]

2.2 Integration of existing models and standards in S4N

The main challenge is integration. Minimum option is to keep SPICE, IEC 61508 and IEC 60880 separate and leave space for ad hoc integration in each assessment and qualification case. This option is more or less the current situation. Assessor should have a portfolio of assessment methods and use them as needed.

The other extreme is tight integration. It would mean that each element and requirement in SPICE, IEC 61508 and IEC 60880 is linked with each other. Also overlaps are removed and combined when possible, to minimize unnecessary recording work during assessment.

S4N tries to implement tight integration. It will be achieved gradually during next years. S4N can take elements and requirements also from other sources, mainly based on regulatory and qualification needs. An example is information security, which is more relevant nowadays. Because current SPICE does not have any process for information security, it will be taken either from ISO/IEC 27001 or ISO/IEC 15504 Part 8.

Integration can be illustrated as a 3-level model, see figure 1. First we can integrate a generic approach in SPICE with functional safety requirements in IEC 61508. This can be further integrated with nuclear domain requirements and classifications. All levels have their own internal structure.

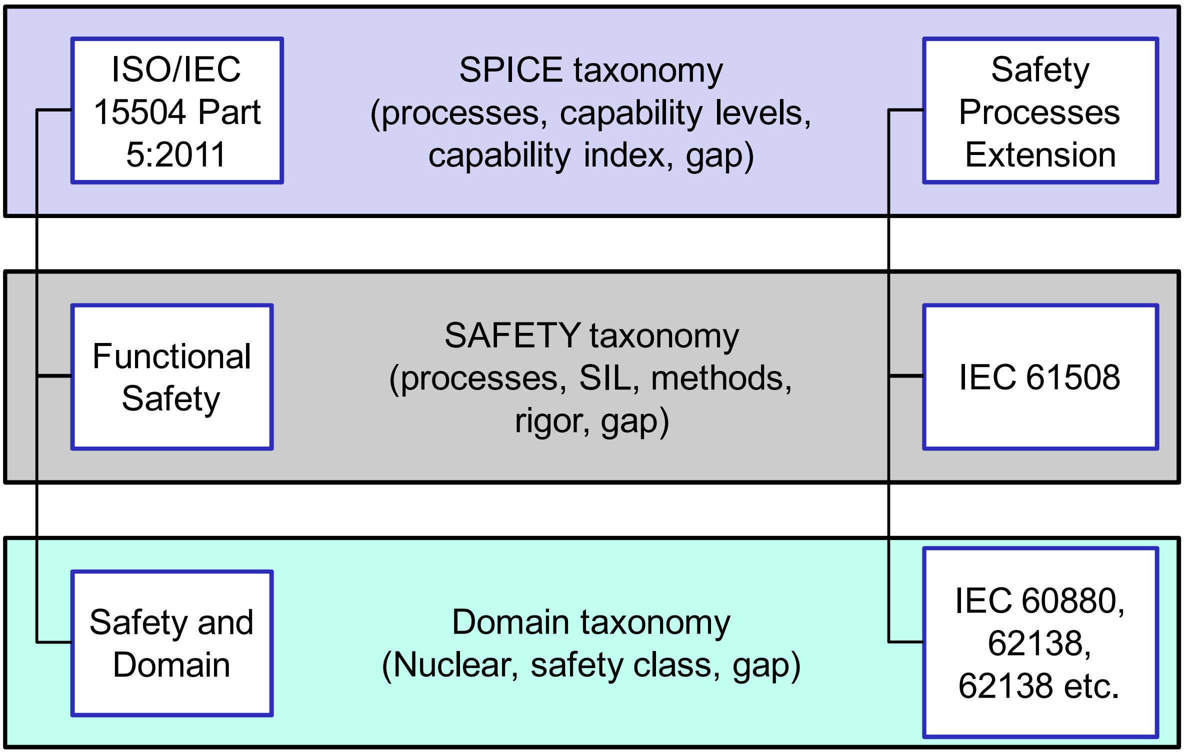


Figure 1: Main levels of integration and classifications by sources in S4N

2.3 *Using different classifications*

Each existing approach has its own classifications. Some examples are:

- SPICE: Processes, process attributes, capability levels
- IEC 61508: Processes, SIL levels, methods, rigor
- IEC 60880: Processes, safety class, method requirements, nuclear domain specific requirements

Integration of approaches means also better harmonization between classifications. For example, a process in SPICE can achieve capability level 3. It should have some relevance also in satisfying process specific requirements in IEC 61508 and in IEC 60880. Also, if some requirement in IEC 60880 is satisfied, it should be an evidence also for functional safety and SPICE. Otherwise, there is no integration.

In the early phase of S4N development, we intentionally keep integration level quite low. Main idea is to get a method that contains all classifications. Later on, we can develop S4N smarter to avoid unnecessary recording work and to fulfill product specific qualification needs. Product evaluation is typically done using a safety case or an assurance case approach. S4N can provide major part of evidences for safety case calculations, but it can only be achieved gradually. Case studies are needed to validate S4N for the safety case.

All different classifications can be expressed as a “taxonomy machine”. It is a smart tool, being able to combine different evidences and (re)using them to satisfy different classifications. In this phase of S4N development, taxonomy machine is just a metaphor and needs further elaboration.

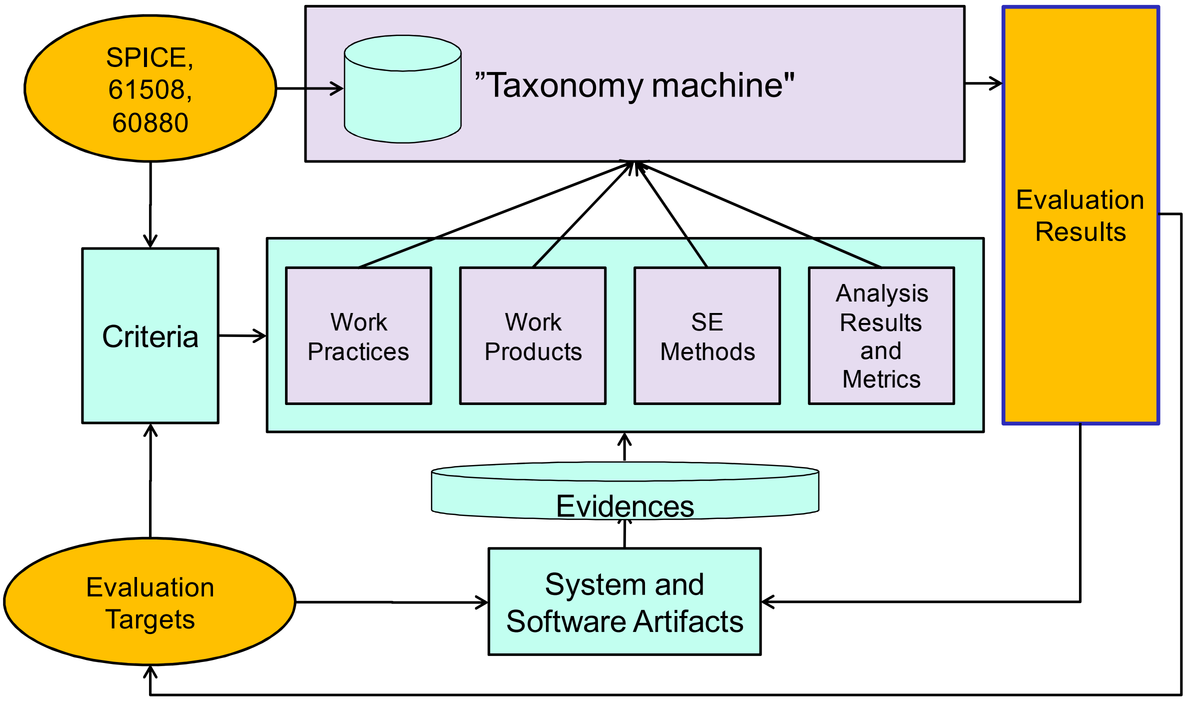


Figure 2.: Taxonomy machine concept to combine different types of classifications and evidences

An essential part of process assessment is to record and classify evidences. They can be categorized by type and source. Figure 2 proposes a 4-type classification system for evidences:

- Work practices: they can be base and generic practices in SPICE or process requirements in IEC 61508 and in IEC 60880.
- Work products: they can be same as in SPICE. Some work products could be mandatory, others optional.
- Software engineering methods. Main source are tables in IEC 61508 Annexes A, B and C. Also rigor and relevance for SIL is here. This evidence type allows also new methods to be included in S4N, if and when some research projects can validate new methods.
- Analysis results and metrics: The key is relevance for validation, using various analyses of measurement results. Examples are PHA, FMECA, reliability calculations and V&V metrics. This evidence type is not implemented in the first phase of S4N, hopefully in coming years.

2.4 The classification from the SPICE point of view

Figure 3 depicts the tentative integration of SPICE safety and nuclear safety process requirements as an UML class diagram, from the SPICE point of view. The white classes illustrate the elements from SPICE and the gray-shaded classes represent its extensions coming from the standards IEC 60880 and IEC 61508. In the SPICE assessment model, there are nine process attributes for each of the processes. One of the attributes is the performance attribute relating to the capability level 1, and the others are capability attributes relating to the capability levels 2-5. The process attributes are associated with assessment indicators of several kinds illustrating the types of evidence. Figure 3 focuses on process performance. In the SPICE assessment model, there are two kinds of indicators for the performance attribute: base practices and work products, which act as the input and output products of the processes.

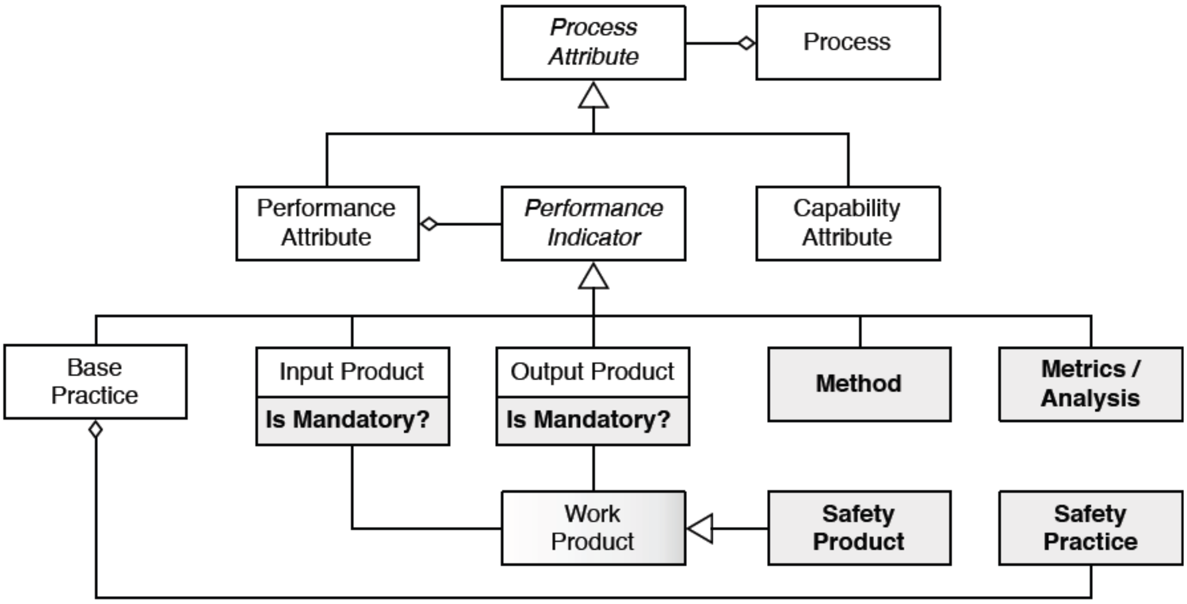


Figure 3: A part of the S4N assessment model as an UML class diagram

The preliminary S4N extends the base practices with safety practices based on IEC 60880 and IEC 61508. The safety practices are grouped by the base practices. In some cases, a safety practice is associated more than one base practice. In addition to extra practices, S4N contains supplementary safety and nuclear safety specific work products, which are denoted by the class safety product as the sub-class of work product in Figure 3. S4N also extends input and output with an attribute specifying if a work product is mandatory for a process or not. Besides the SPICE evidence types, S4N also contains two other kinds of evidence categories, method and metric/analysis based on IEC 60880 and IEC 61508. The added indicators are associated to processes through the process performance attribute.

3 S4N Assessment process

S4N assessment process is more or less same as in SPICE. Defined assessment types for supplier processes are:

- Ability assessment. Main goal is to check that the supplier is able to deliver according to standard requirements. Focus is in document review, using a limited set or product, process and organizational data and records.
- Pre-qualification assessment. Main goal is to identify potential risks and gaps, which can then be included in agreement and delivery project as action items. In ISO/IEC 15504 standard this is called Capability Determination mode. Level of formalism can vary, depending mostly on strictness and class of safety requirements and system size/complexity.
- Qualification assessment, leading to formally defined process profile result. Typically it contains also conformance statement or certificate against required standard, for example IEC 60880.

Assessment of supplier processes is typically a mix of SPICE, IEC 61508 and IEC 60880 requirements. S4N model and method includes also assessment for acquirer processes. It is based fully on the SPICE process set.

Main phases of S4N assessment are presented in figure 4. The illustrated process is most relevant for pre-qualification assessments [6]. Full scale qualification assessment has a more detailed set of phases and tasks.

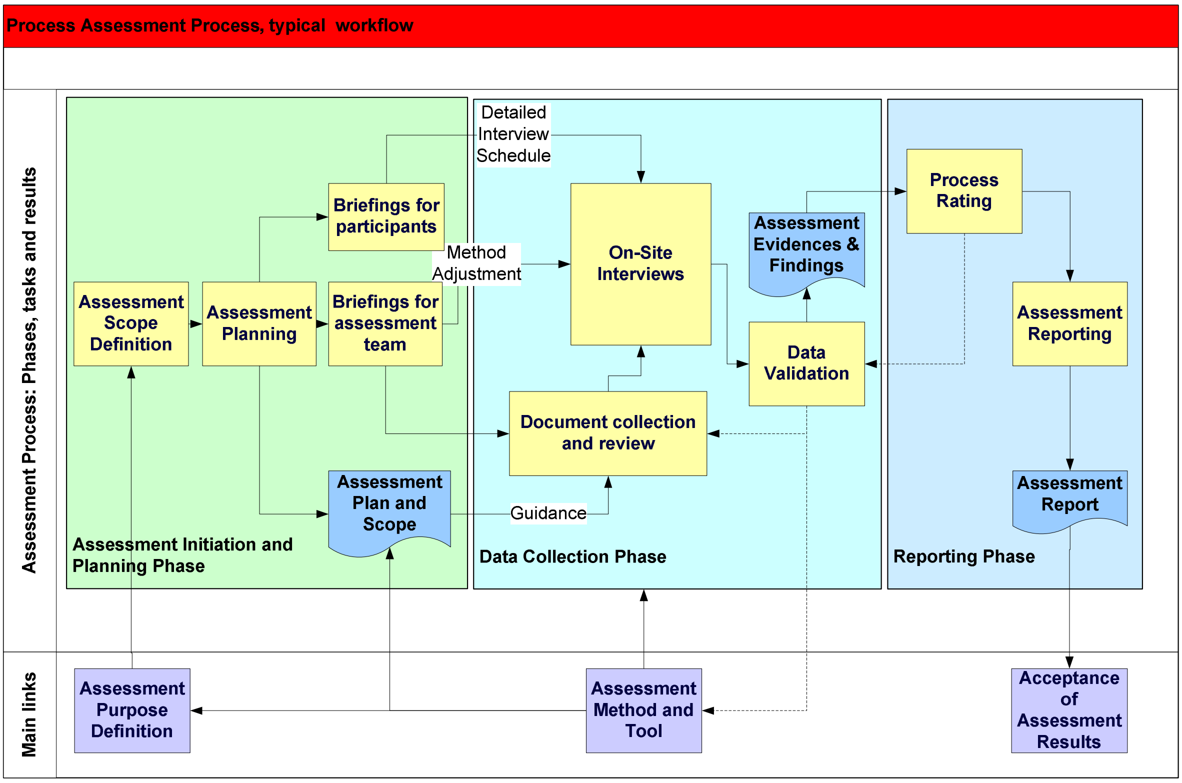


Figure 4:. Typical assessment process in S4N

Figure 5 depicts an example, how S4N extends the SPICE ratings. The gray-shaded elements illustrate extension. As already shown in Figure 3, S4N contains supplementary assessment indicators, including safety practices. They are rated independently while also acting as sub-indicators for base practices. Thus, the safety practices guide an assessor to rate a base practice such as base practices are used to give a rating for the performance of a process.

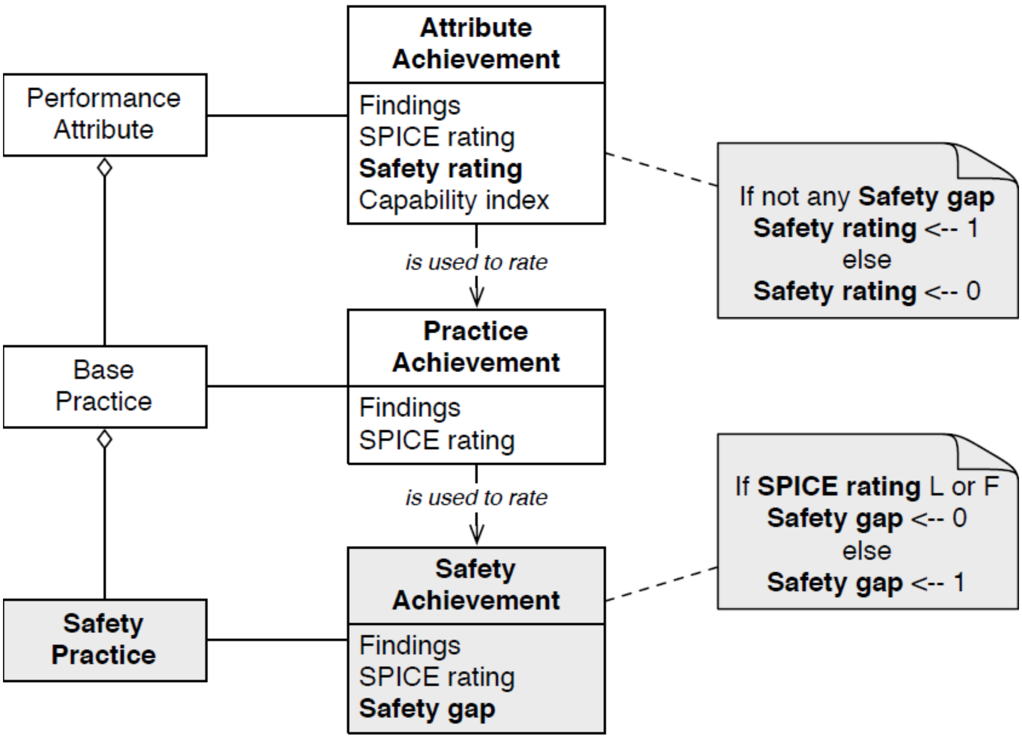


Figure 5: S4N extensions for rating process instances in assessments

In addition to SPICE rating, S4N proposes safety gap to be associated with safety practice, and safety rating related to the performance attribute. Relating to a safety practice, a process instance to be assessed has safety gap (1), if its SPICE rating is N of P, otherwise safety gap does not exist (0). The safety rating of the performance attribute of a process instance is failed (0), if any of the safety practices has safety gap, otherwise it is passed (1).

4 S4N Process sets

The scope of an assessment can vary, based on the assessment type, safety class, organisation size and software type. If the system is mainly software, then software engineering processes are more relevant and will be included in the assessment scope.

Following two tables 2 and 3 have a list of processes. They are categorized in three basic types of assessment scope:

- Core set: processes that are most relevant and will be always in the assessment scope. Otherwise even light integration between SPICE, IEC 61508 and IEC 60880 is not possible.
- Management set: Processes that are needed to cover all lifecycle phases in IEC 61508 and IEC 60880. They are additional to the core set. Management set cannot be assessed separately, but only together with the core set.
- Conformance set: Full coverage of all IEC 61508 and IEC 60880 requirements and also additional qualification needs.

Other main classification is scope by pre-qualification vs. full qualification:

- Scope of pre-qualification. These processes are marked with X and grey color.
- Scope of full qualification. These processes are marked with x and yellow color. Full qualification contains also pre-qualification and may require reassessment.

Category	Process in ISO/IEC 15504-5:2011	Core set	Mgmt set	Conf. set	Acquirer set
Technical Processes (ENG)					
	ENG.1 Stakeholder requirements definition	X			x
	ENG.2 System requirements analysis	X			x
	ENG.3 System architectural design	X			
	ENG.4 Software implementation	X			
	ENG.5 System integration	X			
	ENG.6 Systems qualification testing	X			
	ENG.7 Software installation	x			
	ENG.8 Software acceptance support		x		
	ENG.9A Operational use				x
	ENG.10 Software maintenance	x			
	ENG.11 Software disposal				
Software Implementation Processes (DEV)					
	DEV.1 Software requirements analysis	x			
	DEV.2 Software architectural design	x			
	DEV.3 Software detailed design	x			
	DEV.4 Software construction	x			
	DEV.5 Software integration	x			
	DEV.6 Software qualification testing	x			
Software Support Processes (SUP)					
	SUP.1 Software documentation management	X			
	SUP.2 Software configuration management	X			
	SUP.3 Software quality assurance	X			
	SUP.4 Software verification				
	SUP.5 Software validation				
	SUP.6 Software review	x			
	SUP.8 Software problem resolution	x			x
	SUP.9 Software change management	x			x

Table 2: Engineering, software implementation and software support processes by scope tiers

Category	Process in ISO/IEC 15504-5:2011	Core set	Mgmt set	Conf. set	Acquirer set
Safety extension (SAF)					
	SAF.1 Safety Management		X		
	SAF.2 Safety Engineering	X			
	SAF.3 Safety Qualification		X		
Project Processes (PRO)					
	PRO.1 Project planning		x		
	PRO.2 Project assessment and control		x		
	PRO.4 Risk management		x		
	PRO.5 Configuration management	x			
	PRO.6 Information management	x			
	PRO.7 Measurement	x			
Agreement Processes (AGR)					
	AGR.1 Acquisition				x
	AGR.1A Acquisition preparation 1				x
	AGR.1B Supplier selection 1				x
	AGR.1C Agreement monitoring 1				x
	AGR.1D Acquirer acceptance 1				x
	AGR.2 Supply			x	x
	AGR.2A Supplier tendering			x	
	AGR.2B Contract agreement		x		x
	AGR.2C Product/service delivery and support		x		
	AGR.3 Contract change management		x		x
Organizational Project-Enabling Processes (ORG)					
	ORG.1A Process establishment			x	
	ORG.2 Infrastructure management		x		
	ORG.5 Quality management		x		
Category	**Additional processes (candidates)**				
Additional processes					
	Suitability analysis		X		
	Information security		X		

Table 3: Safety extension, project management, agreement and organizational processes by scope tiers

The main source for processes in tables 2 and 3 is ISO/IEC 15504-5:2011 [8]. It is possible that there will be some minor changes in process names and categories before the final publication of the standard. Information security process can be taken for example from ISO/IEC 15504-8 draft version. Suitability analysis can be a separate process or maybe an extension of the safety management process.

The minimum set of processes in an assessment scope is 10, when only the core set in pre-qualification is covered. The maximum amount of processes is 41, and then all tiers are covered in full qualification.

5 Conclusions and future

This article describes first step in S4N development. Current situation is based on existing standards and models (SPICE, IEC 61508 and IEC 60880). We have also used TVO SWEP method, which is an assessment method for lower safety classes in nuclear domain.

Integration goes further in future. Most important direction of integration is towards product evaluation. Already now we have some such concepts, for example required analyses, data and metrics. We have also some natural overlap between process assessment and product evaluation, because product is an evidence type in S4N.

An important area is to cover process evidence needs for safety case. S4N goes quite far in this, because it covers already process assessment and conformance with standards. Open problem remains in quantitative evicencesevidences for reliability and SIL calculations. Software does not behave similarly as hardware and therefore statistical calculations are difficult. For example, mean time between failures (MTBF) is not quite meaningful for software, because all detected faults have been also latent faults.

Software safety case itself is transforming to software assurance case. It is even more open concept to use all kind of evidences to proof quality claims. Standard ISO/IEC 15026 is becoming important in this context. Maybe one option is that we are seeking for certificate against that standard.

6 References

[1] Software safety standards for the basis of certification in the nuclear domain. Harju, H., Lahtinen, J., Ranta, J,. Johansson, M., Nevalainen, R.. 7th International Conference on the Quality of Information and Communications Technology QUATIC 2010. Porto, Oct. 27 – 29, 2010.

[2] Comparison between IEC 60880 and IEC 61508 for Certification Purposes in the Nuclear Domain. Jussi Lahtinen, Mika Johansson, Jukka Ranta, Hannu Harju, Risto Nevalainen. SAFECOMP 2010 conference, Wien 2010.

[3] Qualification of Safety-critical Systems in TVO Nuclear Power Plants", Juha Halminen and Risto Nevalainen, Software Process Improvement Practice, Wiley Interscience, online, 4th July 2007, pp. 559-567.

[4] Certification of software in safety-critical I&C systems of nuclear power plants. Risto Nevalainen, Juha Halminen, Hannu Harju and Mika Johansson. Article in book "Nuclear Power", SCIYO publisher 2010.

[5] ISO/IEC 15504-2, Information technology – Process assessment - Part 2: Performing an Assessment. 2003.

[6] ISO/IEC 15504 - Part 3: Information Technology - Process Assessment - Part 3: Guidance on Process Assessment, ISO 2004.

[7] ISO/IEC 15504 - Part 4: Information Technology - Process Assessment - Part 4: Guidance on use for Process Improvement and Process Capability Determination, 2005

[8] ISO/IEC 15504-5 FPDAM, Information technology – Process assessment - Part 5: An exemplar Process Assessment Model. 2011.

[9] IEC 61508-3: Functional safety of electrical/electronic/programmable electronic safety-related systems Part 3: Software requirements, IEC 2010.

[10] IEC 60880: Nuclear power plants – Instrumentation and control systems important to safety – Software aspects for computer-based systems performing category A functions, IEC, 2008.

[11] Hevner, A. R., March, S. T., Park, J., and Ram, S. (2004). Design Science in Information Systems Research. MIS Quarterly, 28:75–105.

Author CVs

Risto Nevalainen, Lic.Tech.

Mr. Risto Nevalainen (Lic. Tech.) has long experience in software measurement and quality topics. He has been managing director of Spinet Oy for last 7 years. He is also managing director of Finnish Software Measurement Association FiSMA. His working experience includes position as managing director of Finnish Information Technology Development Center during 1989-1995. Before that he had different research and management positions for example in Prime Minister´s Office, Technical Research Centre (VTT) and Technical University of Helsinki (HUT). Mr. Nevalainen has participated in ISO/IEC 15504 (SPICE) standard development since the beginning. He is Competent SPICE Assessor and ISO9001 Lead Assessor.

Nevalainen has participated in development of SPICE based assessment model for nuclear domain for several years. First version TVO-STEP was intended for safety-related software. Current phase of development is to define an assessment model and method S4N for highest class safety-critical software. Work is part of the Finnish nuclear safety SAFIR 2014 research program.

Timo Mäkinen

Dr. Timo Mäkinen is working as a senior researcher at the Tampere University of Technology, also giving lectures in software engineering and database management. His practical experience consists of engineering and project management positions in software industry. He has published several articles related to software process improvement (SPI). His current research interests include the development of methods and tools for SPI.

Timo Varkoi

Mr. Timo Varkoi (Lic. Tech.) has studied Computer Science at Tampere University of Technology and graduated Master of Science in 1989 and Licentiate of Technology in 2000. His working experience includes both software development and management responsibilities in industrial software organizations. Recently he has been working as a project manager in research and expertise service projects to improve software processes with Finnish software companies. He is a competent SPICE assessor, an active member of the Finnish Software Measurement Association (FiSMA), and the Chair of the SPICE Academy. His current interests include software process assessment and improvement related research, expertise and training combined with development of software intensive organizations. At present he is working as a research project manager at Centre of Software Expertise in Tampere University of Technology, Pori.

Mr. Varkoi is also the editor of ISO/IEC 29110-3 VSE Assessment Guide and the soon to be published update of ISO/IEC 15504-5 Exemplar Software Process Assessment Model, and a member of the Standards Management Group (SWG5) of ISO/IEC JTC1 SC7. He participates in the development of the SPICE based assessment model for nuclear domain, S4N.

TIPA: 7 years experience with SPICE for IT Service Management

Alain Renault, Béatrix Barafort
Public Research Centre Henri Tudor
29, avenur John F. Kennedy,
L-1855 Luxembourg
{alain.renault ; beatrix.barafort}@tudor.lu

Abstract

At the time of the revision of the ISO/IEC 15504 Process Assessment standard for a generic process assessment approach, a R&D project considered the combination of ISO/IEC 15504 and the ITIL de facto standard. Through a Sustainable Service Innovation Process, TIPA (Tudor's IT service management Process Assessment) was developed as a whole framework for assessing IT Service Management processes with "process assessability" concerns. This paper depicts the development of this service innovation framework tackling science-based as well as maturation aspects for market transfer and adoption.

Keywords:

Process assessment, SPICE, service, service management, IT service management, ISO/IEC 15504.

1 Introduction

In the year 2003, the ISO/IEC 15504 has been revised as a generic process assessment standard [15]. At the same time the IT Infrastructure Library (ITIL®) de facto standard was developing quickly and raising more and more interest in the Grand Duchy of Luxembourg. As many companies providing IT services in Luxembourg and the surrounded areas experimented ISO/IEC 15504 software process assessments and needed to strengthen their IT delivering processes, the demand for a common approach for IT process assessment and improvement [3][1][2] emerged.

Public Research Centre Henri Tudor (TUDOR) has been studying and using the ISO/IEC 15504 standard since the mid-nineties for assessing software processes (and using the assessment results for improvement programs), and TUDOR was fast in understanding the interest of applying the generic framework for process assessment to other fields of activity than software and particularly to IT Service Management.

The AIDA research project (AIDA standing for Assessment and Improvement integrateD Approach) was defined in 2003 in order to develop an IT Service Management (ITSM) framework for assessing ITSM processes. The combined use of both standards (ITIL and ISO/IEC 15504) for which TUDOR was an early-adopter became one of the research objectives of the AIDA project [3].

The innovative ideas of the project were born from many issues in companies where the need for improving ITSM processes appeared but there was a lack of an objective and repeatable approach for assessing processes, and a lack of a very structured improvement path. Moreover, similar approaches combining the improvement of software development processes and ITSM ones were missing.

AIDA has since then been renamed as TIPA: Tudor's IT (service management) Process Assessment [16].

This paper is providing a global picture of the R&D activities that have lead to TIPA as an exhaustive framework for the assessment of IT Service Management processes using ISO/IEC 15504.

This paper first explores the research challenges that were tackled for developing compliant but also usable process assessment models. Then it describes the TIPA service development approach before illustrating the use and future of this specific innovation embodied by TIPA. Finally the next steps are presented before concluding.

2 Research Questions

The foundations of the original AIDA project were based on several factors such as:

A. the empiric findings from past use of ITIL and ISO/IEC 15504 standards within companies where software engineering and ITSM aspects were converging
B. The quality approach as experiences by these companies
C. The common terminology provided by standards
D. The process approach promoted by both standards
E. The way ISO standards are implemented on the field

The AIDA project was aiming to bring an answer to the following question: “Does the combined use of ITIL and ISO/IEC 15504 truly increases effectiveness and efficiency and can be adapted to the need of flexibility of today’s organizations?” [3].

In order to address these issues, a first research question was to investigate the possibility to develop a PAM based on ITIL, and to bring value to organizations and consultants in charge of assessing and improving ITSM processes.

The quality aspects of the process models appeared to be very important and also became a research objective in the TIPA context [5]. The process assessment approach is described on a very strong way in the ISO/IEC 15504 standard. However, developing compliant but also usable process assessment models turned to be a real challenge. The standard allows communities of interest to build Process Assessment Models for their own use or particular context. Does the standard, as actually available, really enable anyone to build his own PAM [13]?

So, a second research question was to determine how to improve “*process assessability*” through the design and build of better-formed process assessment models?

3 TIPA service framework development approach

TIPA has been developed, tested, experimented, validated, improved and transferred following the TUDOR Sustainable Service Innovation Process (S2IP) methodological framework for service innovation. In addition, the combined expertise from several fields of activities has enabled to apply specific Requirements Engineering (RE) techniques to support the design of the process models, as described below.

3.1 Framework for the development of a new service

In the service context of Luxembourg, and in the multi-disciplinary approach featuring Services Science, TUDOR has developed S2IP, providing a framework for services managed in a living lab, and then all facilities for several interacting disciplines [6]. S2IP is based on a participatory and collaborative innovation approach in order to sustain deep involvement of the network’s actors in the development of innovation services. The overall structure of S2IP is depicted on **Figure 1.** Each step corresponds to a process by itself that has to be performed and may be pursued in parallel with other processes in a non strict sequence.

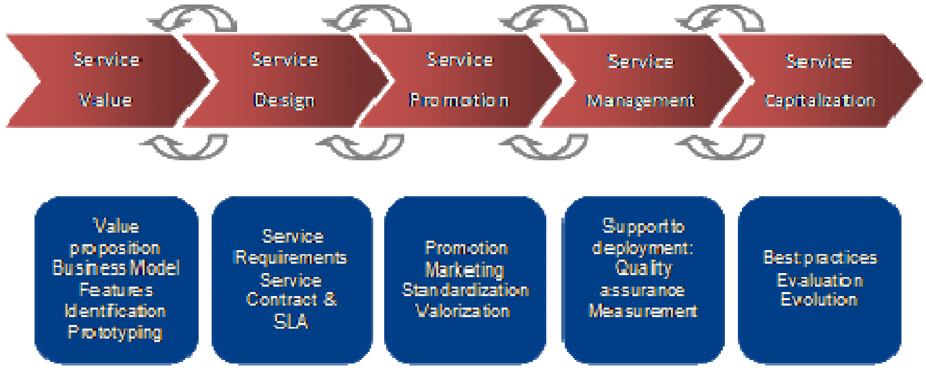


Figure 1 **Tudor Sustainable Service Innovation Process (S2IP)**

1. **Service value** covers the identification of an opportunity for a new service innovation. It covers the study of the technological feasibility of the service (which can require the building of a prototype) as well as a preliminary identification of the business model associated with the value (both expressed in terms of tangible financial elements and of intangible assets).
2. **Service design** covers the definition of the service not only in terms of its business functional objectives but also in terms of all its required qualities. These activities required to elicit the strategies of the different early-adopters stakeholders involved in the final acceptance of the service as well as to understand the constraints associated with the environment (like specific regulations associated with the domain). From this initial elicitation, requirements have to be formally expressed in terms of properties of the services that can be organized in terms of a service contract (or a service level agreement).
3. **Service promotion** comes once early adopters have validated the service contract. It is then important to promote the service to other potentially interested parties. In a network of organizations or for a sector, this promotion can include initiatives regarding the branding of the new service through some label definition and associated certification scheme, or even through national or international standardization.
4. **Service management** is providing tools that can be used by those that will deploy the service for checking and measuring the correctness of its implementation (i.e. metrics associated with the measurement of the quality of the implementation of the services contract).
5. **Service capitalization** is supporting the collection of feedbacks associated with the measures as well as from evaluation performed with the services end-users. The analysis of this feedback will indicate the possible evolution of the service in terms of new requirements, new business model, etc.

3.2 Focus on TIPA design

The Service Design phase was critical for TIPA from a science-based point of view and the quality of performance was a key success factor.

ISO/IEC 15504-2 defines the minimum requirements that a PRM and a PAM shall meet for use in the scope of a process assessment and the exemplar of Process Assessment Model provided in ISO/IEC 15504 5 has often been used as template for building new PAMs as the standard states “it embodies the core characteristics that could be expected of any Process Assessment Model consistent with ISO/IEC 15504-2”. However, there is no guideline on how to build those models, and no guidance is provided to support the transformation process towards a PRM and a PAM if the original description of the process needs sound restructuring, if no process description exists or if the process model is based on a collection of requirements [5].

Being deeply involved in the standardization works, and particularly in the working group dealing with process assessment[1], TUDOR made considerable efforts to come up with compliant, though usable

[1] ISO/IEC JTC1/SC7 W10 Process Assessment

Process Assessment Models. TUDOR has been amongst the first institutions to try and apply ISO/IEC 15504 to non-software development processes [3][4][8], and rapidly, further steps were made to apply ISO/IEC 15504 to non-process description of activities [4][5][7][12]. This experience has helped to the development of a methodology to support the development of ISO/IEC 15504-2 compliant PAMs. This has been applied on the IT service management processes [2] as described by the OGC in ITIL Best Practices.

The first version of this transformation process was considering the original inputs as Process Implementation Model (PIM). After review and experimentation on the PAM, it appeared that having ISO/IEC 15504-2 compliant descriptions of the processes (i.e. with unique process identifier, process name, purpose and outcomes) may not be sufficient and additional work needs to be done to factorize the process descriptions. This can be done thanks to goal-driven requirements engineering (RE) methods [14], which can help analyze, understand, structure, and document better processes in a PRM and in a PAM (**Figure 2**).

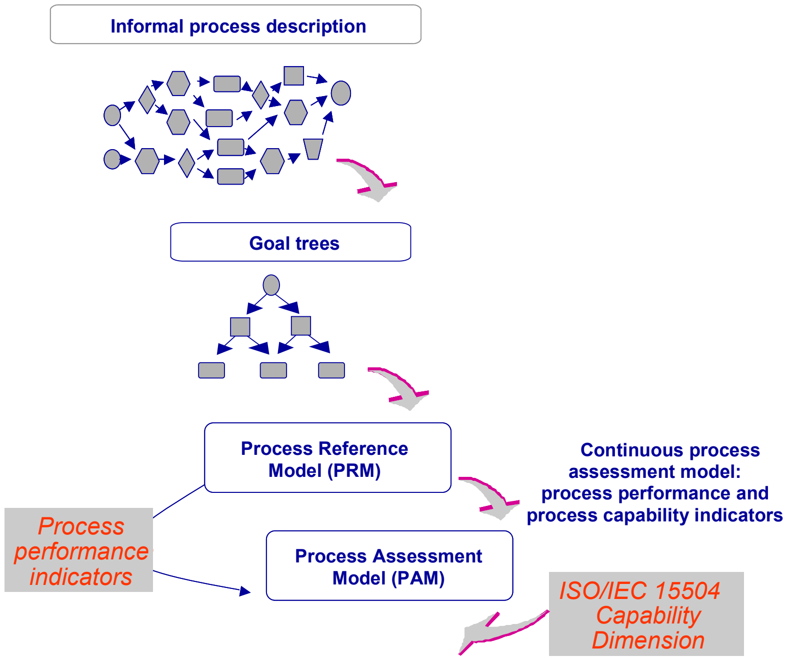


Figure 2 RE goal trees help to structure processes

This technique enables to document a strict traceability of the process models (both PRM and PAM) components via indications of location of information within the original process description document. The use of goal trees appeared to be helpful while negotiating among experts to try to get to a consensus on how a particular process should ultimately be like, as the graphical representation of the purpose decomposed into goals and sub-goals makes it easier to validate each main aspect of the process than with a text paragraph.

Goal-driven RE methods give a simple but effective set of techniques supporting analysis activities conducting to the development of a PRM or a PAM. In addition to the usual benefits of RE methods, the goal-driven style ensures a high-level declarative expression together with rich traceability links; these links are important aspects for analyzing and verifying process models, and can provide helpful support for the identification of processes purpose and outcomes based on a collection of requirements [5].

This transformation process, supported by goal-oriented RE techniques, enables the transformation of original data or requirements into compliant PRM and PAM fulfilling the stakeholders' requirements and needs. It provides the framework for the overall methodology.

As a result, here is an extract from the new TIPA Process Assessment Model for ITIL® v3for Incident Management (**Table 1**):

Process ID	**INC**
Process Name	**Incident Management** *[ITIL v3 - Service Operation:p46]*
Process Purpose	The purpose of the Incident Management process is to restore normal service operation as quickly as possible minimizing the adverse impact on business operations, thus ensuring that the best possible levels of service quality and availability are maintained.*[ITIL v3 - Service Operation:p46]* NOTE 1: normal service operation is defined as service operation within Service Level Agreement -SLA- limits. NOTE 2: The incidents described here can include failures, questions or queries reported by the users, by technical staff, or automatically detected and reported by event monitoring tools. *[ITIL v3 - Service Operation:p46]*
Process Outcomes	As a result of successful implementation of Incident Management : 1. Incident management policies are defined; 2. Incidents are identified and documented; 3. Incidents are investigated to find the most suitable way to solve it; 4. Actions are performed to solve incident and restore the normal service operation; 5. Incidents are tracked all along their life cycle; 6. Customers are kept informed of their incidents progress, and, if necessary, of the service level breaches.
Base Practices	**INC.BP1. Define and agree on incident categories and priorities** Define and agree on incident categories and priorities (including major incidents). *[ITIL v3 - Service Operation:p49]* [Outcome 1, 2] **INC.BP2. Define, agree and communicate timescales for all incident-handling stages** Define and agree on timescales based upon the overall incident response and resolution targets within SLAs. Communicate timescales to all support groups. *[ITIL v3 - Service Operation:p47]* [Outcome 1, 3, 4, 6] **INC.BP3. Detect and log the incidents** Record relevant information about the incident whatever the way of reporting. *[ITIL v3 - Service Operation:p49]* [Outcome 2] NOTE 1: Incidents could be reported by technical staff, users, or communicated by Event Management. NOTE 2: Characteristics about the incident record are detailed in the Work Product characteristics (Annex VI.2). **INC.BP4. Categorize the incidents** Assign the incidents to a type, a category and some sub-categories. *[ITIL v3 - Service Operation:p50]* [Outcome 2] NOTE 3: Example of categories of incident: application, hardware, software, network ... NOTE 4: If Service requests are detected (incorrectly logged as incidents), transfer them to the Request fulfillment process. […]

Input	Output
05_06 Event record [Outcome 2, 3] [INC.BP3,6]	05_07 Incident record [Outcome 2, 3, 4, 5] [INC.BP9,10,4,3,11,6,5]
05_05 Configuration Management System (CMS) [Outcome 3, 4] [INC.BP8,6]	05_04 Incidents knowledge base [Outcome 4] [INC.BP9,10]
08_01 Service Level Agreement (SLA) [Outcome 1, 2, 3, 4, 6] [INC.BP12,2,5,7]	01_02 Incident management tool [Outcome 1, 2, 3, 4, 6] [INC.BP2,1]

05_03 Known Error Database (KEDB) [Outcome 2, 3, 4] [INC.BP6,5]	02_06 Incident model [Outcome 1, 3, 4, 6] [INC.BP2]
05_04 Incidents knowledge base [Outcome 3] [INC.BP6]	02_07 Incident categories [Outcome 1, 2] [INC.BP1]
05_02 Problem knowledge base [Outcome 3] [INC.BP6]	06_05 Customer satisfaction survey [Outcome 4] [INC.BP10]
01_02 Incident management tool [Outcome 1, 2, 3, 4, 5, 6] [INC.BP2,1,4,11]	07_02 Request for Change (RFC) [Outcome 4] [INC.BP9]
02_07 Incident categories [Outcome 2] [INC.BP4]	

Table 1 Extract of the TIPA PAM for ITIL V3

Specific process activities that are obviously connected to a higher level of capability than 1 are mapped to the adequate capability level and process attribute. When this cannot be done, a specific practice is then added to the list of Generic Practices [13], as in the example for Service Reporting (**Table 2**) for the assessment of process attribute PA 3.1: Process Definition:

GP 3.1.6 [ITIL v3: Service Reporting] Define and agree Service Management reports content [PA 3.1: e] Define and agree with the business the lay out, the contents and frequency of the Service Management reports

Table 2 Specific practice for Service Reporting Process

As a result of using this transformation process, the resulting PAM can claim to allow a better "assessability" of process thanks to the following characteristics:
1. Robustness of the process descriptions
2. Distinction between the operational and the organizational (support) activities
3. Traceability to the source information
4. Better mapping to the good capability level
5. Questionnaires built based on additional information provided by the goal trees (like connection between practices)
6. Customization of questions of higher level of capability, depending on specific practices of some processes
7. Validation by a community of international experts

4 *TIPA framework use and future*

The process models are just a small part of the TIPA framework. As already mentioned before, the questionnaires are also playing an important part in the assessment process. Other supporting tools have also been developed to assist assessment and are gathered in the TIPA Toolbox available for all TIPA assessors (Figure 3).

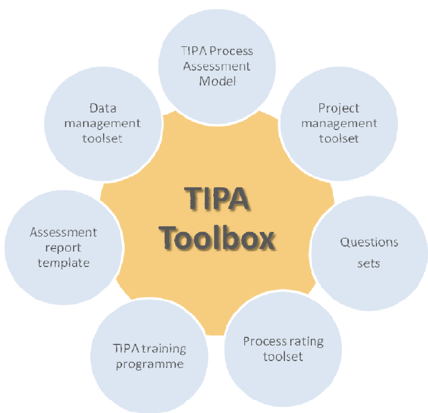


Figure 3 **The TIPA Toolbox**

4.1 How TIPA is used on the market

Validation of TIPA by early adopters worldwide as explained earlier (see 3.1) has largely contributed to the quality of it, and to its international recognition. The actual usage that they have done from the framework illustrates how TIPA can actively support business improvement and innovation.

4.1.1 Global alignment of process capability at Dimension Data

Dimension Data is a specialist IT services and solution provider that helps clients Plan, Build and Support, Manage, Improve and Innovate their IT infrastructures. The company employs 9500 people worldwide. In 2006 Dimension Data began a Service Improvement and Alignment initiative across its Global Service Centers (GSC), which are Dimension Data's "Managed Service" delivery arm. The GSCrs are located across the world and provide 24 hour follow-the-sun Managed Services to regional and international clients. This set of assessments has enabled to:

- Determine the capability level of the processes in the Global Service Centres;
- Identify best practices, to share across Global Service Centres; and
- Propose recommendations for improvement and alignment

But in addition, Dimension Data has also highlighted some interesting advantages for the use of TIPA for such assessment projects [9]:

- Standard and structured approach
- Public domain "methodology"
- Credibility of ISO
- Objectivity of results
- Pertinence of the TIPA model
- Availability of ISO/IEC 15504 resources across the world speaking the same ISO/IEC 15504 language

This first international experience with TIPA, enabled to progressively but fully transfer the mastering of the methodology to the company, the fourth assessment being held without any direct implication of TUDOR in the assessment but the review of the assessment report.

4.1.2 Support to ISO/IEC 20000 preparation at Fujitsu Services Oy

Fujitsu Services is part of the global Fujitsu Group, employing 175000 people in 70 countries. Fujitsu Services is the leading supplier of ICT services in Finland, with 2800 ICT professionals in the Baltic countries.

Fujitsu Services has been using ISO/IEC 15504 consistently to support process improvement since 1997. In 2005, the company decided to gradually replace the traditional internal audits of key processes with ISO/IEC 15504 capability assessments. Since that period, they have performed about ten assessments per year in order to support process improvement projects, internal audits, or to handle customer issues or share knowledge on best practices.

Fujitsu Services has been using the TIPA PAM internally since 2007, three years after they started to implement ITIL. These assessments support their three-year Global Improvement Plan defining improvement objectives. The PAM used has been progressively customized to include some context-specific practices, or requirements from other standards like ISO/IEC 20000-1 and ISO/IEC 27001 to serve the needs of internal audits. It is also used to make a global view prior to its ISO/IEC 20000 certification audits.

Fujitsu Services see many advantages in using TIPA for improving their internal processes [10]:

- It is a structured way to prepare for ISO/IEC 2000 certification;
- It has decreased dramatically the number of internal audits required;
- It helps prioritize the improvement actions;
- It enables internal benchmarking between assessments;
- Assessment results can be easily communicated to customers;
- It highlighted a strong correlation between assessment results and customer satisfaction.

4.2 Other market recognition

With the increasing number of articles and posts published on different websites and forums on the web, TIPA has aroused interest, curiosity and lots of questioning on its added value compared to existing solutions. The critical factor that has really helped in getting wide recognition is the fact that the framework is being based on an international (ISO) standard, offering the level of neutrality awaited by many organizations for such activities or services. This has been surprisingly strong amongst the itSMF community in different countries like France, the USA and the UK.

As a result, an international editor has proposed to support TUDOR in editing a book dedicated to TIPA:" *ITSM Process Assessment Supporting ITIL*" [11].

This market recognition has also lead to the development of a certification program with a partner company (ITpreneurs), giving more value to the competences required to perform such an assessment professionally.

5 Next steps

The certification scheme is designed to develop the knowledge and skills of individuals to enable them to lead and/or participate in IT process assessments based on the TIPA framework. Two training courses and two levels of certification have been logically defined: TIPA Assessor and TIPA Lead Assessor. This will give the guarantee that TIPA assessments are lead and ran by people having an adequate level of expertise for performing ISO/IEC 15504 assessment, together with sound ITIL knowledge and background.

The new TIPA website (http://www.tipaonline.org) will maintain the list of official TIPA certified individuals and companies.

While the original works were based on ITIL V2, TIPA is now available for ITIL V3, and the toolbox is being adapted for the integration of the soon coming ISO/IEC 15504-8 PAM on ISO/IEC 20000.

Other works have also started to allow the integration of multiple standards and frameworks like ISO 9001, ISO/IEC 27001, eSCM and others. This will make TIPA a process governance framework providing mechanisms for a better alignment of business processes and IT, and economy of means.

6 Conclusion

The Sustainable Service Innovation Process has enabled to bring the original AIDA R&D project to a mature framework bringing real value to the market: TIPA.

The framework as documented in TIPA can be used by any ISO/IEC 15504 assessor for structuring and running his assessment projects. The rigorous approach used to develop the process models increases "*process assessability*", bringing more value to both the assessors and organizations undergoing process assessments.

The toolbox supporting the framework can be easily adapted in order to be compatible with any ISO/IEC 15504 compliant PAM. The certification brings credibility to the framework by giving recognition on the level of expertise that is expected from assessors and lead assessors.

The challenge in the near future will be to be able to follow the pace of revisions of the targeted standards to keep TIPA aligned with the evolution of the market needs on one side, and with the state of standardization on the other side

References

[1] Barafort B., Di Renzo B.: Assessment and improvement integrated approach: combined use of the ISO/IEC 15504 (SPICE) and the Information Technology Infrastructure Library (ITIL), Proceedings of the National Conference SPIRAL'2004, Luxembourg, 2004.

[2] Barafort B., Di Renzo B., Lejeune V., Simon J-M, ITIL Based Service Management measurement and ISO/IEC 15504 process assessment: a win – win opportunity, Proceedings of the 5th International SPICE Conference on Process Assessment and Improvement, Klagenfurt, Austria, 2005.

[3] Barafort B., Di Renzo B., Merlan O.: Benefits resulting from the combined use of ISO/IEC 15504 with the Information Technology Infrastructure Library (ITIL). Proceedings of the International Conference PROFES'2002, Rovaniemi, Finland, 2002.

[4] Barafort B., Humbert J-P., Poggi S., Information Security Management and ISO/IEC 15504: the link opportunity between Security and Quality". Proceedings of the SPICE 2006 conference, Luxembourg, May 2006.

[5] Barafort B., Renault A., Picard M., Cortina S. "A Transformation Process for Building PRMs and PAMs based on a Collection of Requirements – Example with ISO/IEC 20000". Proceedings of the 8th international SPICE 2008 Conference, Nuremberg, 2008.

[6] Barafort B., Rousseau A., Sustainable Service Innovation Model: a standardized IT Service Management Process Assessment framework, in Proceedings of the International Conference EuroSPI'2009, Madrid, Spain, 2009.

[7] Di Renzo B., Hillairet M., Picard M., Rifaut A., Bernard C., Hagen D., Maar P., Reinard D., "Operational Risk Management in Financial Institutions: Process Assessment in Concordance with Basel II", SPICE'2005, Klagenfurt, Austria, April 2005.

[8] Di Renzo B., Valoggia Ph., "Assessment and Improvement of Firm's Knowledge Management Capabilities by using a KM Process Assessment compliant to ISO/IEC 15504. A Case Study", proceedings of the SPICE conference 2007, Seoul, South Korea, May 2007.

[9] Hilbert R., Renault A., Assessing IT Service Management Processes with AIDA – Experience Feedback, 14th European Conference for Software Process Improvement EuroSpi, Potsdam, Germany, 2007.

[10] Jokela J., Long Term Utilisation of SPICE in an IT Service Company. Proceedings of the 9th international SPICE 2009 Conference, Turku, Finland, 2009.

[11] ITSM Process Assessment Supporting ITIL®, Van Haren Publishing, ISBN 978-90-8753-564-3, 2009.

[12] Mention, A-L and B. Barafort, "An Open Innovation Framework for Services Process Improvement and Capability Determination in the Banking and Financial Sector Based on ISO/IEC 15504", in: Proceedings of the International Conference SPICE 2008, Nuremberg, Germany, 2008.

[13] Picard M., Renault A., Cortina S., How to improve process models for better ISO/IEC 15504 process assessment, EuroSPI 2010 proceedings (p. 130-141), Springer, Germany, 2010.

[14] Rifaut A. "Goal-Driven Requirements Engineering for supporting the ISO 15504 Assessment Process", EuroSPI 2005 proceedings (p. 151-162), Springer, Germany, November 2005.

[15] Rout T. P., El Emam K., Fusani M., Goldenson D., Jung H-W.: SPICE in retrospect: Developing a standard for process assessment. Journal of Systems and Software 80(9): 1483-1493, 2007.

[16] St-Jean Marc, Mention Anne-Laure: How to evaluate benefits of Tudor's ITSM Process Assessment Proceedings of the SPICE 2009 Conference, Turku, Finland, 2009.

How to assess the software & system testing process:

The Test SPICE approach

Tomas Schweigert, SQS Software Quality Systems AG, Germany

1 Introduction

Since the first complete version of ISO/IEC 15504 was published, it was discussed to enhance the standard by adding exemplar assessment models for system and service engineering and also for security. There were also discussions to introduce an organisational maturity view. Now we have ISO 15504 Part 6 (Systems Engineering) and Part 7 (Organisational Maturity). Up to now no Part # for Software Testing Processes is available. Such a model is needed to transfer the benefits of ISO/IEC 15504 for the industrialisation of the software development to the industrialisation of the software test. In the test business there are 2 major assessment methods available: TPI® NEXT /TMAP® NEXT and TMMI. There is also a standard for the education of test professionals available (ISTQB). The question is: Is one or more of these models compliant to ISO/IEC 15504 II or is there a need for a new model and how it should look like and what needs should it cover?

2 The conformance requirements of ISO/IEC 15504 II

ISO/IEC 15504 II defines a set of conformance requirements. These Requirements address the process reference model (PRM) and the process assessment model (PAM).

2.1 Conformance Requirements for the Process Reference Model (PRM)

To define an ISO/IEC 15504 II conformant PRM, the model must fulfil the formal process description requirements: Describe a process with

- a unique ID,
- a name,
- a purpose
- outcomes and
- prohibit the use of any element of the measurement framework in the process content.

The PRM shall also declare how it fulfils the conformance requirements.

2.2 Conformance Requirements for the Process Assessment Model (PAM)

The PAM shall declare

- the selected PRM(s)

- the selected processes taken from the PRM and the
- capability levels taken from the measurement framework.

It is also required, that the Model describes the mapping between the model and
- The Process Reference Model
- The Measurement Framework

As long as a model for assessing tests will follow these given structure the conformance to ISO/IEC 15504 II is assured. The Test 4 Spice approach will fulfil these essential precondition like shown later.

3 The ISO/IEC 15504 Conformance of current Test Assessment Models

The currently available Test Process Models. There are 3 major models available on the market:
- ISTQB
- TPI®/TMAP®
- TMM®

Let´s have a look if these models fulfil the requirements of ISO/IEC 15504 II

3.1 ISTQB

The International Software Testing Qualifications Board (ISTQB) provides a set of syllabi for the qualification of test people (e.g. foundation level, advanced level: functional tester or test manager and expert level certified test process improver). The ISTQB syllabus provides a fundamental test process: planning and control; analysis and design; implementation and execution; evaluating exit criteria and reporting; test closure activities. The description of processes is heterogeneous. Sometimes a process is described with its purpose, but no explicit description of outcomes is available. Result: The fundamental test process of ISTQB does not meet the conformance requirements of ISO/IEC 15504 II.

3.2 TPI®NEXT/TMAP®NEXT

TPI®NEXT/TMAP®NEXT is the test process assessment and improvement method of a company named SOGETI. The assessment is based on a questionnaire that covers the needs of software testing. The approach uses a staged maturity model. The content is grouped in key areas. Key areas are structured by check points with requirements that are specific for each key area. The Model contains 4 Maturity Levels (initial, controlled, efficient, optimising) Each check point is mapped to one level. So the levels are not generic for every process but specific for each process. As a result a mapping between TPI® NEXT and ISO/IEC 15504 must be available. Due to the characteristic of TPI®Next as a staged model, there is –by using a construct named “enablers” only a little support for mapping to the capability levels and process attributes of ISO/IEC 15504 available in the model. Result: TPI®/TMAP® does only partially meet the conformance requirements of ISO/IEC 15504 II

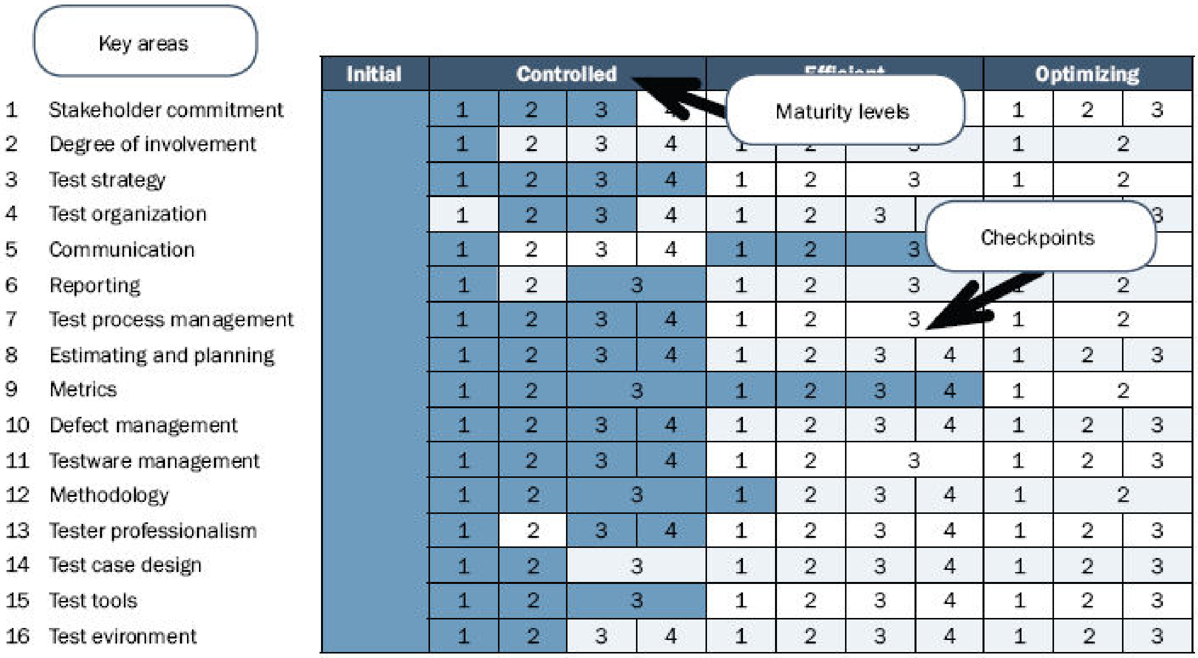
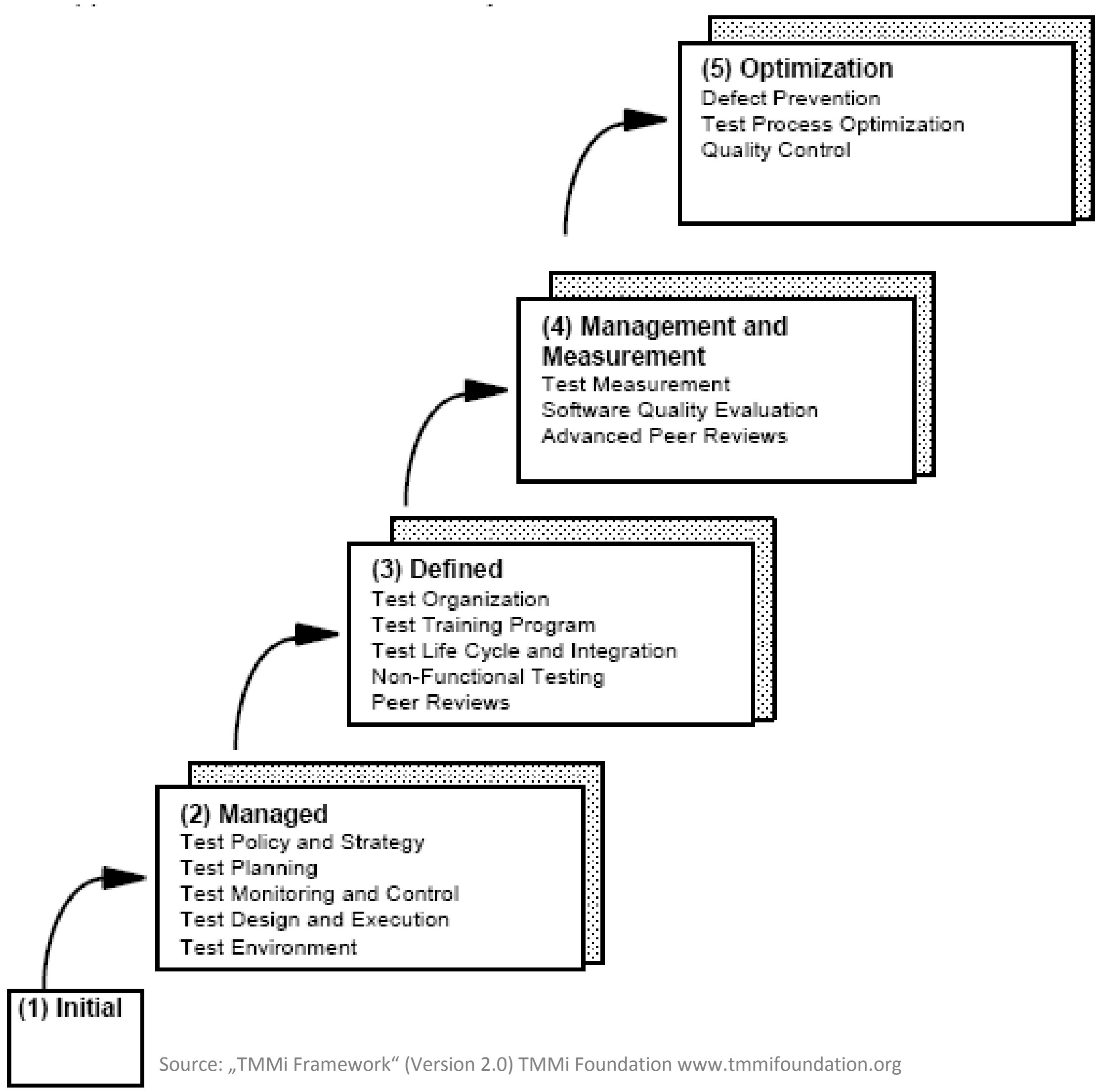


Figure 1: The Test Maturity Matrix of TPI®NEXT

3.3 TMM(sm)/TMMI®

TMM(sm) was initially developed from the Illinois Institute of Technology and is now maintained as TMMi® by the TMMi foundation. The objective of this initiative was to use the CMM®/CMMI® approach for test process assessment and improvement. The current published model is based on the staged approach of CMMi® that means processes are directly linked to maturity levels. Different to CMMI®, TMMi® has no continuous representation. A continuous model allows to define the capability level of each process and to deliver a capability profile. Contrary to the staged model a continuous model has a chance to meet the conformance requirements of ISO/IEC 15504 II. Result: TMMi does not meet the compliance requirements of ISO/IEC 15504 II.



Source: „TMMi Framework“ (Version 2.0) TMMi Foundation www.tmmifoundation.org

Figure 2: The Maturity Levels and Processes of TMMI®

3.4 Intermediate Result

None of the current available test process assessment models meet the conformance requirements of ISO/IEC 15504 II

4 The Test SPICE approach

Objective of the Test 4 Spice approach is, to deliver a PRM and a PAM that both meet the conformance requirements of ISO/IEC 15504 II and cover the processes necessary to effectively and efficiently assure the quality of software products.

4.1 The Basis: ISO/IEC 15504 V

The team developed TEST SPICE with ISO/IEC 15504 V as starting point. This model is structured in process categories, process groups and processes. We decided to use the whole structure.

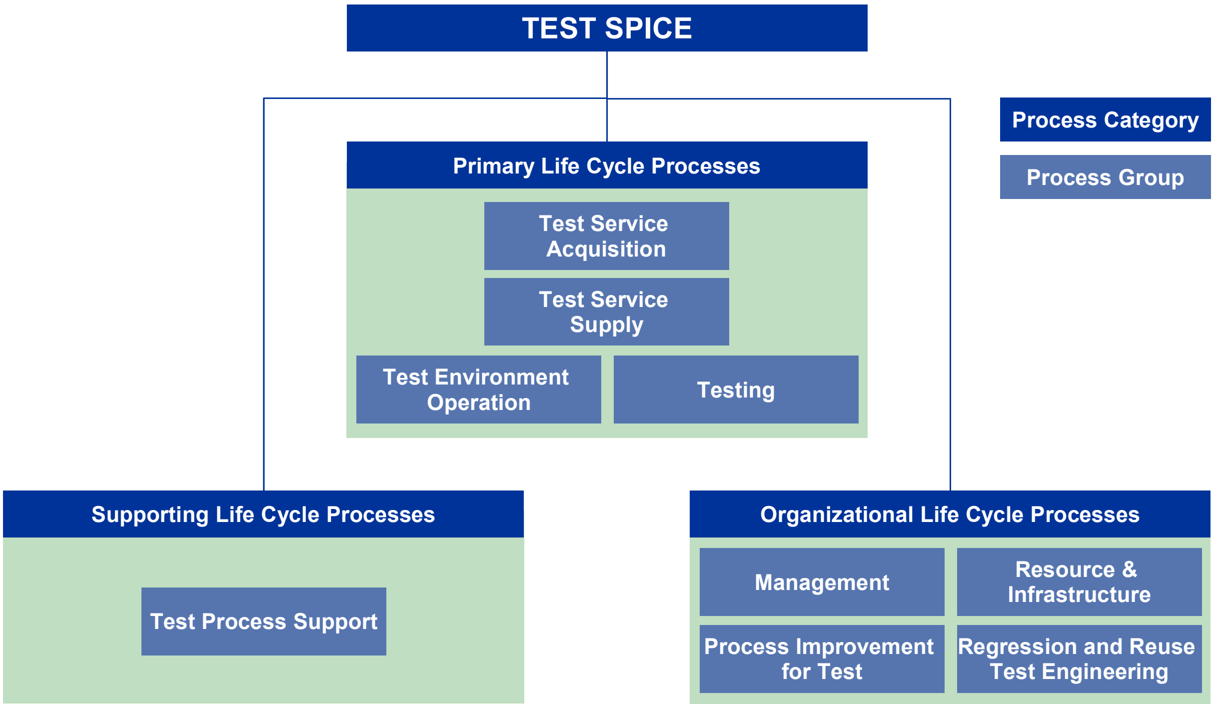


Figure 3: The overall Structure of Test SPICE

To transform the original model we used 3 methods

- Do nothing: a process is transferred 1:1 from ISO/IEC 15504 V to Test SPICE Example: Project Management
- Replace: a process from the original model is replaced by a test process. Example: Domain Engineering was replaced by Regression Test Management
- Rename: Some process groups where renamed to show the focus of their processes. Example: Support was renamed to Test Process Support
- Insert: A new process was inserted Example: Test planning

We did not change the process categories.

Sources

The team interviewed several colleagues from SQS and analysed the current available models (ISTQB, TMAP®/TPI®, TMMI®) and some literature to extract the common ideas about software testing processes.

4.2 The TEST SPICE Model at a Glance

The following figures show the overall content structure of the model:

Primary Life Cycle Processes

Test Service Acquisition
- TAQ.1 Acquisition Preparation
- TAQ.2 Supplier Selection
- TAQ.3 Contract Agreement
- TAQ.4 Test Service Monitoring
- TAQ.5 Test Service Acceptance

Testing
- TST.1 Requirements elicitation
- TST.2 System requirements analysis
- TST.3 Test requirements analysis
- TST.4 Test Analysis & Design
- TST.5 Test Realization and Execution
- TST.6 Test Results analysis and Reporting
- TST.7 Test automation design
- TST.8 Test automation implementation
- TST.9 Test environment testing
- TST.10 Testware maintenance

Test Service Supply
- TSP.1 Test Supplier tendering
- TSP.2 Test Service Delivery
- TSP.3 Test Service Acceptance Support

Test Environment Operation
- TEO.1 Operational Use of Test Environment
- TEO.2 Test Environment User Support

Figure 4: The primary life cycle processes of TEST SPICE

Supporting Life Cycle Processes

Test Process Support
- TPS.1 Quality assurance
- TPS.2 Verification
- TPS.3 Validation
- TPS.4 Joint review
- TPS.5 Audit
- TPS.6 Product evaluation
- TPS.7 Documentation
- TPS.8 Configuration management
- TPS.9 Problem resolution management
- TPS.10 Change request management

Organizational Life Cycle Processes

Management
- MAN.1 Organizational alignment
- MAN.2 Organization management
- MAN.3 Project management
- MAN.4 Quality management
- MAN.5 Risk management
- MAN.6 Measurement
- MAN.7 Organizational Test Strategy
- MAN.8 Test Planning
- MAN 9 Test Execution and Controlling
- MAN.10 Test Closing and Reporting
- MAN.11 Organisational Test Policy

Resource & Infrastructure
- RIN.1 Human resource management
- RIN.2 Training
- RIN.3 Knowledge management
- RIN.4 Test Infrastructure

Process Improvement for Test
- PIT.1 Process Establishment
- PIT.2 Process Assessment
- PIT.3 Process Improvement

Regression and Reuse Test Engineering
- RRT.1 Test Asset management
- RRT.2 Test Work Products Reuse Management
- RRT.3 Regression Test Management

Figure 5: The Supporting and Organizational Life Cycle Processes of TEST SPICE

4.3 Some Aspects of the TEST SPICE Model

The team had to consider some aspects that had influence for the model:

- The best practices of the ISTQB syllabi should be integrated
- The best practices of 30 years testing at SQS should be integrated as well
- The model should be E2E in the sense that you don't need to perform an additional assessment to become aware that a project problem is linked to poor requirements or a lack of reviews.
- The model should reflect the project and the organisational context in a sense that you don't need to perform an additional assessment to become aware that a so called test problem is linked to poor estimating or time planning.
- The model keeps the life cycle integration of ISO/IEC 15504 in a sense that you don't need another PRM to assess the acquisition of test services or the supply of test services.
- The model also reflects, that test automation is a big issue. Some times test automation using a script language means nearly to implement the features in another environment. So test automation has to be required, designed and tested.

4.4 ISO/IEC 15504 II Conformance of TEST SPICE

As TEST SPICE is derived from ISO/IEC 15504 V using the characteristics and indicators of ISO/IEC 15504 V it is easily possible to refer to the conformance statements of ISO/IEC 15504 V

There is one exclusion because the model has another community of interest. So TEST SPICE cannot refer to the community commitment of ISE/IEC 15504 V. As ISO/IEC accepts a statement, that consensus is not yet reached or not planned, the working group decided to keep the conformance as follows:

- For this version no commitment is planned.
- For the next version the community of interest will be the 1500 testing people at SQS their commitment will be assured and documented
- For a future version it might be that a commitment will be reached at ISTQB level. But that is not yet discussed or planned.

5 The current Roadmap of Test SPICE

The PRM and the PAM are published at INTACS and the SPICE User Group.

The conformance of Test SPICE is verified by INTACS

A Test SPICE SIG is formed that held its first meeting in Sindelfingen, Germany

The training syllabus and material will be available latest until the end of July

The Test SPICE SIG is in negotiation with leading providers of assessment tools which aim an agreement that these providers will provide the Test Spice Model to users of these tools.

There are nearly 20 Test SPICE Assessments performed and SQS plans to sell another set of up to 80 Assessments.

6 ISO/IEC Standardisation of software testing and the assessment of testing processes

Currently we see 2 ISO/IEC standards for testing processes in progress

- ISO/IEC 29119 PRM for software and systems testing
- ISO/IEC 33063 PAM for software and systems testing.

Both standards will have an impact on the future development of Test SPICE

- They define the core processes of software testing. The TestSPICE SIG has to analyse this and establish compliance
- They don't cover the whole testing business. Some core features are missing:
  - Test outsourcing from the customer and provider perspective
  - Test automation and test environment management
  - Integration of project and test management

7 Summary

If we consider ISO/IEC 15504 as an open standard for process assessment and improvement especially for the IT industry then this open standard should also be applied for test processes. The first version of this model shows, that this is achievable. As a benefit for the IT industry there is no longer a need to translate the results of proprietary models to the ISO/IEC 15504 measurement framework which saves money for training (one measurement framework fits all), data collection and analysis.

8 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. Tomas 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.

Requirements engineering in large scale European funded projects

Jesper Berthing[1], Timo Feismann[2], Swapnil Gandhi[2], and Zlatko Petrov[3]
(the author list is in alphabetic order)

[1] Danfoss Power Electronics A/S
[2] Delphi Deutschland GmbH
[3] Honeywell International s.r.o.

Abstract

Use of intelligent control systems involving use of micro-controllers is increasing exponentially in various domains. There are multiple applications across industrial, automotive as well as aerospace domain, where correct behaviour of such computers is vital for public/environmental safety. Various national and international standards are used for development of such safety critical systems in order to keep tract of complete product deployment (from concept to production) process. Cost of proving compliance with industry specific standard (IEC 61508 for industrial, ISO 26262 for automotive, DO-178B and DO-254 for aerospace, etc.) is generally very high. With a minor change in the system, a re-certification of complete system is needed, adding significant resource pressure.

To reduce (re-)certification as well as overall system costs, ARTEMIS funded RECOMP [1] consortium is developing methods, tools and reference platforms for emerging Multi-core micro-processor technology. RECOMP aims to capture requirements from various layers of industries (OEMs, component suppliers, tool providers etc) and propose a series of different methodologies which are conformed with domain specific safety standards.

This paper provides detailed information about requirements-engineering phase of RECOMP. It lists challenges faced while collecting requirements from large number of consortium partners (from different industrial domains) and methods adopted (solutions) to make requirements engineering (elicitation, sorting, traceability and verification) conform to highest level of effectively.

1 RECOMP Introduction

1.1 Introduction to RECOMP

The RECOMP (Reduced certification costs for trusted multi-core platforms) research project aims to establish methods, tools and platforms for enabling cost-efficient cortication and re-certification of safety critical and mixed-criticality multi-core based systems. The applications addressed in RECOMP are automotive, aerospace, industrial control systems, lifts and transportation systems.

There was recently a paradigm shift and semiconductor industry seeks nowadays higher performance solutions based on growing number of cores on a die. RECOMP recognizes that it is unlikely that the semiconductor industry will over high-performance single-core systems in the next 5-10 years and the increasing processing power of embedded systems is mainly provided by increasing the number of processing cores. The increased numbers of cores is commonly regarded as a design challenge in the safety-critical area, as there are no established approaches to achieve certification. Furthermore, there is an increased need for flexibility in the products in the safety-critical market. This need for flexibility puts new requirements on the customization and the upgradability of both the non-safety and safety-critical critical part. The difficulty with this is the large cost in both effort and money of the re-certification of the modified software, which means that companies cannot fully leverage the advantages of modular software system.

The RECOMP project will bring clear benefits in terms of cross-domain implementations of mixed criticality systems in all domains addressed by project participants: automotive systems, aerospace systems, industrial control systems, lifts and transportation systems.

The participating and collaborating partners aim to achieve at least the following commercial advantages (for more details one may refer to [1]):

- Lower total system cost
- Lower cost of the certification of first version of the product
- Shorter time-to-market
- Lower re-certification costs
- Supplier-based development process

1.2 RECOMP -Work-Package Structure. WP1 Goals & Organization

RECOMP is structured into 8 work-packages (WP), where WP1 sets out to collect the system level requirements from the Automotive, Aerospace and Industrial automation domain. The requirements are then addressed in WP2, WP3 and WP4, aiming at providing design methods and tools, platforms and certification aspects related to reducing certification costs for multi-core platforms. The results of these work packages are then applied in WP5 where demonstrators for the different domains are developed. The demonstrators are then used in WP6 to validate with particular focus on the main certification improvement aspects and provide feedback to the work packages. WP7 focus is on the dissemination activities of the project. Project management and administration is handled in WP8 and supports the work performed in WP1-WP7. The overall project workflow is depicted in Figure 1, representing the relation and information flow between WP1 to WP6.

The goal of WP1 ”Research Drivers” is to establish both the business case of the approach as well as the exact requirements for the solutions as they apply to different domains. The business cases will establish goals that have to

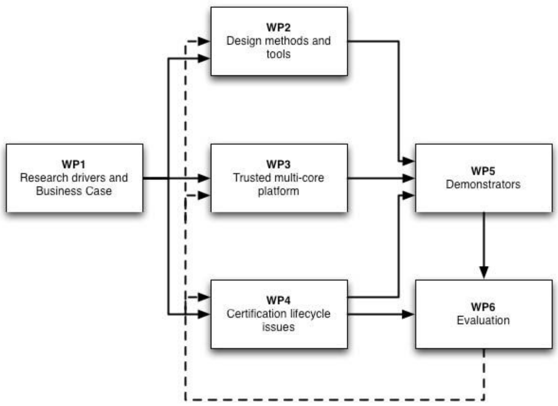


Fig. 1. Overall Project Workflow

be achieved, and that can be assessed in the Evaluation work-package (WP6). The requirements will provide the specific constraints and problems that have to be solved in work-packages WP2, WP3 and WP4. This work-package will also identify the demonstrators that will be used in WP5.

Work package 1 consists of three tasks. The purpose of Task T1.1 ”Business cases and requirements elicitation” is to establish different business cases. Furthermore, these business cases drive the identification of high level requirements for each research driver. Task T1.2 ”Requirements consolidation and scope definition” aims at sorting and rearranging the high-level requirements proposed throughout activities in T1.1 and initial scope definition of the RECOMP project. Finally, Task T1.3 ”Requirements refinement and demonstration goals establishment” breaks down into detailed technical and non-technical (e.g. legal) requirements coming from T1.1 through T1.2. The detailed requirements at the end form a basis for demonstrator development in WP5. T1.3 furthermore proposes metrics and fit criteria for the proposed and refined requirements, such as to form a basis for a validation plan to be worked out in WP6. The process of requirements establishment and refinement form a WP task perspective is exemplified in Figure 2.

2 Requirements Engineering

2.1 Process of Deriving Domain and Cross Domain Requirements

This section provides more information about methods, tools and activities carried out in WP1 in order to address WP1 objectives and meeting WP1 expectations.

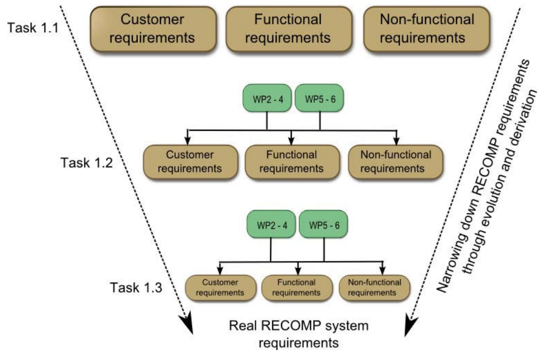


Fig. 2. RECOMP Requirements engineering from WP1 task perspective

Requirements Establishment Process. The main goal of WP1, namely to provide detailed domain-specific and cross-domain requirements for RECOMP multi-core platforms is definitely challenging. It is not a simple matter of gathering and writing down what OEMs from different RECOMP domains want. Indeed this is our starting point, and further through derivation, evolution and consolidation we refine and establish the RECOMP domain-specific and cross domain requirements. Figure 2 exemplifies the process of requirements establishment and refinement from a task perspective. Two main principles have been taken into account in the course of requirements establishment. The first principle ”*Horizontal division of requirements*” imposes that the scope of RECOMP is not limited to any particular type of industry. Methods and processes developed within RECOMP shall be applicable to several domains - automotive, aerospace and industrial automation. The RECOMP consortium is consisting of industrial partners coming from different domains that will bring along requirements particular to their specific application fields. The purpose is to identify, collect and structurally document all functional and non-functional requirements. The so called ”research drivers” or business cases coming from each industrial partner have been used to form the basis for the initial top level requirements establishment. The second principle - ”*Vertical division of requirements*” imposes that requirements should be allocated to the three typical layers in an embedded system, namely application, operating system & middleware, and hardware. The vertical direction is in addition complemented by requirements that the RECOMP toolsets shall meet and/or requirements for certification related processes extensions. Figure 3 exemplifies the discussion so far and gives the vertical split of requirements and their relation to other WPs within RECOMP.

Going into more details, Task T1.1 aims:

- To define the very first version of RECOMP system requirements by determining, analyzing, prioritizing high level system requirements. This phase provides the very first goals, functions and constraints of HW/SW multi-core systems, i.e. to limit the space exploration of WP2, WP3 and WP4

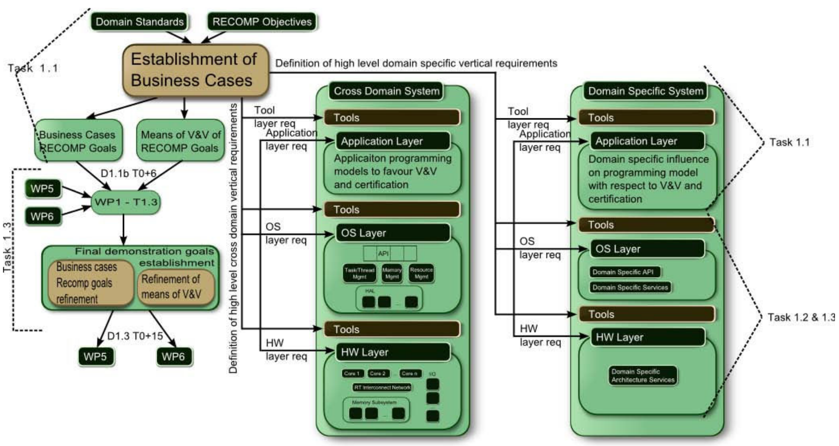


Fig. 3. Vertical split of requirements and relation to other tasks and WPs

- To try to group different domains requirements into 2 groups, as shown in Figure 3 - cross-domain requirements and domain-specific ones. The latter should consists of 3 sub-groups, namely aerospace, automotive, and industrial
- To provide high level system requirements

At the definition of high level goals and constraints that WP5 should achieve through demonstration and means of validation and verification of those goals, given RECOMP's objectives and all application domains being investigated in RECOMP. This is the very first input to WP5 and WP6. Those goals and means of verification and validation are at the disposal of WP5 and 6 working groups for review and assessment of their feasibility from WP5 and WP6' perspectives - see Figure 3. Furthermore, tasks T1.2 and T1.3 aim to refine and detail T1.1's high level requirements by:

- taking into consideration WP's specific requirements as those that T3.1, T3.2, T4.1a, T4.1b establish and additional ones from WP2 that may appear after initial investigation phases on the detailed WP2's work flow

- Partitioning and allocation starting from application layer going down to HW, SSW, and tools layers. Again those should be group into 2 main categories - cross-domain and domain-specific requirements, as depicted in Figure 3.

T1.3 is additionally in charge of refining business goals and means of their verification and validation by taking into consideration additional WP5 and 6' specific requirements that may appear after initial investigation phases on the detailed WP5 and 6' work flows - this process in depicted in both Figure 2 and Figure 3.

Overall the presented work flow and adopted principles should lead to detailed technical as well as non-technical (e.g. legal) requirements. Functional requirements as well as safety requirements that form a basis for research and development work to be done in WP2, WP3 and WP4. Detailed rational requirements form a basis for demonstrator development in WP5. To already existing or planned demonstrators, requirements can be allocated as appropriate.

The work carried out towards establishment and refinement of vertical direction requirements leads to establishment and refinement of demonstration goals along with metrics and fit criteria for requirements verification, such as to form a basis for a validation plan to be worked out in WP6.

Safety requirements will be listed according to the needs of specific industry and will form a basis for the safety case to be certified within WP4.

One may note that the process of requirements establishment is an evolving one and initial requirements will change over project life time. For instance it is very likely that level of details and priorities might be a subject of change, dictated by the necessity of taking into account different constraints constituted in result of deeper tasks and objectives analysis in other WPs or simply due to afford-ability issues. Thus we consider review processes both internal (within WP1 activities) and external (review from other WPs) as substantial step for setting up the right expectations and establishing realistic requirements still fulfilling RECOMP's top objectives and goals.

Excel for Requirements Engineering. At the beginning of RECOMP and the start of WP1 a common tool for requirements engineering was needed that every partner could work with and that was available from the beginning. The usage of either DOORS, Caliber, Word or Excel was discussed. Finally Microsoft Excel was chosen because of the highest availability, lowest cost and easiest to use for requirements management. The first step was to set up a template document that includes all the attribute needed for proper requirements engineering. A set of the requirement attributes that are currently in use are listed in Table 1.

2.2 Requirements Review Process

The requirements collected during WP1 will be addressed and used by WP2, WP3 and WP4, it is therefore of importance that the system level requirements specified in WP1 are communicated and understood in the other work packages. Internal and external review of the requirements has been carried out to ensure the quality and communication of requirements.

The internal review have been carried out at the workshops arranged within WP1 where all the requirements have been reviewed by the WP1 participants, to ensure common understanding and alignment of common domain requirements and domain specific requirements.

The external reviews have been guided by the allocation of the individual requirements to specific work packages, enabling review sessions with each work package participants. The review sessions were arranged using an internet-based conferencing system, to enable feedback from as may of the project participants.

Table 1. RECOMP requirement attributes

Requirement attribute	Description
RequirementID: *<Participant No.>-<Req. ID>*	Unique ID that consists of the domain (IDR - Industrial; ARR - Aerospace; AUR - Automotive; CDR - Common Domain), the RECOMP participant number (to see who originated the requirement) and a running number that must only be used once from each participant. The reuse of number from deleted requirements is also not allowed.
Domain: *<ARR, AUR, IDR, CDR>*	Redundant to the content of the requirement-ID (can be IDR, ARR, AUR or CDR).
Short Description: *<Req. Name>* **Description:** *<Req. Description>*	These attributes contain the text for the requirements. The short description is just to give a brief overview of the content the requirement is dealing with. The description attribute contains the complete text.
Verification Method: *<Description how to verify>*	The verification method gives a short description (or basic thought) on how to verify a requirement. This attribute is only filled if the type is also set to requirement.
Rationale: *<The rationale behind this req.>*	This attribute shall give the reader the rationale behind the requirement in case that the requirement does not give this rationale implicitly.
Importance: *<Low, Medium, High>*	The "Importance" attribute give the implementers the information on what is most important and therefore needs to be implemented first. It is used to prioritize implementation.
Object Status: *<New, Changed, etc.>*	This attribute tell the reader whether this object is accepted, in review, obsolete, new or changed.
Review: *<Comment>*	This attribute is used to write down review comments (with date and originator).
Assigned to WPX *<YES, NO>*	Here the requirement is assigned to one or more work packages by selecting yes or no.

The participants received the requirements and a defect log (supplied by one of the companies) to record and classify the defects found. At the time of review there were 82 requirements allocated to WP2, 204 requirements allocated to

WP3 and 154 requirements allocated to WP4 out of 376 requirements in total. The review sessions resulted in 39 defects for WP2, 159 defects for WP3 and 137 defects for WP4 (See Table 2 for an overview of requirements and defects). The requirements allocated to WP5 at that time were specific to the functionality of the demonstrators and was therefore not part of the external review sessions.

The requirements have then been modified according to the review defect logs by the requirement owners and a response was provided for each defect in the review defect log.

Table 2. Overview of requirements and defects

WP	Requirements	Defects
WP2	82	39
WP3	204	159
WP4	154	137

2.3 Verification Methods

The purpose of WP1 is to gather all top-level requirements for all domains (industrial, aerospace and automotive). Within all these requirements a set of top-level common domain requirements are identified. IEEE STD 830-1993 (Recommended Practice for Software Requirements Specifications) defines nine qualities for requirements specification.

– Complete - All external behaviors are defined

– Unambiguous - Every requirement has one and only one interpretation

– Ranked for stability - Each requirement rated for likelihood to change, based on changing expectations or level of uncertainty in its

– Correct - Every requirement stated is one that software shall meet

– Consistent - No subset of requirements conflict with each other

– **<u>Verifiable</u> - A cost-effective finite process exists to show that each requirement has been successfully implemented**

– Modifiable - Structure and style are such that any changes to requirements can be made easily, completely, and consistently while retaining structure and style

– Traceable - Origin of each requirement is clear, and structure facilitates referencing each requirement within lower-level documentation

– Ranked for importance - Each requirement rated for criticality to system, based on negative impact should requirement not be implemented

– Ranked for stability - Each requirement rated for likelihood to change, based on changing expectations or level of uncertainty in its description

The definitions given in IEEE STD 830-1993 are followed to a high degree in WP1. One very important point in the list above is ”verifiable”. To follow the recommended practice in IEEE STD 830-1993 all the requirements in WP are delivered with a verification method.

Why are verification methods needed at this rather early stage in development although there are almost no systems available now? Safety-critical systems such as the ones developed within the RECOMP project should be developed with thorough verification. When developing systems, it is a good idea to be able to show the customer that the system works. This applies especially if the system is going to be a safety-critical one!

The system development in RECOMP should follow the classic V-Model shown in Figure 4. With this model, testing explicitly starts at the very beginning, i.e. as soon as the requirements are written. This helps in identifying errors very early in the lifecycle and minimizes potential future defects appearing in the code later in the lifecycle.

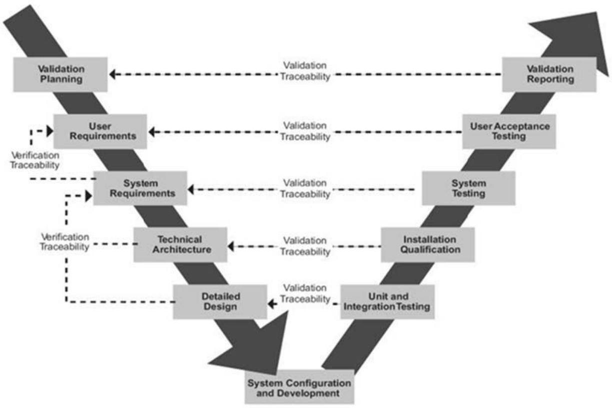


Fig. 4. V development model

That is why inside WP1 all the requirements are listed together with a basic method description of how to verify them. Of course the given verification methods are no direct test case descriptions (because the requirements only show top-level requirements) but they can give guidance where to start when creating the test descriptions for the system.

Below is an example of how a verification method description looks like.

Detailed test descriptions including verification and validation will be done in WP6. The top-level natures of many requirements worked out in WP1 only allow some conceptual verification done in an assessment. Each and every requirement needs to be verified. A requirement is verifiable if an argument can be constructed that tells the tester the system fulfills this requirements. The argument needs to be supported with sufficient objective evidence. The WP1 requirements mainly require two different types of verification methods:

– Definitive

Short Description: *<Req. Name>*	Description: *<Req. Description>*	Verification Method: *<Description how to verify>*
Binding of specific software components to specific hardware hosts	RECOMP component-based design methodology shall support binding of specfic software components to specific hardware hosts	Verify that component-based methodology support binding of specific software components to specific hardware components (e.g. threads to cpu) either using heuristics or a cost-effective allocation algorithms. Allocation shall be done taking into account the criticality levels from different domain standards.
MCP shall support 'Safe-State' transition of individual applications	More than one safety critical application is present on MCP all with different 'Fail-Safe' strategies. It shall be possible for each application to fall back in to its respective fail-safe state without affecting normal functioning of other applications	Test has to be performed using failure injection mechanisms of the μ C. Failure should be injected and the expected reaction is that only the affected function transitions to the respective fail safe state

- Results are quantitative
- Can be compared directly to the requirements
- Results can be stated as pass/fail

- Analytic
  - For requirements that cannot be definitively verified (e.g. requirements that need to be verified by a safety assessment)

2.4 Functional Safety Requiremements

Introduction. One of the most important objectives of RECOMP is to develop methods and technologies that can make implementation of mixed-criticality functions on one ECU. Such an implementation should also help bring down certification and re-certification costs. This objective leaves us with ’functional safety’ as one of the key issues to be taken care of during the tenure of RECOMP. System safety can be achieved through a number of safety measures, which can be implemented with the help of variety of technologies (e.g. mechanical, hydraulic, electrical, electronic, programmable electronic etc.) and is influenced by the development process (V-Process), as well as production and management activities. The international safety standard ISO26262 as well as IEC 61508 define activities to be carried out during concept phase of system development. Such a task is supposed to deliver top level ’functional safety requirements’. These requirements may include item definition, the initiation of the safety lifecycle, the hazard analysis and risk assessment and the functional safety concept. [3] A reference safety life-cycle as specified by ISO 26262 is shown below in Figure 5.

The requirements elicitation task in RECOMP WP1 targets performing hazard and risk assessment as well as definition of functional safety concept. With reference to above diagram, said WP1 involves partners (who wish to provide demonstrator in WP5) providing hazard and risk assessment of respective function, specification of safety goals and specification of top-level functional safety requirements. Said functional safety requirements will be derived from detailed functional safety concept.

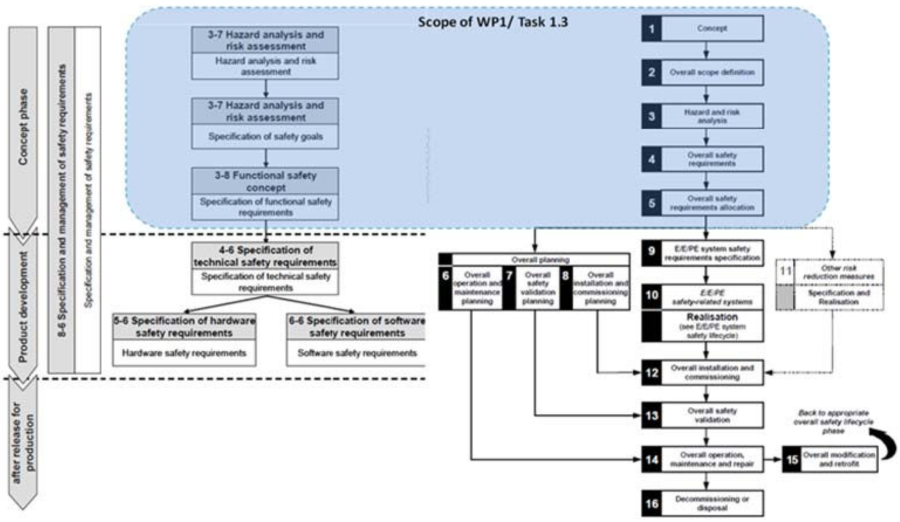


Fig. 5. Safety life-cycle as specified by ISO 26262 [3] and IEC 61508 [2]

Hazard and Risk assessment (H&R). Detailed system description as done in ’item/system’ definition phase is used to perform H&R analysis. While considering the system for H&R analysis, all safety mechanisms (internal/external) that are already implemented are not considered [3]. The task of identifying and analysing hazards consists van be systematically broken down in several phases (stages). Number of phases involved during hazard analysis, as well as required documentation during individual stage can vary with safety criticality of a function [4]. Major hazard analysis activities and related documentation is depicted in figure below 6.

End result of an H&R analysis provides list of potential risks a function carries under different situations, safety criticality classification of every listed risk (SIL/ASIL), and safety goal associated with each risk. If applicable, safe state information of a function (after a fault leading to hazard is detected) is also provided with above mentioned information. Figure 7 below summarizes activities and results of H&R analysis phase.

Results of H&R analysis phase, namely, safety criticality classification of a function, safety goal as well as safe state specifications are used to set up functional safety requirements. Safety criticality level further provides guidelines concerning so called safety metrics. Such safety metrics are domain specific. For example ISO 26262 speaks about 'single point failure metric' and 'latent fault metric', while IEC 61508 speaks about 'safe failure fraction' metric. All RECOMP partners who are willing to demonstrate multi-core design related technology have agreed to provide detailed H&R analysis and related outcomes as described before. In case of aerospace domain, respective standards will be used to derive system requirements. The derivation of functional safety requirement may not look similar to that of industrial and automotive domain but will have work products which should describe use of multi-core related methods developed during RECOMP to achieve required safety of a function in consideration.

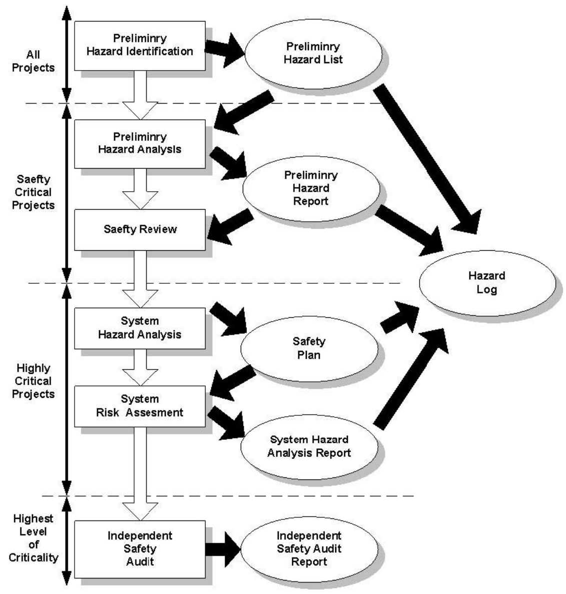


Fig. 6. Hazard and Risk analysis tasks (Source: [4], page 52)

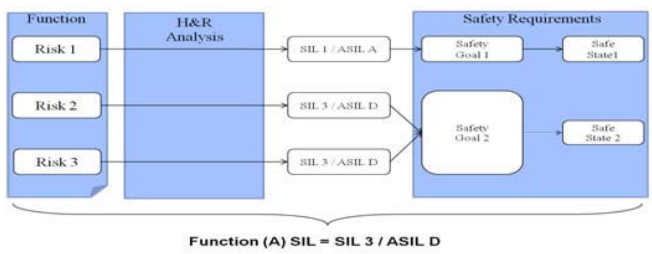


Fig. 7. Results of the Hazard and Risk analysis

Functional safety concept. *According to ISO 26262,* "*the objective of the functional safety concept is to derive functional safety requirements from safety goals (and safe state definitions) and to allocate them to preliminary architectural elements of the item or system*" [3] System architectural requirements are derived such that concerned system conforms to respective safety metrics. Both IEC 61508 and ISO 26262 over substantial help for determining mechanisms of system design in order to design a safety critical system. Respective safety standards also provide development process, hardware design and detailed software design requirements which are based in safety criticality of considered function. In general functional safety concept address requirements concerning

- Ability of system to detect fault and transition to defined safe state (with detailed information about time needed for such a transition)
- If applicable, fault tolerance mechanisms implemented within the system, ensuring thereby that such a fault is not violating any of already set safety goals
- Most appropriate user (e.g. vehicle driver) information mechanism about a detected system fault. Such a mechanism is generally meant to reduce the risk exposure time to some acceptable interval.

Within work-package 1, task 1.3, all the partners showcasing their demonstrators will provide a functional safety concept. Although there are several commercially available tools available in the market to document a safety concept, format for collecting such a safety concept within RECOMP consortium is Microsoft-word.

Functional safety requirements. The functional safety requirements are specified according to industry specific safety standard (e.g. automotive domain according to ISO 26262, industrial domain according to IEC 61508 etc.). According to both standards, at least one safety requirement is needed to be specified for each of defined safety goals. As shown in the Figure 8 below, Functional safety requirements are derived from safety goals and safe states as defined within hazard and risk analysis document. It assumes certain preliminary system-architecture.

Such assumptions are then need to be verified with the help of safety assessment methods. Such requirements then form basis for development of methods and technologies within other work packages of RECOMP.

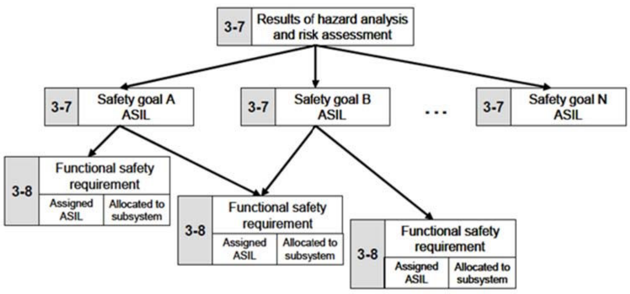


Fig. 8. Safety requirements are derived from safety goals (source: ISO 26262)

Wherever applicable, functional safety requirements are supposed to provide information about system safety mechanisms like function operating modes, fault tolerant time interval, functional safe states, emergency operation interval (if no safe state can be achieved), functional redundancies (e.g. fault tolerance), driver information(warnings) etc. Verification of such a functional safety requirements (and hence functional safety concept) can be performed by methods like FMEA, FTA etc. Such assessment can be documented together with the set of functional safety requirements in order to provide a complete version of functional safety requirements.

2.5 Examples of Requirements in RECOMP

At the time of this paper, WP1 has collected 364 requirements in total. In Table 3, an overview of the requirement allocated to the different work packages (in relation to the domains), as well as total number of work package, common domain and domain specific requirements. It should be noted that a requirement may be allocated to more than one work packages.

The amount of WP5 requirements for AUR and IDR, reflect demonstrator specific requirements for the automotive domain and requirements related to IEC 61508:2010 for the industrial domain.

The number of common domain requirements reflects the effort the WP1 participants to specify requirements that are common for the three domains.

Table 3. Allocation of requirements to work packages and domains

	WP2	WP3	WP4	WP6	Total
CDR	37	96	91	11	142
ARR	33	21	13	0	56
AUR	2	29	11	50	64
IDR	4	74	71	60	102
Total	76	220	186	121	

A selection of common domain and domain specific requirements have been selected from WP2, WP3 and WP4 (See Table 4) as an example of the requirements collected in WP1 for RECOMP. The requirement ID (Req. ID) indicates the relation to the domain and the work package.

3 Conclusions

A paradigm shift in the way semiconductor industry seeks nowadays higher performance solutions imposes multi-core processors as key components for the design of future embedded systems. However, current platforms, design methodologies and tools make the deployment of product based on multi-core technology hosting safety- and mixed-critical applications very complex if not impossible to get certification credits.

Table 4. Example requirements from WP2, WP3 and WP4

Req. ID	Short description	Description
CDR-WP2-001	Seamless design flow	RECOMP component-based design methodology shall support a seamless design flow - from initial requirements specification down to implementation. Modelling of different levels of abstraction with refinement relations between abstracts is a must
CDR-WP2-002	Binding of specific software components to specific hardware hosts	RECOMP component-based design methodology shall support binding of specific software components to specific hardware hosts
CDR-WP2-003	Support software of different criticality	The design methods and tools shall support separation of software with different criticality, e.g. non-safety critical software from safety-critical software
AUR-WP2-001	Tools qualification	Tools used for development of MCP shall show their compatibility in fulfilling 'Tooling requirements' as specified by ISO26262
IDR-WP2-001	Safety Life Cycle	The Safety Life Cycle and relating activities used to develop multi-core systems shall comply with the guidelines set out within IEC 61508:2010 parts 1-4
CDR-WP3-001	Mixed criticality	Mechanisms shall be provided to enable the execution of s/w tasks with different safety criticality levels on a single core
CDR-WP3-002	Non-safety critical upgrade	An upgrade or update of the non-safety critical application shall not require either functional or timing related changes to safety critical applications on the same multi-core application platform
IDR-WP3-001	Hardware Fault Tolerance	HFT shall be 1 to fulfil SIL3
CDR-WP4-001	Certification guidelines	Provide certification guidelines for enabling mixed criticality application to execute on the same core
CDR-WP4-002	Documentation guidelines	Specify required documentation level for certification of software component allocation on multi-core devices

Furthermore multi-core based solutions are an ideal candidate for hosting mixed-critical application partitions while offering reduced size, weight and power. Since the non-critical application partition is often a subject of updates and upgrades, then a solution for product re-certification by simply reusing certification credits that have been already received for critical application partitions is considered to be a key factor for a successful adoption of multi-core technology in mixed-critical applications and lowering the total system development cost. The RECOMP project fills in these gaps by proposing, developing and validating design methods, tools and reference platforms for multi-core design of safety- and mixed-critical applications in areas of automotive, industrial control and aerospace. The RECOMP project starts by requirements gathering, elicitation and refinement from various stakeholders in the entire product deployment chain. The proposed requirements afterwards drive the design methods, tools and reference platforms being a subject of research and development in another project work-packages. This paper discusses and proposes an approach for requirements engineering in large scale projects as the ARTEMIS project RECOMP. We have discussed some of the challenges faced and how these have been handled. Simple, but yet well working tools have been proposed for requirements capturing, elicitation, sorting, tracing and verification. Internal and external requirements review processes have been used in order to achieve consortium-wide deep understanding and acceptance of requirements, while refining towards final versions of requirements. A verification method for each requirement have been established to be a prerequisite for requirements acceptance, thus achieving traceability.

In the context of safety- and mixed-critical applications functional safety is of utmost importance. The paper illustrates the process of deriving functional safety requirements in automotive domain by initial discussion about hazard and risk assessment and further discussion about functional safety concept. This paper gives at its end a summary of proposed requirements, their allocation to other RECOMP work-packages and their split per domain, including cross domain requirements.

Acknowledgements

This work has been partially funded by the National Authorities involved in RECOMP and the Advanced Research & Technology for Embedded Intelligence and Systems (ARTEMIS) within the project RECOMP under Grant agreement no. 100202. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of funding agencies.

The authors would like to thank all partners involved in WP1 and all WP leaders for constructive review feedback.

References

1. R. C. http://atc.ugr.es/recomp/. Reduced certification costs using trusted multi-core platforms, 2011.
2. International Electrotechnical Commission. *IEC 61508 - Functional safety of electrical/electronic/programmable electronic safety-related systems, Edition 2.0, 2010-04.*
3. International Organization for Standardization (ISO)-26262, March 2010. *Road vehicles Functional safety Part 3: Concept phase. ISO26262-Part 3.*
4. N. Storey. *Safety Critical Computer Systems*. Addison-Wesley, 2001.