

Eur SPI'2004

European Software Process Improvement

10.-12.11.2004, SAS Radisson Hotel, Trondheim, Norway

<http://www.eurospi.net>

Eur  SPI'2004

EuroSPI 2004 Industrial Proceedings

Partnership

ASQ, <http://www.asq.org>
ASQF, <http://www.asqf.de>
DELTA, <http://www.delta.dk>
ISCN, <http://www.iscn.com>
SINTEF, <http://www.sintef.no>
STTF, <http://www.sttf.fi>

Supporters

Norwegian Technical University
SPIKE Project
Talent@IT Project

Technical Report 07/04

ISSN-NO: 1503-416X

NTNU Norges teknisk-naturvitenskapelige universitet

Fakultet for informasjonsteknologi, matematikk og elektroteknikk

Institutt for datateknikk og informasjonsvitenskap





TECHNICAL REPORT
07/04
ISSN-NO: 1503-416X

TITEL EuroSPI 2004 Industry Proceedings	DATE 10.-12.11.2004
	NO. OF PAGES 284
SUPERVISOR Prof. Dr Tor Stalhane, NTNU, Norway	
AUTHOR/ Editor Richard Messnarz (ISCN) and Mads Christiansen (DELTA)	
PROJECT EuroSPI 2004 – European Software Process Improvement	

ABSTRACT

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

EuroSPI conferences present and discuss practical results from improvement projects in industry, focussing on the benefits gained and the criteria for success. Leading European industry are contributing to and participating in this event. This year's event is the 11th 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.

3 KEYWORDS

Software Process Improvement
Management
Measurement

EuroSPI 2004 Proceedings

Proceedings

The papers in this book comprise the industrial proceedings of the EuroSPI 2004 conference. They reflect the authors' opinions and, in the interests of timely dissemination, are published as presented and without change.

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

EuroSPI

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

EuroSPI conferences present and discuss practical results from improvement projects in industry, focussing on the benefits gained and the criteria for success. Leading European industry are contributing to and participating in this event. This year's event is the 11th 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.

Publisher

ISSN-NO: 1503-416X , Norwegian Technical University

EuroSPI Chairs

General Chair Dr Richard Messnarz, ISCN

Industry Chair Bernd Hindel, ASQF

EuroSPI 2004 Local Chair Torgeir Dingsoyr & Nils Brede Moe, SINTEF

Scientific Programme Committee Chair Torgeir Dingsoyr & Nils Brede Moe, SINTEF

Industrial Programme Committee Chair Jorn Johansen & Mads Christiansen, DELTA

Industrial Programme Committee Co-Chair Risto Nevalainen, STTF

Exhibition Chair Robert Treffny, ASQF

Tutorial Chair Dr Richard Messnarz, ISCN

Industrial Programme Committee

Geir Amsjøl, UiO, Norway

Bo Balstrup, Danfoss, Denmark
Gualtiero Bazzana, Onion, Italy
Andreas Birk, SD&M, Germany
Mads Christiansen, DELTA, Denmark
Ioana Ene, Onion, Italy
Tom Gilb, Norway
Uwe Hehn, Methodpark, Germany
Tim Hind, Axa Sun Life, UK
Greta Hjertø, HiST, Norway
Janne Jarvinen, SolidTech, Finland
Jørn Johansen, DELTA, Denmark
Conny Johansson, Ericsson, Sweden
Kurt Crone Jørgensen, Grundfos, Denmark
Even-Andrè Karlsson, Q-Labs, Sweden
Peter Læssøe, Nykredit, Denmark
Richard Messnarz, ISCN, Ireland
Edda Mikkelsen, DNV, Norway
Kim Møller, Ericsson DiAx, Denmark
Risto Nevalainen, STTF, Finland
Mark Paulk, Carnegie Mellon University, ISRI, USA
Poul Grav Petersen, Novo Nordisk Engineering, Denmark
Jan Pries-Heje, IT University of Copenhagen, Denmark
Ian Seward, JP Morgan, UK
Rini van Solingen, LogicaCMG, Netherlands
Harald Störrle, University of Munich, Germany
Lisa Tipping, QinetiQ, UK
Tor Ulsund, Bravida, Norway
Niils Jakob Villmones, Kongsberg Spacetec, Norway
Otto Vinter, DELTA, Denmark
Hans Westerheim, SINTEF ICT, Norway
Capers Jones, Software Productivity Research

EuroSPI Board Members

ASQ, <http://www.asq.org>
ASQF, <http://www.asqf.de>
DELTA, <http://www.delta.dk>
ISCN, <http://www.iscn.com>
SINTEF, <http://www.sintef.no>
STTF, <http://www.sttf.fi>

Editors of the Proceedings

Dr Richard Messnarz, ISCN

Mads Christiansen, DELTA

Sonja Koinig, ISCN GesmbH

Welcome Address by the EuroSPI General Chair



Dr Richard Messnarz

EuroSPI 2004 has grown in size and content. EuroSPI has also developed from purely a conference series to an initiative with 3 major goals (www.eurospi.net):

1. An annual EuroSPI conference supported by Software Process Improvement Networks from different EU countries.
2. Establishing an Internet based knowledge library, newsletters, and a set of proceedings and recommended books.
3. Establishing an effective team of national representatives (in future from each EU country) growing step by step into more countries of Europe.

EuroSPI established a an experience library (library.eurospi.net) which will be continuously extended over the next years and will be made available to all attendees. EuroSPI also established an umbrella initiative for establishing a European Quality Network in which different SPINs and national initiatives might join mutually beneficial collaborations. Thus I expect for 2005 that through EuroSPI partners and networks, in collaboration with ISQI (International Software Quality Institute) a number of IT professions can be trained and officially certified (e.g. Certified ISQI project manager, certified ISQI software architect, etc.). Also I expect that EuroSPI partners will closely collaborate to form a group of national institutions in Europe representing accredited ISO 15504 training and certification institutions. And EuroSPI 2005 will re-use the successful structure of EuroSPI 2004 and offer a 4 stream conference with a mixture of research and industry presentations, with research and industry proceedings. Finally, keep in mind what companies stated about EuroSPI :” ... the biggest value of EuroSPI lies in its function as a European knowledge and experience exchange mechanism”.

Welcome to Trondheim and NTNU by Our Major Supporter



Tor Stalhane

PC Member

It is a great pleasure to welcome you all to EuroSPI 2004 here in Trondheim, the home of the Norwegian University of Science and Technology - NTNU. The EuroSPI conference, with its combined focus on academic theory and industrial practice, goes well together with the goal of our university - to produce people with a solid academic background, dedicated to solve practical problems for the industry.

As one of the main sponsors of this year's EuroSPI conference, we feel sure that the event's presentations with a mixture of industrial and academic papers will create interesting discussion and new insights that you can take with you to improve your organisation or use as a starting point for new research.

Last but certainly not least: even if it is pretty dark outside most of the time now in November and you have a lot of important SPI related things to do, you should still take some time off to sample the life of Trondheim - the capital of Trønderlag. It is definitively worth a visit.

Welcome Address by the EuroSPI Industrial Programme Chairs



Jorn Johansen



Mads Christiansen

It is a great pleasure to welcome you the first Industrial Track of EuroSPI. The EuroSPI 2004 has been redesigned compared to the previous EuroSPI conferences with one of the major differences being the division into two tracks: one for research and one for industrial experiences. We note with pleasure the extensive contribution to the Industrial Track as we have been able to accept 32 papers - a lot more, than we hoped for from the start.

As consultants in software engineering, DELTA has made more than 70 BOOTSTRAP assessments (mostly in Denmark but also in e.g. Poland, USA and Germany) and we are now also introducing CMMI to assist our customers to focus on improving their most critical processes. We can see a pattern that matches many of the papers in this proceeding i.e. we do find common problems and solutions. This is important and it is vital that the software community share their experiences to inspire and to improve our SPI efforts.

Welcome to Trondheim and SINTEF by the EuroSPI Local Chair



Nils Brede Moe

Local Chair

As head of the local Chair of EuroSPI'04, it is a great pleasure to welcome you to Trondheim and SINTEF. The SINTEF Group is the largest independent research organisation in Scandinavia. We generate new knowledge and solutions for our customers, based on research and development in technology, the natural sciences, medicine and the social sciences. The EuroSPI conference with its combined focus on research and industrial practice goes well with our wish to contribute to the creation of value and to a society in healthy sustainable development. The conference will cover all the nice and important topics in SPI from Process Improvement Methodologies and Technologies, Business Strategies, Human Factors, Knowledge, and Innovation. Hopefully you will get new insight and share some experiences with others. I hope you will have some time to experience both the charm and intimacy of the small town and the plethora of choices of big-city life in Trondheim. You will find the people of Trondheim friendly and sociable. Trondheim is a very hospitable city, with its doors open wide to the world.

Contents

Session I1-A – SPI and Management

Project manager's guide to the Galaxy . The ultimate tool for running software development projects?	I1-A.1
<i>Nils Jakob Villmones</i>	
ManagEUr – Success Criteria for EU Project Management	I1-A.15
<i>Miklos BIRO and Richard MESSNARZ</i>	
Experiences in the Assessment of the Requirement Management Process	I1-A.23
<i>Jose A. Calvo-Manzano, Gonzalo Cuevas, Tomas San Feliu, Ariel Serrano, Magdalena Arcilla</i>	

Session I1-B – SPI and Management

SPI Managers' Experiences and Opinion On SPI In Practice	I1-B.1
<i>Nathan Baddoo and Tracy Hall</i>	
From Waterfall to Evolutionary Development (Evo) Or How to create faster, more user-friendly and more productive software	I1-B.21
<i>Trond Johansen</i>	
AXA IT Organisation & Process Maturity Assessment Framework	I1-B.31
<i>Tim Hind</i>	

Session I1-C – SPI and RUP

Improving the object-oriented design process	I1-C.1
<i>Javier Garzás and Mario Piattini</i>	
Adopting RUP to defined project types: a preliminary experience report	I1-C.5
<i>Gunnar Nordseth and Geir K. Hanssen</i>	
Tailoring the Unified Process	I1-C.9
<i>Christian Stensholt</i>	

Session I1-D – SPI Case Studies

Beyond Software process improvement: A case study on change and knowledge management	I1-D.1
<i>Jan Pries-Heje & Pouya Pourkomeylián</i>	
The Software Development Culture of Northern Ireland	I1-D.13
<i>Donald McFall and Fergal McCaffery and George Frederick Wilkie</i>	

Session I2-A – SPI and Engineering

A Proposal for an Integrated Advanced Engineering Process	I2-A.1
<i>Uwe Hehn and Bernd Hindel</i>	
Continuous tailoring activities in soft-ware engineering	I2-A.11

P. Saliou and V.Ribaud

Merging Agile Methodologies - The Case for DSDM and XP I2-A.19

Andrew M Goulding and Rory V O'Connor

Session I2-B – SPI Case Studies

User Expectations for an SPI Knowledge Management System I2-B.1

Ilmari Saastamoinen, Markku Tukiainen, Timo Mäkinen, Timo Varkoi

Stages of Software Process Improvement - Based on 10 Year Case Studies I2-B.7

Miklós Biró, Katalin Balla, János Ivanyos, Richard Messnarz

Evaluation of using a UML methodology in a large embedded development project I2-B.19

Hanne Kristin Thorsen, Bente Anda, Kai Hansen and Ingolf Gullesen

Session I2-C – SPI and Organisational Strategies

Building a learning organisation: experience with establishing communities of practice I2-C.1

Hege Dreiem, Rune Myrdal, Kjetil Jørgensen-Dahl

Process Improvement is Not Enough – Do not Forget Innovation Processes I2-C.9

Richard MESSNARZ, Romana Vajde Horvat, Katja Harej, Eva Feuer

Prioritizing SPI actions based on business needs and impacts I2-C.17

Matti Haavikko, Mika Johansson, Risto Nevalainen

Session I3-A – SPI and Estimation

Software Productivity estimation based on Association Rules I3-A.1

S. Bibi, I. Stamelos, L. Angelis

Navigating the Minefield -- Estimates Before Requirements I3-A.7

Carol A. Dekkers

Quantitative Project Management . Integration of Balanced Scorecard and CMMI Maturity Level 4 I3-A.15

Carsten Højmosse Kristensen and Peter Voldby Petersen

Session I3-B – SPI and Requirements Management

Developing a RE Process Improvement Model I3-B.1

Sarah Beecham, Tracy Hall and Austen Rainer

How Evaluation Techniques Influence the RE-tool Evaluation: An Experiment I3-B.11

Raimundas Matulevicius, Lena Karlsson, Guttorm Sindre

A Business-Driven Requirement Process I3-B.17

Rolf Nergaard

Session I3-C – SPI and Risk Management

Risk-driven Software Process Improvement - a Case Study I3-C.1

Jakub Miler , Janusz Górski

Increased Value through Reduced Risk in Joint Industry /External Researcher Projects – Towards Guidelines	I3-C.9
<i>Lars Bratthall, Torbjørn Skramstad</i>	
Software Quality Models: Decisions by Indian companies	I3-C.15
<i>Brendan Keane & Ita Richardson</i>	
Session I3-D – SPI and Systems	
NOEMI, a collaborative management for ICT process improvement in SME: experience report	I3-D.1
<i>Bernard DI RENZO, Christophe FELTUS, Sylvie PRIME</i>	
Practical Experience of Virtual Team Software Development	I3-D.9
<i>Valentine Casey & Ita Richardson</i>	
Process improvement through development of an extended electronic process guide - From electronic process guide to integrated work tool	I3-D.17
<i>Ken Rune Nilsen</i>	

Project manager's guide to the Galaxy

The ultimate tool for running software development projects?

Nils Jakob Villmones, Kongsberg Spacetec
nils@spacetec.no

Abstract

Project management is a crucial part of software development, and in order to improve software development, software project management needs to be improved as well. To create a computer/intranet based tool for this, one need to have a vision of better overview, simpler planning, follow-up and reporting to management as objectives for the project manager.

Kongsberg Spacetec has developed a Project Web, which is based on an underlying Process Guide, describing all relevant processes for software development and other project types in the company. This presentation outlines how the project manager can use the Project Web as a tool to accomplish, not all, but most of his/her tasks running a project. The requirements a project manager has to put to such a tool are briefly covered.

The article also gives some statistics on how much the Project Web has been used since introduction in August 2003 and trends in use, off-springs like the Maintenance Project Web and the Proposal Web which are connected to the Project Web in a time dependent manner. Also, productivity figures for company project management efficiency will be displayed.

The plans for further development of the Project Web will be outlined – what does it lack, what needs to be improved?

The author's view is that this tool is the ultimate help a project manager may wish to have to do his/her job thus defending the title – “Project Manager's Guide to the Galaxy.”

1 Background

1.1 The company

Kongsberg Spacetec AS of Norway is one of the leading producers of receiving stations for data from meteorological and Earth observation satellites (optical and SAR). Since Kongsberg Spacetec was founded in 1984 its products, including the MEOS™ (Multi-mission Earth Observation System), has been delivered to a number of clients around the world. Our export share is currently 85%.

Our speciality is operational systems for remote sensing, and associated services. Kongs-

berg Spacetec has expertise in electronics-, software development and applications. Currently Kongsberg Spacetec has 60 employees, 80 % of the staff has master degree in physics or computer science.

1.2 From engineering to product development

From the start in 1984 the main tasks of the company was engineering through customer specific projects. More specifically the main customer was ESA. The methodology and the processes was instructed and supervised by the customer. Kongsberg Spacetec adapted and followed the ESA PSS-05 engineering standards for all projects even when the customer did not have any requirements.

During the 1990s the market situation changed for Kongsberg Spacetec. A new kind of customers became increasingly important. The new kind of customer was not interested in how the product was developed or how we performed quality assurance. Instead of providing detailed requirements specifications they expected off-the-shelf products that could be delivered on a short notice. These customers were not interested in buying a unique product, on the contrary they preferred products that had a history and a widespread use. In return for lack of uniqueness the customer demanded a much lower price, it became impossible to demand enough for a product to cover the complete development costs.

The obvious consequence for the company was that it became necessary to develop generic products through internally financed and managed projects.

2 Introduction: A tool for better Project Management

Project management is a crucial part of software development, and in order to improve software development, software project management needs to be improved as well. To create a computer/intranet based tool for this, one need to have a vision of

- better overview
 - simpler planning
 - tighter follow-up
 - easier reporting to management
- as objectives for the project manager.

Project Web

Kongsberg Spacetec has developed a *Project Web*, which is based on an underlying Process Guide, describing all relevant processes for software development and other project types in the company. This presentation outlines how the project manager can use the Project Web as a tool to accomplish, not all, but most of his/her tasks running a project. The requirements a project manager has to put to such a tool are shortly covered.

To aid the project manager in establishing and planning his/her project, a Blast-Off process divided into sub processes like preliminary Handover ending with Kick-Off has been defined to aid the project manager in establishing and planning his/her project. This concerns the establishment of a Work Breakdown structure, how to estimate work packages, use case analysis, definition of schedule, tailored phase related check lists, milestones, resource planning, organisation and composition of the project team including the different roles to fill. Risk analysis is supported and later maintained through status and progress reporting. The project is divided into phases that can be made iterative, and each phase has its process description which is instantiated in the Project Web; e.g. phases for specification, design, testing, system

realisation, delivery/installation and project termination with optional post mortem analysis at the end. Email registry for the project is linked into the Project Web.

Of phase independent processes, progress reporting is streamlined as long as possible with all relevant information available and auto generated, like remaining work estimates, resource allocation changes, payment plan, cost budget, deliverables status and graphs for man hours budgeted and used, prognosis for future work and process progress planned and actual by the checklists that have to be checked out for each sub process. Special Alert Status exists for technical problems. Progress may be reported at any time, not rigorously periodically, but may be each week, each third month or even every day if so needed. The Project Web has full communication with the company time accounting system providing updated man-hours recorded at any time, and interacts with the purchasing database and invoicing archive.

Project meetings are supported with Minutes of meetings template and actions lists connected, and requirements management and economy/budget insight are provided. Milestones planning and follow-up are included.

Several attached databases service the project, like Product database, Delivery database and Requirements database for tracing requirements through project lifetime. A special database for bug reporting is available for use in the acceptance testing and (later) warranty stages.

The project web shows project status at any time concerning milestone and overall progress; reporting status and economy figures, risks and actions statuses. Colours are used to flag deviations like red for alerts, delay and alarming conditions, yellow for concern and warning and green for status as planned. Also there are hyperlinks to all relevant documents like Proposal, Contract, Architectural Design, and test documents. Templates to all necessary documents are provided for use.

The Project web gives total visibility of the project including e.g. detailed economy figures to all project members, but also to middle and upper management and all interested employees. This is a deliberate strategy for total openness to provide full information to anyone concerned.

MEOS KSPT Project WEB

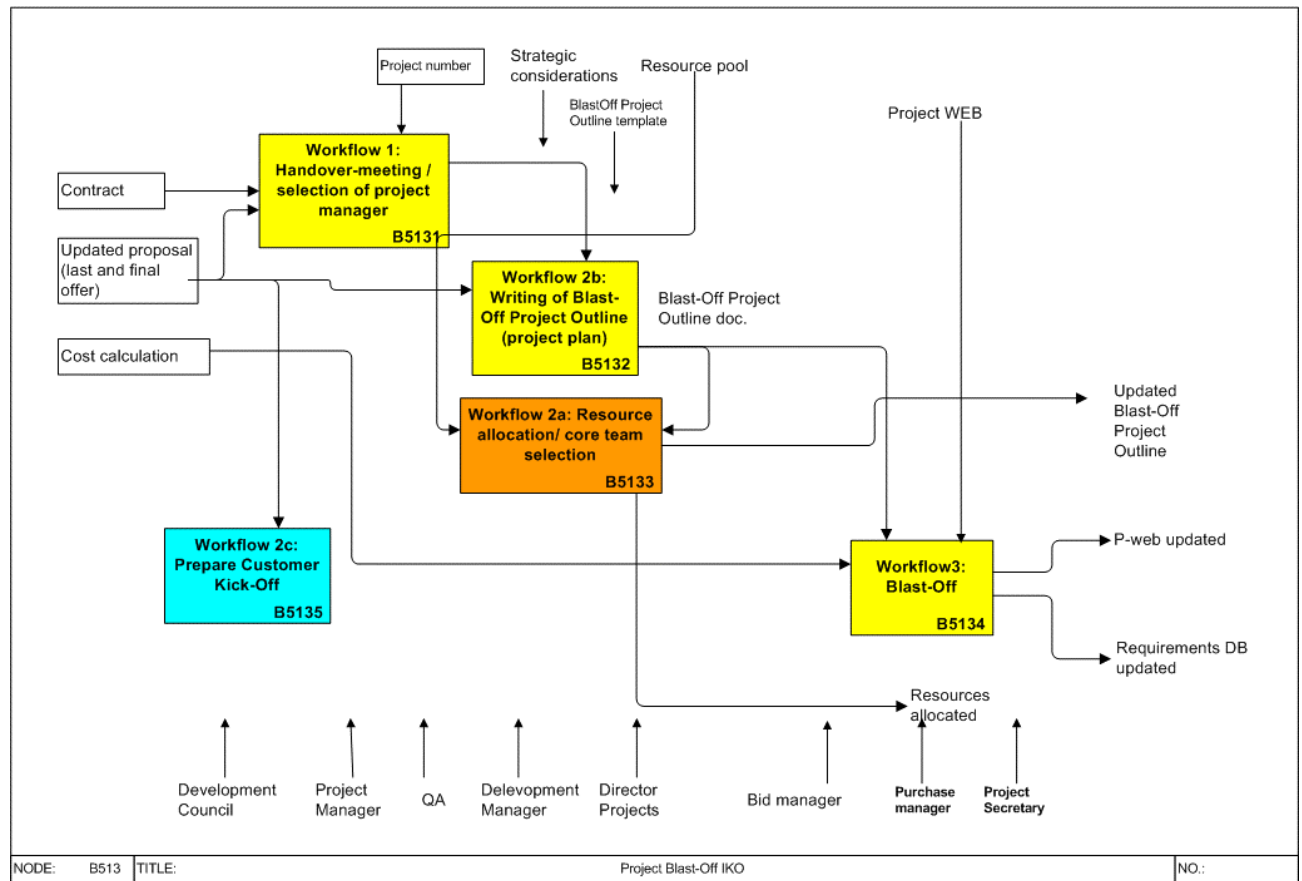
41 Open Projects

Status	Project Name	Project Nr.	Started	End	P. Report	Economy	Alert	Team	Customer	Risk	Act.
OK	ICEMCON Kwa 1	8339	20030310	20040901	OK	OK	OK	IT (PM), HRS CORN TQA: JACOT GEVER	ESRIN	0.0	
OK	PADARSA T 2	8333	20030221	20041201	OK	OK	Warning	GLR (PM), GLR LFG MS IA NMI NEM (RH) (TR) TQA: JACOT GEVER	KSAT	0.1	
OK	ADA S-DUEEP	8341	20030319	20041001	OK	Warning	Warning	AMY (PM), HG, SS, HB, AM, JAJ, SH, AJH TQA: SE	Alcatel Bell Space	0.4	
OK	IPAS	8342	20030420	20031212	OK	Warning	OK	SH (PM), KAS, KI TQA: EKO	CSD	1.0	
OK	MODIS operasionalizing	8117	20030901	20041030	OK	Warning	OK	Frode Dinesen (PM), HD, II TQA: Tere Tawarasa	NE.S	0.0	
OK	20030914	20041001	OK	Warning	OK	ston (PM), DRK, KI, E& MH, GEO, JMS			

<Figure 1> Project Web front page

3 Starting up the project: The Blast-Off process

To aid the project manager in establishing and planning his/her project, a Blast-Off process, divided into sub processes, like preliminary Handover, ending with Kick-Off have been defined to aid the project manager in establishing and planning his/her project.



<Figure 2> BlastOff process

3.1 Planning the project

3.1.1 Work Breakdown Structure

A Work Package *standard* is the default Work Breakdown Structure, containing general work packages for project management, configuration control and quality assurance, but also development specific, like Requirements Analysis, Architectural Design and Code and Unit Testing.

The project manager edits the WBS to satisfy his needs for his projects, decides dates for start and finish (duration) of all individual work packages and may estimate work effort in man-hours (if so not already done).

Project workpackets:

ID	Task	Resp.	Effort	Used	Remain	Description	Status	Dep
2000	Quality Assurance	TQA	21		10	Project Technical QA	Open	
3000	Requirements Analysis	TBD	1		10	Requirements analysis WPs	Open	
4000	Architectural Design	ANY	33		10	Architectural design iteration	Open	
4800	AD Test	KRN	10		10	Test architectural design (Review)	Open	
4900	ATP Elaboration	TBD	1		1	Create outline ATP	Open	
5000	Component DD	TBD	1		1	Detailed design of one component	Open	

<Figure 3> Work Breakdown Structure

3.1.2 Use case analysis

A use case analysis is carried out to demonstrate the project manager’s understanding of the project objectives. To accomplish this, a simple tool is provided for describing the Use Case together with a diagram showing the UC and the actors (Figure 4).

New UC:

UC no	UC name	Actor(s)
8	The name of the UC	Actor A
Description	Basic Event Flow	Alternative Event Flow
A short description	1. Actor initiates something 2. Something happens 3. And then the response is made	E1: Illegal access
Precondition	Post condition	Assumption

Add new UC Export

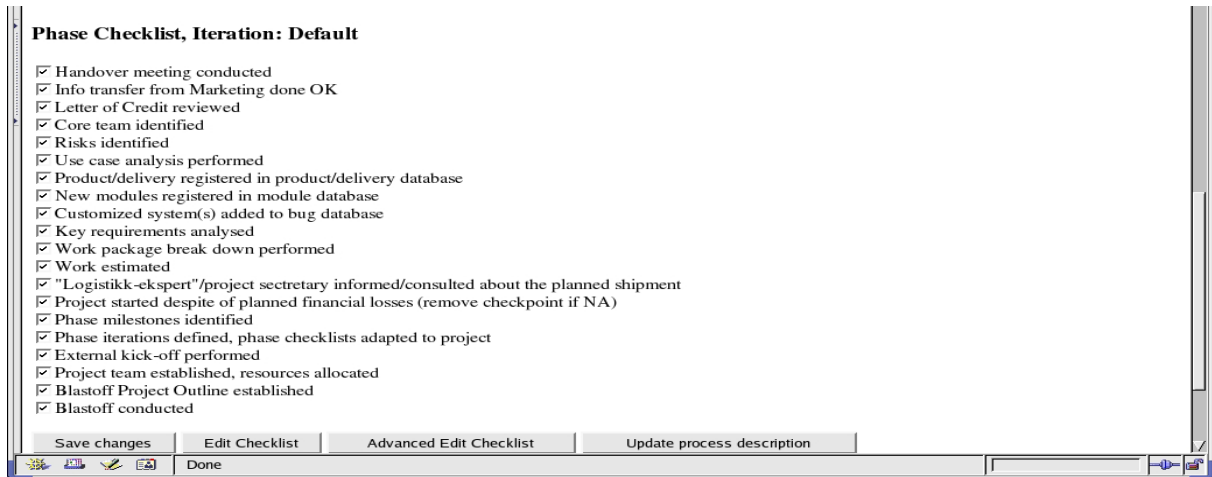
<Figure 4> Use Case Analysis support

3.1.3 Schedule and milestones

In connection with work planning, the schedule of the project containing the different phases with definition of milestones must be done. These milestone dates are later used by the Project Web to automatically signalise delay in project progress when over due and not reached.

3.1.4 Process checklists

Each process description is set up with default checklists for process, project manager and Technical QA. The project manager must edit the checklists to suit the project’s needs.



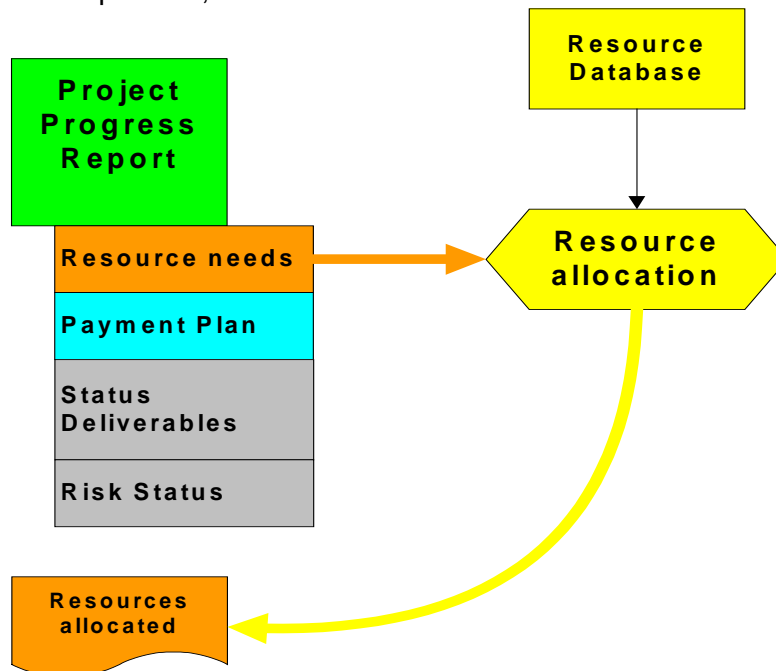
<Figure 5> Check list (BlastOff process)

3.1.5 Resource planning

In connection with the work packages, the project manager must decide what competence requirements each work package has, and make a resource plan for the whole project period for the needed manpower to do the work.

This resource plan is submitted via the BlastOff process to the resource allocation function (headed by the Director Projects) for decision on resource allocation

If later in the process lifetime changed, the resource needs for the remaining are received by the resource allocation process, which consults the Resource Database.



<Figure 6> Resource allocation process

A decision is taken whether to either hand over the requested manpower or to deny or delay such allocation.

3.1.6 Organisation of the project

The project manager must decide the organisation of the project and obtain an optimal composition of the team with roles and responsibilities necessary to reach the objectives.

3.2 Risk analysis

Risk analysis is supported and later maintained through status and progress reporting.

Risk Management may be divided into five main activities:

1. Identification of risk
2. Analysis of identifies risks (probability, seriousness, consequences)
3. Establish contingency plans
4. Preparations
5. Monitoring of identified risks

The tool will bring out a risk table:

Risk: 2 jhsre vw Consequence:	High	High	dshcew
-------------------------------------	------	------	--------

Last operation:

#	Risk Description	Prob.	Sev.	Measure
	New risk:			
3	Consequence:	High	High	

Add new risk Export

<Figure 7> Risk analysis tool

During project lifetime, the project manager may message changes of risks on the Project Web in a dedicated column for this, thereby alerting upper management if needed.

3.3 Phase orientation

The project is divided into phases that can be made iterative, and each phase has its process description, which is instantiated in the Project Web; e.g. phases for software

- specification
- design
- testing
- system realisation
- delivery / installation
- project termination with optional post mortem analysis at the end.

Project ADAS-DIFEP (8341) making "ADAS-DIFEP" for Alcatel Bell Space

Task description

Develop and deliver Data Ingest and Front End Processor to the ADAS

Processes and Project Status

Blastoff	Specification	Elaboration	Component construction	System integration	System delivery
Process (12/12 closed)	Process (3/5 closed)	Process (10/10 closed)	Process (34/47 closed)	Process (3/14 closed)	Process (0/4 closed)
	TQA (3/3 closed)	TQA (4/4 closed)	TQA (1/3 closed)	TQA (0/3 closed)	TQA (0/4 closed)
	PM (3/5 closed)	PM (1/1 closed)	PM (0/2 closed)	PM (0/3 closed)	PM (0/4 closed)
100% Completed	100% Completed	100% Completed	67% 17 of 52 remaining	25% 15 of 20 remaining	0% 12 of 12 remaining
Milestone: 20030519	Milestone: 20030820	Milestone: 20031010	Milestone: 20041020	Milestone: 20041120	Milestone: 20041220

Project end-date according to front-page: 20041001

<Figure 8> Phase process diagram

Email registry for the project is linked into the Project Web.

Checklists have to be checked out for each sub process divided into one for Project Manager, one for Project Team and one for Technical Quality Assurance – a role assigned to one individual partly independent of the project management.

The checklists may be edited in several ways. E.g. if a phase is run in several iterations, checklists may be duplicated so that each iteration has independent checklist.

4 Running the project

4.1 Phase Independent Processes

4.1.1 Progress Reporting

Of phase independent processes, progress reporting is streamlined as much as possible with all relevant information available and auto generated, like

- remaining work estimates
- resource allocation changes
- payment plan, cost budget
- deliverables status
- graphs for man hours budgeted and used
- prognosis for future work
- process progress *planned* and *actual*

Special Alert Status exists for technical problems. Progress may be reported at any time, not rigorously periodically, but may be each week, each third month or even every day if so needed.

The Project Web has full communication with the company time accounting system providing updated man-hours recorded at any time, and interacts with the purchasing database, invoicing records and contracts archive.

<Figure 9. > Project progress reporting interacting checklist

Project reports - Netscape

File Edit View Go Bookmarks Tools Window Help

http://www2.spacetec.no/process_guide/project_web/5912/prc... Search

SAS Braathens Fly Norwegian- Billige Flybilletter Fra Nor... Project reports

Date: 20040826

Checklist:

Yes No NA - [Remaining work estimated in WP list](#)

Yes No NA - [Resource allocation plan up to date](#)

Yes No NA - Change in resource needs

Yes No NA - Changes in resource needs communicated to Prosjektsjef

Yes No NA - Changes in resource needs approved by Prosjektsjef

Yes No NA - [Status payment plan reviewed](#)

Yes No NA - [Direct cost budget is correct](#)

Yes No NA - [Status on project orders reviewed](#)

Yes No NA - [Status on project accounting reviewed](#)

Yes No NA - [Status SW deliverables updated](#)

Yes No NA - [List of HW deliverables still correct](#)

Yes No NA - Phase checklists up-to-date

Yes No NA - Phase milestones up-to-date

Yes No NA - [Risk status up-to-date](#)

Yes No NA - [Alert status up-to-date \(project problems flagged!\)](#)

Project managers view of the project:

[Free text description of how the project manager consider the project status
[Optionally also additional explanation of items in this project report.]
Example: "Project has been running according to plan, however resources unavailable the last week due to problems in another project, if this status it might have schedule consequences ."

Workpackets that will be closed the next 30 days:

[* Item]

Time off identified by project manager:

Only when time off will have consequences to the project schedule or budget
[* Team member, <from-date> to <to-date>; Reason]

Remaining direct costs (Budget: 0,
Invoiced: 46.729): -46729.95

Start 10:39

4.1.2 Project meeting support

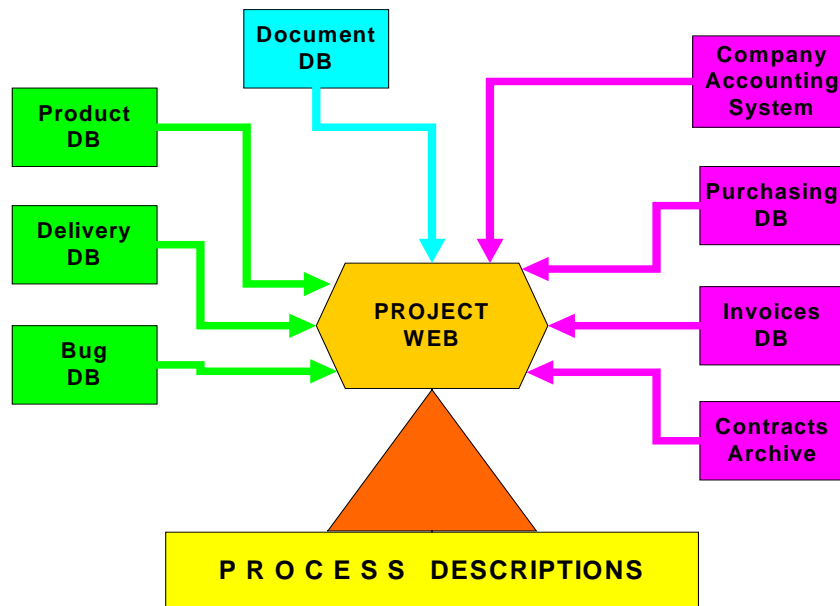
Project meetings are supported with Minutes of meetings template and actions lists connected, and requirements management and economy/budget insight are provided. Milestones planning and follow-up are included.

4.1.3 Supporting databases

Several attached databases service the project, like

- Product database
- Delivery database
- Requirements database for tracing requirements through project lifetime.

A special database for *bug reporting* is available for use in the acceptance testing and (later) warranty stages.



<Figure 10> Project Web access to relevant databases

4.2 Status visibility

The project web shows project status at any time concerning

- milestone and overall progress
- reporting status and economy figures
- risks
- actions statuses

Colours are used to flag deviations like red for alerts, delay and alarming conditions, yellow for concern and warning and green for status as planned.

Hyperlinks are defined to all relevant documents like

- Proposal
- Contract
- Architectural Design
- and test documents.

Templates to all necessary documents are provided for use.

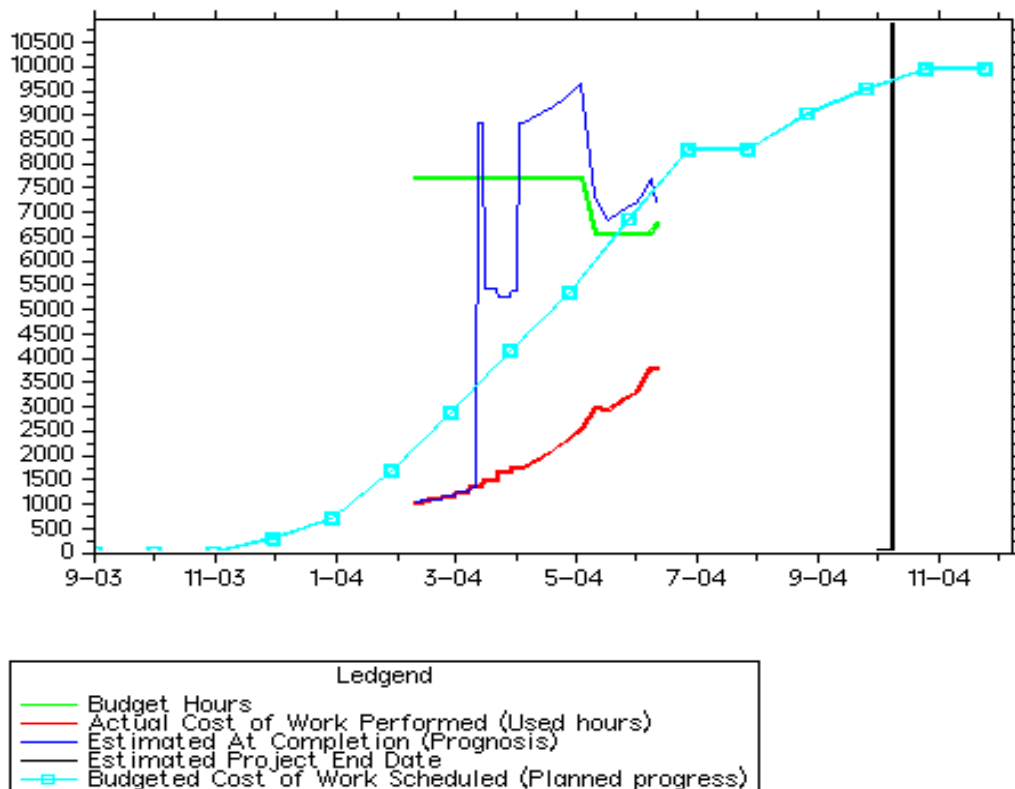
4.2.1 Management view

The Project web gives total visibility of the project including e.g. detailed economy figures to all project members, but also to middle and upper management and all interested employees. This is a deliberate strategy for total openness to provide full information to anyone concerned.

In addition to the numerical and textual information in the Project Web, also graphs being automatically updated in time exist for monitoring project progress and efficiency.

The following figure shows graphs for:

- Budgeted man hours
- Actual man hours
- Prognoses for completion figures
- Estimated end date
- Budgeted cost
- Process progress based on fulfilled check lists

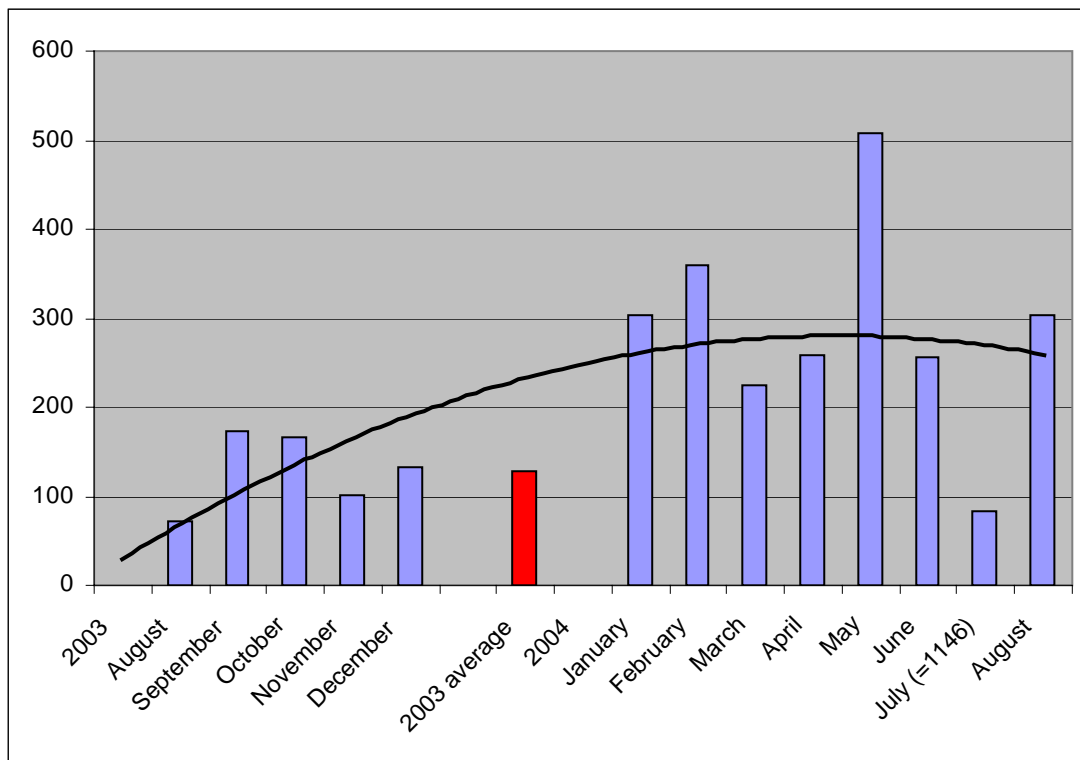


<Figure 11> Status on resource usage.

5 Statistics

5.1 Usage by project managers and developers

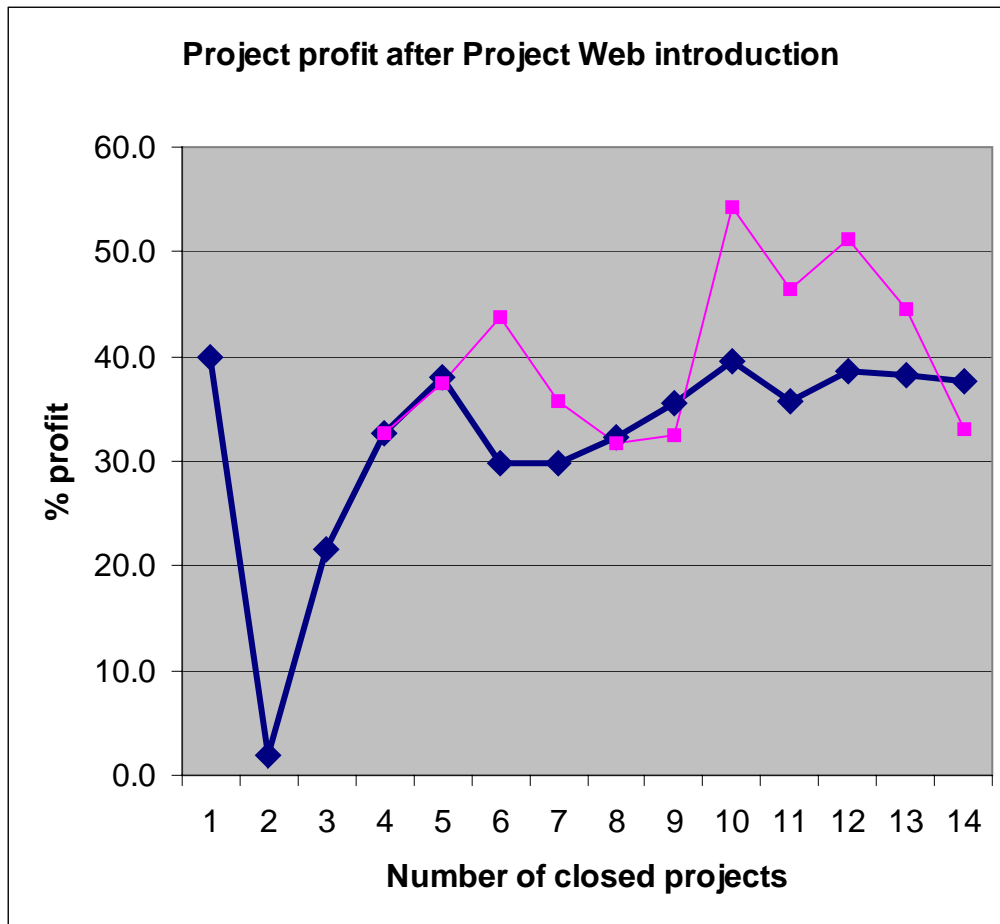
The diagram shows how much the Project Web has been used since introduction in August 2003 (one year) measured by number of web page hits per working day,



<Figure 12> P-web - # of web page hits per working day

5.2 Productivity

Productivity figures for company project efficiency is shown in next diagram, displaying project profit development after introducing P-web (counting only closed (terminated) projects).



<Figure 13> Project profit development after introducing P-web

Literature

Tore Dybå, Torgeir Dingsøy, and Nils Brede Moe, *Process Improvement in Practice - A Handbook for IT Companies*. Boston: Kluwer, 2004, ISBN 1-4020-7869-2.

Nils Brede Moe, Torgeir Dingsøy, Tore Dybå, and Trond Johansen, "Process guides as software process improvement in a small company," EuroSPI, Nürnberg, Germany, 2002.

Author CVs

Nils Jakob Villmones Kongsberg Spacetec, Quality Assurance Manager

Born: 22 May 1951
Nationality: Norwegian
Languages: Norwegian, English
Education: Master of Science (M.Sc.) (cand. real) University of Tromsø, 1982, Computer Science.
 Thesis on: "Open Systems Interconnection, Systems Network Architecture and DATEX: An analysis of models and adaption of an implementation".
 B. Sc. (høgskolekandidat) Møre og Romsdal Distriktshøgskole, EDP, 1974

Experience:

2002-	Participated in the SPIKE process improvement project
1999-2002	Participated in the PROFIT process improvement project
1995-1998	Participated in the SPIQ process improvement project Project Manager for EUCLID CEPA 9.8 Western Union research project (Italy/Germany/Holland/Norway)
1998	Responsible ISO 9001 and TickIT Certification for Kongsberg Spacetec
1992-	Quality Assurance Manager, Spacetec a.s Executing Quality Assurance on several Spacetec projects: XRAIS, POEM, AR5 SFL, AR5 ISF, IDUN BDDN, JERS, ENVISAT etc.
1984-1992	System Manager, Process Control systems, Elkem Aluminium, Mosjøen, Norway Experience with DEC VAX and PDP-systems for process control of aluminium production cells: system programming and management, system development FORTRAN
1982-1984	Senior System Programmer, Data Communication systems, Tromsø Datasentral
1974-1979	System Engineer, Data Communication systems, Tromsø Datasentral, Tromsø, Norway Experience with IBM mainframes and communication systems (system programming); programming: PL/1, assembler languages, system development

ManagEUr – Success Criteria for EU Project Management

Miklos BIRO¹, Richard MESSNARZ²

¹*Budapest University of Economic Sciences and Public Administration,
Veres Palne u. 36. H-1053 Budapest, Hungary*

Tel: +36-1-218-4665, Fax: +36-1-218-4665, Email: miklos.biro@informatika.bke.hu

²*ISCN LTD, Bray, Co. Wicklow, Ireland*

Tel: +43 316 811198, Fax: + 43 316 811312, Email: rmess@iscn.com

Abstract

This paper discusses success criteria of distributed multinational EU projects based on trials performed at 59 organisations in 13 countries in the EU IST-2000-28162 project TEAMWORK. In this project the typical working processes, the teamwork structures in multinational distributed EU projects have been analysed and configured into e-working spaces. These e-working spaces were then used in the trials.

The experiences (extracted success criteria) from these trials are leveraged in the design of an EU project manager skill card being developed in the framework of the EU Leonardo da Vinci project ManagEUr (2003 – 2006). This skill card is currently being configured into a skills portal and allows managers to browse the required skills, to do self assessment, and to receive guidance. It is a distinctive feature of ManagEUr that it puts equal emphasis on technology, content and human factors necessary for successful application.

The paper promotes to use these success criteria in largely distributed projects to improve the project management, performance, and team motivation .

Keywords

Project Management, Success Criteria, Human Factors

1 Introduction and Needs

ManagEUr (European Project Manager) is a pilot project which is funded by the EU Leonardo da Vinci Programme under contract number HU/B/03/F/PP-170028.

ManagEUr addresses needs originating in two current developments.

One is the accession of new countries to the European Union. These countries have already been partners in many EU programmes for a couple of years. According to the experiences however, their participation was limited. When organisations from these countries did participate, while their role was often significant from the content point of view, it was marginal as far as the management of the projects was concerned.

The second one is the use of outsourcing partnerships and supply chains in the industry which often lead to multinational distributed development teams. Exactly the same situation happens in distributed EU funded projects.

The major question is therefore in which areas we need to improve traditional project management (as additional success criteria added to the known ones from e.g. the PMBOK – Project Management Body of Knowledge) to be successful in such a multinational, distributed management situation.

2 Leveraging Results of Earlier Projects

2.1 Focussing on networking and team-working processes

A study in 1998 (EU Leonardo da Vinci Project BESTREGIT – Best Regional Innovation Transfer, 1996 – 1999) analysed how successful innovative organisations involved in EU projects operate and compared 200 organisations in Europe ([1], [6],[7]). The study outlined that innovative organisations

- invest time and money into the understanding of the fundamentals of the forces of change,
- understand the different cultures through personal contacts using networking at a personal level,
- study trends and they are always up-to-date,
- concentrate their energy in areas where they excel or where no one else can operate,
- outsource all other non-core activities,
- are practical users of information technologies and have information technology strategies in place.

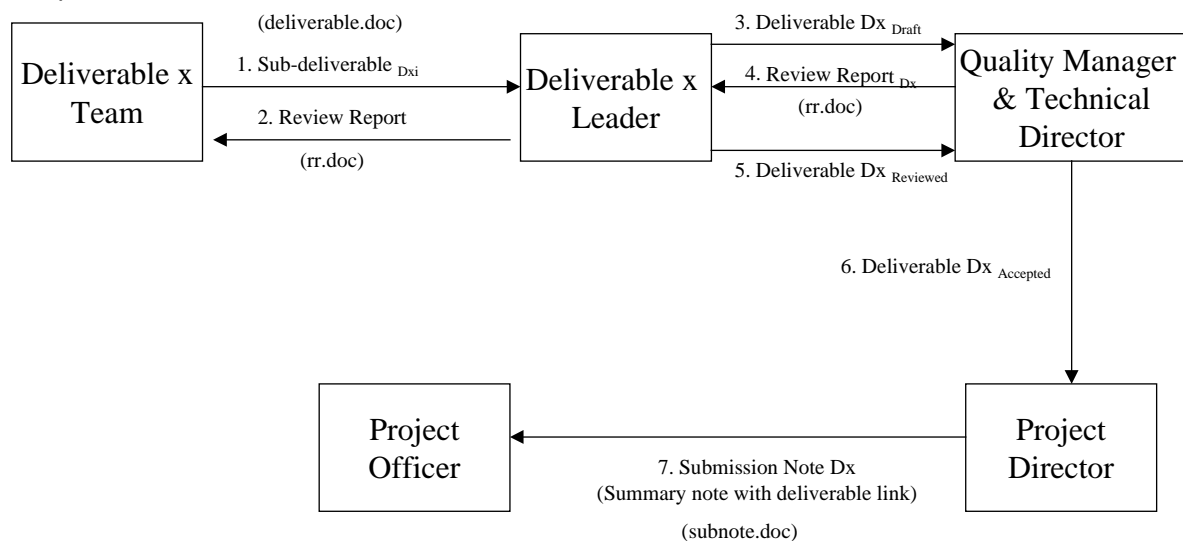


Figure 1: Example BESTREGIT Result – Deliverable Production and Review Team Scenario

Therefore the BESTREGIT project developed a learning organisation model in which a traditional organisation step by step transforms into a dynamic , goal oriented, and interdisciplinary teamwork (human skill based) smart organisation.

2.2 Focussing on e-working and smart organisations

Another result from the 1998 study was that innovative organisations understand the different cultures through personal contacts using networking at a personal level, and are practical users of information technologies and have information technology strategies in place .

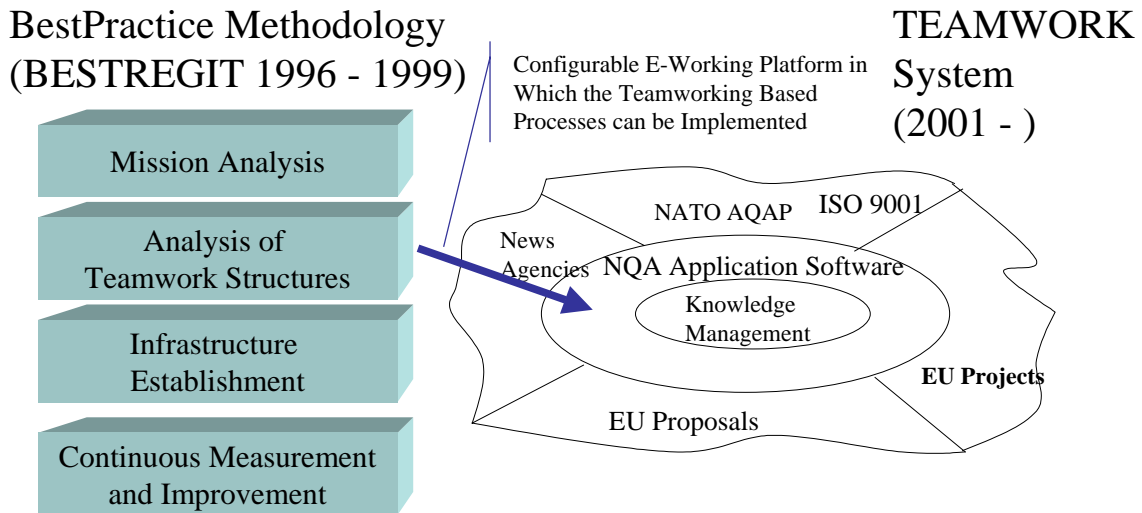


Figure 2: TAMWORK Concept – Generating E-Working Environments

:: Submission Dialog:

Current Role(s) Assigned: QM - Quality Manager, WGM - Working Group Manager, WPL - Workpackage Leader

Version Control: *checked in* ::Check Out to Edit ::Version List

Document: ::Link Documents/Objects

Document Status: *Draft*

:: Home :: Deliverable Design

ManagEUR Skill Card leo_pilot_del(2003/12/15 17:50:54)

ManagEUR Skill Card leo_pilot_del(2003/12/15 17:50:54)

Deliverable Review: Meeting Valladolid Del Review [leo_pilot_delrev Nr.1 (2004/4/27 11:51:27)]
Input form Joseph Bremer [leo_pilot_delrev Nr.3 (2004/5/4 16:32:27)]

NQA Link Destinations

- Skill Card Design with Visio [leo_pilot_subdel Nr.1 (2004/4/24 21:24:40)]
- PMBOK [leo_pilot_subdel Nr.2 (2004/4/24 21:28:37)]
- PRINCE2 [leo_pilot_subdel Nr.3 (2004/4/24 21:32:20)]
- TEAMWORK [leo_pilot_subdel Nr.4 (2004/4/24 21:34:52)]

Figure 3: Example TEAMWORK Result – Deliverable Linked With Sub-Deliverables

Therefore from 2001 onwards a partnership from 7 EU countries received funding to establish, test, and refine a generic and configurable working platform for implementing BESTREGIT scenarios . It was a distinctive feature of that project that social experts were involved to analyse social teamworking behaviour and success criteria.

Applying these best practices ManagEUR supports the following concepts:

Scenario Thinking. Problems are solved in a team. Team members play defined roles in which they produce and submit materials to other roles. People are assigned to roles. A set of roles and results focused towards a specific objective is grouped into a team-working scenario.

Role Thinking. One person can play many roles just as many persons might be assigned to the same role. Roles determine ownership of materials and people identify with the processes underlying

the team-working scenario through roles.

Project Types / Work Spaces. A demo system which allows for the use of a wizard to configure the structure of workspaces, which we also call project types. Each project type involves a set of team-working scenarios with predefined documentation structures, roles, and flows.

Wizard. The configuration wizard allows for the creation of new project types, structures and minutes and for the restructuring of operational e-working services according to needs.

Version Management. All materials published and submitted by the roles fall under version control with defined check in and check out processes, and a version history.

Group Management. Project workspaces based on project type structures are created using an administrator menu. This includes the establishment of a group with user accounts, which are then assigned to roles within the defined project type environment.

Privacy. Created project work spaces are only accessible by the team members who belong to the groups assigned to the projects.

Knowledge Retrieval. The server includes a search and query service, which is also able to search in binary files. Results are presented in a search engine like style with specific content recommendations.

Forward Trace-ability. The configuration wizard allows for definition of relationships between different types of documents. With the creation of a document, the documents and reports can automatically be linked forward and backward. The need for this feature came from engineering disciplines.

Discussion Forum. Each created project workspace automatically contains a discussion forum. In this forum different topics can be created, discussion topics linked, and files exchanged. It is intended for more informal communication.

Submission / Notification Management. Submission lists are configured per type of document and in the form of roles. Once the submission takes place, the system extracts the users who match the roles and submits a standard notification to the recipients.

Re-Use Pool. The system has already been configured with ISO 9001:2000, ESA ECSS and ESA PS 005 standards, EU project management, EU proposal writing, Software Development, etc. Once a project type has been configured, projects of that type can be created by just a click.

2.3 Focussing on social criteria

The TEAMWORK platform was tested with teams from 59 organisations in 13 countries of Europe. The working behaviour of the users (team-working and team-learning members of the networked platform) has been analysed and a study with key success factors for social team-learning and team-working has been produced as a project deliverable. There were 42 different projects running through the system using the defined environment and being managed by a virtual team leader. The team size of the projects varied from 13 down to 2 different organisations.

The trials (2002 – 2003) included the configuration of the TEAMWORK platform for

- EU proposal writing and EU project management, and involved IST NCPs (National IST Contact Points) from Hungary, Austria, and Ireland.
- Contact Points and consulting units for the EU Leonardo da Vinci Programme from Spain, Austria, Hungary, Ireland, Slovenia,
- as well as national innovation funding programme units from Ireland and Austria.

Questionnaires and interviews have been used by a social team to receive feedback about the system usage and user opinions. This led to a social guidebook with typical (using statistical significance analysis) situations, findings and recommendations. Two types of findings could be differentiated:

- Social team factors influenced by the organisational style patterns
- Social team factors influenced by the management style patterns
- Social team factors influenced by the cultural patterns

The analysis was based on the PATTERNS approach. "Patterns and Pattern Languages are ways to describe best practices, good designs, and capture experience in a way that it is possible for others to reuse this experience." ([5], Patterns Homepage, <http://hillside.net/patterns/>).

A list of (social) success criteria and guidance about how to manage this has been elaborated (and will be part of the course programme).

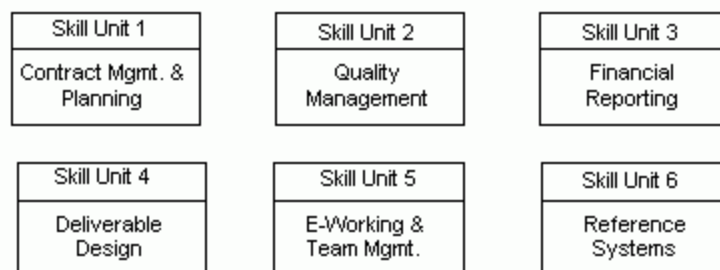
3 Content Provided by ManagEUr

In the project ManagEUr we use all this previously gathered know how and

- Defined a skills card for EU project managers,
- Offers online browsing and self assessment of skills (based on results from CREDIT MM 1032 project , 1998 – 2001).
- Established a demo best practice e-working system to manage distributed multinational EU projects (based on TEAMWORK)
- Developed a course programm for the required skills of EU project managers.

The job role of an EU Project Manager (see Figure 5) therefore is able to apply team scenario thinking, to use e-working systems based control of team scenarios, to enforce version and submission control in the team, manage conflict situations with extensive knowledge about human factors and social team criteria, etc. For structuring a skills set the EU leonardo da Vinci project ManagEUr followed the EU standards for skills cards [3].

The CREDIT project (MM 1032 multimedia project, 1998 – 2001) and its exploitation in a Capability Adviser (http://www.iscn.com/projects/piconew_skill_cards/index.html) platform system offers (see Figure 6) online access for browsing skill cards, performing self assessment online, collecting evidences, and receiving a formal assessment of the evidences online, as well as accessing reference materials and generating learning portfolios/plans. (see Figure 6)



Select Learning Elements from Unit 1 Planning Measurement of Objectives Exploitation & Integration Risk Management Contract Management	Select Learning Elements from Unit 2 Quality Planning Realization Monitoring Documentation Internal Review External Review
Select Learning Elements from Unit 3 Effort Reporting and Control Deviation Analysis Trend Analysis Corrective Management Estimation Models Financial Audit	Select Learning Elements from Unit 4 Deliverable Team Management Deliverable Design & Integration Deliverable Review Deliverable Definition
Select Learning Elements from Unit 5 Team Building Teamworking Processes & Tools Social Team Factors Cultural Aspects	Select Learning Elements from Unit 6 Introduction to EU Projects PMBOK PRINCE2 TEAMWORK

Figure 5: Skills Card of an EU Project Manager Job Role

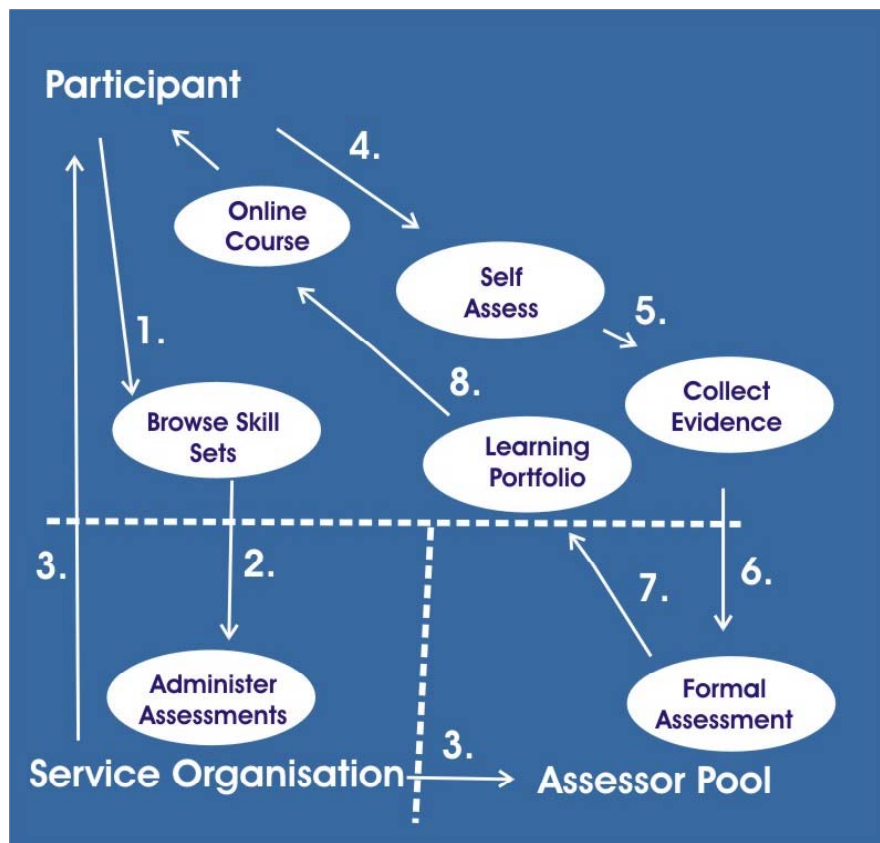


Figure 6 : Skills assessment and learning guidance process model

4 Literature

- [1] Biró,M; Messnarz,R; Davison,A.G. The Impact of National Cultural Factors on the Effectiveness of Process Improvement Methods: The Third Dimension. *Software Quality Professional (ASQ~American Society for Quality) Vol.4, Issue 4 (September 2002) pp.34-41.* (http://www.asq.org/pub/sqp/past/vol4_issue4/hiro.html)
- [2] Biró,M.; Messnarz,R. Key Success Factors for Business Based Improvement. *Software Quality Professional (ASQ~American Society for Quality) Vol.2, Issue 2 (March 2000) pp.20-31.* (http://www.asq.org/pub/sqp/past/vol2_issue2/hiro.html)
- [3] Feuer E., Messnarz R., *Best Practices in E-Commerce: Strategies, Skills, and Processes*, in: *Proceedings of the E2002 Conference, E-Business and E-Work, Novel solutions for a global networked economy*, eds. Brian Stanford Smith, Enrica Chiozza, IOS Press, Amsterdam, Berlin, Oxford, Tokyo, Washington, 2002
- [4] Feuer E., Messnarz R., Wittenbrink H., *Experiences With Managing Social Patterns in Defined Distributed Working Processes*, in: *Proceedings of the EuroSPI 2003 Conference, 10-12 December 2003*, FTI Verlag, ISBN 3-901351-84-1
- [5] Project EASYCOMP (IST Project 1999-14191, homepage: <http://www.easycomp.org/>)
- [6] Messnarz R., Nadasi G., O'Leary E., Foley B., *Experience with Teamwork in Distributed Work Environments*, in: *Proceedings of the E2001 Conference, E-Work and E-commerce, Novel solutions for a global networked economy*, eds. Brian Stanford Smith, Enrica Chiozza, IOS Press, Amsterdam, Berlin, Oxford, Tokyo, Washington, 2001
- [7] A Learning Organisation Approach for Process Improvement in the Service Sector , R. Messnarz. C. Stöckler, G. Velasco, G. O'Suilleabhain, *A Learning Organisation Approach for Process Improvement in the Service Sector*, in: *Proceedings of the EuroSPI 1999 Conference, 25-27 October 1999, Pori, Finland*
- [8] Hofstede, G. (1994). *Cultures and Organizations, Software of the Mind: Intercultural Cooperation and its Importance for Survival*, McGraw-Hill, London, 1994.

5 Author CVs

Dr Miklos Biro

Dr. Miklós BIRÓ (miklos.biro@bkae.hu) is a professor at the Department of Information Systems of the Budapest University of Economic Sciences and Public Administration with 27 years of software engineering and university teaching (including professorship in the USA), and 17 years of management experience. He has a Ph.D. in mathematics (operations research) from the Loránd Eötvös University in Budapest, an Executive MBA (Master of Business Administration) degree from ESC Rouen, France, and a Master of Science in Management degree from Purdue University, USA. He is fluent in Hungarian, English, and French.

He is a SPICE (Software Process Improvement and Capability dEtermination - ISO/IEC 15504) assessor. He gives Ph.D. courses and company training courses on software quality management, and on the Capability Maturity Model - Integrated (CMMI - service mark of Carnegie Mellon University, USA). He designs and delivers software technology courses tailored to the requirements of the program in business informatics of the Budapest University of Economic Sciences and Public Administration.

He initiated and managed the Hungarian participation in numerous European multinational projects and organisations committed to software process improvement (European Software Institute, Bootstrap Institute). He was the initiator and head of the Information Society Technologies Liaison Office in Hungary for the European Union's 5th Framework Programme. He is invited as expert consultant by Hungarian and international organizations (European Commission; Irish National Policy and Advisory Board for Enterprise, Trade, Science, Technology & Innovation~Forfás; Communications Authority of Hungary; Hungarian Committee for Technological Development; Investment and Trade Development Agency of Hungary; Hungarian Air-

lines; United Nations Industrial Development Organization-UNIDO; International Software Consulting Network;...).

He has numerous publications in international scientific and professional journals ([Software Process Improvement and Practice](#), [Software Quality Professional](#) (1, 2), [Software Process Newsletter](#), [European Journal of Operational Research](#), [Zeitschrift für Angewandte Mathematik und Mechanik](#), [Optimization](#), [Information Processing Letters](#), [Discrete Mathematics](#), [Journal of Advanced Transportation](#), [Acta Cybernetica](#)) and conference proceedings. He is the co-author of Hungarian and English language books on operations research models, software engineering, software process improvement and business motivations.

He is member of the [Editorial board of the journal Software Process Improvement and Practice](#) published by John Wiley & Sons, and founding president of the professional division for Software Quality Management of the John von Neumann Computer Society. He is the Hungarian member of [Technical Committee 2 \(TC-2\) Software: Theory and practice of the International Federation for Information Processing \(IFIP\)](#) . He is member of several other professional bodies and societies.

Dr Richard Messnarz

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

- PICO - Process Improvement Combined Approach 1995 - 1998,
- Bestregit - Best Regional Technology Transfer, 1996 - 1999,
- TEAMWORK - Strategic Eworking Platform Development and Trial, 2001-2002,
- MediaSF - Eworking of media organisation for strategic collaboration on EU integration, 2001-2002

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

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

He is a lead ISO 15504 assessor. He has worked as a consultant for many automotive firms, such as BOSCH, ZF TE, ZF N, Continental TEMIC, Audi/VW, etc. He is a member of the ITACS accreditation board, he is an initiator of the German MDISQ (www.mdismq.de) initiative, and he is the technical moderator of the SOQRATES initiative (www.socrates.de).

Acknowledgements

We acknowledge the support and contributions from A. Fenz, APS (A), J. Ivanyos, Memolux (HU), A. Tiron Tudor and M. Lutas, Babes-Bolyai (RO), J. Bremer, iCUBE (LU) , G. Velasco, FGUVA (E),

Experiences in the Assessment of the Requirement Management Process

Jose A. Calvo-Manzano¹, Gonzalo Cuevas¹, Tomas San Feliu¹, Ariel Serrano¹, Magdalena Arcilla²

¹ Faculty of Computer Science, Polytechnic University of Madrid (UPM), Madrid, Spain

² Department of Computer Science, Spanish Open University (UNED), Madrid, Spain

Abstract

The focus of this paper is to outline the practical experiences and the lessons learned derived from the assessment of the requirements management process in two industrial case studies. Furthermore this paper explains the main structure of an alternative assessment approach that has been used in the appraisal of the two case studies. The assessment approach helped us to know the current state of the organizational requirement management process. We have to point out that these practical experiences and the lessons learned can be helpful to reduce risks and costs of the on-site assessment process.

Keywords

Capability maturity model, CMMI, requirement management, SCAMPI, software process assessment, software process improvement

1 Introduction

The main objective of this paper is to present the practical experiences and the lessons learned derived from the assessment of the requirements management process in two industrial case studies using the Capability Maturity Model Integration (CMMI) [1, 2] as a reference model. However, in order to understand the problem, the objectives, and the alternative assessment approach structure, this paper is divided into five main sections.

Section 1 presents a quick view of ISPI's software process improvement model [3], and a quick view of the requirements management process. Section 2 presents the problematic of in the use of SCAMPI [4] and describes the structure of an alternative assessment approach. Section 3 present the data analysis of the two industrial case studies. Section 4 present the practical experiences and the lessons learned derived from the assessment of the requirement management process. Section 5 presents the paper conclusions.

1.1 Software Process Improvement Models

During the last few years, several Software Process Improvement (SPI) models have been developed to increase the quality and productivity of software. Models like IDEAL [5] or ISO/IEC TR 15504-7 [6] have been useful to initiate a software process improvement effort in many organizations. However, such models are expensive and time consuming, and hence small organizations find some difficulties to perform them.

To solve the inconveniencies associated with the high cost and time consuming the Institute for Software Process Improvement (ISPI) developed a new simplified SPI model. It takes the advantages of IDEAL and ISO but reduces the number of stages at only four [3].

Stage 1: *Commitment to improvement*, its objective is to obtain senior management commitment to carry out the improvement project. Stage 2: *Software process assessment*, its objective is to obtain strengths and weaknesses of the process with respect to a software process model. Stage 3: *Infrastructure and action a plan*, it has two objectives, the first is to provide the necessary infrastructure for the improvement project, and the second is to design the improvement action plans. Stage 4: *SPI Implementation*, its objective is to implement and institutionalize the software process improvements.

A group of professors from three Spanish universities: Polytechnic of Madrid (UPM), Carlos III (UC3M) and Open University (UNED) has been using ISPI's model since 1994 with good results [7]. The main objective of the group is promoting SPI initiatives among Spanish organizations. Consequently, the professor's group is still using the ISPI's model and has enhanced it with their projects experiences.

1.2 Requirements Management Process

The requirements are the foundation, upon which the software process is built, and the Requirements Management (RM) process emerges as a systematic approach to find, document, organize, and track all system's requirements during the life cycle. But unclear requirements and the inability to manage requirements change are the cause of the most part of delays on the software development process [8].

According to a SEI's Study [9], the top two out of ten factors that contribute to the failure of system development projects are requirements problems. These problems are mainly associated with an inadequate requirements specification and/or an insufficient requirements change management. Also the Standish Group's CHAOS report [10] found that the major factors that cause software projects to fail are: lack of user input, incomplete requirements specifications, and changing requirement specifications.

Therefore, RM process is considered the cornerstone of the software lifecycle and CMMI identifies the enormous importance of the RM, granting the category of “Process Area” and placing it in the CMMI, staged representation, maturity level two [2]. According to the CMMI, the Requirements Management major aim is establishing an agreement between the customer and the software team on the meaning of the requirements [1, 2].

2 Assessment Method

2.1 SCAMPI

The Software Engineering Institute (SEI) establishes that one of the first steps of a SPI project is to understand the current state of the process [5]. Therefore the SEI developed SCAMPI [4] as the assessment method to know the current state of the organizations software process using the CMMI as reference model.

SCAMPI describes the activities to assess the software process, however, it is an expensive method because consumes a lot of resources like: team size, training and cost (time and money). In some organizations the SCAMPI assessment cost could be too high from 40,000 to 100,000 USD [11]. The cost and effort of a SCAMPI assessment might be too big particularly in small organizations.

2.2 Alternative Assessment Approach

To determine the perform level of the requirements management process, we use the ISPI’s model and develop an alternative assessment approach. This approach is based on the use of a questionnaire for data collection and the subsequent data analysis. The questionnaire is based on the two types of practices of the requirements management process area (REQM) described in the CMMI [1, 2] and it is divided into two sections: The first one is related to the specific practices and describes the series of steps that have to be followed to perform the process. The second one is related to the generic practices and describes the series of steps that have to be followed to institutionalize the process.

The questionnaire contains closed questions with five possible answers to know the extent to which practice is performed. The answer options and the numeric value to calculate the arithmetic mean are: almost always (1), more often (0.75), sometimes (0.5), rarely if ever (0.25), and never (0) (Figure 1).

between: and:	(PI <= 100) (PI > 75)	(PI <= 75) (PI > 50)	(PI <= 50) (PI > 25)	(PI <= 25) (PI > 0)	PI = 0
Documented	Almost always	More often	Sometimes	Rarely if ever	Never
Not documented	More often	Sometimes	Sometimes	Rarely if ever	Never

Figure 1. Answer option perform level (PI)

For realizing the data analysis, the arithmetic mean (Σ) is calculated with all valid answers for each question (practice or subpractice). Subsequently, those practices which arithmetic mean is smaller than 75% are considered as candidates for improving (weakness), but only, if the standard deviation is smaller or equal to one. On the other hand, those practices which arithmetic mean is greater or equal to 75% could be consider process strengths, but only, if the standard deviation is smaller or equal to one. So, if the standard deviation is greater than one, there are strong discrepancies between the answers. Therefore, it is necessary to analyze the answer-question in more detail with the use of interview techniques. (Figure 2).

	standard deviation ≤ 1	standard deviation > 1
$\Sigma \geq 75\%$	STRENGTH	INTERVIEWS
$\Sigma < 75\%$	WEAKNESS	INTERVIEWS

Figure 2. Data analysis table to establish practice perform level

This assessment approach may be helpful to obtain information that reduce risks and costs of the on-site assessment process and to classify practices in those that are performed but not documented, those that require prioritizing and those that are not implemented.

3 Case Studies

The assessment approach proposed in this paper was used in the assessment of the requirements management process of two industrial case studies with the purpose of validating it.

We have to point out that firstly is very important to obtain senior management commitment with regard to people, time and other resources needed for the software process improvement project. Senior management commitment must be embedded in all decisions to be made throughout the improvement project, and these decisions must be communicated to all involved people.

3.1 Data Analysis

3.1.1 Specific Practices

The first section of the assessment analysis focuses on determining the current state of the activities of the RM process using as reference the five specific practices of REQM. The analysis of the values obtained from the answers of two case studies found that none of the five Specific Practices (SP) achieves the minimal performance level (75%) to be considered as process strengths. However, there are three practices in the case study 1 and two in the case study 2 between 50% and 75% of performance level. This suggests that the improvement effort must be focus on documenting the process. On the other hand, the values obtained for the remaining practices were under 50%. This suggests that for these practices it is necessary a greater improvement effort (Figure 3).

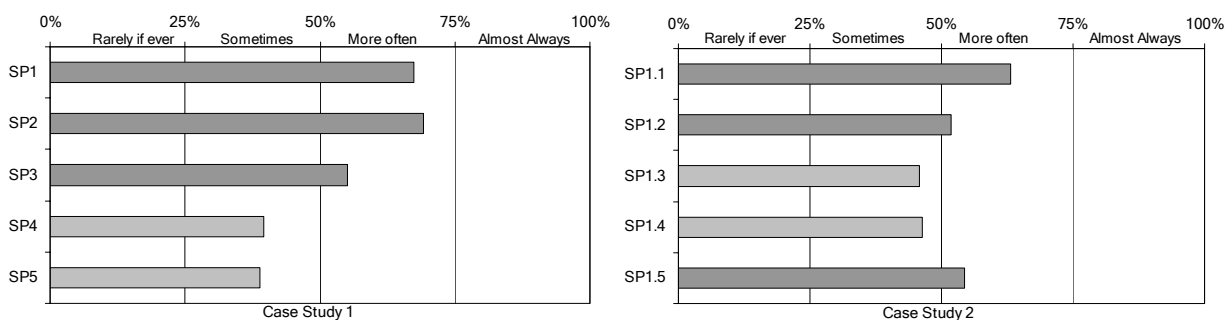


Figure 3. Requirements Management specific practices perform level

3.1.2 Generic Practices

The second section of the assessment analysis focuses on determining those practices related to the process institutionalization. These are labeled Generic Practices (GP) by the CMMI because are used in all process areas. The analysis of the obtained values from the answers of two case studies found that none of the ten GP achieves the minimal performance level (75%) to be considered as an institutionalized

process. This observation was expected since none SP was rated equal to 75% of perform level. Nonetheless, there is one practice in case study 1 between 50% and 75% performance level. This suggests that the improvement effort for this practice must be focus on documenting the process. On the other hand, three practices in the case study1 and two in the case study 2 are fewer than 25% of performance. This suggests that these practices are poorly performed and require a great effort to implement them (Figure 4).

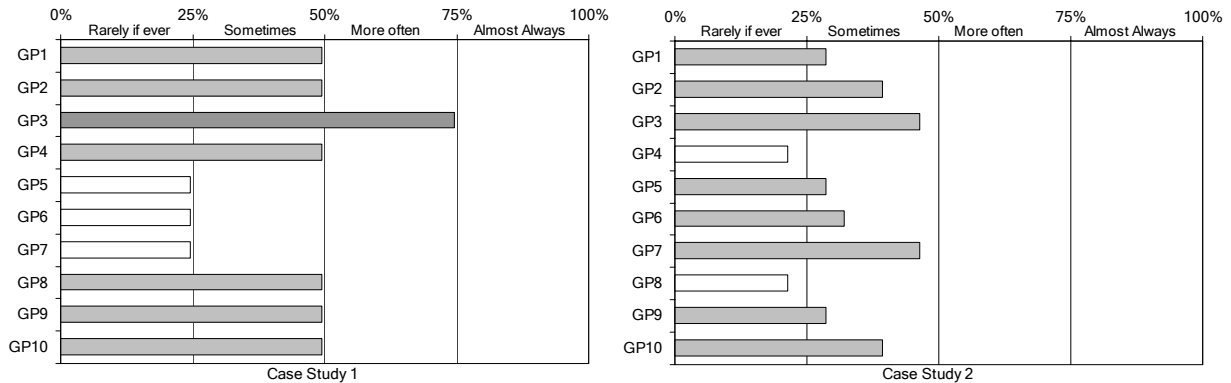


Figure 4. Requirements Management generic practices perform level

4 Practical experiences and lessons learned

In this section, the lessons learned from the assessment of the requirements management process in two industrial case studies are presented.

- In the two case studies we discover an initial resistance to change from almost all the affected people. It is important to select people which are interested in the process improvement to decrease the change resistance.
- It is essential to designate a project manager who will be in charge of the improvement project.
- Senior management must give explicitly samples of commitment like attending the most important meetings, inform to all people involved in the project about the results of the assessment, inform to all organization's personal about the improvement policy, etc.
- The assessment has to be collaborative and it is necessary training the members of the organization that are going to participate in the assessment.
- People who answer the questionnaire must have sufficient organizational and process knowledge.
- Each question must be answered after each specific or generic practice is explained.
- Answers to questionnaire will be focus on the way organization works and not on specific projects.
- The application of the questionnaire to all the people involved in the process allows reducing the number of interviews. However, all the ambiguous issues from the questionnaire answers must be prepared in detail for the interview stage.
- As a result of the assessment, a draft of the action plan is included, defining short-term priorities and activities to continue with the process improvement effort.

5 Conclusions

This work of investigation proposes an alternative approach to help in the assessment of the requirements management process, using a questionnaire and a simple method that allows obtaining results in a fast form. The results derived from their application of two case studies not only demonstrate the effec-

tiveness of the approach, but also takes the assessment experiences to be considered in other similar projects.

We have to point out that assessment does not in itself provide any improvement but it provides valuable information of the current state of the process and it lays the foundation for making the right choices about the changes you should make.

6 Literature

- [1] Product Development Team, "Capability Maturity Model Integration (CMMI), Version 1.1, Continuous Representation", CMU/SEI-2002-TR-011, Software Engineering Institute, Carnegie Mellon University, 2002.
- [2] Product Development Team, "Capability Maturity Model Integration (CMMI), Version 1.1, Staged Representation", CMU/SEI-2002-TR-012, Software Engineering Institute, Carnegie Mellon University, 2002.
- [3] Calvo-Manzano, J.A., Cervera, J., and San Feliu, T., "Software process improvement: MESOPYME model", *Journal of Computing and Information Technology*, vol. 43, 1997, pp. 159 - 165.
- [4] Members of the Assessment Method Integrated Team, "Standard CMMI Appraisal Method for Process Improvement (SCAMPI), Version 1.1", CMU/SEI-2001-HB-001, Software Engineering Institute, Carnegie Mellon University, 2001.
- [5] McFeeley, R., "IDEAL: A User's Guide for Software Process Improvement", CMU/SEI-96-HB-001, Software Engineering Institute, Carnegie Mellon University, 1996.
- [6] ISO/IEC TR 15504-7, "Information technology -- Software process assessment -- Part 7: Guide for use in process improvement", International Organization for Standardization, 1998.
- [7] Calvo-Manzano, J.A., Cuevas, G., San Feliu, T., Amescua, A.d., García, L., and Pérez, M., "Experiences in the Application of Software Process Improvement in SMES", *Software Quality Journal*, vol. 10, 2002, pp. 261-273.
- [8] Sommerville, I. and Sawyer, P., *Requirements Engineering: A good practice guide*, 1st. ed. England: John Wiley & Sons, 1997.
- [9] Siegel, J., Stewman, S., Kondra, S., Larkey, P., and Wagner, W., "National Software Capacity: Near-Term Study", CMU/SEI-90-TR-012, Software Engineering Institute, Carnegie Mellon University, 1990.
- [10] The Standish Group International, "The CHAOS report" http://standishgroup.com/sample_research/, 1994.
- [11] Walden, D., "A Business Case for CMMI-Based Process Improvement", presented at Sixth Annual PSM Users' Group Conference, Keystone, Colorado, USA, 2002.

7 Author CVs

José Antonio Calvo-Manzano Villalón

He has a Ph. D. in Computer Science and is assistant professor in the Faculty of Computer Science of the Polytechnic University of Madrid, Spain. He is teaching in the area of Software Engineering. He is author of several international papers related to the software engineering area. He is author of books in the software process improvement and software engineering topics. He has collaborated in several European projects oriented to software risk management, software process improvement in requirements management. <jacalvo@fi.upm.es>

Gonzalo Cuevas Agustín

He has a Ph. D. in Telecommunications and a MS in Computer Science. He is actually full professor in the Faculty of Computer Science of the Polytechnic University of Madrid, Spain. He has occupied senior management positions during more than 30 years in the university and industry always in the environment of information and communication technologies. Founder and Director

of Master in Software Engineering in association with Carnegie Mellon University in the Computer Science of the Polytechnic University of Madrid. He is reviewer of the European Commission and consultant in software process improvement. Current research interest: models and methods for software process improvement. Senior member of IEEE and member of ACM, author of numerous publications and books on software engineering and speaker at numerous conferences, seminars and symposium. <gcuevas@fi.upm.es>

Tomás San Feliu Gilabert

He has a Ph. D. in Computer Science. He is assistant professor in the Faculty of Computer Science of the Polytechnic University of Madrid, Spain. He teaches Software Engineering in this Faculty. He is author of several international papers related to the software engineering area. He is author of books in the software process improvement and software engineering topics. He has collaborated in several European projects oriented to software risk management, software process improvement in requirements management. <tsanfe@fi.upm.es>

Ariel Edgar Serrano Rico

He has a BSc. in Computer Science, Universidad Iberoamericana, México. He was assistant professor in the same university. He has occupied many management positions during more than 10 years in the industry. Nowadays, he is a PhD student in the Faculty of Computer Science of the Polytechnic University of Madrid, Spain. <aserrano@zipi.fi.upm.es>

Magdalena Arcilla Cobián

She has a BSc. in Computer Science in the Polytechnic University of Madrid, Spain. She got her Advanced Studies Diploma in the Department of Languages & Computer Science of the UNED (Spanish Open University). She is assistant professor in the Department of Languages & Computer Science Department of the UNED. She has also worked in the Computer Science Department of the Universidad Carlos III de Madrid, Spain, and she has 8 years of experience as software developer for CustomWare Company. <marcilla@lsi.uned.es>

SPI Managers' Experiences and Opinion On SPI In Practice

Nathan Baddoo and Tracy Hall

School of Computer Science, University of Hertfordshire

Abstract

We present results from a study of SPI managers' perception of the factors that motivate practitioners to support SPI. In this study we present SPI managers as change agents whose experiences and perceptions are critical to the success of SPI. We show that a majority of SPI managers generally have positive experiences of SPI. The most experienced managers tend to be most positive about SPI. We also show that a high number of SPI managers report optimistic expectations of SPI success. SPI managers are of the opinion that SPI does improve software quality. They are also of the opinion that SPI is cost beneficial. However, we report that SPI manager have the perception that senior managers are not supportive of SPI in their companies. Also, some SPI managers have the perception that developers are not enthusiastic about SPI. Overall, we report that SPI managers indicate that there is insufficient support for SPI in companies. We suggest that these findings make it important to establish, from software practitioners, what the factors are that can improve their support for SPI.

1 Introduction

SPI implementation requires competence in change management. Paulk et al [1994a] and Curtis in (ESEPG 1999) suggest that companies that excel at SPI are those that are effective at change management, amongst other things. One of the critical success factors of change management is the vision and drive of change agents (McCalman and Paton 1992). If change agents are not committed to the programmes through a lack of vision or lack of motivation, then such programmes are bound to be less effective (McCalman and Paton 1992). Change agents can also be good indicators of how well programmes are doing in companies.

SPI managers are change agents in SPI programmes (Moitra 1998). We suggest that as a result, it is possible to assess the state of SPI practice in companies by understanding the perception of SPI managers. No studies have actually examined SPI managers' perception of SPI. However, (Nicholson et al. 1995) and (Warden and Nicholson 1995) investigated the perception of quality improvement personnel. Findings from these previous studies suggest that the agents of change themselves are demotivated in their jobs. Nicholson and Warden suggest that such a finding presents a bleak prospect for quality improvement programmes in companies.

We assert that important reasons for understanding SPI managers' perception of SPI are to:

- Check on the motivation and experiences of people running SPI in companies.
- Provide a barometer for the state of SPI practice in companies.

In this paper, we present a study of Software Process Improvement (SPI) managers' perceptions of the factors that motivate software practitioners to support SPI. The study was conducted by surveying the experiences and opinions of eighty SPI managers from a wide range of UK companies. In this paper, we present the results of the survey in simple statistical analysis. We aim to answer the following research question:

What perceptions do SPI managers have of the factors that can motivate practitioners to support SPI?

We address the research question by investigating SPI managers' experiences and opinions of SPI. It also highlights the expectations that SPI managers have of SPI programmes. Overall, this study

provides a 'snapshot' of the issues that underpin software practitioners' motivators and de-motivators for SPI in UK companies.

This rest of this chapter is structured as follows: Section 2 presents the background and rationale for this study. Section 3 presents the study method, a profile of the sample of SPI managers and explains some of the organisation of the analysis. Section 4 presents analysis of survey data. Section 5 discusses the analyses. Section 6 summarises the results of this study and concludes this paper.

2 Literature background

The SPI literature reveals certain fundamental issues about SPI. To understand the state of SPI practice in companies through the perception of SPI managers, it is necessary to understand their perception on the fundamental issues. These fundamental issues, can be summarised into the inputs, implementors and outputs of SPI according to systems theory (Curtis 1989). The following is an overview of these issues.

- Experiences of SPI

Implementing quality programmes of any sort is about people (DeMarco and Lister 1987). Implementing SPI programmes, in particular, requires close attention to the experiences and expectations of the practitioners who manage and carry out the SPI practices. This research suggests that it is important to survey experiences of SPI managers to establish if it impacts on their opinions of SPI. Other studies have suggested that practitioner experiences forge attitudes and subsequent behaviour towards SPI (Humphrey 1998).

- Expectations of SPI

It can be argued that practitioners' expectations of SPI also contribute towards their perceptions of SPI. Practitioner expectations of SPI may be formed by several factors, but most importantly, experiences of SPI play a significant part in forming such expectations. Humphrey explains that practitioners tend to have negative expectations of an improvement programme if their experiences of that programme are also negative (Humphrey 1998). It is therefore useful in this research to understand the expectations of SPI managers for SPI to assess whether such expectations have been formed from their experiences.

- Product quality improvement

The fundamental ethos of SPI is improving software quality through focussing on development processes (Deming 1986; Humphrey 1989). This means that SPI success can be judged through the quality improvement achieved in the product. Although some recent empirical studies on the impact of SPI report substantial quality improvements, for example in (Ahuja 1999; Fitzgerald and O'Kane 1999; Krasner 1997; Paulk et al. 1994), an overwhelming body of evidence has yet been established to support this claim. SPI managers' perception of the impact of SPI on product improvement can offer some insight into the extent of product improvement.

- Cost-effectiveness

Crosby's assertion that "*quality is free*" (Crosby 1979) has often been quoted by companies embarking on SPI programmes to justify the substantial start-up costs of such programmes. Indeed, high profile SPI success stories like Hughes Aircraft Company (Willis et al. 1998) and Raytheon (Haley 1996) report long term cost gains. Some companies have reported substantial increases in the Return On Investment (ROI) of between 5:1 to 9:1 (Krasner 1997). Overall, reports from companies practising SPI in the last ten years indicate long term sustainable cost benefits (Fox and Frakes 1997; Haley 1996; Herbsleb et al. 1994; Krasner 1999; Willis et al. 1998).

However, in the UK there is a lack of independent evidence, anecdotal or otherwise, to either support or counter published accounts of the cost effectiveness of SPI as in, for example (Krasner 1997), where companies like Hewlett Packard report "*savings of \$20Million in one financial year alone due to inspections*". SPI managers' perception can provide some evidence of cost benefits in UK companies.

- Senior management support

Many case studies consider senior management support critical to the success of SPI programmes (Krasner 1999; Wilson, Hall, and Baddoo 2001). Telcordia's SPI success reports visible senior management support as critical to this success (Pitterman 2000). Laporte and Trudel [1998] suggest that by showing understanding and a full commitment to process issues, and displaying this through their day to day activities, senior managers are sending a 'positive' signal to middle and lower ranked practitioners about their commitment to SPI. Such a signal is instrumental in gaining the buy-in of other practitioner groups for SPI (Laporte and Trudel 1998).

However, with the exception of the handful of successful case studies, there are few independent accounts reporting on the level of senior management support, generally, for SPI programmes. This research suggests that understanding SPI managers' perception on this issue can provide some indication of the level of senior management support for SPI in practice.

- Developer buy-in

Successful SPI accounts like (Goldenson and Herbsleb 1995; Herbsleb and Goldenson 1996; Krasner 1997; Paulk et al. 1994; Paulk 1999; Paulk, Goldenson, and White 2000; Willis et al. 1998) suggest that to achieve high maturity in development processes, it is necessary to transfer ownership of such processes to the people who actually perform the functions. This indicates that it is vital to let the practitioners who conduct the improvement effort to have ownership of those processes. However practitioners must buy-in to SPI first before this stage can be reached. The successful case studies report that buy-in can be achieved through consultation with practitioners where their views are encouraged and incorporated into company-specific improvement initiatives.

Understanding SPI managers' perceptions of developer buy-in in companies can provide a strong indication of the level of potential grassroots support for SPI in companies.

- Implementation approaches

Recent accounts of SPI programmes in companies question the merits of the classical management approaches to implementing SPI (Hammock 1999; Paulk 1999). In fact there are increasing calls to software engineering managers to move away from top-down and bottom-up approaches of implementing programmes to more adaptive forms of implementation. For example, inside-out or "*growing programmes in situ*" (Hovenden et al. 1996). There is also an increasingly popular assertion that whole improvement initiatives should be run from within, not fostered by external agencies (Krasner 1997). SPI managers' perception of the favoured implementation approaches can provide insight into how SPI is being implemented in practice.

Overall, this study of SPI managers' perceptions provides insight to SPI practice in companies. In particular, it provides insight into how SPI is being supported in companies, and it helps contribute some assessment of the effectiveness of SPI uptake in a sample of UK companies.

3 Study Method

In this study we used a questionnaire survey to collect both subjective data - SPI managers' perception of SPI in their companies - and objective data - demographic and background information about SPI managers.

A sample of SPI managers was identified using public domain information about software development companies. This information included relevant mailing lists and conference attendance lists. Questionnaires were mailed to SPI managers at one thousand companies and two hundred replies were received of which eighty were fully completed questionnaires. A response rate of 20% is normally considered acceptable for a survey of this nature, however, considering many UK software companies have no formal SPI programme, a total response of two hundred questionnaires could, in this case, be considered very good.

Questionnaire responses were measured on a Likert scale (Likert 1932) of 1 to 5, where 1 indicates strong disagreement with a statement and 5 strong agreement. Appendix A presents a copy of the complete questionnaire.

The following sub-sections discuss the profile of SPI managers in this study and explain the use of sub-sample analysis to support findings.

3.1 Respondent profile

This section presents some demographic features of SPI managers who responded to this survey.

- Job titles

Table 1 shows that people who assume SPI management roles in companies have a variety of job titles. This is often because their SPI mandates happen to be additional responsibilities. So the other titles reflect other responsibilities. However, throughout this research, these people are referred to as SPI managers.

Job title	No. in sample	% in sample
Quality director/manager/co-ordinator	28	35
Development director/manager/co-ordinator	12	15
SPI managers	10	12.5
Software manager/director	6	7.5
Managing Director/General manager/Partner	6	7.5
Project director/manager	4	5
Senior staff/software engineer	4	5
Technical manager/specialist	4	5
Programme manager	2	2.5
Other	4	5
Total	80	100

Table 1: People with SPI responsibility

- Career history

Figure 1 shows that more than half of respondents in this study have over fifteen years experience in the software industry. Figure 1 also shows that many have not been with their present companies for longer than five years. This suggests that most SPI managers are experienced computing professionals who have made the transition to SPI later in their career. The late up-take of SPI responsibilities could be attributed to the fact that the concept of SPI is a relatively recent phenomenon.

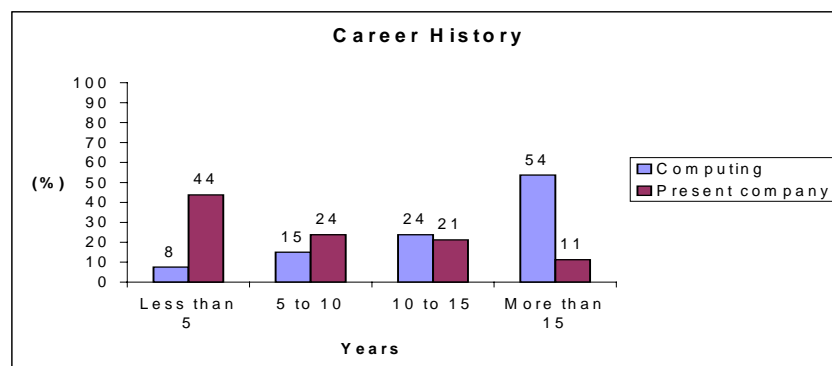


Figure 1: Distribution of years served in computing and present company

- Education.

More than 80% of SPI managers have obtained at least Bachelors degree, but only 35% of these degrees are from computing disciplines. A majority of the remaining 65% had degrees from other science disciplines with engineering and the natural sciences being most predominant. This indicates that a high percentage of SPI managers in this study have a technical background, albeit not necessarily all in Computer Science. Our later studies show that some practitioner groups are de-motivated by managers who do not have a technical background.

- Professional body membership

Overall, 50% of respondents were affiliated to professional technical bodies. This research suggests that this may indicate that SPI managers' educational background may be technical too.

3.2 Sub-Sampling

One of the questions in this study asks SPI managers to rate their familiarity of SPI on a five-point scale. We conduct further analysis of the opinions elicited in this study by separating the responses of practitioners who report that they are highly familiar with SPI from those who report a low familiarity. We have done this because we suggest that software practitioners' familiarity with SPI qualifies their opinions on SPI, implying that respondents who are more familiar with SPI are likely to have more reliable opinions. Consequently, 62% of the sample in this study are familiar with SPI whilst 18% are less so, with the rest remaining neutral.

Despite this breakdown, the perceptions presented in the results section are generally of the whole sample, because familiarity with SPI makes significant differences to only a few of the results reported. As a result sub-sample perceptions are only presented in cases where the difference between familiar and unfamiliar managers is significantly different.

The three sets of opinion: from overall sample, 'familiar managers' and 'unfamiliar managers' are presented in a comparative table as Appendix B

4 Analysing SPI managers' perceptions

Analyses of SPI managers' perceptions are presented in terms of the inputs, implementors and outputs that are reported to influence SPI success.

4.1 Analysis of the inputs

The following summarises responses to questions on the inputs of SPI.

- Familiarity with SPI

"I am familiar with implementing Software Process Improvement initiatives"

Figure 2 shows that the majority of managers with SPI responsibility are familiar with implementing SPI initiatives. 62% of respondents indicated either strong or very strong familiarity with implementing SPI programmes. However, 18% said they were not familiar with SPI. Closer analysis shows that these 'unfamiliar managers' are not necessarily new to their positions. For example, some managers who report least SPI familiarity had spent between fifteen and twenty-seven years in their present jobs. Often these managers do not have official SPI job titles despite having responsibility for SPI. This may mean that they have recently been given SPI remits within an existing role. This may be related to other factors like the lack of senior management support that these managers also report. Indeed one respondent emphasised this by saying...

"I have found that many SPI managers are "lumbered" with the position and are expected to cope rather than being given the support and resources to make it worthwhile"

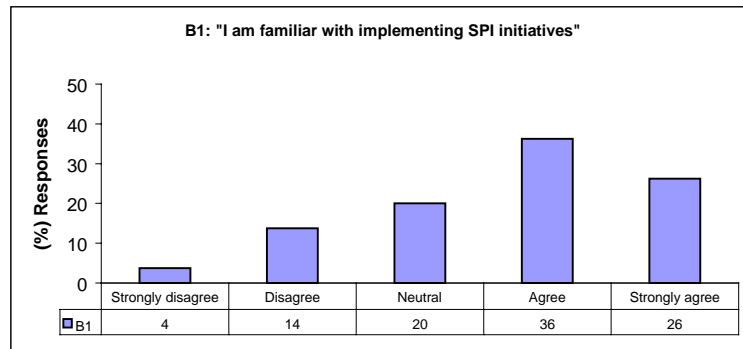


Figure 2: Familiarity with SPI

- Experience of SPI

“My experience of Software Process Improvement has been positive.”

More than 65% of respondents reported positive experiences of SPI. Figure 3 shows that the percentage of negative experiences of SPI was negligible at 7%. Furthermore, 82% of managers who are familiar with SPI report positive experiences of SPI, whilst 14% of managers who are unfamiliar with SPI report negative experience of SPI. Overall, these results suggest that SPI managers' general experience of SPI is positive.

However, a few managers who responded positively to also had issues to raise about SPI. For example, one manager said:

“My experience of championing SPI has been mixed. Initially we had an excellent SPI programme. However changes in the company have made implementing this far harder.”

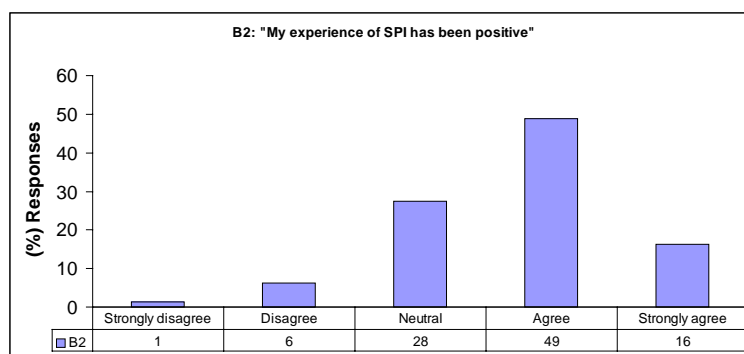


Figure 3: Experiences of SPI

- Difficulty of implementing SPI

“Implementing an effective SPI programme is difficult”

SPI managers generally agree that it is difficult to implement an effective SPI programme. Figure 4 shows that only 8% of respondents disagree with this statement. Sub-sample analysis based on reported familiarity with SPI indicates that both familiar and unfamiliar managers agree that SPI programmes are difficult to implement, with 80% of familiar managers in agreement with the statement. This finding indicates that even though a strong majority of respondents have reported positive SPI experiences, they are still objective about how difficult it is to implement SPI and therefore have realistic expectations of SPI.

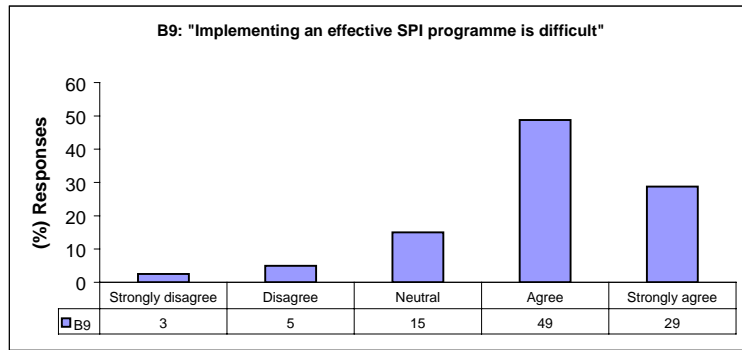


Figure 4: The difficulty of implementing SPI

4.2 Analysis of *implementors*

The following is a summary of responses to questions on the *implementors* to SPI programmes.

- Approaches for implementing SPI

“Software Process Improvement should be implemented via a Top-Down approach.”

Figures 5 and 6 show that SPI managers' perceptions of approaches to implementing SPI are varied. Less than 45% agree that SPI should be implemented by a top-down approach. The largest single percentage of responses, 30%, was neutral. Indicating that there is a sizeable proportion of SPI managers who are undecided as to whether top-down is the appropriate approach to implementing SPI or not.

Further analysis of responses from managers familiar with SPI reveals varied opinions. 48% agree that SPI should be implemented top-down, 30% disagree and 22% remained neutral. Again, this implies that SPI managers' opinions are not overwhelmingly strong as to whether SPI should be implemented top-down or not.

“Software Process Improvement should be implemented via a Bottom-Up approach”

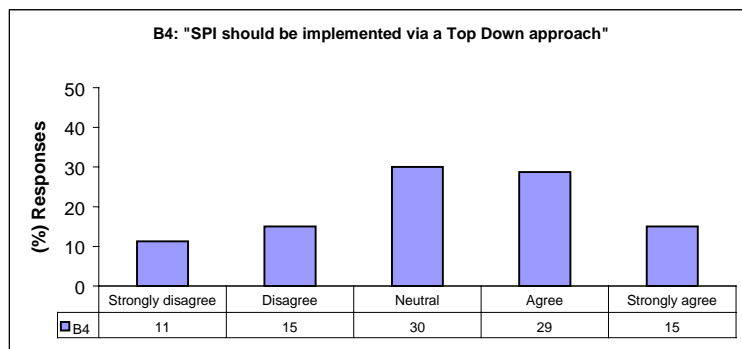


Figure 5: Top-down implementation of SPI

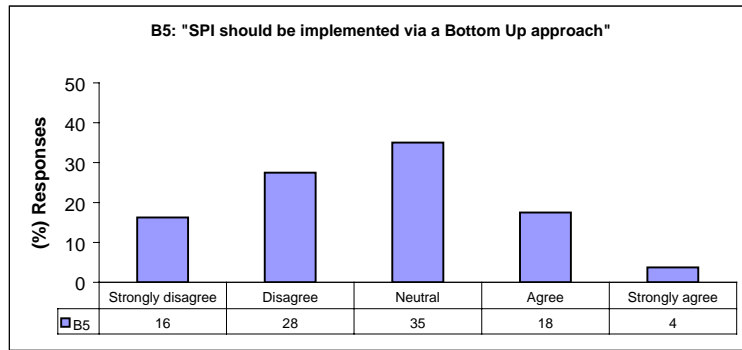


Figure 6: Bottom-up implementation of SPI

SPI managers are also undecided about bottom-up approaches to implementing SPI. Only 4% of respondents strongly agreed that such an approach should be used, with the single highest percentage, 35%, remaining neutral.

This result may reflect a problem with the way the statements were presented in the questionnaire. SPI managers may prefer a combined top-down/bottom-up approach to implementing SPI. In fact this is substantiated by annotations provided by some respondents:

"I think you need to use both approaches".

"It depends on the culture. No 'right' answer".

- Management and grassroots support

“Senior managers are very supportive of SPI.”

SPI managers' perception is that senior management is not supportive of SPI. Figure 7 shows that less than 50% of respondents indicated that their senior managers were supportive of SPI.

This finding is important as successful accounts of SPI suggest that senior management support and commitment for SPI is critical.

This research suggests that senior management motivation for SPI may often be short term and profit motivated, whereas SPI managers may perceive SPI as long term investment. This potential conflict between these two perspectives may explain why most SPI managers in this study consider senior managers as not supportive of SPI.

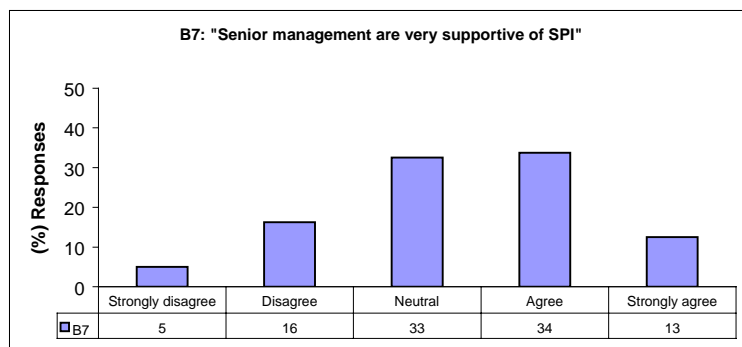


Figure 7: Senior management support of SPI

“Software developers are enthusiastic about SPI”

Figure 8 shows that SPI managers' perception of developers' support for SPI is weak. Less than 40% of respondents felt that grassroots practitioners were enthusiastic about SPI. One respondent added that developers were *"sceptical if it means more work, but want to see improvement"*.

This is another important finding in view of what the literature and published case studies advocate about grassroots support and buy-in for SPI. Our findings indicate problems with grassroots support for SPI in UK companies.

We suggest that because senior management commitment and grassroots support for SPI are two critical factors for SPI success in companies, the above findings make it necessary to conduct more studies to establish why these two group of practitioners appear not to be supporting SPI. We suggest that these findings on senior management commitment and grassroots support are major justifications for further studies in this area.

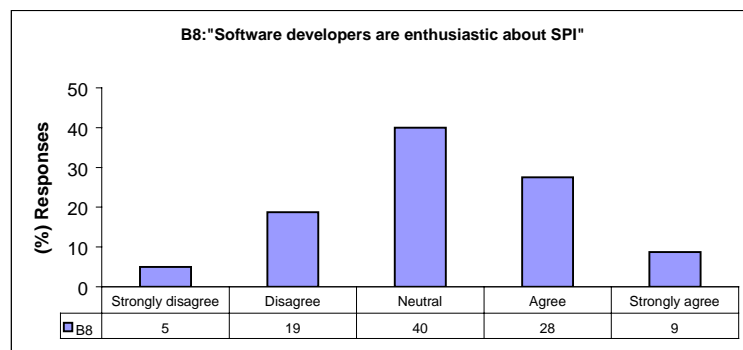


Figure 8: Developers' enthusiasm for SPI

4.3 Analysis of SPI outputs

The following are summary responses to questions on outputs of SPI.

- Impact of SPI on software quality

"Software Process Improvement is an effective approach to improving the quality of the software product."

Figure 9 shows that most SPI managers acknowledge that SPI is an effective approach to improving the quality of software. This is one of the strongest positive responses in the study. More than four out of five respondents say that SPI leads to better software quality. The proportion of negative responses is negligible and is not from the same respondents as reported negative experiences of SPI in question B2. This shows that even managers whose experience of SPI has not been positive, still believe that SPI can lead to better quality software. It suggests that the negative experiences reported in question B2 are not negative evaluations of SPI as such, but rather of particular implementations of SPI. It is also interesting because it supports the view that quality problems in software development can be solved by focusing on the processes (Humphrey 1989).

Further analysis reveals that 90% of managers who reported familiarity with SPI agree that SPI leads to improved software quality, with a further 64% of unfamiliar managers also agreeing with the statement.

Overall, this finding presents a strong indication of SPI managers' conviction about the effectiveness of SPI in improving software quality.

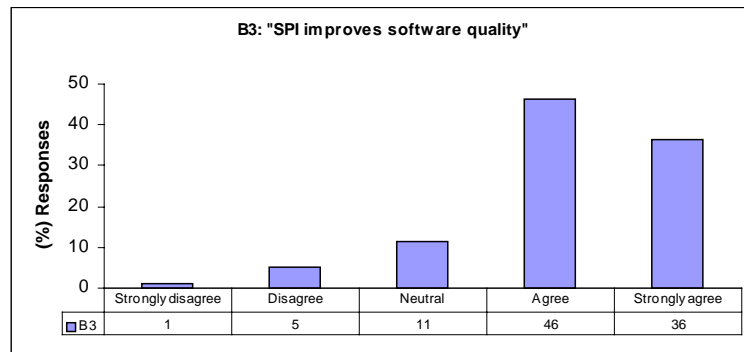


Figure 9: Software quality and SPI

- Cost/Benefits of SPI

“In the long term, the cost of setting up an SPI programme is compensated for by the cost savings made elsewhere in the development process.”

Figure 10 shows that 74% of SPI managers acknowledge that SPI has benefits that outweigh costs. Only 7% of respondents disagree with this.

This is an important finding for two reasons. Firstly, because it supports what companies undertaking successful SPI programmes have been reporting over the years. In fact many successful companies report a healthy return on investment. Secondly, it is a view that reflects the general literature on cost effectiveness of SPI programmes.

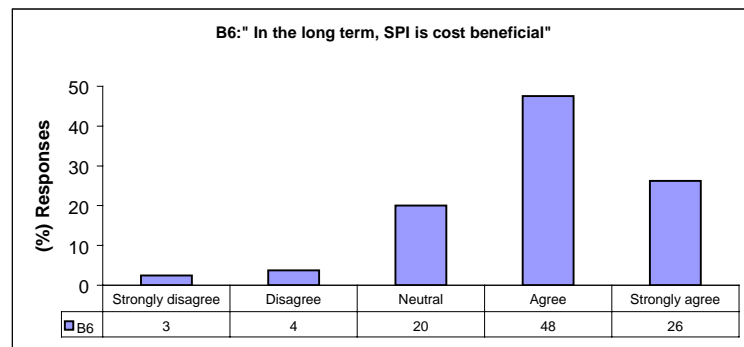


Figure 10: Cost benefits of SPI

4.4 Process maturity

Correlation analyses were conducted to establish whether the maturity of a company’s processes influenced the responses offered by its SPI managers. This analysis was carried out with fifty out of the eighty respondents because the other respondents declined to provide data on the maturity of their company's processes.

No correlation coefficients were significant. This result indicates that SPI managers' responses to questions do not relate to the maturity of the processes in their companies. This may further suggest that the success or otherwise of SPI in individual companies has not biased the perceptions collected in this study. SPI managers from companies with more mature processes are not being more optimistic and neither are managers from companies with less mature processes being less optimistic about the prospect of SPI.

5 Discussion

In this section, we present a summary of the study findings and show how these findings answer the research question:

What perceptions do SPI managers have of the factors that can motivate practitioners to support SPI?

- Experience of SPI

SPI managers generally report positive experiences of SPI. The findings also show that SPI Managers with a high familiarity of SPI are more positive about their experiences of SPI.

On the whole, opinions of SPI managers who are familiar with SPI appear stronger than the opinions of the overall sample. Managers who are familiar with SPI consistently confirmed the general view of the whole sample by reporting stronger percentages. This shows some consistency in the perceptions elicited in this study and also affirms the notion that SPI managers' experiences and familiarity with SPI informs their perception of SPI.

- Expectations of SPI

SPI managers are optimistic about the success of SPI. They generally have positive expectations of SPI, but acknowledge that there are significant difficulties with managing SPI programmes. Some SPI managers indicate that some of the difficulties with SPI are caused because software projects often take priority over improvement practices. On the whole, SPI managers have the perception that there are effective ways of managing SPI programmes successfully.

- SPI impact on software quality

SPI managers' positive expectations of SPI generally also translate into their perceptions of the impact of SPI on software quality. This study shows that SPI managers perceive that SPI leads to improved product quality. This represents a positive reflection of SPI managers' expectations of SPI.

- Cost benefits

SPI managers hold the view that SPI has cost benefits. This finding is encouraging because justifying the cost for SPI is often the most difficult hurdle to overcome in implementing SPI. This research suggests that if SPI managers are convinced of the cost benefits of SPI, then it is likely to make the process of convincing senior management easier. Again, this finding indicates that SPI managers have positive expectations of SPI.

- Senior management support

In this study, SPI managers do not think that senior management is supportive of SPI. Less than half of respondents indicated that senior management was supportive of SPI. This research suggests that this is an important finding because of how critical senior management support has been reported to be to SPI success.

- Developers' support for SPI

SPI managers also indicate that developers are not enthusiastic about SPI. This finding shows that developers are not supportive of SPI. This is another important finding because of the importance of grassroots support for SPI success.

Overall, SPI managers' perceptions indicate that there is insufficient support for SPI from practitioners at both grassroots and senior management levels of their companies. These findings are important indicators for the remaining studies in this research.

- Process maturity

Overall, SPI managers' responses bear no relation to the current maturity of the processes in their companies. This finding may imply that the responses provided by SPI managers are drawn from their longer-term experience of SPI as opposed to what their current circumstances with SPI may reflect. As a result, this research suggests that this finding gives more confidence to the data provided by SPI managers.

6 Conclusion

Overall, the findings from this study raised some important questions. Two of these questions relate to support for SPI from senior managers and developers. Our findings suggest that:

- 1 Senior managers may not be supportive of SPI
- 2 Developers may not be as enthusiastic as they could be about SPI

We suggest that these two findings are the strongest indication of the state of motivators and de-motivators for SPI discerned from this study. They indicate that software practitioners of all levels are not sufficiently motivated to support SPI and are being de-motivated from supporting SPI. These findings suggest that:

- Factors that can improve practitioners' support for SPI have not been sufficiently addressed.
- There may be factors that are hindering practitioners from supporting SPI

It is therefore important to identify, from software practitioners, what the factors are that can improve their support for SPI.

References:

- Ahuja, S. 1999. Process Improvement In A Rapidly Changing Business And Technical Environment. Paper read at Fourth Annual European Software Engineering Process Group Conference, at Amsterdam, Holland.
- Crosby, P. B. 1979. *Quality Is Free*. New York: McGraw-Hill.
- Curtis, Graham. 1989. *Business Information Systems: Analysis, Design and Practice*. London: Addison-Wesley Publishing Company.
- DeMarco, T., and T. Lister. 1987. *Peopleware - Productive Projects And Teams*: Dorset House.
- Deming, W. E. 1986. *Out Of Crisis*. Cambridge, Mass.: MIT Press.
- ESEPG. 1999. Conference Proceedings. Paper read at Proceedings Of The Fourth Annual European Software Process Group Conference, June 6-10, at Amsterdam, Holland.
- Fitzgerald, B., and T. O'Kane. 1999. A Longitudinal Study Of Software Process Improvement. *IEEE Software* May/June:37-45.
- Fox, C, and W Frakes. 1997. The Quality Approach: Is It Delivering? *Communications of the ACM* 40 (6):25-29.
- Goldenson, D. R., and J. D. Herbsleb. 1995. After The Appraisal: A Systematic Survey Of Process Improvement, Its Benefits And Factors That Influence Success: Software Engineering Institute.
- Haley, J. H. 1996. Software Process Improvement At Raytheon. *IEEE Software* November:33-41.
- Hammock, N. 1999. Critical Success Factors In Software Process Improvement. Paper read at Proceedings Of The Fourth Annual European Process Group Conference, June 6-10, at Amsterdam, Holland.
- Herbsleb, J. D., A. Carleton, J. Rozum, J. Siegel, and D. Zubrow. 1994. Benefits Of CMM-Based Software Process Improvement: Initial Results: SEI Technical Report.
- Herbsleb, J. D., and D. R. Goldenson. 1996. A Systematic Survey Of CMM Experience And Results. Paper read at 18th International Conference on Software Engineering - ICSE, 25-29 March, at Berlin, Germany.
- Hovenden, F. M., H. C. Sharp, S. D. Walker, and M. Woodman. 1996. Building Quality Into Scientific Software. *Software Quality Journal* March (5):25-32.
- Humphrey, W. S. 1998. Why Don't They Practice What We Preach? *Annals Of Software Engineering* 6 (4):201-222.
- Humphrey, W. S. 1989. *Managing The Software Process*: Addison Wesley.
- Krasner, H. 1999. *Accumulating The Body Of Evidence For The Payoff Of Software Process Improvement -1997* URL, 1997 [cited May 1999]. Available from <http://www.utexas.edu/coe/sqi/archive/krasner/spi.pdf>.
- Krasner, H. 1999. Using The Cost Of Quality Approach For Software. Paper read at Proceedings Of The 12th International Software Quality Week Conference, May 24-28, at San Jose, California.
- Laporte, Claude. Y., and Sylvie. Trudel. 1998. Addressing The People Issues Of Process Improvement Activities At Oerlikon Aerospace. *Software Process - Improvement And Practice* 4:187-198.
- Likert, Rensis. 1932. A Technique For The Measurement Of Attitudes. *Archives Of Psychology* 22 (140).

- McCalman, James, and Robert A Paton. 1992. *Change Management: A Guide To Effective Implementation*. London: Paul Chapman Publishing.
- Moitra, Deependra. 1998. Managing Change For Software Process Improvement Initiatives: A Practical Experience-Based Approach. *Software Process - Improvement And Practice* 4:199-207.
- Nicholson, I., D. Holmes, J. Richards, and R. Warden. 1995. Motivation Of Quality Practitioners. In *Quality Management III*, edited by M. M. Ross, C. A. Brebbia, G. Staples and J. Stapleton: Computational Mechanics Publications.
- Paulk, M, C Weber, B Curtis, and M Chrissis. 1994. A High Maturity Example: Space Shuttle Onboard Software. In *The Capability Maturity Model: Guidelines For Improving Software Process*: Addison-Wesley.
- Paulk, M. 1999. Practices Of A High Maturity Organisation. Paper read at Proceedings Of The Fourth Annual European Software Engineering Process Group Conference, June 6-10, at Amsterdam, Holland.
- Paulk, M.C. , D. Goldenson, and D.M. White. 2000. The 1999 Survey of High Maturity Organizations: Software Engineering Institute, Carnegie Mellon University.
- Pitterman, B. 2000. Telcordia Technologies: The Journey To High Maturity. *IEEE Software* 17 (4):89-96.
- Warden, R., and I. Nicholson. 1995. IT Quality Initiatives At Risk. *Software Quality Management New Year* (24):24-27.
- Willis, R. R., R. M. Rova, M. D. Scott, M. I. Johnson, J. F. Ryskowski, J. A. Moon, K. C. Shumate, and T. O. Winfield. 1998. Hughes Aircraft's Widespread Deployment Of A Continuously Improving Software Process: Software Engineering Institute, Carnegie Mellon University.
- Wilson, D., T. Hall, and N. Baddoo. 2001. A Framework For Evaluation And Prediction Of Software Process Improvement Success. *Journal of Systems and Software* 59 (2):135-142.

Appendix A: Sample questionnaire - SPI managers' perception of SPI

Thank you for dedicating some time to completing this questionnaire. We guarantee that all the information given will be treated in the strictest confidence.

Your name: _____

Your job title: _____

E-mail address: _____

Company: _____

Address: _____

Date: _____

Section i : Personal Background

This section is concerned with information about your background and experience as a software process improvement manager. This information will be treated in the STRICTEST CONFIDENCE and any publication of this study will present information in aggregate form and such information will be anonymous and unattributable to individual organisations or individual respondents.

A.1 How long have you worked in this company? _____ years

A.2 How long have you worked in computing/software engineering/IT? _____ years

A.3 What is your educational background?
(You may tick more than one)

- a. Bachelors degree
- b. Masters degree
- c. Doctorate degree
- d. None of the above
- e. Other

Please specify _____

A.4 Are any of your educational qualifications in the following areas?

- a. Computer science
- b. Software engineering
- c. Information Systems
- d. Information Technology
- e. Other

Please specify _____

A.5 Of which of the following professional bodies are you a member?
(You may tick more than one)

- a. British Computer Society (BCS)
- b. Institute of Electronic Engineers (IEE)
- c. Institute of Electronic and Electrical Engineers (IEEE)
- d. Association for Computer Machinery (ACM)
- e. Institute for the Management of Information Systems
- f. Association of Chartered Engineers
- g. None of the above
- e. Other

Please specify _____

Section ii: Personal Opinions

This section is concerned with information about your opinions as a software process improvement manager. This information will be treated in the STRICTEST CONFIDENCE and any publication of this study will present information in aggregate form and such information will be anonymous and unattributable to individual organisations or individual respondents.

In response to the following statements, please indicate your level of agreement by circling the appropriate number on the scale: where 5 indicates a strong agreement with the statement made and 1 indicates a strong disagreement.

- i.e. 5 = Strongly agree
 4 = Agree
 3 = Neutral
 2 = Disagree
 1 = Strongly disagree

B.1 I am familiar with implementing Software Process Improvement initiatives.

Agree 5 ----- 4 ----- 3 ----- 2 ----- 1 Disagree

B.2 My experience of Software Process Improvement has been positive.

Agree 5 ----- 4 ----- 3 ----- 2 ----- 1 Disagree

B.3 Software Process Improvement is an effective approach to improving the quality of the software product.

Agree 5 ----- 4 ----- 3 ----- 2 ----- 1 Disagree

B.4 Software Process Improvement should be implemented via a Top Down approach.

Agree 5 ----- 4 ----- 3 ----- 2 ----- 1 Disagree

B.5 Software Process Improvement should be implemented via a Bottom Up approach.

Agree 5 ----- 4 ----- 3 ----- 2 ----- 1 Disagree

B.6 In the long term, the cost of setting up an SPI programme is compensated for by the cost savings made elsewhere in the development process.

Agree 5 ----- 4 ----- 3 ----- 2 ----- 1 Disagree

B.7 Senior managers are very supportive of SPI.

Agree 5 ----- 4 ----- 3 ----- 2 ----- 1 Disagree

B.8 Software developers are enthusiastic about SPI.

Agree 5 ----- 4 ----- 3 ----- 2 ----- 1 Disagree

B.9 Implementing an effective SPI programme is difficult.

Agree 5 ----- 4 ----- 3 ----- 2 ----- 1 Disagree

Appendix B: Sub-Sample percentages based on reported familiarity with SPI

Questionnaire statements	Whole Sample (%) Responses					Familiar Managers (%) Responses					Unfamiliar Managers (%) Responses				
	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
B1: "I am familiar with implementing SPI initiatives"	4	14	20	36	26	0	0	0	58	42	21	79	0	0	0
B2: "My experience of SPI has been positive"	1	6	28	49	16	0	4	14	60	22	7	7	72	14	0
B3: "SPI improves software quality"	1	5	11	46	36	0	6	4	44	46	7	0	29	43	21
B4: "SPI should be implemented via a Top Down approach"	11	15	30	29	15	14	16	22	28	20	14	14	43	22	7
B5: "SPI should be implemented via a Bottom Up approach"	16	28	35	18	4	16	26	34	18	6	21	36	36	7	0
B6: "In the long term, SPI is cost beneficial"	3	4	20	48	26	0	2	16	50	32	14	0	29	43	14
B7: "Senior management are very supportive of SPI"	5	16	33	34	13	2	14	36	32	16	14	14	36	29	7
B8: "Software developers are enthusiastic about SPI"	5	19	40	28	9	2	20	38	30	10	14	14	43	29	0
B9: "Implementing an effective SPI programme is difficult"	3	5	15	49	29	0	4	16	46	34	14	0	14	65	7

NB: Percentage figures have been rounded up for presentation purposes

Biographies: Dr Nathan Baddoo and Dr Tracy Hall

Nathan Baddoo

Nathan Baddoo is a senior lecturer in the School of Computer Science at the University of Hertfordshire. His research focuses on human factors in software processes. He holds many publications in this area. Nathan completed a PhD in Motivators and De-motivators in Software Process Improvement, in 2001. His current research interests are in the general areas of software quality and software process improvement. Nathan is also exploring how to construct prediction models of software engineers' motivation by using concepts like Agent-based simulation and fuzzy logic.

Tracy Hall

Tracy Hall is a principal lecturer in the Department of Computer Science at the University of Hertfordshire. She heads the Systems and Software Research group (SSR), previously she headed the Centre for Systems and Software Engineering (CSSE) at South Bank University. For the past eight years she has been active in the area of software quality, measurement and software process improvement. She has many publications in this area. All of the research that Tracy undertakes is in direct collaboration with companies in the software industry. To date she has collaborated with over 25 companies. Over the last four years she has been a programme committee member of the British Computer Society's annual Quality Management conference.

She is also an active member of the UK think tank: the Centre for Software Reliability Council. She is the co-editor of a book on professionalism of software practitioners, 'The Responsible Software Engineer', Springer, 1997 and has chaired highly successful international conference in that area.

From Waterfall to Evolutionary Development (Evo) Or How to create faster, more user-friendly and more productive software

*Trond Johansen
QA & Process Manager, FIRM
Trond.Johansen@firmglobal.com*

Abstract

1 The shift of focus: from waterfall to evolutionary development

Peter Myklebust, FIRM CTO, and I heard Tom Gilb speak about evolutionary project management (Evo) at a software conference autumn 2003. We saw that Evo attacked many of the flaws in our waterfall process; most importantly it gave us a high focus on quality attributes that we felt could have been better in our latest release.

We decided to do an Evo pilot with a development phase of 3 months.

1.1 FIRM's interpretation of Evo: Basis for the 3 month trial period

After the one day crash course with Tom and Kai Gilb and a literature study ("Competitive Engineering" by Tom Gilb and other material on the subject), our overall understanding of Evo was this:

- Find stakeholders
- Define the stakeholders' *real* needs and the related product qualities
- Identify past/status of product qualities and your goals, and goal levels
- Identify possible solutions for meeting your goals
- Develop a step-by-step plan for delivering improvements with respect to Stakeholder Values & Product Quality goals:

And most importantly:

- Deliveries every week (approximately)
- Measure weekly: are we moving towards our goals?

1.2 Working with requirements the Evo way

With Evo, our requirements process changed. Previously we focused mostly on function requirements, and not on quality requirements. It is the quality requirements that really separate us from our competitors.

1.3 Find Solutions that take you closer to your goals

For every quality requirement we looked for possible Solutions (Design Ideas)

We evaluated all these, and specified in more detail those we believed would add the most value (take us closer to the goal)

1.4 Working evolutionarily, the FIRM Evo week

We decided that every step should last one week, and organized the week in a special way.

On Friday we plan deliverables for version N, at the same time as we build and deploy version N-1 on the test server. Monday to Thursday is dedicated to design, code and test. During the week, the project collects feedback from stakeholders based on the previous step/week.

We collected the most promising solutions and included them in an Evo plan (also called Impact Estimation Table: IET). The solutions were evaluated with respect to value for clients/cost of implementation, choosing the ones with the highest value to be implemented first.

2 *Impacts on our product: experiences and conclusions*

2.1 The method's Impact on Conformat product qualities

The impact described in this paper is based on internal usability test, productivity tests, performance tests carried out at Microsoft Windows ISV laboratory in Redmond USA, and direct customer feedback. Only highlights of the impacts are listed here

Description of performance requirement and corresponding work task	Before Evo	Current Status
Usability.Productivity: Time for the system to generate a survey	7,200 sec	92 sec
Usability.Productivity: Time to set up a typical specified Market Research-report (MR)	65 min	20 min
Usability.Productivity: Time to grant a set of End-users access to a Report set and to distribute report login info.	80 min	5 min
Usability.Intuitiveness: The time in minutes it takes a average experienced programmer to define a complete and correct data transfer definition, with Conformat Web Services, without any user documentation, or any other aid	15 min	5 min
Performance.Runtime.Concurrency: Maximum number of simultaneous respondents executing a survey with a click rate of 20 secs. and a response	250 users	1350

time of <500 ms, given a defined [Survey-Complexity] and a defined [Server Configuration, Typical]		
--	--	--

Table 1: Improvements to product qualities

2.2 Conclusion

Overall, the whole organization has embraced Evo. The release of Conformat 8.5 showed some of Evo's great potential, and we will work hard to utilize it to the full in the future. In June 2004 we had Tom and Kai Gilb for a 4 days course for the whole R&D department and related resources and we hope the next version of Conformat will prove that we have matured in our understanding and execution of Evo.

Keywords

Evolutionary development (Evo), product qualities, Evo week, Impact Estimation Table, measurable requirements, solutions, feedback, stakeholder values, open architecture, continuous integration CI

1 About the company

FIRM was established in 1996, and has 70 employees in 4 offices (Oslo, London, New York and San Francisco). FIRM delivers one software product: Confirmat. Confirmat is a web-based application which enables organizations to gather, analyze and report key business information across a broad range of commercial applications. Confirmat can be applied to any information-gathering scenario, but its three main data sources are: Customer Feedback, Market Feedback and Employee Feedback.

The FIRM R&D department consist of about 20 people, including a Quality Assurance department of 3 people. These people are mainly involved in product development of Confirmat, but we also do custom development for clients who fund new modules of the software.

2 Development background & history

In the very beginning, when FIRM only had a couple of clients, our development was very ad-hoc and customer driven. We didn't follow a formal development process. The software was updated nearly on a daily basis based on client feedback. You can say that we had one of the important elements in Evo: Deliver stakeholder value **fast**.

This ad-hoc development resulted in nice features for the few dedicated clients we had, but it also resulted in a lot of defects, long stressful nights, and little control.

As our client base grew, we felt a need to introduce more-formal processes in order to increase our quality standards. Larger clients started to ask questions regarding our development processes.

We formalised the development process according to a waterfall model, and started climbing the CMM ladder. The reason for choosing the waterfall model was that it was the only development process we knew about.

After a few years with the waterfall model, we experienced aspects of the model that we didn't like:

- Risk mitigation was postponed until late stages.
- Document-based verification postponed until late stages.
- Attempts to stipulate unstable requirements too early: change of requirements is perceived as a bad thing in waterfall.
- Operational problems discovered too late in the process (Acceptance testing)
- Lengthy modification cycles, and much rework.
- Most important; the requirements were nearly purely focused on functionality, not on quality attributes.

Our experiences is backed up by statistics

- In a study of failure factors on 1027 IT projects in the UK, scope management related to waterfall practices was cited to be the largest problems in 82% of the projects. Only 13 % of the projects didn't fail. (Thomas, M.2001. "IT project Sink or Swim," British Computer Society Review)
- A large study showed that 45 % of requirements in early specifications were never used (Johnson, J. 2002. Keynote speech, XP 2002, Sardinia, Italy)

3 The shift of focus: from waterfall to evolutionary development

Peter Myklebust, FIRM CTO, and I heard Tom Gilb speak about evolutionary project management (Evo) at a software conference autumn 2003. We had just released a new version of our software that contained a lot of new nice features, but it had limitations with respect to usability, productivity and performance (e.g. throughput and response time). We found the ideas very interesting, and Tom and Kai Gilb offered to give a more detailed introduction to the concept. They spent one day in our offices, giving a very compressed introduction to Evo. We saw that Evo attacked many of the flaws in our waterfall process; most importantly the high focus on quality attributes that we felt could have been better in our latest release.

We decided to do an Evo pilot with a development phase of 3 months. We decided to do a literature study ourselves and then use Evo as best as we could for the next release (Confermit 8.5), without further Evo courses.

3.1 FIRM's interpretation of Evo: Basis for the 3 month trial period

Evo is in short: Quickly evolving towards stakeholder values & product qualities, while learning through early feedback. The beauty lies with the simplicity of the method, combined with advanced methods of measurement and control.

After the one day crash course with Tom and Kai Gilb and a literature study ("Competitive Engineering" by Tom Gilb and other material on the subject), our overall understanding of Evo was this:

- Find stakeholders (End users, super-users, support, sales, IT Operations etc)
- Define the stakeholders' real needs, and the related product qualities
- Identify past/status of product qualities and your required goal level (how much you want to improve).
- Identify possible solutions for meeting your goals
- Develop a step-by-step plan for delivering improvements via the identified solutions, with respect to Stakeholder Values & product quality goals:

And most importantly:

- Deliveries of measurable stakeholder-valued results every week (every Evo cycle)
- Measure weekly: are we measurably moving towards our goals?

3.2 Working with requirements the Evo way

With Evo, our requirements process changed. Previously we focused mostly on function requirements, and not on quality requirements. It is the quality requirements that really separate us from our competitors. E.g. spell checker in MS Word, why was this a killer application? There was no new functionality; authors of documents have been able to spell check with paper dictionaries for ages. The real difference was superior product qualities: **speed of spell checking and usability**.

We tried to define our requirements according to a basic standard [1]

- Clear & Unambiguous
- Testable
- Measurable
- No Solutions (Designs)

- Stakeholder Focus

Example taken from our requirements in Conformat 8.5:

Usability.Productivity

Scale: Time in minutes to set up a typical specified Market Research-report (MR)

Past: 65 min, **Tolerable:** 35 min, **Goal:** 25 min (end result was 20 min)

Meter: (how to measure if we are moving towards our goal): Candidates with knowledge of MR-specific reporting features performed a set of predefined steps to produce a standard MR Report. (The standard MR report was designed by Mark Phillips, an MR specialist at our London office)

The focus is here on the day-to-day operations of our MR users, not a list of features that they might or might not like. We *know* that increased efficiency, which leads to more profit, will please them.

After one week we had defined nearly all the top level quality and performance requirements for the next version of Conformat; and we were ready to start on our first Evo step. We decided that one Evo step should last one week; because of practical reasons, even though we violate the general Evo policy of not spending more than about 2 % of project schedule in each step. The rationale behind the 2% rule is not to spend more time than you can afford to lose. After one week, you'll find out whether you are on the right track (by getting feedback from stakeholders).

3.3 Find Solutions that takes you closer to your goals

For every quality requirement we looked for possible Solutions (Design Ideas)

E.g. for Quality Requirement: Usability.Productivity we identified the following Solutions: (identified by their name, not their description here)

- Solution.Recoding
- Solution.MRTotals
- Solution.Categorizations
- Solution.TripleS
- ..and many more

We evaluated all these, and specified in more detail those we believed would add the most value (take us closer to the goal level)

3.4 Working evolutionary, the FIRM Evo week

We organized the week in a special way.

On Friday we plan deliverables for version N, at the same time as we build and deploy version N-1 on the test server. Monday to Thursday is dedicated to design, code and test. During the week, the project collects feedback from stakeholders, based on the previous Evo step/week.

	Development Team	Users (PMT, Pros, Doc writer, other)	CTO (Sys Arch, Process Mgr)	QA (Configuration Manager & Test Manager)
Friday	<ul style="list-style-type: none"> ✓ PM: Send Version N detail plan to CTO + prior to Project Mgmt meeting ✓ PM: Attend Project Mgmt meeting: 12.00-15.00 ✓ Developers: Focus on general maintenance work, documentation. 		<ul style="list-style-type: none"> ✓ Approve/reject design & Step N ✓ Attend Project Mgmt meeting: 12-15 	<ul style="list-style-type: none"> ✓ Run final build and create setup for Version N-1. ✓ Install setup on test servers (external and internal) ✓ Perform initial crash test and then release Version N-1
Monday	<ul style="list-style-type: none"> ✓ Develop test code & code for Version N 	<ul style="list-style-type: none"> ✓ Use Version N-1 		<ul style="list-style-type: none"> ✓ Follow up CI ✓ Review test plans, tests
Tuesday	<ul style="list-style-type: none"> ✓ Develop Test Code & Code for Version N ✓ Meet with users to Discuss Action Taken Regarding Feedback From Version N-1 	<ul style="list-style-type: none"> ✓ Meet with developers to give Feedback and Discuss Action Taken from previous actions 	<ul style="list-style-type: none"> ✓ System Architect to review code and test code 	<ul style="list-style-type: none"> ✓ Follow up CI ✓ Review test plans, tests
Wednesday	<ul style="list-style-type: none"> ✓ Develop test code & code for Version N 			<ul style="list-style-type: none"> ✓ Review test plans, tests ✓ Follow up CI
Thursday	<ul style="list-style-type: none"> ✓ Complete Test Code & Code for Version N ✓ Complete GUI tests for Version N-2 			<ul style="list-style-type: none"> ✓ Review test plans, tests ✓ Follow up CI

Figure 1: FIRM Evo week

3.5 Evolutionary project planning

We collected the most promising Solutions and included them in an Evo plan (expressed by using an Impact Estimation Table: IET. See example below). The solutions were evaluated with respect to *value for clients* versus *cost of implementation*: choosing the ones with the highest value first. Note that value can sometimes be defined as *removing risks* by implementing technically challenging Solutions early.

The IET is our tool for controlling the qualities, and delivering improvements to real stakeholders: or as close as we can get to them. (E.g. support people, using the system daily, acting as clients)

Example: IET for MR Project – Conformat 8.5, Solution: Recoding

Description: Make it possible to recode variable on the fly from Reportal. Estimated effort: 4 days

	A	B	C	D	E	F	G	BX	BY	BZ	CA
1											
2		Current Status	Improvements		Goals			Step9			
3								Recoding			
4								Estimated impact		Actual impact	
5		Units	Units	%	Past	Tolerable	Goal	Units	%	Units	%
6					Usability.Replacability (feature count)						
7		1,00	1,0	50,0	2	1	0				
8					Usability.Speed.NewFeaturesImpact (%)						
9		5,00	5,0	100,0	0	15	5				
10		10,00	10,0	200,0	0	15	5				
11		0,00	0,0	0,0	0	30	10				
12					Usability.Intuitiveness (%)						
13		0,00	0,0	0,0	0	60	80				
14					Usability.Productivity (minutes)						
15		20,00	45,0	112,5	65	35	25	20,00	50,00	38,00	95,00
20					Development resources						
21			101,0	91,8	0		110	4,00	3,64	4,00	3,64

Figure 2: Example of Impact Estimation Table, MR project in Conformat 8.5 development.

4 Impacts on our product, experiences and conclusions

4.1 The method's Impact on Conformat product qualities

The method's impact on Conformat product qualities are *not* measured statistically, by doing a scientific correct large-scale survey, although we are currently considering this. The impact described in this paper is based on internal usability test, productivity tests, performance tests carried out at Microsoft Windows ISV laboratory in Redmond USA, and direct customer feedback. Only highlights of the impacts are listed here. No negative impacts are hidden.

Description of requirement/work task	Past	Status
Usability.Productivity: Time for the system to generate a survey	7200 sec	15 sec
Usability.Productivity: Time to set up a typical specified Market Research-report (MR)	65 min	20 min
Usability.Productivity: Time to grant a set of End-users access to a Report set and distribute report login info.	80 min	5 min
Usability.Intuitiveness: The time in minutes it takes a medium experienced programmer to define a complete and correct data transfer definition with Conformat Web Services without any user documentation or any other aid	15 min	5 min
Performance.Runtime.Concurrency: Maximum number of simultaneous respondents executing a survey with a click rate of 20 sec and an response time<500 ms, given a defined [Survey-Complexity] and a defined [Server Configuration, Typical]	250 users	6000

Table 1: Improvements to product qualities

These leaps in product qualities would not have been achieved without Evo. We have received many pleasant emails regarding these quality improvements from our customers:

"I just wanted to let you know how appreciative we are of the new "entire report" export functionality you recently incorporated into the Reportal. It produces a fantastic looking report, and the table of contents is a wonderful feature. It is also a HUGE time saver."

4.2 Feedback from developers and project managers within FIRM R&D

Evo has resulted in increased motivation and enthusiasm amongst developers because it opens up for empowered creativity. EVO and Continuous Integration is a vehicle for innovation and inspiration. The developers get their work out on test servers, and receive feedback. Every week.

Even though they embraced the method there are parts of Evo they found difficult to understand and execute at first:

- Defining good requirements can be hard.
- It was hard to find meters (ways of measuring numeric qualities) which were practical to use, and at the same time measure real product qualities.
- Sometimes it takes more than a week to deliver something of value to the client.
- Testing was sometimes postponed in order to start next step, some of these test deferments were not then in fact done in later testing.

4.3 Lessons learned with respect to the method

Some of the lessons we learned after the trial period are:

- We will have increased focus on feedback from clients. We will select the ones that are willing to dedicate time to us. Internal stakeholders can give valuable feedback, but some customer interaction is necessary.
- Demonstrate new functionality with screen recording software or early test plans. This makes it easier for internal and external stakeholders to do early testing
- Tighter integration between Evo and the test process is necessary
- "Be humble in your promises, but overwhelming in your delivery

4.4 Conclusions

The method's positive impact on Conformat product qualities has convinced us that Evo is a better suited development process than our former waterfall process, and we will continue to use Evo in the future.

What surprised us the most was the method's power of focusing on delivering value for clients versus cost of implementation. Evo enables you to re-prioritize the next development-steps based on the weekly feedback, what seemed important at the start of the project may be replaced by other solutions based on gained knowledge from previous steps.

The method has high focus on measurable product qualities, and defining these clearly and testable requires training and maturity. It is important to *believe* that everything can be measured and to seek guidance if it seems impossible.

One pre-requisite related to the method for using Evo is an *open architecture*.

Another pre-requisite is *management support* for changing the work process, and this is important in any software process improvement initiative.

The concept of Continuous Integration (CI)/daily builds was valuable with respect to deliver new version of the software every week.

Overall, the whole organization has embraced Evo. The release of Conformat 8.5 showed some of Evo's great potential, and we will work hard to utilize it to the full in the future. In June 2004 we had Tom and Kai Gilb for a 4 days course for the whole R&D department and related resources and we hope the next version of Conformat will prove that we have matured in our understanding and execution of Evo.

4.4.1 Literature

- [1] Tom Gilb, "Competitive Engineering: A Handbook for Systems and Software Engineering Management Using Planguage", Elsevier, to be published End 2004-Beginning 2005. Manuscript, until publication, at www.gilb.com (free).
- [2] Kai Gilb, "Evolutionary Project Management & Product Development, Evo, how successful people manage projects". Unfinished book manuscript for project planning to be found on www.gilb.com)
- [3] Craig Larman, "Agile and Iterative Development, a Manager's Guide"

5 Author CVs

Trond Johansen

M.Sc. at the Norwegian Institute of Technology, University of Trondheim (NTNU).
1998-1999: Hydro Data, system consultant SAP
1999-2000: Det Norske Veritas Software, Product Manager Nauticus CMC
2000- Future Information Research Management, FIRM. QA & Process Manager
So far my work has been related to process management and project management within software development as well as management in general.

AXA IT Organisation & Process Maturity Assessment Framework

Tim Hind, AXA UK IT CIO Office

Abstract

Introduction

AXA Group is a major international Financial Services company, with over 7,000 IT staff worldwide. In 1999 it started a global IT process improvement programme. It recognised the need for maturity measurement but needed to be able to implement something worldwide at a reasonable cost within different size companies.

CMM was seen to have too narrow a focus and would have been expensive to implement – especially in the smaller companies. AXA decided to build a framework that was wider than CMM although with less depth.

It started with 7 models (4 SDLC models for Build, Buy, Project Management, Programme Management; 1 Technology Planning model (Architecture Management); 1 Service Delivery model and an Overall Improvement & Measurement model).

AXA decided on some basic principles. The models should

- Have a CMM-ish look and feel.
- Be linked to an industry standard where possible
- Be home-made where no industry standard established.
- Cover the full scope of IT Activity to ensure the possibility of maximum efficiency and effectiveness.
- Be used to facilitate identification of Best Practice to optimise leveraging within AXA Group
- Be used to facilitate Process Improvement within individual companies.

Over the years there has been much change – many models have been refreshed, some have been suspended and new ones emerged.

The Development and Maintenance of the models is the responsibility of the Group IT Organisation and Process (ITOP) Team. This team supports the operating companies collaboratively in the implementation of Process Improvement.

Models Overview

Recognising that there is a Frameworks Quagmire (© 2001, Software Productivity Consortium) it was quite an interesting job to create an AXA Assessment Methodology which would be consistent. AXA decided to use the CMM Pattern of levels; models were developed by picking from the various standards that were available at the time. The industry standards were adopted and adapted.

Our current process scope covers 12 process areas in 4 groups :-

1. Central : (Governance, Sourcing, Architecture Management, Reuse Behaviour, Overall Improvement & Measurement),
2. Relationship : (Business Relationship Management, Service Management),
3. Management : (Project & Programme Management),
4. Delivery : (SDLC Build & Buy, Maintenance).

Of these 12 models there are 9 that are currently operational. These cover all the Central Processes (with the exception of Overall Improvement & Measurement model), all Management Processes and all Delivery Processes.

The Relationship processes are under construction with a pilot for Service Management in the middle of 2004. The Overall Process Improvement & Measurement model has been withdrawn pending review.

Recognising that not all companies are sufficiently mature to warrant being assessed in every model, a set of six (6) Core models have been identified. There is a very strong recommendation to use them.

The Process

Medium to Small Companies within the Group Self-Assess annually. Larger companies are Co-Assessed, with Group IS participation on site, biennially and use either the Self-Assessment method in the middle year or a Light Co-Assessment if preferred.

The questionnaires for each model are available during the autumn as an on-line tool via the Group intranet. Company assessors, appointed by Senior Management, plan the completion of the surveys. They carry out interviews with Subject Area Experts (SAEs). The ITOP Team organise and deliver training to Company assessors, supporting documentation and supporting tools.

The results are recorded in on-line questionnaires and the maturity level calculated for the Key Process Areas, Models and Overall IT Maturity Level score.

A feedback report is produced by the Company Assessors for the SAEs. This forms the basis of a Company report, giving a SWOT analysis, key observations and recommendations. It is created by the Company Assessors together with the company Process Improvement Manager. The report is presented to the company IS Board for approval. If required a CMM Equivalence can be given through a recently developed tool.

The ITOP team gather and select Best Practices to populate a Knowledge Base to promote reuse around the Group to allow Operating Companies to leverage the strength of the Group and to accelerate process improvement.

In the spring a Group report is produced for Group Management of all co-assessed companies and the overall results are published annually for all companies at an Exchange Seminar which takes place in the summer.

Companies, supported by the ITOP team, analyse in more detail the assessment results to identify the Company improvements. Companies subsequently implement Improvement Programmes.

The models are reviewed and refreshed, as necessary, ready for the autumn and the next cycle.

Keywords

SPI; Process Assessment; CMM; ITIL; CoBIT; IEEE; Questionnaire;

1 AXA IT Organisation & Process Maturity Assessment Framework

1.1 Context

In a large international group of companies, especially one which had grown significantly in a comparatively short period of time through merger and acquisition, there is the opportunity for many and diverse practices to emerge, as each subsidiary creates its own processes and procedures. It was clear that AXA Group was no exception to this and it was realised that it was important to understand what was being used within IT around the globe.

AXA Group started an initiative to improve its IT capability in 1999. Linked to a scorecard which measured a number of aspects of IT performance, this included the initial attempt to grapple with a capability assessment survey. The original “Kennedy Statement”¹ was for AXA to become recognised as an Innovative Company within 4 years.

The Group consists of a number of companies in all four corners of the world. They range in size and complexity. The IT provision ranges from in-house “traditional” organisations to those which have outsourced much of their provision. The type of outsourcing itself varies from the use of external software houses to a full transfer of undertakings for all or part of the development portfolio. Individual sites range from a dozen to many thousands of staff.

It was natural with such diversity that a “one-size fits all” approach would be ineffective. There was also a need to look at not just IT Development capability. Out in the market place, ISO 9000 and SEI SW-CMM were well established and SPICE was emerging. These however covered only a part of the disciplines that were to be assessed. The search was on to find or develop a framework whereby all of our organisation and process could be evaluated.

1.2 Creation of the Framework

There was a conscious decision to lay down some basic principles for the framework.

- The models would have similarities to the SEI SW-CMM 5 level approach.
- Wherever possible an industry standard would be adopted and adapted for use within the Group.
- Where it was not possible to use an industry standard – for example where no such standard existed – the Group would develop its own.

It was important that the models would enable the identification of best practice within the Group to act as part of its emerging Knowledge Management philosophy and that the output from the surveys would facilitate process improvement within each company. Knowing where you are, is one thing, knowing how to move from where you are, is quite another.

It was hoped to include, within the framework, models for every aspect of IT activity. This meant not only the factory floor where project teams build software but also the disciplines associated with the Business Relationship, Vendor Management, Outsourcing, Governance etc.

As a starting point, 5 areas of IT were chosen for consideration.

- Project & Programme Management
- Software Delivery Lifecycles for Build & Buy
- Technology Planning (Architecture Management)
- Service Delivery
- Overall Improvement & Measurement

The Group understood that there was a Frameworks Quagmire² (Figure 1) to be navigated. It was considered that the pragmatic way was to develop its own framework (AXA IT Maturity Level Survey – known as the AXA ITML) with a view to making it obsolete within a period of 4 years

The Group's ambition was to move to SEI SW-CMM once the maturity recorded by the Group companies had reached a certain level within the AXA ITML (Figure 2).

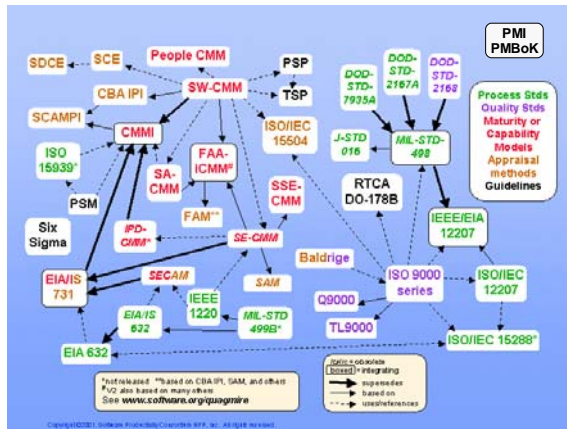


Figure 1: Framework Quagmire

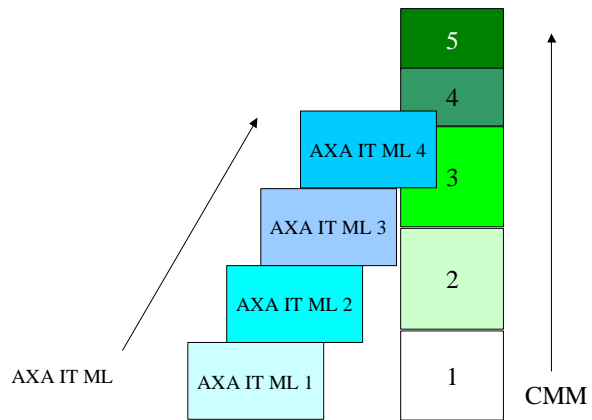


Figure 2: AXA IT ML - CMM Equivalence

The five areas referred to above resulted in the creation of 7 models due to the need for one each for Project & Programme Management and one each for Build & Buy delivery lifecycles.

Under the heading of Software Development Improvement (SDI) the models were rolled out in 2000 and used as self-assessment tools during early autumn of that year. A review of the results took the form of a series of 3 seminars later that year, each lasting a day and a half, to tease out what the learning points should be for the Group. The SDI Steering Group was formed in the spring of the following year (2001) and this was to form the basis for the IT Organisation & Process (ITOP) function which exists today.

One of the first things that the new SDI Steering Group did was to begin the process of regular Exchange Seminars (similar to Software Process Improvement Network (SPIN) meetings). The first took place in Nice in 2002 and has successfully been followed up with Cologne (2003) and Paris (2004). These give the opportunity for AXA IT staff from all over the world to meet together over a two day period to hear the results of the surveys, hear key note speeches from external speakers and to share amongst each other.

The original 7 models ran with only a few alterations for 3 years. The Service Delivery model was withdrawn after its first outing. In 2002 a small additional model was added to cover Reuse Behaviour. In 2003 a major overhaul of the survey took place.

A formal map of the extent of the 12-model target framework had emerged (Figure 3).

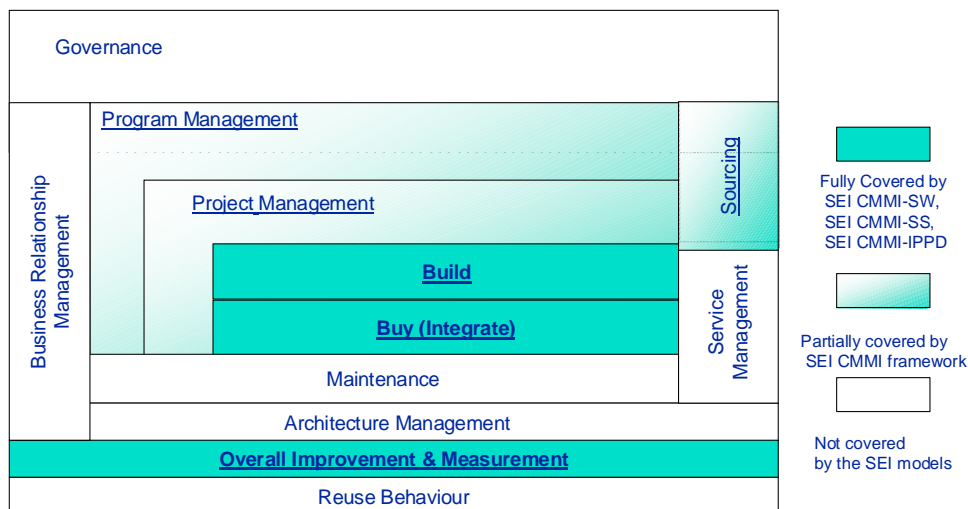


Figure 3: ITOP Framework

This enabled the organisation to see fully the cover that SEI SW-CMM provided and they recognised where the gaps were in terms of the rest of the models. Some consistency check across models showed that the Architecture Management Model and the Overall Improvement & Measurement models needed some attention. The latter was withdrawn for review and the former extended to balance it better against the other models.

Three new models were trialled in the year (Governance, Sourcing and Maintenance) and incorporated in the surveys in time for delivery in the autumn of 2003. The Service Management model was trialled in the summer of 2004 and it is intended to develop a Business Relationship Management model during the year for trial in 2005. A discussion document was issued in July 2004 regarding the revision of the Overall Improvement & Measurement model.

Currently there are, therefore, 9 active models, one in the pipeline, one in development and one pending review.

1.3 The Current Models

Each of the current models conforms to a common pattern. The on-line survey instrument consists of a series of Key Process Areas (KPAs). Each KPA has a number of questions with the tick-box answers. Each question is created with a specific maturity level in mind and will look at whether or not the practice is in place. Where it is, the question will also seek to establish how extensively the practice is used. From this a maturity level for each model can be established.

As an example, Project Management has a KPA which relates to Risk Management. Question 3.4.1 (Figure 4) is designated as a level 3 question if established for all projects. Less coverage would result in a level 2 score. If the KPA practice is not implemented it would result in a level 1 score. A classification scheme is in place to decide what Large, Medium and Small mean in respect of projects.

3.	Project Management	In Place		Scope of Use			
		Yes	No	80% of Small Projects	80% of Medium Projects	80% of Large Projects	80% of All Project Types
3.4.3	Has your organization established a standard for risk plans and management?	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Figure 4: Sample Question

The table (Figure 5) gives a flavour of all the Project Management KPAs and shows the maturity that can be achieved. Some practices, e.g. Scope Management, will achieve level 2 by being in place and level 4 through full deployment. Others, e.g. Human Resources Management, will achieve level 3 by virtue of being in place. Project Management Improvement is a level 4 activity.

ITOP Model	Key Practice Area (KPA)	2	3	4
Project Management	3.1 Classification Scheme			
	3.2 Stakeholders			
	3.3 Scope Management			
	3.4 Risk Management			
	3.5 Time Management			
	3.6 Cost Management			
	3.7 Quality Management			
	3.8 Human Resources Management			
	3.9 Communications Management			
	3.10 Plan Integration Management			
	3.11 Project Management Improvement			

Figure 5: Project Management KPAs and Maturity Levels

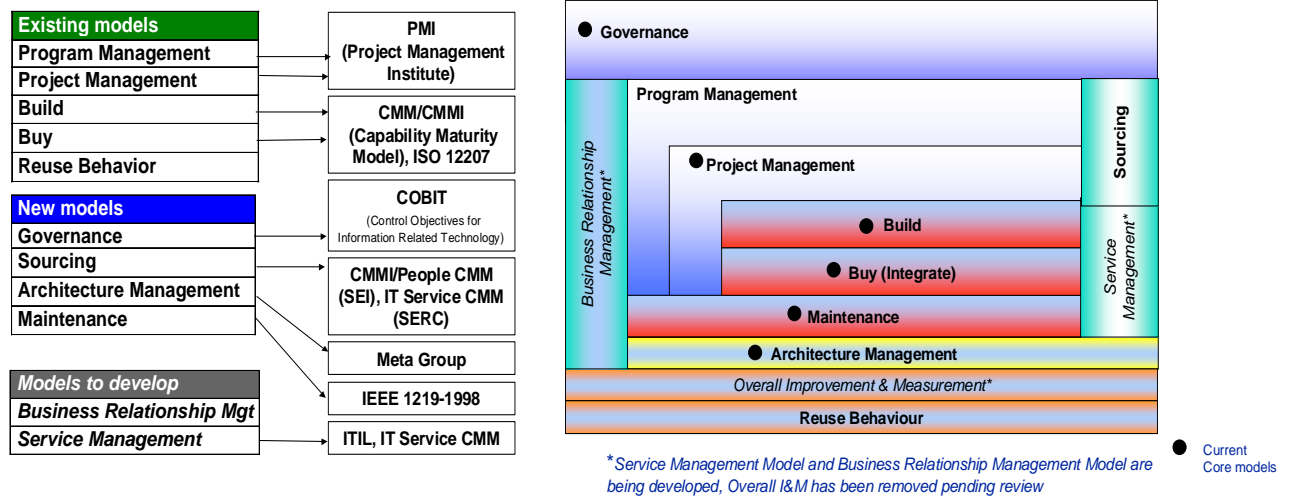
For the central, i.e. not project-based, models, such as Governance & Architecture Management, the

questions are answered according to the practice being in place for a percentage of cases. The answers can be No, Some or Yes.

Ensuring that the models had this commonality was not easy given the variety of standards that were used to develop them. It can be seen from the table (Figure 6) that there were sometimes a number of standards informing a single model.

The current framework has to be useful in all of the AXA Group subsidiaries. Each model is tailorable by virtue of the classification scheme where Large, Medium & Small mean different things for different sized companies. However, not every model is applicable to be assessed in every company and a decision needs to be made as to which models are employed.

A number of the models are deemed core and are shown below (Figure 7). It is strongly recommended that they are assessed. In fact a new word (voludatory) has been coined to describe this –



voluntary but (really) mandatory!

Figure 6 : Reference Table for Standards used. Figure 7: ITOP Framework

1.4 The Annual Cycle

The surveys take place annually in each AXA subsidiary. Medium-to-small companies self-assess their processes whereas large companies are co-assessed every other year. In the interim year, large companies will self-assess or use a light co-assessment.

The co-assessment process is naturally much more rigorous, than self-assessment, with Group IS participation on site. This process enables a balanced view to be created between companies ensuring that Group results can be compared on a like-for-like basis.

Recognising that some medium-to-small companies may wish to be part of the co-assessment process, a “Light Co-Assessment” process has also been established, where Group verify the answers. The ability of Group to respond to requests for this process may be limited by availability within the relatively small but efficient ITOP Team.

At Group level the process starts in the autumn of each year with a training session for all assessors. This is principally for the Company assessors but new group assessors are also involved. The ITOP Team prepare and deliver this training to representatives of each subsidiary company. Depending on the size of company these representatives may follow up this initial training with cascade training of the assessors within their own companies.

The ITOP Team works with the companies that are to be co-assessed to build a schedule for the assessment. They come on-site to work with the company assessors. They conduct the assessment in partnership and produce the Company Co-assessment Report with involvement from the Company Assessors.

Once all of the co-assessments have been completed, in early spring, the ITOP Team prepare a

Group Co-Assessment Report which identifies trends, reuse opportunities and any significant areas where it might be appropriate for Group to be involved in developing something centrally to assist all companies. This feeds into Group Strategic Plans.

Workshops can be proposed when several companies would benefit from one company's best practice.

Any good practices or artefacts that are discovered during the assessment are gathered together and published on a Knowledge Management website, which is accessible to all of AXA employees.

In the summer of each year, the Exchange Seminar takes place where assessors and practitioners meet with senior Group Executives to receive the Group Co-assessment Report and to share experiences.

1.5 The Assessment Process

The Co-Assessment process is the most comprehensive of the three assessment processes, during which the key people are the Sponsor, Group IS Assessors, Company Assessors and Subject Area Experts.

Visibility is key to ensuring that the assessments work well. The co-assessment process has 7 steps and Preparation & Launch are two of the key steps to get right.

Legend

öö Lead Person

ö Support Person

- Preparation
- Launch
- Identify Resources
- Schedule Assessment Interviews
- Research
- Conduct Assessment Interviews
- Issue Co-Assessment Report

	Sponsor	Group IS Assessors	Company Assessors	Subject Area Experts
Preparation	öö	ö		
Launch	öö	ö	ö	
Identify Resources	ö	ö	öö	
Schedule Assessment Interviews		ö	öö	ö
Research				öö
Conduct Assessment Interviews		öö	öö	ö
Issue Co-Assessment Report	ö	öö	ö	

The Self-Assessment process is similar to the above with the two assessor columns combined. The Light process is a compromise between the two.

1.6 The Process Improvement Process

Once the report has been issued the Process Improvement Process can start. This has 5 steps. During this process the key people are the Sponsor, Improvement Programme Manager, Assessors (Company or Group) and Process Owners. It is vital to have someone who has overall responsibility for improvement. If this is not possible, a co-ordinator, who manages the activity of a number of process owners, can work just as well but needs strong executive sponsorship.

Legend

öö Lead Person

ö Support Person

- Set Objectives and Target Maturity
- Analysis & Research
- Identify Potential Improvements
- Prioritise Improvements
- Plan Improvement Activities

	Sponsor	Improvement Programme Manager	Assessors	Process Owners
Set Objectives and Target Maturity	öö	ö	ö	
Analysis & Research		ö	öö	ö
Identify Potential Improvements		ö	öö	ö
Prioritise Improvements	ö	öö		ö
Plan Improvement Activities	ö	öö		ö

One of the clear shifts of emphasis that occurs here is the transfer of activity away from the Assessors to the Improvement Programme Manager. The models, however transparent and user-friendly, still need some interpretation to ensure that the messages from the assessment can be translated into the language used by practitioners. The Improvement Programme Manager needs to be able to be part of the identification of improvements but then to be allowed to get on with it.

Once the Action Plan has been developed this produces the framework for improvement activity over a period. Sometimes this will have a short-term focus; on other occasions it may lead to a programme of activity over one or more years. It feeds into the Company IT Strategic and Action Plans.

Understanding the landmarks (milestones) that are to be achieved, enables targets to be set for subsequent assessments.

The results of the assessments are included in company scorecards and dashboards.

1.7 Conclusion

The ITOP Maturity Assessment Framework is always developing. Through its operation it has demonstrated sufficient stability to enable useful trends to become visible over a number of years. It has already added an enormous amount of "Best Practice" material to the Knowledge Management database and more importantly through the annual seminar has meant that the growth of knowledge around the group has been accelerating.

In the Group a wide variation of maturity is in evidence and the 2004 results for 5 of the Core Models are shown in the graph (Figure 8). Over the years significant improvement in maturity has been witnessed and this is demonstrated in the graph relating to Project Management (Figure 9).

The strength of the framework is in the comprehensive nature and intellectual integrity of its design. The work done to define, redefine and refine the content over 5 years has created a robust tool for process improvement for use across the globe in companies of all sizes.

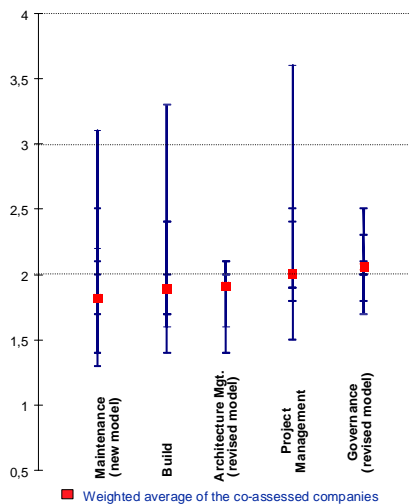


Figure 8 : Selected Results 2004.

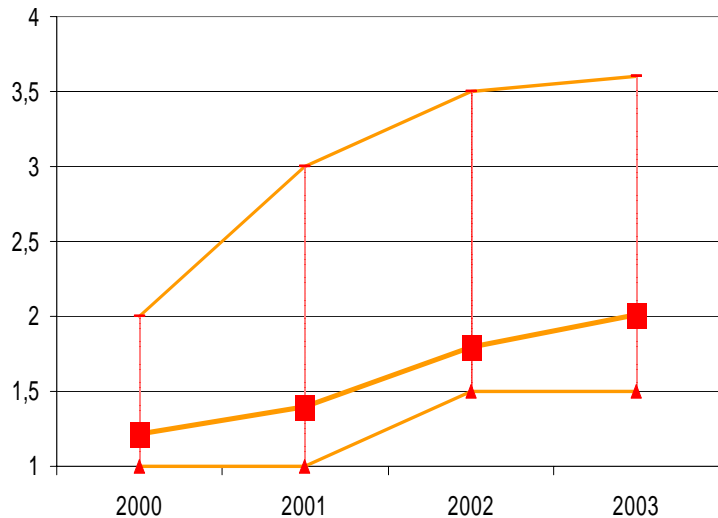


Figure 9 : Project Management Progress 2000-3

2 Literature

1. President John F Kennedy to Congress 25th May 1961: "I believe that this nation should commit itself, before this decade is out, to the goal of landing a man on the Moon and returning him safely to Earth."
2. The Frameworks Quagmire, Sarah A Sheard, Software Productivity Consortium 2001, <http://www.stsc.hill.af.mil/CrossTalk/frames.asp?uri=1997/09/frameworks.asp>
3. AXA IT ML Training-AXA IT ML Overview, July 2000

3 Author CV

Tim Hind

AXA UK IT, Bristol, UK



Born 1950 and educated at Watford Boys Grammar School and St John's College Cambridge where he studied Mathematics – in particular Fluid Dynamics.

Worked briefly for Royal London Mutual in the Actuarial Department but on leaving University joined Sun Life in London as a Pensions Administrator. In the mid-70s he moved into Data Processing as a Systems Analyst and became a Project Leader in 1980.

His formative years in IS were spent as an analyst on Pensions systems but he has dabbled in almost every other system within the company.

He moved into project statistics in an attempt to assist the cost accountants to split IS costs to business units. After different spells co-ordinating IS Security and working as a Budget Analyst, he joined the IS Metrics Team for AXA Sun Life at its inception in 1996. He has delivered industry experience papers at a number of conferences in UK and in Europe over the last 5 years and is a member of the EuroSPI Industrial Programme Committee.

Currently part of the IT Governance Team within the newly formed AXA UK IT, he is responsible, amongst other things, for Process Strategy, Assessment & Improvement Co-ordination for IT in AXA across UK.

Spare time is taken up with family and church activities as a member of the General Synod of the Church of England.

He is grateful to the AXA Group IS ITOP Team for their support in the preparation of this paper and in particular to Kae Palmer for coordinating their support.

Improving the object-oriented design process

Javier Garzás¹ and Mario Piattini²

¹ mCentric

Project Manager

C.\ Gabriel García Márquez 4, 28230, Las Rozas, Madrid, Spain

Javier.Garzas@m-centric.com / javier31175@terra.es

² Alarcos Research Group

Escuela Superior de Informática - University of Castilla-La Mancha

Ronda de Calatrava, s/n. 13071, Ciudad Real – SPAIN

Mario.Piattini@uclm.es

Abstract. It has been a long time since appeared of the Object Oriented (OO) paradigm. From that moment, the designers have accumulated much knowledge in design and construction of OO systems, but this large body of knowledge is still not organized or unified. In order to improve OO designs (OOD), using all the accumulated knowledge in a more systematic and effective way, we have defined an ontology. We present a controlled experiment about if the usage of the Ontology for OOD Knowledge really improves the OOD process. **Key Words:** Design Process, Object-Oriented Design, Micro-Architectural Design, Design Knowledge, Ontology, Design Patterns, Principles, Heuristics, Best Practices, Bad Smells, Refactoring.

1. Introduction

The software engineering field still suffers from a lack of structured and classified chunks of knowledge (McConnell, 2003). Object-Oriented (OO) designers have accumulated a large body of knowledge that it is still not organized or unified. In OO Micro-Architectural Design Knowledge, design patterns are the most popular example of accumulated knowledge, but other elements of knowledge exist such as Principles, Heuristics, Best Practices, Bad Smells, Refactorings, etc., which are not clearly differentiated, indeed, many are synonymous and others are just vague concepts.

For principles the main contributions are Meyer (1988), Gamma *et al.* (1995), etc., but the study of design principles is limited, leading to them being used in an isolated way or even being ignored; any examples of principles are the Dependency Inversion Principle, Don't Concrete Super class Principle, etc. With regard to heuristics, we can refer to Riel (1996) and Booch (1996). Refactorings are rapidly gaining acceptance, the main works in this area are Fowler's (2000), Tokuda and Batory (2001) and Opdyke (1992). Patterns are, without doubt, the most refined OOD Knowledge: Coad (1992), Gamma *et al.* (1994), Buschmann *et al.* (1996), etc.

In order to improve OO designs, using all the OOD accumulated knowledge in a more systematic and effective way, we have defined an ontology (figure 1), which unifies Principles, Heuristics, Best Practices, etc., as "Rules", show the relationship among these "Rules" and Patterns and Refactorings, etc., and problems (Patterns) in designs.

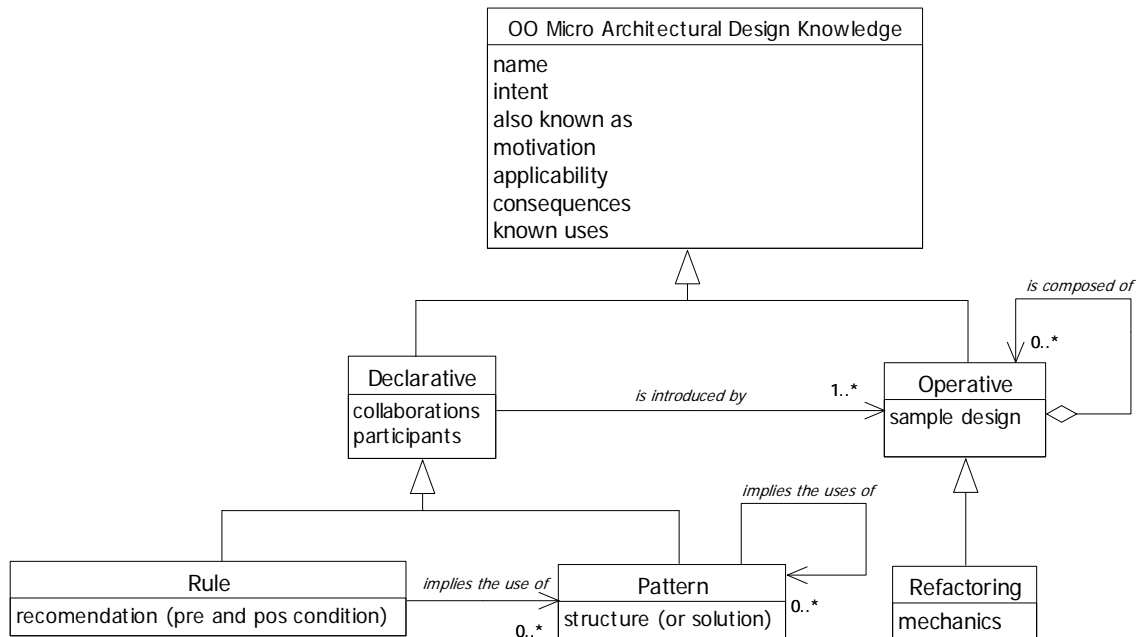


Fig. 1. OO Micro-Architectural Design Knowledge Ontology

We may observe as the entities Rule and Pattern can be introduced in the design for Refactorings and Rules implies the use of Patterns. We have also defined an improved OOD process which takes into account this ontology and the OOD knowledge, helping OO designers in their activities. We present in this paper an controlled experiment (approach based on Wholin *et al.* (2000) and Prechelt *et al.* (2000) and Bernárdez (2004)), we wanted to ascertaining if the usage of the Ontology for OOD Knowledge really improves the OOD process, helping in the detection of defects (Rules violated)

2. Experiment Description

The main purpose of this controlled experiment is to compare effectiveness and efficiency to dispose of “Ontology of OOD knowledge”. Moreover, we wanted to analyze if to dispose of a catalog of rules and its relations with other patterns (according to the OO Micro Architectural Knowledge) has influence on the effectiveness and efficiency in the improve the quality of OO micro architectural design.

2.1 Planning

We classified the subject in two groups and in a previous session of 30 minutes, we explained to one group some notions about Ontology, Rules, and its relationships with Patterns; and how to apply the Rules Catalog. The subjects were asked to fill out a questionnaire about your expertise. According to the responses collected in this questionnaire, we form the two groups of subjects, trying to have the same number of subjects with good marks and bad marks in each group.

We have to consider what independent variables or factors were likely to have an impact on the results. These are: OO Micro Architecture. We considered two dependent variables:

- **Effectiveness:** Number of Defects Found / Total Number of Defects. This is the percentage of the true improvements found by a reviewer with respect to the total number of defects in the inspected.
- **Efficiency:** Number of Defects found / Inspection Time. Where Inspection Time is related to the time that subjects spent on inspecting the micro architecture, it is measured in minutes.

For each subject, we had prepared a folder with the experimental material. Each folder contained one micro architectural diagram and a questionnaire for answers; additionally, a group of subjects had a summary of the rules presented in section 2 and a summary of Gamma's *et al.* (1995) design patterns, the other group had only the catalog of patterns.

2.1.1 Hypotheses Formulation

We want to test two groups of hypotheses, one for each dependent variable.

- Effectiveness Hypotheses

- $H_{0,1}$. There is no difference in effectiveness in detecting the violation of rules using a catalog of rules and its relationship with patterns as compared to subject without catalog or rules. // $H_{1,1} : \neg H_{0,1}$
- $H_{0,2}$. There is no difference in effectiveness of subjects in detecting the application of patterns implicated by rules using a catalog of rules and its relationship with patterns as compared to subject without catalog or rules. // $H_{1,2} : \neg H_{0,2}$
- $H_{0,3}$. There is no difference in effectiveness of subjects in detecting the application of patterns not implicated by rules using a catalog of rules and its relationship with patterns as compared to subject without catalog or rules. // $H_{1,3} : \neg H_{0,3}$

- Efficiency Hypotheses

- $H_{0,4}$. There is no difference in efficiency in detecting the violation of rules using a catalog of rules and its relationship with patterns as compared to subject without catalog or rules. // $H_{1,4} : \neg H_{0,4}$
- $H_{0,5}$. There is no difference in efficiency of subjects in detecting the application of patterns implicated by rules using a catalog of rules and its relationship with patterns as compared to subject without catalog or rules. // $H_{1,5} : \neg H_{0,5}$
- $H_{0,6}$. There is no difference in efficiency of subjects in detecting the application of patterns not implicated by rules using a catalog of rules and its relationship with patterns as compared to subject without catalog or rules. // $H_{1,6} : \neg H_{0,6}$

2.2 Operation

This section we describe the preparation, execution, and data validation. Before the day of the experiment execution, we gave a seminar to the subject of group that had a summary of the rules. In this seminar, we explain to the subject how to apply the rules catalog. The subjects work without speaking mutually. The subjects had to perform the following experimental tasks: Manually fulfill the form registering and Writing down the start and end time of the activity. We collected the forms filled by the subjects, checking if they were complete.

2.3 Analysis and Interpretation

The purpose is to analyze how the independent variables have influence in dependent variables. The general purpose is to determine whether using the Ontology of OOD Knowledge is more effective than not use that. Table 1 shows the averages obtained from the experiment. Outliers have not been identified. In order to decide how to testing the validity of the hypotheses we evaluated if the data follow a normal distribution, the result was normal; and we decide to use a t-student test.

	Group with patterns catalog and with catalog of rules and its relation with patterns	Group without patterns catalog and catalog of rules and its relation with patterns
Efficacy in Detection of Rules.	0,64	0,12
Efficacy in Detection of Patterns not implicated by rules.	0,64	0,71
Efficacy in Detection of Patterns implicated by rules.	0,68	0,18

Efficiency in Detection of Rules.	0,26	0,04
Efficiency in Detection of Patterns not implicated by rules.	0,08	0,07
Efficiency in Detection of Patterns implicated by rules.	0,18	0,04

Table 1. Averages obtained form the experiment

The results confirmed by means of t-Student test that there was a significant difference between the two groups, and that the new approach seems to be more effective and efficient for carry out the OOD Process. The statistical assumptions of each test were verified, so that, the conclusion validity was fulfilled.

3. Conclusions

We are conscious that in this experiment we have chosen to experiment with an individual technique that could interact with many other development techniques and procedures (“the life-cycle issue”, Kitchenham et al., 2002), such as the life-cycle model (light or weight), the modeling language used to express the design artifacts, etc. So, large-scale empirical studies are needed to get conclusions about the effect of the knowledge systematization in the OOD improvement.

4. References

1. Bernárdez B., Genero M., Durán A., Toro M. (2004). A Controlled Experiment for Evaluating a Metric-Based Reading Technique for Requirements Inspection. *Software Metrics*, 10th International Symposium on (METRICS'04)
2. Booch G. (1996). *Managing the Object-Oriented project*. Addison-Wesley
3. Buschmann F., Meunier R., Rohnert H., Sommerlad P. and Stal M. (1996). *A System of Patterns: Pattern-Oriented Software Architecture*, Addison-Wesley.
4. Coad P. (1992). *Object-Oriented Patterns*. *Communications ACM*, vol. 35, n 9, pp. 152-159.
5. Fowler M. (2000). *Refactoring improving the design of existing code*. Addison Wesley
6. Gamma E., Helm R., Johnson R. and Vlissides J. (1995). *Design patterns: Elements of Reusable Object Oriented Software*. Addison-Wesley.
7. Kitchenham B.A., Pfleeger S., Pickard L., Jones P., Hoaglin D., Emam K., Rosenberg J. (2002). Preliminary Guidelines for Empirical Research in Software Engineering. *IEEE Transactions on Software Engineering* archive, Volume 28, Issue 8
8. Meyer B. (1988). *Object Oriented Software Construction*. Prentice Hall.
9. McConnell S. (2003). *Professional Software Development*. Addison-Wesley.
10. Opdyke W. (1992). *Refactoring OO frameworks*. PhD Thesis, Department of Computer Science. *University of Illinois*.
11. Prechelt L., Unger B., Tichy W. y Bossler P. (2000). A controlled Experiments in Maintenance Comparing Design Patterns to Simpler Solutions. *IEEE Transactions on Software Engineering*.
12. Riel A. J. (1996). *Object-Oriented Design Heuristics*. Addison-Wesley.
13. Tokuda L. and Batory D. (2001). Evolving Object-Oriented Designs with Refactorings. *Kluwer Academic Publishers - Automated Software Engineering*, vol8, N°1, pp 89-120
14. Wohlin C., Runeson P., Höst M., Ohlsson M., Regnell B., and Wesslén A. (2000). *Experimentation in Software Engineering – An Introduction* published. Kluwer Academic Publishers.

Adopting RUP to defined project types: a preliminary experience report

Gunnar Nordseth, Kantega AS, gunnar.nordseth@kantega.no
Geir K. Hanssen, SINTEF ICT, ghanssen@sintef.no

1. Introduction

The Unified Process [3] and the commercialized variant Rational Unified Process (RUP[1]) is a comprehensive framework for software development projects. RUP defines a software development project as a set of disciplines running from start to end through a set of project phases (one or more iterations). A project is performed by a group of actors, each having one or more roles. Each role participates in activities producing artifacts (results such as documentation, models, code etc.).

However, RUP is as stated a framework, meaning that it is a more or less complete set of process elements that has to be tailored to each case (project), no project needs the complete set.

Jacobson, Booch and Rumbaugh says in [3] p.416:

"It [RUP] is a framework. It has to be tailored to a number of variables: the size of the system in work, the domain in which that system is to function, the complexity of the system and the experience, skill or process level of the project organization and its people." Further on they say: "Actually, to apply it you need considerable further information."

So, it is clear that RUP needs to be tailored, downscaled and specialized to the context of use, based on knowledge. There are at least three overall strategies for doing so:

1. Specialize the framework for each case, that is, for each project.
2. Specialize the framework to organizational specific needs.
3. Specialize the framework to defined types of projects in the organization.

The first strategy require little or no specializing in advance, it implies a comprehensive job initially in each project. This can be justified for large projects where the methodology specializing constitutes only a small part of the total work and where the type of projects vary a lot. The second is the most commonly used strategy. However, this strategy only gives you a framework tailored to the organization in general. This is suitable if most projects are quite similar, otherwise there is still a need of comprehensive specialization to specific projects. The third strategy has the highest up-front cost that starts with identifying and describing common project types in the organization. However, this reduces the need for the last step adjustment to project specific context. It also ensures that focus is kept strongly on project specific topics.

This paper focuses on this third strategy and reports the practical work that so far has been done in a Norwegian software company.

2. Case context

The case company is a software consultancy house with 45 employees. We have a high competence in RUP and some of the employees are RUP mentors. An initial interview with five employees with varying project background indicated that the use of RUP was variable and that some felt that there was a high threshold for adapting RUP to the project and taking it into use. As a consequence, Kantega decided to identify and define their typical project types and develop one RUP specialization for each type.

3. Defining project types

As a basis for specializing the RUP framework into a few variants for typical project types we first needed to identify the types and describe them. We found no relevant experiences in the literature for doing this so we decided to go for a simple approach gathering a team of experienced people in a workshop (half a day). The requirements for being a member of this group was that they together would represent practically all aspects of software development projects: project management, development, maintenance, documentation, customer relations, requirements, architecture and so on. This group pretty quickly agreed upon three distinct types of projects. It seemed like this was already known to them, although it had never been described specifically. The three types was given descriptive names and explained using keywords:

1. Greenfield
 - A lot of research
 - Unknown size (time, economy, staff)
 - High risk
 - Never fixed price
2. Push-button
 - Known technology
 - Low risk
 - Known size
 - Usually fixed price
3. Mainstream
 - Uncertainty with respect to system integration issues
 - Known technology
 - Unknown size
 - Moderate risk

This simple overview now represents a common understanding of the project types characteristics and points of distinction based on experience. This process of defining these types is a knowledge management process that summarizes experience that may be difficult to express otherwise.

This can now be used as a tool to support the process of defining a set of variants of the methodology framework, in this case RUP.

4. Adapting RUP to defined project types

By adaptation (of a methodology framework) we mean to deselect unnecessary process elements, alter existing ones and maybe add new ones. This job must be supported by the following types of knowledge;

- about RUP itself
- organization experience
- about the project

The project type definitions come into use with respect to the last issue.

After initially having defined the project types, the next step was to define one instance of the framework for each defined type. We decided to start out with the Mainstream type because this is the most common project type. We arranged a second workshop (2 short days) where the participants were actors with varying and complimentary experience with this type of projects. We started the workshop by presenting and discussing the project type description to establish a common view of what is important for this type and what is not. We also discussed each phase and the participants made post-it notes with important issues. Examples for the inception phase are: "requirements to external systems (integration)" and "change management". We did this to make it absolutely clear what the real challenges, and need for methodology support are for this project type. After having established a common mindset about Mainstream-type of projects we started with a prepared list of all process elements from RUP (artifacts, roles and activities), deciding (through group discussions) what to keep, remove or alter. We soon experienced that we were better off just focusing on artifacts, the roles and activities followed based on which artifacts the group decided to keep/remove/alter. We were not able to complete this group process the first day so we had a continuous workshop a few days later. The second day, we also involved a RUP-mentor because we had some time consuming and difficult discussions the first day when the group had to decide upon process elements that were unknown to the group. Remember, the group was put together to have maximum experience on this particular project type, they were not necessarily experts on RUP. For example, the group did not know what the artifact "capsule" was. Having spent about ten minutes of guessing and searching in the RUP documentation we found it to be relevant for real-time system development. It was then easy to decide that it should be dropped. Having a RUP-mentor present the second day made the whole select/deselect-process much more agile and less time consuming. Having learned these lessons, we believe that this workshop could have been done more efficiently, one day should do it.

After this workshop, we had wider discussions in the company, involving more actors to mature the results. The outcome is a RUP adaptation tuned for Mainstream projects. Out of the original list of 187 process elements, the specialized list has 83 elements.

The next step of the adaptation process is to take this into use in projects. We have decided to keep it simple and will not make a RUP Online presentation of the Mainstream instance. We believe that just having this list is sufficient (as a checklist), RUP templates are found in the full version of RUP Online.

This is how far we have come today. The next step is now to repeat the process for the two last project types: 'Greenfield' and 'Push-button'. Having done this once, we now believe that the two next workshops will be less time consuming.

5. Conclusion

Having first decided to use a methodology framework, such as RUP, as a basis for defining a specialized software development process you first need to consider your development practice. Do you need a common framework for your organization or do you need variants for your most common and recurring project types? This is also a question of how much effort you want to put into it, up front.

Describing project types can be done using workshops with experienced participants with thorough and complimentary experiences with software development in the organization. Establishing a common mindset and understanding of the features and characteristics of each defined project type is important.

To run the select/deselect process, an expert on the methodology framework is a valuable resource to run the process speedy and smoothly. If this resource is not available within the organization it may be brought in from the outside, e.g. engagement of a RUP mentor.

Still being within the adaptation process we have no empirical data to validate how well the specialized versions of RUP work, which at the end is the interesting question here. This work is going on and results from this work are expected in 2005.

6. References

- [1] Krutchen, Philip, An introduction to the Rational Unified Process
- [2] Karlsson, F., Ågerfalk, P.J., *Method configuration: adapting to situational characteristics while creating reusable assets*, Information and Software Technology, Elsevier 2003
- [3] Jacobson, I., Booch, G., Rumbaugh J., *The Unified Software Development Process*, Addison-Wesley, 1999

Tailoring the Unified Process

Christian Stensholt, ErgoSolutions

Abstract

The Rational Unified Process, RUP, is a process framework aimed to support the design and development of software. The creators of RUP are offering a very comprehensive toolbox in the framework. The idea is that the organisation using RUP should tailor the process framework to the actual development projects. This tailoring should be based on the technology used in the project, the customer and type of customer, the domain in which the software will run as well as other characteristics by the design and development. This tailoring is an issue in many organisations, IBM/Rational offers though little practical guidance on how to plan and conduct this tailoring at that time.

ErgoSolutions has created its own variant of RUP, named ESUP; ErgoSolutions Unified Process. The current version is at the moment covering all the different projects run by ErgoSolutions. This paper will describe ESUP as it exists by now. The paper will also describe the process within ErgoSolutions leading to the current version of ESUP. Researchers from the SPIKE project have studied the use of ESUP, and the paper will present some of the results. Finally the paper will describe Fisken ("The Fish")¹, which is the project management process framework in use in ErgoSolutions. The motivation for doing this is that Fisken and ESUP are both supporting the accomplishment of software development projects in ErgoSolutions. The paper will emphasize the important to have good implementation plans and stick with the plans.

Keywords

Unified Process, RUP, method adaptation.

1 Introduction

The development division in ErgoSolutions has used Rational Unified Process for several years since the break through for objective oriented software development in the middle of the nineties. The use was fractional and mostly used by enthusiasts. There were discussions between supporters of more traditional project methods as Goal Directed Project Management (GDPM) and supporters of RUP.

In the spring 2002 we started the work to tailor the Unified Process to our organization. We have two processes in our quality management system concerning software development, a project management process framework (Fisken) and ESUP based on RUP.

This article describes the work of creating the processes, how we implemented the processes and some experiences with the implementation and use of the processes.

¹ „The Fish“ is a Cause and Effect Diagram also known as the fishbone diagram, hence the name

2 Company description

ErgoSolutions is a company in ErgoGroup, owned by Norway Post. ErgoGroup has about 2000 employees and ErgoSolutions about 300 employees.

ErgoSolutions develops, integrates and administers business critical IT systems for major organisations. The company provides services spanning the entire lifecycle of the IT project: from initial advice and analysis to project management, software development and efficient application management.

ErgoSolutions offers customers solutions for structuring and improving their system architecture. We also offer integration between legacy systems and new system architecture. Another important service area is administration of existing and new application portfolios, which contributes to increased quality, productivity and cost-effectiveness.

ErgoSolutions has skills and experience within content administration, secure electronic interaction and messaging.

Our customers are large companies in the public sector as Norway Post, banking and finance. We also operate within the private sector.

3 ESUP – the specialized UP

ErgoSolutions has been certified to ISO 9001 for several years. In 2002 we were recertified according to ISO 9001:2000. This standard promotes the adoption of a process approach when developing, implementing and improving the effectiveness of a quality management system and continuous improvement of the processes.

The ISO-process approach made it easier to get support for a software development process. Today ESUP is one of four core processes in our quality management system. The other three core processes are sales and marketing, project management (The Fish) and application management.

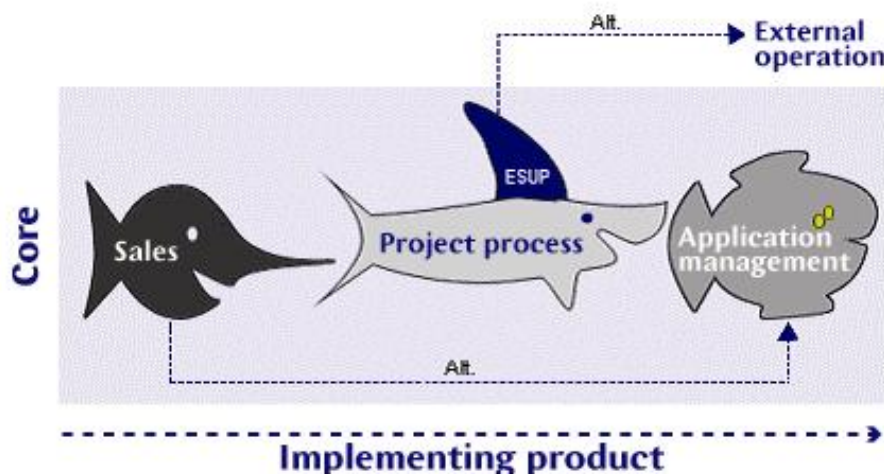


Figure 1

Some of the criticism against RUP has been that it is very comprehensive and has been growing bigger and bigger and is covering larger parts of software development. It might seem that IBM Rational has taken this criticism into account since the latest release of the software has focus on tailoring the process to the individual project.

At the time ErgoSolutions started to develop ESUP this was not supported in the current version of the RUP.

RUP has a large number of artifact templates, description of roles and disciplines. The making of ESUP started with a workshop. We gathered resources with knowledge and experience with RUP and project experience from the whole organisation. The objective for the workshop was to bring forward our own experience with RUP and bring down the number of artifacts and roles and assess the necessity for all the disciplines. The workshop did not have the authority to decide, only to propose.

Several views were put forward and everything was documented on big sheets of wrapping paper on the wall and post-it notes. Sometimes opposite views were voted on to eliminate artifacts and roles.

After the workshop we had collected much experience from project work and software development in our company.

The next step was to refine the result of the workshop. This was done by an internal group consisted of senior project managers, the project director in ErgoSolutions and a web-designer.

The result was a website describing the development process. Very much is similar to RUP, the same four phases; we omitted the Business Modelling discipline used in RUP. The changes are bigger when it comes to roles and activities. First of all we have tried to reduce the number of roles and tailor the description of the role to our own experience and needs.

All the artefact templates are offered, but only a small number is made mandatory. Some more are translated to Norwegian.

ESUP is mandatory for all projects with more than 2000 work hours. For small projects (between 500 and 2000 work hours) we have a simple description formed as a software development plan. Projects less than 500 work hours don't have to use ESUP.

4 *Fisken* – the project management framework

While ESUP describes the SW-development process, *Fisken* describes a project management process. The project process contains a time axis and shows the steps we have defined to carry out in a project in ErgoSolutions regardless of the type of development process. The steps are shown as fishbone in a fish (Cause and Effect Diagram).

The purpose of a standardized and uniform project process is primarily to contribute to an efficient way of conducting projects in ErgoSolutions. To carry out projects within a big company it is necessary that the principles to establish, implement, do quality assurance and complete the project are uniform and practical. It is also necessary that some important routines with templates are used. Good processes are also a good sales argument.

The Project Management Office in the company defined this process.

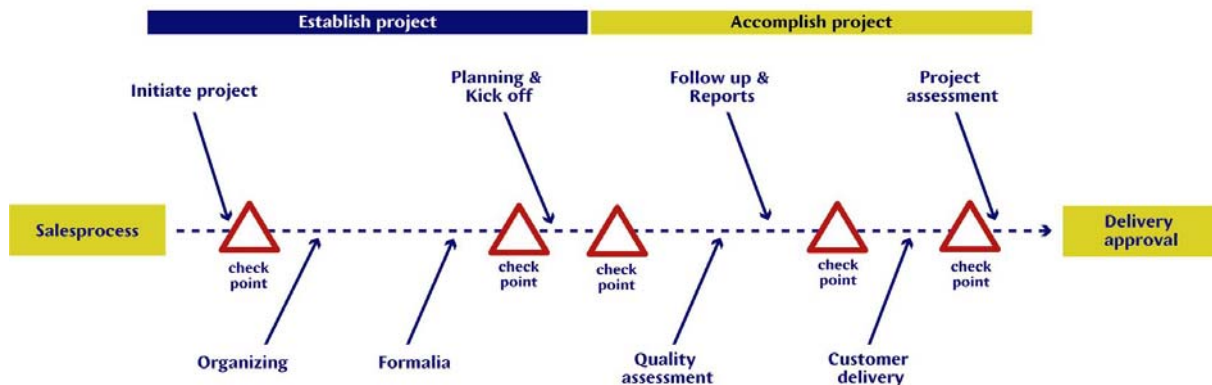


Figure 2 A simplified version of the Fish

All projects in ErgoSolutions are imposed to follow the guidelines for project management and project work as described in the Process. That however, doesn't mean that all projects have to do everything as described in the process. The idea is that each project has to adjust the use of the process to the size of the project and its complexity.

On each fishbone activities, marked with red text if mandatory and black text if optional. Behind every activity there is a link to documents and specifications (The activities are not shown in the simplified figure. Some examples of mandatory activities are "Election of project manager", "kick-off meeting" on the fishbone "Initiate project").

For smaller projects (between 500 and 2000 workhours) we use a smaller process called Gullfisk (Goldfish). For projects less than 500 workhours, we use an even smaller process called Reka (Shrimp)

Fisken is the administrative project process in ErgoSolutions. ESUP is the software development process. They are both accommodated to Goal Directed Project Management method, which the Fisken is based upon. We tried to make the processes independent of each other to make it easy to exchange one of the processes without having to change the other. We also looked for inconsistency and overlap between the two processes.

Fisken is a tool for the project manager through the lifecycle of a project providing guidelines for planning, managing, reporting and to manage deviation. Through the lifecycle the project manager has to go through five mandatory checkpoints to assess the project (represented by the triangles). Unlike the Fish, ESUP describes the process for developing the system to be made for all of the project members.

5 Experiences

To implement the processes in the organizations we pointed out three assumptions:

- It is necessary to have a firm foundation in the management on all levels
- Measure the process improvement
- Efficient tool support

We made a plan to implement the process in the organization, what we did:

- Performed a mini-assessment with external help based on the Best Practices in RUP. The assessment focused on three areas:

- Requirement management,
- Change and configuration management
- Iteration planning

- Selected three pilot projects
- In-house training for the whole organization and specific training for the pilot projects
- Carried out 8 workshops for the pilot projects based on the mini-assessment
- Hired a mentor for the biggest pilot project (>35 000 work hours)

The pilot projects got a good start. The biggest project was loyal to ESUP through the whole project (Finished in June 2004).

Soon after we had started the implementation of the processes the IT-crisis came. Few new projects were started and the focus changed from processes to cost reduction. The consequence was lack of foundation in the management and the plans for measurement of process improvement disappeared. And of course there was no budget for tool investments.

We also lost the focus and new projects got no support in implementing ESUP. We did not lose focus on "Fisken" in the same way, because one of the purposes with "Fisken" is to report project status and

keep an eye on the project economy.

So we kept focus on economy, if the project was on time and budget, but lost focus on efficiency in the projects.

After a period of almost two years we have experienced that fewer people mastered ESUP than expected.

We have tried to adjust ESUP to each project and to adapt new changes and experience to ESUP. We have also seen that projects change from traditional development project til portal project and integration projects. ESUP has to adapt these changes as well.

Despite the reduced focus on intruducing ESUP and evaluating the effects, some projects have used it lately. The decision to use ESUP, in each case, is based on personal interests and a belief in that it will support the project, it is not mandatory

By early 2004 ErgoSolutions joined the research program SPIKE and initiated an activity to improve ESUP and increase the use of it. As a part of this work an investigation of the latest experienceses with ESUP was conducted. Eight recent projects were analyzed by interviewing project participants, mostly project leaders. The scope of the interviews was to document the degreee of use and how well it had worked, or possibly, how bad it had worked to identify areas of improvement. The interviews also looked into the use and experiences with Fisken, the project management framework.

Of the eight projects, six of them used ESUP. The interviews were analyzed to determine the *degree of use* which was defined as either *extensive, medium, little* or *none*. Similarly, the interviews was analyzed to determine the degree of use for Fisken, using the same scale.

ESUP degree of use		Fisken degree of use	
Extensive:	3	Extensive:	8
Medium:	2	Medium:	0
Little:	1	Little:	0
None:	2	None:	0

The respondents of the six projects using ESUP was also asked to evaluete *how well* ESUP and Fisken worked in the projects. We registered both statements that indicated that it had worked well and worked not so well. This evaluation was categorized using a scale from *very well, medium* or *bad*. Of the six projects using Fisken, we only got data on 5 projects (one of the interview respondents was not a PL and could not answer).

ESUP, how well it worked (6 projects)		Fisken, how well it worked (5 projects)	
Very well	4	Very well	2
Medium	4	Medium	3
Bad	1	Bad	1

These figures gives us a rough overview of the use and effect of ESUP and Fisken.

6 Conclusion

We carried out a good process making ESUP, engaging av lot of people. We made good plans to introduce the process to the organization. We had lectures, pilot projects, hired mentors and had assessments to point out the main tasks.

Processes as RUP has a tendency to become too comprehensive because software development is complicated and processes as RUP have to take many considerations. That makes it even more important to tailor the process to your own organization. It is important both to take care of the basic idea of Unified Process and take into consideration own experience and history and to collect viewpoints from other corresponding processes as agile processes (agile modeling and XP)

It is very important to have good introductions plans and to point out assumptions to succeed. And more important, stick to the assumptions.

Our mistakes were that we did not kept a firm foundation in the management on all levels. We did not got approval to measure the effects of the process by using methods as CMM or SPICE.

The IT-crisis contibuted very much to change the focus from efficient software development to cost reduction.

Both ESUP and Fisken are mandatory in all project with more than 500 work hours, but the use of ESUP is really voluntary in ErgoSolutions today. The motivation for using ESUP is very individual among the project managers. Our experiences with ESUP (and Fisken) so far indicates that it definitely has a value, but that it still can be better adapted to the organization. Ergo will tailor ESUP further to different project types within the company.

7 Literature

Phillipe Krutchen, "The Rational Unified Process ,An Introduction, Second Edition", 2000

Walker Royce, "Software Project Management: A Unified Framework", 1998

Scott W. Ambler, "Agile Modeling", 2002

8 Author CVs

Christian Stensholt is working as a project manager in ErgoSolutions. He graduated at the University of Oslo in 1974.

Stensholt has worked as a consultant since 1986. Since 1996 he has worked especially with object-oriented techniques and methods within project management and software development processes.

He is the project manager for the SPIKE project at ErgoSolutions.

Beyond Software process improvement: A case study on change and knowledge management

Jan Pries-Heje & Pouya Pourkomeylian

Abstract

This paper gives an account of a case study in the software developing part of a large pharmaceutical company. Three improvement efforts were undertaken. The first two failed mainly due to rigid process focus and too much reliance on normative models. The third effort succeeded because the effort went beyond traditional software process improvement and included change management and knowledge management mechanisms. The paper discusses the findings from the case in relation to literature on organisational change and knowledge management. Finally the paper concludes with four lessons learned: Lesson 1: SPI is a knowledge creating process and should be treated as such. Lesson 2: Keep the SPI effort oriented towards the practitioners. Lesson 3: The SPI developers shall see themselves as service providers. Lesson 4: If you don't measure you don't know!

Keywords

Software Process Improvement, Case Study, Organisational Change, Knowledge Management.

1 introduction

Software Process Improvement (SPI) has become a very popular way of improving a software developing organization's ability to develop software. According to the common wisdom on SPI the way to do it is the following. First you assess your organizations current capability to develop quality software. Second, based on the assessment results, you derive a stepwise and focused strategy for improving the organizations capability. You then make improvement efforts an integral part of your organization's long-term goals. And then the result will be in an increase in both the organization's quality of services and the productivity.

This common wisdom of SPI is certainly appealing, and it has inspired many software organizations to engage in improvement initiatives. However, many organizations struggle to meet expectations, and many eventually fail.

If we take a closer look at what the organizations actually do we find that quite many organizations use a normative model to guide their SPI effort. The most popular normative SPI model seems to be the Capability Maturity Model – often just called CMM (Paulk *et al.*, 1993) having more than 2800 appraisals in organizations officially reported. (Software CMM, 2003).

The CMM describes the process capability of software organisations at five levels. The higher the process maturity, the lower the risk, and the higher the productivity and quality become. The idea is that an organisation at a higher maturity level will perform better than one at a low maturity level. The model provides a roadmap for moving from an ad hoc process culture to a culture of process discipline in which there is continuous process improvement. Each maturity level in the CMM is composed of several KPA and each KPA is organised into five sections called common features. The common features specify the key practices that, when collectively addressed, accomplish the goals of the key process area.

1.1 The maturity levels in CMM

The first level is the *initial* level, which has no requirements. This level is consequently often described as an ad hoc level. The software process capability of a level 1 organisation is unpredictable because the software process is constantly being changed or modified as work progresses. Schedules, budgets, functionality, and product quality are generally unpredictable.

The second level is the *repeatable* level. Here, the focus is on getting software project management under control to track costs, schedules, and functionality. The necessary process discipline should also be in place to repeat earlier successes in projects with similar applications and to avoid past failures. The software process capability of level 2 organisations can be characterised as disciplined because the planning and tracking of the software project is stable and earlier successes can be repeated.

Level 3 is the *defined* level, and the focus is on standardising the processes used in the software projects across the entire organisation. This requires the creation of defined processes as a basis for consistent implementation and better understanding. All projects use an approved, tailored version of the organisation's standard software process for developing and maintaining software. The software process capability of level 3 organisations can be characterised as standard and consistent because both software engineering and management activities are stable and repeatable. At level 3 management has good insight into the technical progress of all projects. Costs, schedules, and functionality are under control, and software quality is tracked.

The most significant quality improvements, according to the CMM, can begin when the organisation at level 4 (the *managed* level) has initiated comprehensive quantitative measurements to ensure that the software processes operate within statistically predictable limits. At this level, the software process

capability of the organisation can be characterised as predictable because the process is measured and operates within measurable limits. Detailed measures of the software process and product quality are collected. Both the products and software processes are quantitatively understood and controlled. On level 3 the software products are expected to be of a predictably high quality.

At level 5 (*optimised* level), the entire organisation is focused on continued process improvement based on quantitative data from level 4. The software process capability of level 5 organisations can be characterised as being in a state of continuous improvement. Improvement occurs both by incremental advancements in the existing process and by innovations using new technologies and methods. At this level, the organisation is typically characterised by a focus on continuous process improvement.

1.2 The IDEAL model

On top of the CMM model the IDEAL model was developed (McFeeley 1996). In this model we find a suggested way to organize an SPI initiative in five steps: 1) *Initiating* the SPI effort, 2) *Diagnosing* the capability of the organisation, 3) *Establishing* the needed resources for the further improvement efforts, 4) *Acting* for improving or creating new processes, 5) *Learning* from the SPI efforts done through this circle. The idea is that the SPI efforts should be planned and organised as one project going through the IDEAL phases to complete one circle and continue again in a new circle of improvement activities later on.

1.3 How the remainder of the paper is organised

The paper will be developed as follows. In section 2 we will give an account of the case study research methodology that we used. In section 3 we give an account of the case study at AstraZeneca R&D (AZ) Mölndal in Sweden within a software development department called DevIS (Development IS) in three parts, one for each improvement effort. In section 4 we discuss the learning from the case, and relates our findings to change management and knowledge management literature. And finally in section 5 we summarise and conclude the paper.

2 research methodology

The rationale for the research methodology arises from the exploratory nature of this study: The discovery of concepts in an arena that is not well studied. Originally we were concerned about efficient and effective SPI. However, the rationale shifted toward the need to study the emergence of organisational change and knowledge management. These requirements were satisfied by a research design that involved a longitudinal, embedded, single case setting (Yin, 1994). The setting is a single case, centered on software process improvement in a large Research and Development Unit within AZ. One of the authors worked as Head of Quality Management in the organisation whereas the other author had no other relationship with this company outside of the study.

Data was collected by the author working in the organisation in the period 1998-2003; less systematically in the beginning and more elaborate and systematic towards the end of the period. We analyzed this set of data seeking an answer to the question "Why is this SPI initiative failing despite the fact that common wisdom on SPI was followed?" As the case developed over time, the researchers were increasingly drawn to focus on organizational activities that implemented organisational change and

knowledge management processes. The following narrative of our findings describes the case, and then discusses the activities beyond common wisdom on SPI.

3 Case: SPI at AZ

AZ is a large pharmaceutical company that was formed in April 1999 by the merger of two companies. AZ has more than 50,000 employees worldwide. Research and development (R&D) takes place in Scandinavia, UK and the USA. The company has close to 10,000 R&D people. In 1999 the investment in R&D was US \$2 billions. Thus it is fair to say that AZ is a very research intensive organisation.

In the pharmaceutical sector many regulatory authorities require pharmaceutical companies and their software organisations to comply with GXP (Good *Manufacturing* Practice, Good *Clinical* Practice, and Good *Laboratory* Practice) rules. GXP rules are the authorities' quality requirements to pharmaceutical companies for ensuring patient health, the quality of processes (e.g. clinical studies or software development) and the quality of products (e.g. tablets or software).

One fundamental requirement is that software organisations must be able to show the authorities, in documented evidence, that software development activities (e.g. software change control, software validation, and data processing and storage) are being performed in compliance with quality requirements. Therefore every software project regulated by GXP requirements should carefully apply all quality rules and be able to show by documented evidence that the software is compliant with the related GXP requirements.

Years ago AZ adopted standard operating procedures (SOPs) that explicitly describe how the company's software quality rules should be addressed in software projects. These SOPs should be applied in all software projects regulated by GXP requirements.

3.1 The first improvement effort

The first attempt to improve quality started early in 1998. The idea was quite simple; gather all quality rules in use in the organisation and make them available to everyone through AZ's intranet. The work was organised and performed as two main activities: 1) gathering the quality rules, 2) designing the intranet solution.

The first activity was mainly focused on studying authority's home pages, AZ company's quality standards, and to a certain degree describing how to handle some quality issues. The intranet solution was designed as a very long e-document which was made available through the intranet.

The results of this first improvement effort was presented in its first version (in reality more of a prototype) to the management team of the software developing part of AZ - called Development IS (DevIS). The feedback was to carry on with the improvement effort, trying to focus on specific issues. For example the management team asked for issues related to software maintenance with specific focus on quality management to be addressed.

However, the first improvement effort stopped here. The reason was that software developers and people responsible for software quality couldn't use the first version due to its prototypical appearance. For example large amounts of information were at too high a level of abstraction, and the architectural design – one long e-document – was not user friendly.

3.2 The second improvement effort

Later the same year the second improvement effort took place. This time the effort was focused on studying a part of the AZ organisation doing software maintenance and documenting their work as best practice.

This second improvement effort was organised differently, with three main parts. Interviews and direct observation was used to gather data on how software maintenance was done. Another part of this second effort was focused on analysing and structuring the information that had been gathered in the first – failed - improvement effort. The third part was focused on designing a more user-friendly intranet solution for making the gathered information available.

The project group carrying out this second improvement effort was quite focused on documenting the work of the maintenance organisation and designing a better intranet solution. They were so excited about the efforts and the results that they never asked: what shall we do when we have the results, how shall we go further, what do we need to maintain the product and so on.

The second improvement effort took 4 months and resulted in an intranet page including two parts:

- 1) The best practice part, describing the best way of working with software maintenance activities.
- 2) A more structured information database addressing quality rules and to some degree how to deal with some specific quality issues.

The result was presented both for the management team of the organisation and for all practitioners in a department meeting. Some people started looking at the intranet solution to get ideas for addressing practical problems in different software projects, and probably a few found what they were looking for. But use of the intranet solution was very sparse, and after some months updating and maintaining the site was stopped due to the lack of resources and management support. So also the second improvement effort was a failure.

3.3 The third improvement effort

Early in 1999 the results of an informal problem analysis showed that the software organisation's software project practice needed improvement. There was also a need to provide guidelines to understand the SOPs and the GXP rules. Many practitioners working in different software projects pointed to this subject for improvement by sending e-mails to an analysis group responsible for gathering software professionals' ideas for improvement. Management at that time did not have detailed knowledge of the depth of the problem or how to improve the software project practice.

At this point AZ had the experience of the two unsuccessful attempts and wanted to "do it right" this time. People in AZ had heard about successful results in other organisations using SPI and the CMM for improving the capabilities of software organisations. After further study of the SPI literature the need of using two approaches for the improvement activities became clearer: 1) a structured and systematic model for planning organising, analysing, and improving software practices, 2) a model for focusing specifically on software project problems.

After meetings with the director of DevIS and an SPI expert to discuss different approaches to improvement activities it was decided to start the third improvement effort using the IDEAL model for planning, organising, and running SPI activities, and using the CMM to focus on level 2 key process areas.

The first step was to establish the SPI project organisation and diagnose the current maturity level of DevIS's software projects through a CMM-based appraisal. On the basis of the results of the appraisal a report with suggested improvements were produced in October 1999. The report suggested six improvement activities, with a specific focus on software documentation and software validation to sat-

isfy one important quality requirement, namely that the pharmaceutical industry must document all software engineering activities to comply with health authority regulations. In the case of inspections the company must be able to show documented evidence that a specific task, e.g. that implementation of change in a software product, has been done in accordance with predefined standard procedures.

As the next step, an improvement plan was created and the SPI working group started to work on improvement activities, which resulted in the creation of three new guidelines: a software documentation guideline, a software validation guideline, a software change control and document version control guideline - and developed the template library.

The newly created software guidelines were presented to the steering committee and further modified on the basis of the committee's feedback. Furthermore a MAP analysis (Aaen *et al.*, 2001) was conducted to better understand the nature of our SPI project and to be able to plan and conduct further SPI activities.

An implementation plan was created to address the issues related to change management and presented to the steering committee. The steering committee of the project accepted the refined software processes and the implementation plan and decided to implement the processes throughout DevIS.

The acceptance of the software processes and the decision to implement them throughout the whole organisation was a first important milestone for the initiative. The implementation activities were scheduled to take place between August 2000 and June 2001. One important aim was to change the context in which the new software processes should operate. Therefore, among other activities, a trainee program was scheduled for all practitioners at DevIS. The implementation phase also included further improvement activities in which the processes would be enhanced on the basis of experience of using them in practice. The implementation phase aimed to result in a new version of the software processes in June 2001.

To support knowledge sharing in the organisation a knowledge management strategy was adapted to address issues related to transferring different types of knowledge. This strategy included the use of such diverse things as training sessions, networking facilities, and an electronic process library.

4 Discussing case findings

After the third improvement effort a second CMM appraisal was carried out to find out whether the new processes had resulted in any improvement. The appraisal results showed that the maturity of the organisation had improved by 36% in comparison with the first appraisal in 1999.

Thus the third improvement effort was declared successful. The reason for the success was presumably the definition of a change process and a knowledge management strategy on top of the traditional SPI effort.

Furthermore this inclusion of change management and knowledge management beyond SPI has now become the way of working within the AZ organisation. Today the second version of the software quality framework has been developed as the result of a fourth improvement effort. And a culture of continuous improvement work has been cultivated within the software organisation.

4.1 Change management as part of the improvement effort

The primary goal of SPI efforts is to change the practice of software work to the better. Software work is often organised as different size projects including different numbers of people following a process aiming to reach a goal within a certain period of time. Changing from one situation to another requires changing the behaviour of the software developers.

Many researchers have approached change and discussed different models for addressing how organisational change should be planned and performed (see Collins 1998, Hall 1997, Weinberg 1997, and Burnes 1992). According to Collins (1998) change models generally can be divided into two broad categories: 1) *under-socialised* models, which have a rationalistic and planned stepwise view to change. Through such models problems are identified, the possible solutions are selected and preferred, action is planned and occurred, and change has been monitored. This approach reminds of McFeeley's (1996) IDEAL model through which problems are identified and prioritised as result of an assessment and action is taken to plan and perform improvement efforts. 2) *Over-socialised* model, which have an interpretative approach to change through which goals should be clarified, communicated and understood by people. This approach is focused on creating an engaging environment in which individuals are the engine behind the change process for overcoming resistance to change (Collins 1998, Weinberg 1997).

Ebrahimpur (2002) means that in order to cope with dynamics of change organisations should improve their ability to effectively manage and apply knowledge. She proposes that change should be analysed, not just from a rationalistic stepwise point of view, but also from a point of view to understand *What* is to be changed, *Why* things should be changed, and *Who* is to be benefit from this change. She further argues that a consequence of this approach would be that improved routines and processes are designed with consensus as the road map, resulting in improvements as part of practitioners' daily work.

Looking closer at the improvement efforts at AZ we realise that both the IDEAL and the CMM model support addressing Ebrahimpur's first question to some extent, i.e. *What* should be changed? Using the IDEAL and the CMM one can establish the current maturity level of the organisation and identify the problem areas and further prioritise the improvement activities. But neither the IDEAL nor the CMM go beyond this step and discuss in detail the importance of preparing activities and steps needed to be taken to make clarity for the practitioners and the whole organisation about: 1) *Why* this change process has been initiated and, 2) *What* is the overall goal of the change programme setting in relation to other factors like organisations business goals and company's overall goals. This effort needs involvement of higher management level into the SPI effort and certain communication channels for performing *pre-SPI* efforts to create clarity for all practitioners within the organisation about the means of such initiative.

The learning from the AZ case, and one important reason why the third improvement effort succeeded, is that you should ask and answer the following questions before engaging in an improvement effort:

What is the relation between the SPI-effort's and the organisation's and company's overall business goals?

What is the reason that this SPI effort has been initiated?

What is the expected impact of the SPI effort on the organization?

What is the expected impact of the SPI effort on the software developer' daily work?

Furthermore when the improvement effort have started and the process improvement activities have resulted in the creation of new or improved processes it is important to identify *Which* are the target groups and *Which* services are needed to facilitate the implementation of new processes in reality, i.e. *How* to transfer the new created knowledge to *Whom* and *When*?

4.2 Knowledge management as part of the improvement effort

An organisation's software development practices are based on the knowledge and competencies of its practitioners and managers (Arent and Nørbjerg 2000). Mathiassen *et al.* (2001) argue that SPI efforts depend on the implicit, individual knowledge of practitioners in an organisation. To change software development practices, the organisation should improve the practitioners' existing knowledge (both

theoretical and practical) of its software practices. Knowledge about the new processes should thus be made available on different organisational levels.

One major challenge for organisations working with SPI efforts is hence to create strategies and mechanisms for managing knowledge about software development. Other studies have argued for and illustrated the usefulness of applying Knowledge Management (KM) to SPI (cf. Arent and Nørbjerg 2000, Nonaka and Takeuchi 1995, Baskerville and Pries-Heje 1999, Kautz and Thaysen 2001, Mathiassen *et al.* 2001, Arent *et al.* 2001). These studies indicate that the issues related to knowledge creation, modification and sharing have important roles in SPI initiatives, but to understand where and how to use KM insights to improve SPI practice we need to experiment with and further develop different KM insights in SPI practice.

Our finding from the AZ case is that an SPI effort should start by identifying and documenting the fundamental reasons of change. It is important to understand why the change process is initiated. This identifying-reasons-for-change work should be initiated by the management and the result should be communicated to all practitioners. In the identification of reasons the four questions above (see section above) will also be useful. Until the answers are communicated, understood, and accepted by the practitioners you don't have a stable ground for planning and performing an SPI project.

Generally in an SPI effort two types of knowledge are created:

- 1) the process knowledge which deals with the fundamentals of SPI such as: management of SPI initiatives, how such an initiative should be guided and issues related to SPI's focus on target (s) (see Aaen *et al.*, 2001), and
- 2) Software Engineering knowledge such as: project planning, quality assurance, change control, and configuration management (see Pressman 1997).

As long the SPI project goes on the amount of knowledge being created is rising. For better managing and applying the created knowledge our AZ case gave evidence that a knowledge management strategy should be developed early in the project. This strategy should be based on organisation's knowledge management needs and we suggest that it includes both *codified* and *personalised* approaches (cf. Hansen *et al.* 1999, Mathiassen & Pourkomeylian 2003).

A *codification* strategy is based on a cognitive model through which the focus is on codification of knowledge objects that are stored on databases from which they easily can be accessed and used by practitioners in the organisation (Swan *et al.* 1999).

In contrast, the *personalised* strategy is based on a community model which provides a perspective in which knowledge is closely tied to the person who creates it and is mainly shared through direct person-to-person contacts (Swan *et al.* 1999). Further, on the basis of a KM strategy different facilities should be developed to support knowledge sharing in the organisation, e.g. role based electronic process libraries including a search mechanism can support the knowledge sharing activities for codified knowledge and make right information available and accessible to the right person within a very short time. Networking facilities, such as training sessions, workshops, coaching efforts can be adopted to support sharing of personalised knowledge within the organisation.

5 Conclusion

Insights from the analysis of our case at AZ shows that organisational change management and knowledge management can be used to support traditional approaches to SPI. In short we found the following four lessons to be important:

Lesson 1: SPI is a knowledge creating process and should be treated as such. An SPI effort is a huge knowledge creator process. Therefore it is important to put knowledge management in focus rather than process improvement. In practical terms this means more focus on addressing issues for knowledge creating, modification, and sharing. Addressing KM issues will lead to better understanding of

the Whats, the Whys, and the Hows in the organisation and gains practitioners support and commitment to change and reduce resistance to change.

Lesson 2: Keep the SPI effort oriented towards the practitioners. To better aim the right knowledge at the right person using the most effective knowledge sharing mechanism the knowledge sharing needs of practitioners should be identified. Further the organisations KM strategy should be evaluated and based on the evaluation and identification of the characteristic features of the knowledge being shared a KM strategy should be developed and implemented. In this way different effective knowledge sharing channels will be developed to support knowledge transformation to different practitioners to satisfy their specific knowledge requirements.

Lesson 3: The SPI developers shall see themselves as service providers. To facilitate an effective implementation the SPI team should become a service provider offering different services to support an effective implementation of the new processes in the whole organisation. The team should offer different services e.g. training, consultation, advisory, coaching, and measurement to make new processes work in reality. The benefit of this action is that the SPI efforts continuous working near practitioners, support an effective change process, and creates an environment for further improvement efforts.

Lesson 4: If you don't measure you don't know! To improve the practice of SPI and further improve the quality of the new or improved processes measurement should be performed in continuously way. At least two levels of measurements should be performed to support continuous improvements: 1) measuring the effectiveness of the KM mechanisms in facilitating the knowledge sharing process, 2) measuring the effect and the usability of the new or improved software processes in reality. Furthermore improvements in the organisation should be based on the results of these measurements, which support cultivation of a continuous improvement culture in the organisation.

6 Literature

- Aaen, I., Arent, J., Mathiassen, L. and Ngwenyama, O. (2001). A Conceptual MAP of Software Process Improvement. *Scandinavian Journal of Information Systems, Special Issue on Trends in the Research on Software Process Improvement in Scandinavia*, Vol. 13, pp. 123-146.
- Arent, J. and Nørbjerg, J. (2000). Organizational Knowledge Creation: A Multiple Case Analysis. *Proceedings of Hawaii International Conference on Systems Science*.
- Arent, J., Pedersen, M. H. and Nørbjerg, J. (2001). Strategies for Organizational Learning in SPI. In: Mathiassen, L., Pries-Heje, J. and Ngwenyama, O. (Eds.): *Improving Software Organisations - From Principles to Practice*, Addison-Wesley.
- Baskerville, R. and Pries-Heje, J. (1999). Managing Knowledge Capability and Maturity. In: Larsen, T., J., Levine, L., and DeGross, J., I. (Eds.): *Information Systems: Current Issues and Future Change*. Norwell, MA: IFIP/Kluwer Academic Publisher.
- Burnes, B. (1992). *Managing Change: A Strategic approach to organisational development and renewal*, London: Pitman Publishing.
- Collins, D. (1998). *Organisational Change: Socialigical perspectives*, New York: Routledge.
- Ebrahimpur, G. (2002). *Changing in Large Firms, Aligning Change and Agility*, Ph.D. Thesis. Department of Project Management, School of Technology Management and Economics, Fenix Research Program, Chalmers University of Technology, Gothenburg, Sweden.
- Hall, D. J. (1997). Organizational Change: Kinetic theory and organisational resonance, *Technovation*, 17(1).
- Hansen, T., Morten, N. and Tierney, T. (1999). What's Your Strategy for Managing Knowledge? *Harvard Business Review*, pp. 106-116.
- Kautz, K. and Thaysen, K. (2001). Knowledge, Learning and IT Support in a Small Software Company. *Proceedings of the European Conference on Information Systems*, Bled, Slovenia.
- Mathiassen, L., Pourkomeylian, P. (2003), Managing Knowledge in a Software Organisation, *Journal of Knowledge Management*, Vol 7 No. 2.
- Mathiassen, L., Pries-Heje, J. and Ngwenyama, O. (Eds.) (2001). *Improving Software Organisations - From Principles to Practice*. Addison-Wesley.
- McFeeley, B. (1996). *IDEAL. A User's Guide for Software Process Improvement*. The Software Engineering Institute, Carnegie Mellon University, Pittsburgh, Handbook CMU/SEI-96-HB-001.
- Nonaka, I. and Takeuchi, H. (1995). *The Knowledge-Creating Company*. Oxford University Press.
- Paulk, M., Curtis, B., Chrissis, M. and Weber, C. (1993). *Capability Maturity Model for Software Version 1.1*. The Software Engineering Institute, Carnegie Mellon University, Pittsburgh, Technical Report CMU/SEI-96-TR-24.
- Pressman, R. S. (1997). *Software Engineering, a practitioner's approach*. Fourth edition, McGraw-Hill, International Editions, Software Engineering Series.
- Software CMM (2003). *Process Maturity Profile, CBA IPI and SPA Appraisal Results, 2003 Mid-Year Update*, September 2003. Downloaded from: <http://www.sei.cmu.edu/sema/pdf/SW-CMM/2003sepSwCMM>.
- Swan, J., Newell, S., Scarbrough, H. and Hislop, D. (1999). Knowledge Management and Innovation: Networks and Networking. *Journal of Knowledge Management*, Vol. 3, No. 3, pp. 262-275.
- Thomson, H. E. and Mayhew, P. (1997). Approaches to Software Process Improvement. *Software Process-Improvement and Practice*, Vol. 3, No. 1, pp. 3-17.
- Yin, Robert K. (1994). *Case Study Research: Design and Methods*. 2nd Edition. Sage Publications.
- Zahran, S. (1998). *Software Process Improvement: Practical Guidelines for Business Success*. Addison-Wesley.

7 Author CVs

Jan Pries-Heje & Pouya Pourkomeylian

Jan Pries-Heje holds M.Sc. and Ph.D. degrees from Copenhagen Business School, Denmark. He is now Associate Professor at [The IT University of Copenhagen](#), Denmark, and belongs to the research group: Design and Use of IT (DUIIT). He is also (part-time) Professor in Software Engineering and Management at the IT-university of Gothenburg.

Pouya Pourkomeylian holds B.Sc. in System Analysis and M.Sc. in Informatics from Lund University in Sweden. He received his Ph.D. in Informatics from Göteborg University, Sweden. He works today for AstraZeneca as Head of Quality and Security Enabling Services in Sweden and collaborates with the IT University in Göteborg, Sweden as lecturer and researcher within the Software Engineering and Management programme.

The Software Development Culture of Northern Ireland

Donald McFall, Fergal McCaffery, George Frederick Wilkie

*Centre for Software Process Technologies, University of Ulster at Jordanstown,
Shore Road, Newtownabbey, Northern Ireland BT37 0QB*

Email: d.mcfall@ulster.ac.uk, f.mccaffery@ulster.ac.uk, fg.wilkie@ulster.ac.uk

Abstract

The Northern Ireland (NI) software development survey was developed by the Centre for Software Process Technologies (CSPT). CSPT was established in September 2002 and is jointly funded by Invest Northern Ireland (an economic development agency) and the University of Ulster. One of the centre's early priorities was to obtain an understanding of current local software engineering practice. The vehicle chosen to obtain this information was a questionnaire, administered through face-to-face interviews. Key personnel in fifty-six software development organizations were surveyed in this way. This paper presents a subset of our findings with a focus on quality and software process improvement issues in NI software development organizations.

Keywords

Software industry survey, process area issues, process improvement and quality issues.

1 Introduction

From a business perspective there have been studies of the NI software industry carried out by organizations such as local trade associations [1] but no detailed technical perspective of the industry has ever been produced. While some surveys provided a clear indication of 'who was doing what' in terms of software production there was little about how this was achieved in a technical or quality sense. The CSPT needed to understand these processes because one of the centre's objectives is to assist software development organisations in NI with their software process related needs; the survey then seemed like an obvious initial task.

2 The Structure of the Survey

During the period September 2002 to June 2003 the survey data was collected. It was then analysed over the period from July 2003 to September 2003. The survey targeted only companies who engaged in working on all or part of the software development lifecycle. It was also important that we engaged the senior people in the company as the survey may not have been completed accurately by other staff.

NI has approximately 110 software companies that develop software either as a prime business activity or as an important part of their business function. The following categories of company participated in the survey:-

- Indigenous: NI (home grown) software company for which the main part of their business is software development;
- Multinational companies whose main business is software development;
- Any company whose main business is not software development but nonetheless engage in life-cycle activities (these might include banks, civil service or smaller businesses developing software in a more limited capacity).

The survey was extensive in its scope encompassing more than half of all of NI's software development companies, which together employ 80% of NI's software engineering employees.

3 Survey Findings

3.1 General Issues.

It emerged that even though 71% of the software companies in NI are indigenous, they only employ 40% of the entire software engineering workforce. Other companies employ 60% of the NI software engineering workforce. It appears that even though the multi-nationals make up only 12% of the number of overall software companies, significantly they employ 52% of the NI software engineering workforce. The majority of software companies in NI tend to be small, with two thirds of the software companies in NI employing less than 20 people on lifecycle activities.

Using the Republic of Ireland (ROI) [2] software industry as a yardstick it is possible to envisage what growth NI might have achieved within the software industry. The NI software industry employs approximately 7600 people across approximately 110 software companies. By way of context, the ROI has 800 software development companies that employ more than 25,000 people. In terms of population, NI has 1.5 million people and the ROI has 4 million people. Accepting the recent downturn in the software economy since the 'September 11th' disaster, and applying ROI statistics to NI, one might expect NI to have 9400 people employed in the software industry. Therefore, the NI software industry is only 75% of the size that it should be if it had realized the same growth.

3.2 Quality and Software Process Improvement Issues

Due to the slowdown in the world economy training is one of the areas that companies have cut-back. In terms of the fifty-six software companies surveyed the average number of days training per year planned for each software employee was approximately six days. In the case of indigenous companies one-third assign training only on demand, with over half receiving no more than three days training, and only 13% receiving more than six days training (see figure 1).

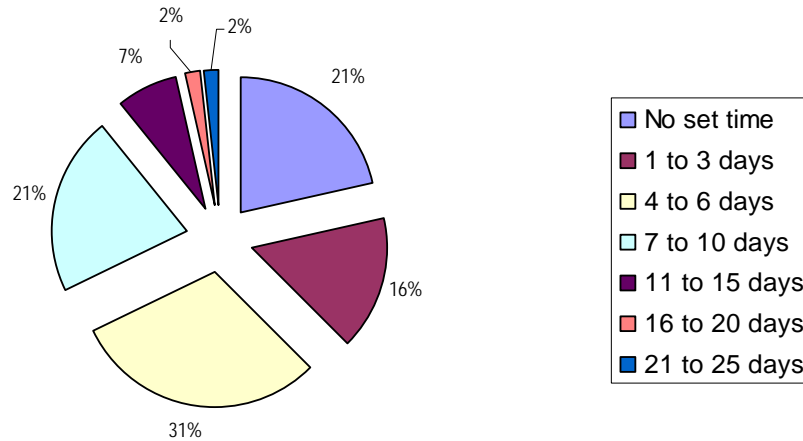


Figure 1: Training

This presents an issue for maturity, especially as management principles need to be in place for any software company to be at the lowest levels of maturity in any of the capability maturity models used in the global software industry. The reality is that despite some good intentions and the position taken in strategic documents, most employees in NI's indigenous software companies have received little or no training within this past two years.

Approximately half of the software development companies surveyed have a team or person who is dedicated to ensuring quality assurance throughout the organization, with only 43% of indigenous companies having a dedicated quality team. These statistics initially give the impression that quality is not given enough priority within the NI software industry. However upon analysis of these statistics it emerges that 86% of companies with more than ten software employees have a team responsible for quality, therefore reflecting that assigning someone to concentrate entirely on quality within a small software development organization is a large overhead in comparison.

For small organizations each person may be given individual responsibilities for ensuring quality. The exception to this is in small companies where there is a mandatory standard and in this case it is considered important to have a dedicated quality team. There is a generally a low awareness of the need to address quality and significant differences between how different companies define quality. Larger companies and companies with mandatory standards place more emphasis on adhering to quality standards and procedures than smaller companies without any procedures in place, for some of these smaller companies quality is simply based on meeting the requirements.

38% of the companies surveyed adhere to a mandatory standard. A mandatory standard means that products can only be marketed if compliance and approval have been obtained from the appropriate regulatory standards bodies. Mandatory standard compliance varied greatly depending upon which market sector the company belongs to. Sectors adhering to these standards typically tend to produce life critical, real-time or highly secure systems.

All of those companies requiring a mandatory standard have of course obtained that standard and in addition three-quarters of these companies have achieved ISO9001 accreditation. But the situation is

different when it comes to other non-compulsory standards, with 50% of all companies and 47% of indigenous companies being certified to ISO9001. Evidence obtained during the survey suggests that ISO9001 is a hurdle for many indigenous companies and is not viewed as having a continuous improvement effect and is instead viewed as simply a badge for marketing.

Large software development companies stressed the fact that it is essential to have efficient generic processes in place in order to cope with the complexities of managing a large number of employees. It was also emphasized that process is vital whenever companies experience a large growth over a relatively short period of time. Medium to large companies expressed a degree of contentment with the processes they had in place but many had concerns in particular with project management and control.

Many NI companies are small and in such environments it is of course much easier to communicate. When communication is straightforward it takes away the urgency for processes. This explains why for some, process is not an issue of importance and of course for some companies, where software development is not their main business, software processes are never going to be an issue.

So even though almost all recognise the importance of process, for some it has not become imperative to do anything about it and for many, the war of day to day trading moves process down the priority list. It appears that only when communication becomes very problematic and the lack of processes makes expansion and diversification difficult is any action taken over it and even then NI's smaller software companies find it difficult to navigate the road forward. Many simply do not know how good or bad their processes are let alone how to improve them.

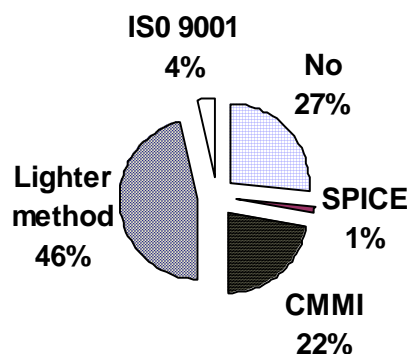


Figure 2: Desire for Software Process Improvement frameworks

A number of the medium to large companies have some awareness of software process improvement (SPI) frameworks. Most of the small companies had little awareness of such approaches. Figure 2, shows that 27% of companies had no desire at all to improve software processes through the use of a framework even though many of these companies identified processes as being important.

4% of companies were content with the process improvement brought about by ISO9001 (even though ISO9001 was not actively used for its improvement potential). Only one company expressed an interest in SPICE [3] in our survey, though other NI companies participated in the SPICE trials and reported some gain through their participation. Encouragingly, 69% of companies had a desire to improve their processes through the use of a framework with 22% of companies firmly in favour of CMM/CMMI [4] based process improvement; several of these companies having experienced problems in the marketplace as a direct result of not being involved in a CMMI programme while their competitors were. Almost half of the companies in our survey desire some form of software process improvement although they expressed doubt about its applicability and concern over cost.

While some companies currently utilise the SPI aspects of ISO9001, the dominant SPI framework is CMM/CMMI, accepting of course that the overall numbers involved are small.

4 Conclusion

In the NI software industry, awareness of standards that can be applied to software is limited and while there is some awareness of SPI, there is a distinct lack of familiarity with it. Several NI software companies have already experienced difficulties in competition with software development organizations worldwide, for work in the U.S.A. Interestingly the prime factor here was not simply cost, but the quality agenda as a whole, with competitor companies having engaged in process improvement programmes.

The larger companies and multinationals, attracted by an educated workforce, employ a large proportion of the software engineers in NI and are very conscious of good standards and practices in software engineering. Many of the multinationals import a sense of urgency in the adoption of best practice from their parent structures and this has had a positive effect on some indigenous software organizations.

Most of the NI software companies are convinced of the importance of process in their working environments and many want to engage in SPI. Although only proper assessment can be conclusive, for the majority of NI companies, particularly small indigenous companies, the characteristics of low maturity are evident. Reliance on individuals in a fire fighting environment, low awareness of standards and problems experienced at the managerial and technical level. Many of the larger companies employing most of the software engineers are process focussed and have a much clearer understanding of the need for process improvement with a better appreciation of the global picture.

A 'titanic' analogy is a natural one as the ship was built in the once great shipyard industry in Belfast. NI's software industry is mimicking the storyline of the titanic but the difference between now and then is that the quality iceberg can be seen from a distance, it's there waiting in the seas ahead. Yes, many countries have acted a little bit earlier and steered another course away from the quality iceberg. Maybe in a strange sense, it is the 'improvement' action of NI's software industry competitors across the world in countries such as India [5] and China [6], that has enlarged the ice berg. The NI software industry, particularly the indigenous industry, needs to minimise the threat of this ice berg by paying due attention to software development processes and doing what is possible to improve them so that they can be competitive in a global marketplace. The cultural attitude of 'Steady-as-she-goes' is not an option.

5 Literature

- [1] Momentum Survey 2001. Advancing Northern Ireland through I.C.T.
- [2] Cochrane, R., IEEE Software, March/April 2001, pp 87-89.
- [3] ISO/IEC TR 15504:1998(E), Information Technology – SPICE, Software Process Assessment, Parts 1-9, Type 2 Technical report.
- [4] Capability Maturity Model[®] Integration (CMMISM) for Software Engineering (CMMI-SW, V1.1, Version 1.1, August 2002)
- [5] Moitra, Deependra, India's Software Industry, IEEE Software January/February 2001. pp 77-80
- [6] Ju Dehua., China's Budding Software Industry, IEEE Software May/June 2001. pp 92-95

6 Acknowledgements

The Centre for Software Process Technologies is supported by the EU Programme for Peace and

Reconciliation in Northern Ireland and the Border Regions of Ireland (Peace II).

7 Author CVs

Dr Donald McFall

Don is involved in the Software Process Improvement Frameworks area. He is in the process of acquiring CMM/CMMI qualification as a lead Assessor and trainer. He has been a member of academic staff at UU since 1989. He has worked with a variety of European and North American software companies including O2, MARI, DataMat, ITC and Nortel on a range of sponsored projects involving software maintenance and re-engineering. Don has been a Visiting Research Scientist at Carnegie Mellon University for the past 2 years.

Dr Fergal Mc Caffery

Fergal is a full-time research associate in the CSPT. His research is focused in the areas of Requirements Management and Project Management. He has 6 years experience as a Team leader/senior software engineer for Nortel Networks and 1 year as a software engineer for Aldiscon Ltd. This period in industry was preceded by 2 years as an academic member of staff at the University of Ulster, during which time he published various papers within the area of adaptive multimedia systems.

Dr George Frederick Wilkie

George is the Director of the CSPT. His principal area of involvement is with Software Process Improvement Frameworks. He is in the process of acquiring CMM/CMMI qualification as a lead Assessor and trainer. George has 12 years experience with Object Technology initially gained through a 3 year period of employment at the Institute of Software Engineering. He has lectured and consulted world-wide on object technology to organisations including NYNEX, AT&T, Ericsson, HP, ITT, KPMG, Lucas Management Systems and J.P.Morgan. Prior to this, George spent 5 years as a Software Engineer with British Telecommunications plc. He joined the University of Ulster in 1992 and has been engaged for the most part in research into the use of object oriented complexity metrics to support software engineering project management. George has been a Visiting Research Scientist at Carnegie Mellon University for the past 2 years. His publications include one book and a range of papers on complexity metrics for object oriented software systems. Fergal is a full-time research associate in the CSPT. His research is focused in the areas of Requirements Management and Project Management. He has 6 years experience as a Team leader/senior software engineer for Nortel Networks and 1 year as a software engineer for Aldiscon Ltd. This period in industry was preceded by 2 years as an academic member of staff at the University of Ulster, during which time he published various papers within the area of adaptive multimedia systems.

A Proposal for an Integrated Advanced Engineering Process

Uwe Hehn

method park Software AG, Wetterkreuz 19 a, 91058 Erlangen, Germany
uwe.hehn@methodpark.com

Abstract: An observation from recent consulting projects is that the definition of advanced engineering processes often serves as a first step on defining engineering processes at all. Although positive in principle this typically leads to a development-biased view at advanced engineering processes. Opposed to this the traditional research-oriented approach to advanced engineering is seen as very costly and rather unspecific. In the paper a proposal is made, that focuses on an Integrated Advanced Engineering Process that integrates in two ways at least:

- The acquisition of knowledge, know-how and ideas and the engineering- and production-oriented aspects are treated as equally important.
- The integration of an essentially independent advanced engineering process and the traditional engineering and production processes. This integration is based on goals and metrics which allow controlling the advanced engineering process and its sub-processes in an effective and efficient way.

1 Introduction

The overall rapid technical and technological progress forces companies to know the newest technologies and to understand their impact on future products and production. Therefore, many companies have implemented appropriate processes to supplement the Product Development and the Production processes. Such processes can be found e.g. in the automotive, the automation or the medical care industries.

Obviously, there is a need to develop strategies and policies:

- How to gain knowledge of the newest research and developments?
- How to evaluate this knowledge with regard to its use for new products and for production?
- How to transfer the knowledge to real products and real production?

There are two principal approaches that are implemented in practice, but both have their limits:

- The classical research-oriented approach – setting up research and development centers – is not favored any more because of the high cost and the often broad but unspecific results.
- The classical engineering-oriented approach – setting up so-called Advanced Engineering departments or expanding existing engineering departments to do pre-series development. Here, new ideas and concepts are tested, studies performed and prototypes built. The results are then available for series development and serial production. The disadvantage of this approach is that advanced engineering and product development or production are coupled too tightly. This often restricts the potential of Advanced Engineering, because the focus typically is on short-term results, waiving medium-term or long-term perspectives.

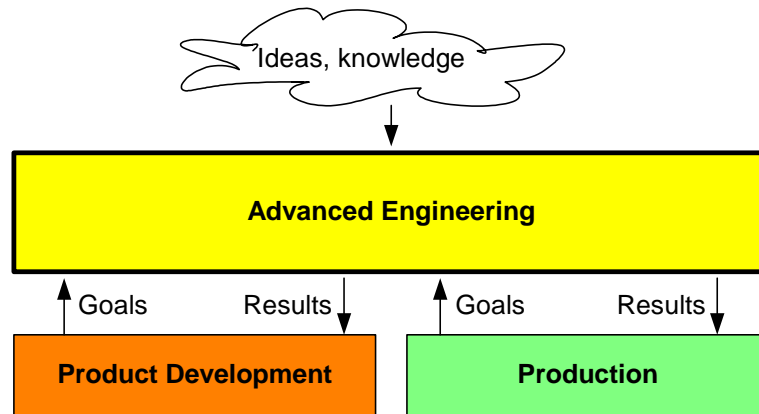


Fig. 1. Overview

The proposal for an *Integrated Advanced Engineering* strategy described here efficiently and effectively combines the strengths of both strategies

- the research-oriented strategy, as well as
 - the product development and production-oriented strategy
- while avoiding their respective weaknesses.

This is made possible by the following measures:

- The overall goals are set by Product Development and Production, ensuring that Advanced Engineering will produce information and know-how most suitable for later use and exploitation.
- The interface to the client processes (Product Development and Production) as well as to the information suppliers is described clearly, and conditions are defined for information and control flows that help to control the knowledge acquisition and evaluation process.
- Acquisition of ideas, definition of proposals for studies and prototypes, selection of Advanced Engineering projects, and conducting studies and pilot projects, as well as the development of prototypes, are essentially considered independent processes.
- The performance of the processes is controlled by metrics, which are deduced from the overall goals, and which are used to decide how the resources are distributed among the Integrated Advanced Engineering Process's processes and sub-processes.
- The metrics may be changed in the course of time.

2 The Integrated Advanced Engineering Process

The proposal described here is called Integrated Advanced Engineering Process:

It is understood as an independent process, which produces

- Ideas
- Project proposals, and
- Prototypes of products

based on demands or goals defined by Product Development and Production.

2.1 Goal

The overall goal for the Integrated Advanced Engineering Process is the evaluation and selection of new technology ideas as an input for the Product Development and the Product Production processes.

The Integrated Advanced Engineering Process covers all activities from the birth of a new technology idea up to feasibility studies including proposals for concepts and solution.

This comprises:

- collecting of ideas
- selection of new technologies based on technical and economical criteria
- feasibility studies
- new technology product ideas
- evaluation, choice and provision of technologies
- development and verification of new solution principles
- system architecture platforms, as well as
- production processes.

2.2 Inputs

One of the main tasks of the Integrated Advanced Engineering Process is to find, gather, preprocess and evaluate available information that could be useful for the purposes of Product Development and Product Production.

The essential input sources for the Integrated Advanced Engineering Process are:

- research results, dissertations, patents (drawn off of institutions and universities)
- technology information in the context of benchmarking (gained by competition)
- customer's and supplier's projects and claims, customer's problems (by customers and suppliers)
- proposals, suggestions, data base of knowledge (as communicated by employees)
- lectures, brochures, technology information about the market and competitors (from congresses and trade shows)

The sub-process of the Integrated Advanced Engineering Process that effectively and efficiently exploits these sources is called *Research and Conception Process*.

2.3 Outputs

The expected outputs of the Integrated Advanced Engineering Process are valuable information to be used for the product development and the series production processes:

- new technology, results of feasibility studies, basic solutions (platforms, system concepts), patents (to be used by Product Development and Production)
- information about the manufacturing of new technology or basic solutions (to be used by Production)

The sub-process of the Integrated Advanced Engineering Process that provides this information effectively and efficiently is called the *Piloting and Prototyping Process*.

3 General structure

3.1 The main processes

In general, the Integrated Advanced Engineering Process implements a workflow between processes, which starts with acquiring information from various sources and ends with reporting the outcome to development and/or production.

1. Generate

Based on the acquired information ideas are generated, evaluated, rated and stored in the *Pool of Ideas*

Ideas

- 2. *Create* *Project*
Proposals
 Based on the information available in the *Pool of Ideas* proposals for interesting projects are created that should/could be conducted as prototype or piloting projects, and stored in the *Pool of Project Proposals*
- 3. *Select Project Proposal*
 One project is selected from the Pool of Project Proposals.
- 4. *Perform* *Project*
 The selected project is carried out and the results of and experiences with this project are reported to development and/or production.

3.2 Controlling the Integrated Advanced Engineering Process

The four main processes run parallel in the sense that as long as input is available for a process the process works with this input generating process outputs.

The processes are controlled by

- observing the goals defined by development and production
- observing metrics associated with the process nodes (at different levels of granularity) and taking appropriate action, if the range of allowed values for a metric is left
- providing feedback loops between the major sub-processes of the Advanced Engineering Process

It is possible

- to “reschedule” a process if by evaluation of appropriate metrics it was found that the task running should not be processed any longer, or
- to stop a process, changing boundary conditions (e.g. given by new goals or changes in metrics) and starting the process anew to adapt it to the changes

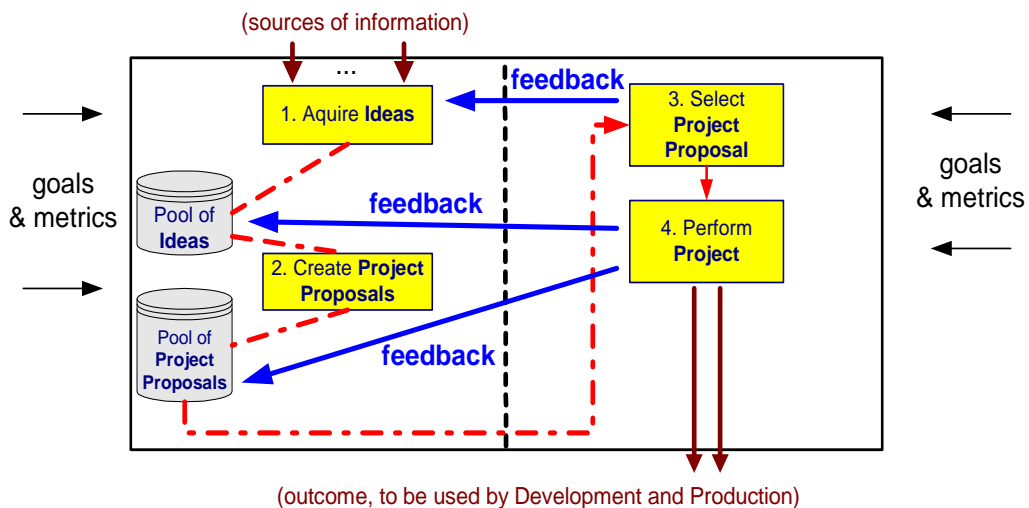


Fig. 2. General structure of the elements of Integrated Advanced Engineering Process (including goals, metrics and feedback)

3.3 Interfaces

The Integrated Advanced Engineering Process consists of two process groups

- 1. “Research and Conception” (responsible for Generating Ideas and for Creating Project Proposals), and

2. “Piloting and Prototyping” (responsible for Selection of Project Proposals and Performing Projects)

There are internal interfaces between these two process groups. We can differentiate

- interfaces, through which information flows that has been processed in earlier stages, and
- interfaces that are used for feedback between the processes of the Integrated Advanced Engineering Process

The interfaces are described in detail in the description of the processes of Integrated Advanced Engineering Process.

3.4 Goals and Metrics

For each instance of the four main processes 1-4 of the Integrated Advanced Engineering process the following information has to be provided:

- One or more goals that have to be considered by the project management of every Integrated Advanced Engineering project

For each instance of an Integrated Advanced Engineering process the following information has to be provided:

- One or more metrics that have to be used by the project management to decide if the given goals are fulfilled to what extent, This allows to recognize if projects may continue or should be changed or stopped.

4 Detailed structure of the Integrated Advanced Engineering Process

4.1 Research and Conceptions

Research and Conceptions is divided into the following groups of processes:

- Generate Ideas
- Create Project Proposals

The task of the *Generate Ideas* group of processes is to:

- Acquire Ideas
- Preprocess Ideas
- Evaluate Ideas

The task of the *Create Project Proposals* group of processes is to:

- Select Ideas and create new Project Proposal
- Estimate Project Proposals Roughly

Both groups of processes are largely independent of each other.

Input to the process

The input to the process is provided by the source-specific sub-processes of *Acquire Ideas*.

Internal links

The link between these groups is a Pool of Ideas that is filled by the process group *Generation of Ideas* and that is used by the second process group *Creating Project Proposals*.

External links

The main link between *Research and Conception* and *Prototyping and Piloting* is provided by the Pool of Project Proposals.

In addition, feedback links supply the following information to Advanced Engineering:

- Suggestions on which domains need new ideas
- Proposals for new projects
- An “a posteriori” rating of the ideas and the projects proposals that are tested in the prototype project

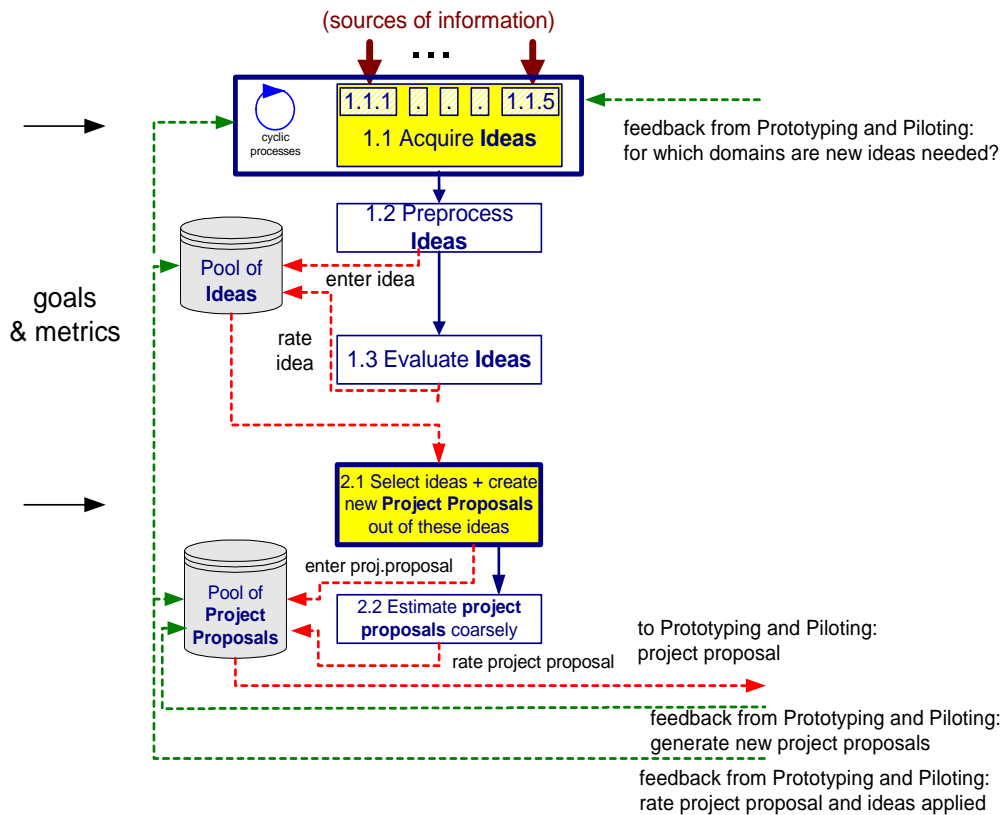


Fig. 3. Detailed data flows of the process group "Research and Conceptions"

Description of detailed processes: Acquire Ideas

The first group of processes (1.1 - 1.3) takes care of the acquisition and first evaluation of ideas that could be valuable for Product Development and Production in the medium or long term.

Acquire Ideas

According to given criteria or strategies inputs are selected that are to be investigated. Naturally, the collecting of ideas is done in varied ways because of the differing sources. The sub-processes are:

- Acquire research results
- Acquire competitor’s information
- Acquire customer’s and supplier’s information
- Acquire employees’ information
- Acquire congress and fair information

In this context there will typically arise further ideas, as for example the description of novelties contained in the inputs, explanations of differences compared to procedures already used or products already in the portfolio, or documented improvement suggestions.

Preprocess Ideas

Ideas of the various sources are prepared, written up in a suitable representation, and at last stored in the Pool of Ideas.

Evaluate Ideas

Each entry in the Pool of Ideas will be rated according to defined criteria (such as “expected usefulness”, “maturity” etc.). This facilitates and supports the selection of suitable ideas and the creation of new project proposals.

Description of detailed processes: Project Proposals

The second group of processes (2.1 – 2.2) is designed for selecting and evaluating project proposals taken to the rated ideas stored in the Pool of Ideas.

Select Ideas and Create New Project Proposal

By selecting suitable ideas from the pool of ideas – of respective technical background and knowledge – project proposals are created and stored in the Pool of Project Proposals.

Estimate Project Proposals Roughly

The elements of the Pool of Project Proposals are preliminarily and roughly rated (quick study) with regard to their potential usefulness, to facilitate the selection of advanced development projects.

4.2 Prototyping and piloting

The process Prototyping and Piloting consists of two groups of processes:

- Select Suitable Project
- Perform Project

The task of the process group *Select Suitable Project Proposal* is selecting a suitable project proposal.

The task of the *Perform Project* group of processes is to instantiate a project, run the project, evaluate the project according to the relevant criteria, and finally elaborate the results with regard to their usefulness for the Product Development and Production processes. This holds even if the project is aborted prematurely.

Process output

The output – reports and appraisals - to be used by Product Development and Production is produced.

Internal links

The selected project proposals are delivered to the project execution process.

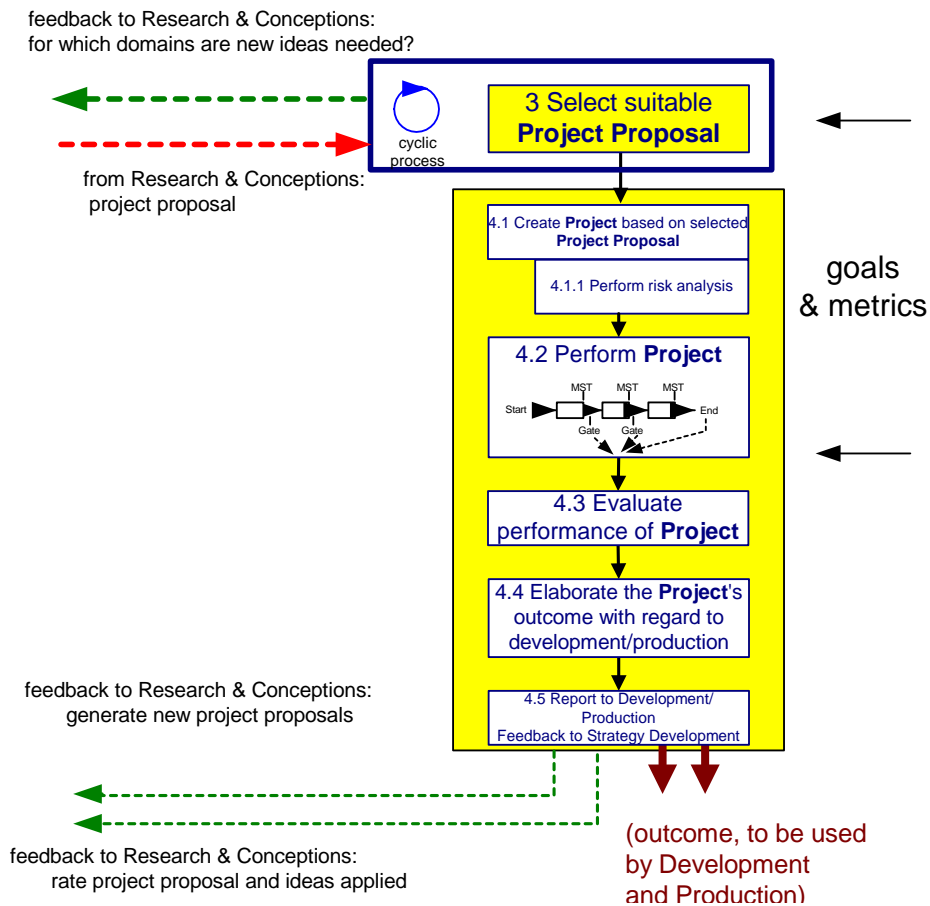


Fig. 4. Detailed data flows of the process group "Prototyping and Piloting"

External links

The main input link between Research and Conception and Prototyping and Piloting is the Pool of Project Proposals.

Feedback links supply the following information to *Research and Conception*:

- Suggestions, which domains need new ideas
- Proposals for new projects
- An “a posteriori” rating of the ideas and of the projects proposals, which are tested in the prototype project

Workflow

The process *Select Suitable Project Proposal* produces as many proposals as needed. Depending on the resources available to the *Perform Project* processes there may be one or more projects running and evaluated at the same time.

Every project is conducted accordingly. Quality gates will be defined which have to be passed in order for the project to continue. The quality gates will typically be associated with a project milestone.

The project may be stopped at any quality gate, if the quality characteristics (functionality, performance, etc.) are not fulfilled or if the boundary conditions of the project have changed. This check is supported by metrics, which are predefined and which help to decide if it is worth to keep the advanced development project running.

Description of detailed processes: Select Suitable Project Proposal

A proposal is selected from the Pool of Project Proposals. The choice considers the interest of the processes Product Development and Production as well as further boundary conditions (like expense, available resources etc.).

The Advanced Engineering actually performed does not have to adopt the proposal minutely. Adaptations to the actual boundary conditions are possible.

This process is cyclic, i.e. the selection of a project proposal may happen at any time, so that several advanced engineering projects may be under way at the same time – if the resources are available.

Description of detailed processes: Perform Project

If a project proposal is selected, a project will be instantiated, run and subsequently rated. The results will be communicated.

Every project implies a feasibility study. The feasibility study contains an economic part and a technical part.

- The feasibility study (economic part) is performed in the context of the project launch
- The feasibility study (technical part) is carried out as the first phase of the project proper. Subsequently the project may be continued as a development project (e.g. development of a prototype)

Create Project based on selected Project Proposal

Based on the selected project proposal a project will be instantiated. Here the project proposal may be modified and supplemented, in order to start a promising advanced development project.

Instantiation includes the elaboration of a project plan – containing goals, milestones, break-off and success criteria – as well as the provision of the necessary resources. Part of the project launch is a risk analysis.

The decision, if an advanced engineering project will actually be carried out depends on the project plan, among others.

Perform Project

The project will be performed in accordance with the project plan.

Principally the first project phase is a feasibility study (technical part), in order to be able to recognize success or failure of the advanced engineering project very early.

At the end of each milestone there is a check, if the project shall be continued or aborted.

Evaluate Performance of Project

After completion the performance of the project is examined. It is determined which benefit can be expected for Product Development or Production.

Elaborate the Project's outcome with regard to development/production

If the project was - at least partially – successful, it is worked out how the results may be used for Product Development or Production. This elaboration represents a significant part of the report that is provided to Product Development or Production.

Report to Product Development or Production, Feedback to Research and Conception

The results relevant to Product Development or Production are made available.

Here feedback is given to either pools of the process Research and Conception, which allows a confirmation, enhancement or modification of the rating of the ideas or project proposals used.

4.3 Integration of the sub-processes

Combining the *Research and Conception* and the *Prototyping and Piloting* structures defined above we get the following integrated structure:

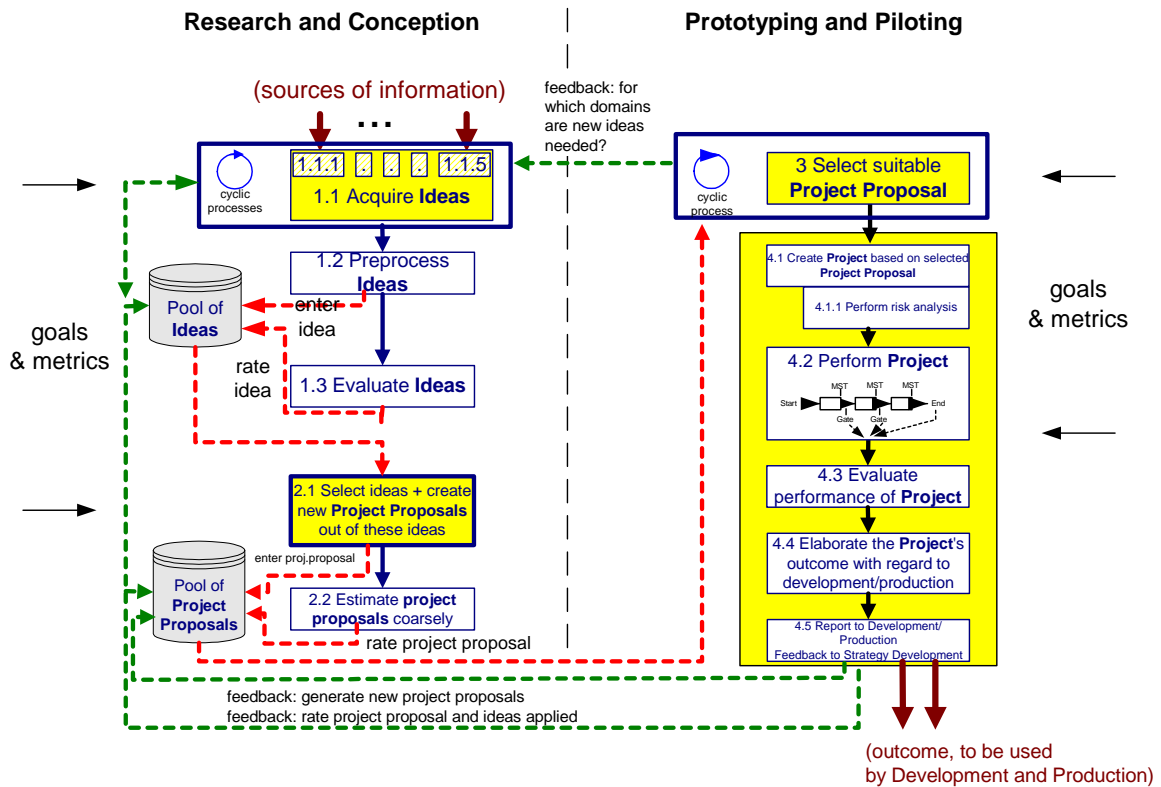


Fig. 5. The combined flows

5 Summary

In the paper we explain how an Integrated Advanced Engineering Process could be setup that covers as well the acquisition of useful ideas for new or the improvement of existing products and their evaluation by feasibility studies and prototype projects. Feedback loops care for the adjustment of the rating of knowledge gathered in Pools of Ideas and Pools of Project Proposals, so improving the quality of the material for future advanced engineering projects.

Engineering and production, which are interested in the results of advanced engineering projects, define goals and metrics for the main advanced engineering processes. This allows the advanced engineering project management to operate rather independently while guaranteeing that acquisition of ideas and their assessment are always focused on their relevance for real projects and products

References

1. Pressman, Roger S., Software Engineering - A Practioner's Approach, McGraw-Hill (2004)
2. Robert B. Grady, Practical Software Metrics For Project Management And Process Improvement, Prentice Hall (1992)
3. Internal documentation (documentation of consulting projects concerning definition and improvement of advanced engineering processes of a variety of our customers), 1999-2004

Continuous tailoring activities in software engineering

P. Saliou and V. Ribaud

*Thales Information Systems, Brest & Département d'Informatique, Faculté des sciences, Brest
E-mail : Philippe.Saliou@univ-brest.fr, Vincent.Ribaud@mythales-is.com*

Abstract

Software activities belong to different processes. Tailoring software processes aims to relate the operational software processes of an organization to the effective project. With the information technology industry moving ever faster, established positions are undergoing constant evolutionary change. The failure of a complex tailoring process of a management information system is reported. There is a need to adopt software processes that can operate under constant change. We propose to add a new dimension called continuous tailoring which is applied at any moment, to each software activity and for each actor. This process is validated in an immersion system intended for young post-graduates before they begin their careers in the software industry. Relationships between tailoring and improvement are considered.

Keywords

Tailoring, software process improvement, quality management, information systems.

1 Introduction

Software engineering can be minimally defined as the set of activities involved in developing, operating and maintaining software. Software engineering standardization is an attempt to integrate, regulate and optimize existing best practices and theories [Wang02]. Software companies use software engineering and software quality standards as the foundation of their quality assurance process or of their quality management system. On the other hand, innovation and change occur faster in the software industry than in any other industry. "Faster, better, cheaper" : people are expected to deliver products faster, of better quality, and at lower cost.

The experience presented in this paper attempts to reduce the gap between operating under the control of stable, well-defined and controlled processes and surviving in a rapidly evolving information technology industry. Faced with this challenge, each project should answer with a kind of continuous adaptation. Hence, the larger question of software process improvement is also addressed.

Quality management baseline is the set of process requirements, standards, corporate commitments and best practices controlling the software processes. The quality management baseline needs to be tailored to a project baseline. The tailoring usually begins at the tendering phase and is refined at the start of the project. Section 2 describes a project where strong tailoring was envisaged and defined at the beginning, but where it did not work as expected.

In section 3, we propose to consider tailoring software activities as a continuous task, managed by the project manager and supported by the whole software team. This adds a new dimension to the software processes we have called harrow processes, using the metaphor of a tine harrow from ecological and sustained agriculture. Tine harrows are used to control weeds; tines can be raised or lowered individually to cultivate specific areas.

Section 4 presents the framework and the projects where we are experimenting and validating the

continuous tailoring. Students in the French technological education system graduate after 4 years of studies which include at least three work placements. Thanks to a strong partnership with Thales Information Systems, students obtain a Master diploma within a software engineering apprenticeship by immersion system [RS03]. Young graduates are immersed in a team led by an experienced software manager and have to deliver real software products under the control of Tempo, the quality management system of Thales-IS. Continuous tailoring is applied at each phase of the project and some observations are presented in this section.

Continuous performance improvement is one of the main issues of ISO 9000:2000 as well as ISO/IEC 15504 (SPICE). In section 5, we try to place continuous tailoring in the light of continuous improvement.

2 Tailoring in practice

The authors both worked for nearly ten years at Thales Information System (formerly Syseca Inc), a software services company. We led projects and developed several management information systems under the control of TEMPO, Thales Information System corporate baseline. « TEMPO is the foundation of the company corporate culture and is the basis for sustaining good working practices. It is a set of procedures, guides and instructions defining how the company operates, and how it is organised, providing a framework for project management, software development and system integration activities [TIS02]». The quality system model adopted is the ISO 9001 standard. From a software development point of view, TEMPO defines a general lifecycle (requirements analysis, preliminary design, detailed design, coding and unit testing, integration and integration testing, software testing).

TEMPO needs to be tailored to each project. The methods and documents provided have a broad range. Tailoring is a vital component of the response to solicitation phase and early project phases. This tailoring process defines the activities to be performed and products to be developed and delivered. With the SPICE terminology, TEMPO is the company's set of standard processes, and tailoring TEMPO to a project results in its instanciated process. TEMPO is somewhat too general, and depending on the domain/size/methods/tools/... of the project, there is a lack of more precise and dedicated standard processes.

Some lessons learned

Ariane was the last project led by the primary author before leaving the Thales Information Systems company. Ariane processes were peer-reviewed by the second author. Ariane (which is operational and still growing) is a Management Information System for procurement by the textile distribution branch of several regional purchasing centres (each supplying around 40 stores). The solution implemented is based on an n-tier architecture, with a central database, local servers in the stores and tablet pen-based computers (without keyboards) radio-linked to the local servers [Sal98].

Oracle Designer was chosen in the perspective of a highly-generated implementation from models. Models are produced according to CADM, the companion method of Oracle CASE tools suite. Moreover, the information system should be commissioned with several stages through successive work packages. In terms of technique, development tools and organization, highly innovative and strong choices were made. As well-disciplined project managers, we started the project with the tailoring of the TEMPO baseline. We spent about a month writing the Project Plan, especially to define the managerial and the technical processes. The main difficulty was to tailor the software development process with a partial knowledge of the Designer tool and its associated CADM method, but without any real practice. The project team received several weeks of technical training. So, the project started with an optimistic software development process, in accordance with TEMPO requirements and intended to address two different issues :

- the definition of milestones and reviews in order to ensure contractual commitments (cost, lead-times and performance) and deliveries of work packages;
- the use of a model-driven method built onto an uppercase tool with each project engineer working on a different functional area.

The optimistic software development process is depicted in figure 1. Unfortunately, the envisaged

tailoring did not work. We returned from our incremental and iterative process to a traditional "V" model of lifecycle. We came back to a traditional client-server (and centralized) architecture. At the same time, we abandoned the generation of code modules (we kept the highly-generated process for the data) to go back to a traditional analysis and design phase followed by a traditional implementation phase using Oracle lowercase tools suite. The main reasons for this reversal were :

- we did not pay enough attention to technical architecture;
- we underestimated the constraints and the complexity of Oracle underlying framework;
- tailoring at the beginning requires real practice and at least one other similar experience;
- the model-driven process is more powerful but more difficult to master than a programming approach;
- the proposed process relies on the idea that it would be possible to change mindsets in one big step instead of building on a progressive and continuous sequence of steps in order to achieve progress.

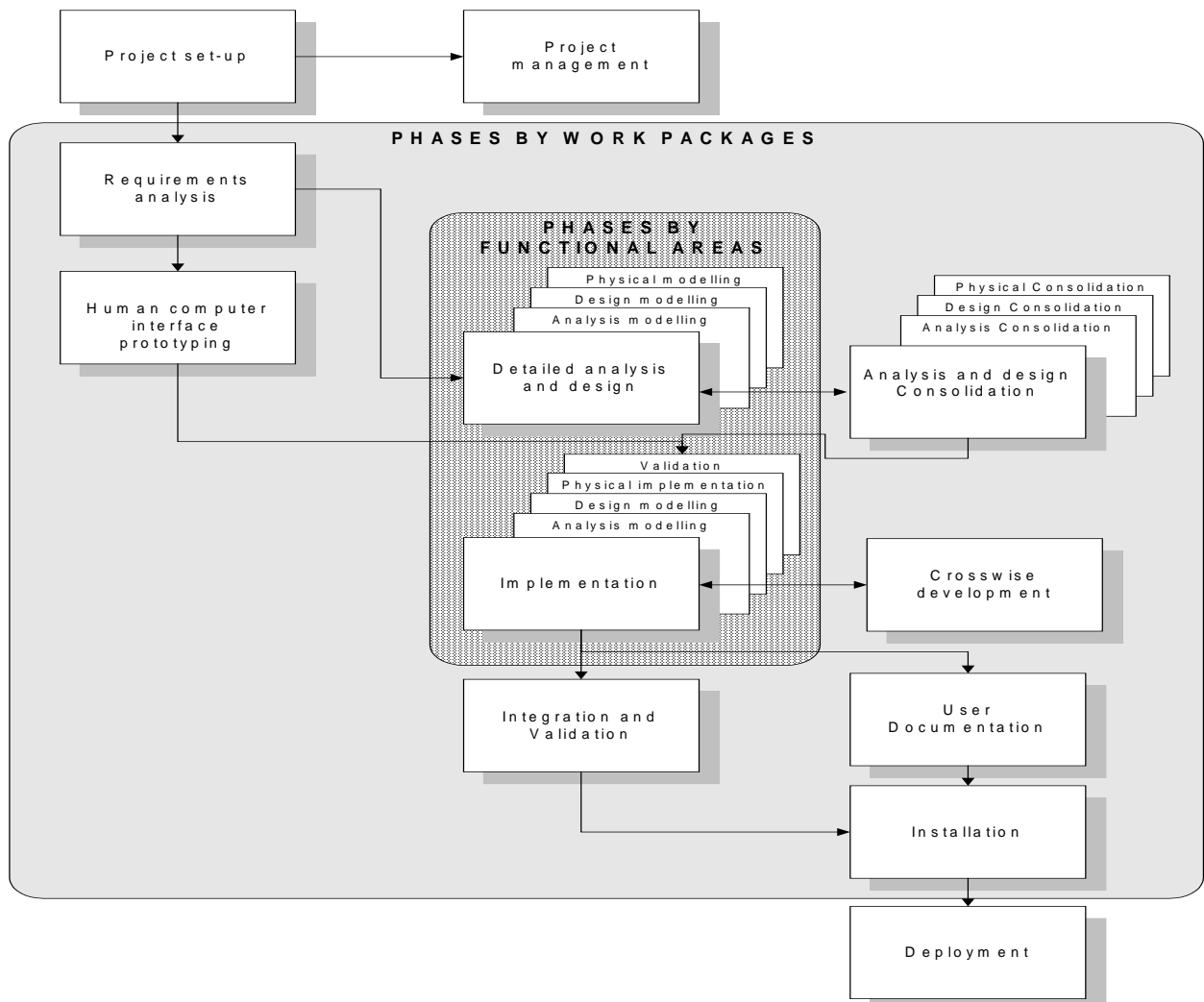


Figure 1 : Ariane envisaged development process

3 Continuous tailoring

In 2000, the primary author left his industrial post to join a professional education institute. The second author also works in this institute, and is a part-time consultant at Thales. We have kept a strong rela-

tionship with Thales Information Systems by providing it with consulting services on projects as well as methodological research on TEMPO baseline. We wanted to solve the identified problems within Ariane, following changes in technology (UML, n-tier architecture, complex frameworks and associated tools). It took us 6 months to build dedicated standard processes for current management information systems, including “on demand” tailoring activities. Then after experimenting with these processes during one year, we chose to structure these activities in a separate dimension called continuous tailoring or “harrow process”. One year on, we now believe that this new dimension (adding adjustable tines to suppress weeds over the desirable area) is essential to achieve modern management information systems requirements.

Standard processes for current management information systems

The starting point is TEMPO, Thales Information Systems baseline. In fact, we have built a specialization of TEMPO with the following constraints : a medium-sized team (from 3 to 8 people), a “Management Information System (MIS) orientation”, an n-tier architecture, an ISO 9001-like quality management system, a software development process belonging to the Unified Process family, a model-driven approach, uppercase tools suite and complex frameworks.

Our software development process relies on the truly specific features of the unified process that is use case driven, architecture-centered, iterative and incremental [JBR99]. Nevertheless, UML is used when it makes sense to do so. For example, analysis is carried out through a hierarchical functional decomposition rather than an object-oriented analysis.

We kept the main idea of the 2TUP (2 Track Unified Process) that is, to separate functionalities and technical architecture. « 2 Track literally means that the process follows two paths... At the close of evolutions concerning both functional model and technical architecture, system implementation consists in merging results issued from the two paths. This merging leads to a Y-development process [RV02]».

Continuous tailoring : a new dimension in project management

Innovation and change occur ever faster in the software industry. In a rapidly changing environment, pressure for change acts on at least four complementary lines :

- the team;
- software development process, methods and tool suites based on a model-driven approach;
- complex technical frameworks and n-tiered architectures;
- last but not least, the market place requires the delivery of products faster, at lower cost and with better quality.

Dealing with these orthogonal constraints generally falls to the project manager. This requires time and continuous effort. Unfortunately, these problems are rarely considered together, or if they are, only at the beginning of the project when most of the problems are unknown. Hence, the idea is to perform a “just-in-time” tailoring when the need occurs but as a planned activity. This adds a new dimension which plays a crucial role in dealing with technological and methodological evolutions as well as enhancing team performances.

The project manager will have to : 1) identify activities for which a previous tailoring is needed; 2) define as well as possible these activities dealing with cost and delay, anticipation (sow at the right time in order to harvest later), evaluate team skills and performances; 3) control and validate the activities results either alone or with help from peers.

4 Experimenting continuous tailoring

Experimentation framework

Students in the French technological education system graduate in 4 years, having completed at least three training periods in a firm. Thanks to a strong partnership with Thales Information Systems, we

had the opportunity to immerse young graduates in an imitation of the real world. The plan of action is built on a 6-month project. Young graduates make up two teams of 5; each team is led by one of us acting as project manager. A contract defines the customer-supplier relationship. The TEMPO specialization depicted in section 3 is the project baseline, including the continuous tailoring dimension. Each project team has its own office with individual working post and common installations. Each project team uses a different and complete software engineering tools suite (Oracle and Rational/IBM Websphere). At the end of the academic year, the team members obtain a Master's degree in Software Engineering.

The first year, we considered tailoring practices (activities) as part of the related software processes along with the other practices (e.g. "Adjust the design" with "Preliminary design"). The second year, we placed these practices in the harrow-dimension.

Definition of continuous tailoring

Tailoring practices (activities) as other practices are roughly-defined, estimated and scheduled in the project plan. Continuous tailoring applies to four process categories (in the SPICE sense) : Customer-Supplier (CUS), Engineering (ENG), Project (PRO) and Support (SUP). Each process category is a set of processes addressing the same general area of activity. Each process is a set of practices that address the same purpose [Spi95, part 9]. Tailoring practices refer to a process; they are intended to improve the process but mainly to contribute to the process realization, i.e. to achieve the process goals and to deliver required outputs.

Let us present a list of the tailoring practices, defined by the first author for his project, applied to the engineering process. The table 1 illustrates the new dimension added to ENG process category. The first column relates to the ENG main processes. The second column relates to the base practices of each process. The third column relates to the tailoring practices to be performed for each process. The inclination and variable size of the third column symbolizes the times of the harrow : times need to be adjusted in order to kill weeds over the desired areas and to the required depth.

Software development engineering		
Requirement capture	Functional requirements capture Technical requirements capture Requirements consolidation	Requirement capture definition and set-up
Technical architecture	Generic design Technical prototype	Technical architecture validation Development framework exploration
Analysis	Requirement analysis Human-computer interface mock-up	Analysis definition and set-up
Design	General design Detailed design Relational database consolidation	Design ajustement Database modelling and implementation guide
Coding - Unit testing	Coding - Unit testing	
Integration-Qualification	Software test plan elaboration Software test description elaboration Integration Internal validation Qualification	Corporate test procedures learning

Table 1 : Harrow-dimension of the Engineering process category

Performing continuous tailoring

The tailoring practices, as well as the base practices, are described in the project plan at an abstract level. Some well-known practices are detailed while others remain general. Before the related process starts, the software manager should write a work card describing in detail how to carry out the required tailoring.

Let us illustrate continuous tailoring practice on the design process. "The purpose of the Develop software design process is to establish a software design that effectively accommodates the software requirements; at the top-level this identifies the major software components and refines these into lower level software units which can be coded, compiled, and tested [Spi95, part 2]". The baseline offers a set of base practices intended to accomplish this goal. These base practices are described at an abstract level identifying "what" should be done without specifying "how". The main difficulty for a project is to be provided with a response to "how ?" suitable for the project specificities : technique, methodology, ...

When the development environments are stable and mature, the way to use it can be found "off-the-shelf" or at least progressively built from successive projects and capitalized in the corporate baseline. When methods and technologies are continuously evolving, the "software component" definition differs according to frameworks, models and tools. Then we need to adapt constantly the "how to design" issue to innovations and changes. That is precisely what the tailoring activities should provide answers to. For example, adjusting the design is aimed at understanding, preparing and defining the design model that should be used during the design phase. Thus, the instructions given in the "Design adjustment" card are intended to answer to these questions :

- which role does the design play in our software process ?
- what are the determining elements and relevant models useful in elaborating the design models ? why are they used ? how are they elaborated ?
- how do project design constraints (modelling language e.g. UML, process e.g. 2TUP, technical framework e.g. J2EE, ...) correlate with corporate baseline requirements ?

The approach envisaged to answer to these questions is to carry out the retro-design of the technical requirement prototype (a previous tailoring activity) used to validate the technical architecture. The output of this activity is a design document which follows TEMPO guidelines.

Benefits of continuous tailoring

Inexperienced graduates greatly benefit from these tailoring activities : very often, they found it difficult to start the related work and carrying out the tailoring product gave them the required boost.

The project manager has difficulties in conciliating production constraints (mainly cost and lead-time) with the need to survive in the market place. This requires constant adaptation to technological and methodological innovations and team competence enhancement.

The continuous tailoring dimension is a valuable tool of the project manager. It helps to materialize cost and lead-time of tailoring activities; to involve everyone in the team to the improvement process; to promote team members' creativity and self-training; to show improvement initiatives and results to the top-level managers.

5 Continuous tailoring and/or continuous improvement

ISO 9000 advocates the continuous improvement principle illustrated by the Deming wheel (Plan-Do-Check-Act) [Mit04]. ISO/IEC TR 15504 (SPICE) develops a 2-D process capability assessment model with both a process and a capability dimension including a level 5 : continuously-improving level. Both approaches rely on a quantitative understanding of process capability and management.

Continuous process improvement and continuous tailoring are two different but complementary approaches with the same goal. The main difference is that tailoring is undertaken while producing the software in order to deal with unanticipated project constraints while in process improvement, change is based on a quantitative understanding of the effectiveness of process changes.

The ISO 9000 continuous improvement principle supposes that there are no stable states for a quality management system while the harrow-dimension supposes that there are no stable states for an MIS development process. There are always ways of improvement intended to emphasize productivity as well as enhancing the quality of the products. Thus, there is a strong need to adjust the process while performing it.

In the software process improvement approach, discipline and creativity are often in opposition [CF02]. It is important to balance discipline and creativity, that is one of the goals of the continuous tailoring dimension : adopting software processes that can operate under constant change.

Continuous process improvement requires commitment from all the staff. Unfortunately, the effective processes allow many actors (developers, system administrators, ...) to stay outside of the management quality system. In our proposal, continuous tailoring can be triggered at each moment on each activity that leads to a natural commitment of the concerned actors.

6 Conclusion

Unfortunately, tailoring usually begins at the tendering phase and is refined at the start of the project. Moreover tailoring is usually carried out by the project manager and the software team is not involved. We argue that tailoring software activities is a continuous task, managed by the software manager and supported by the whole team. We believe that it is a good way to manage the complexity of change and innovation and to motivate a team in order to improve their processes and performance.

7 Literature

- [CF02] Reidar Conradi, Alfonso Fugetta, Improving Software process Improvement, IEEE Software – july/August 2002 , IEEE, 2002
- [JBR99] Ivar Jacobson, Grady Booch, James Rumbaugh, The Unified Software Development Process, Addison-Wesley Longman, 1999
- [Mit04] Henri Mitonneau, ISO 9000 Version 2000, Dunod, 2004
- [RV02] Pascal Roques, Franck Vallée, UML en action, Eyrolles, 2002
- [RS03] Vincent Ribaud, Philippe Saliou, Software Engineering Apprenticeship by Immersion, International Workshop on Patterns in Teaching Software Development, ECOOP'03, 2003
- [Sal98] Philippe Saliou, Ariane – gestion de l'approvisionnement textile du groupe Leclerc, Le Bulletin Technique – Vol. 6, Num. 3 , Thales Information System, 1998
- [Spi95] Software Process Improvement and Capability dEtermination (SPICE), Software Process Assessment - Version 1.00, <http://www.sqi.gu.edu.au/spice/docs/baseline>, 1995
- [TIS02] TEMPO, information systems development under control, Thales Information System, 2002
- [Wang02] Yingxu Wang, Software engineering standards : review and perspectives, Handbook of Software Engineering and Knowledge Engineering, World Scientific, 2002

8 Author CVs

Philippe Saliou

Philippe Saliou, 40 years old, B.S., M.S. and Ph.D. in Computer Science, has been an Associate Professor in the Computer Department of Brest University, France, since 2000.

Before entering the academic world, he spent thirty years in the software industry, essentially as an information system project manager. His last industry job was in UNICOPA, one of the leading agri-food co-operatives in France (4200 employees, 30 industrial sites, 18000 farmers), where he was the Computer Project Director. He spent more than eight years in Thales Group, a world leader in electronic systems serving aerospace, defense and information technology, where he led important projects such as the French Hydrographic and Oceanographic Institute information system, the display system of the Charles de Gaulle aircraft carrier, the textile procurement information system for the supermarket chain Leclerc (15 regional purchasing centers, 600 stores), etc.

He teaches software engineering, information systems through model-driven approach, software configuration management, requirements capture, software analysis and design and other software engineering activities.

His research interests include information system and software engineering. He is currently working on ways to improve processes by adopting software processes that can operate under constant innovation and change in a rapidly evolving technology industry. He is also interested in developing new pedagogical approaches for improving teaching and learning in software engineering education. Over the last two years (2002-2004) he designed and set up a new Master Degree in software engineering relying on an apprenticeship by immersion paradigm, in cooperation with Vincent Ribaud.

Vincent Ribaud

Vincent Ribaud, 45 years old, B.S in Mathematics, M.S and Ph. D. in Computer Science has been an Associate Professor in the Computer Department of Brest University, France, since 1994, and partly senior consultant at Thales Information System, a software services company .

He began as a lecturer at Rennes and Brest Universities before spending six years in the software industry, essentially as an MIS project manager and technical architect. Since 1996, he has led the software engineering research group at Brest University.

His research interests include software engineering, information system and generative design and programming. He is currently working on software process improvement and on semantic integration of software components.

He teaches software engineering, distributed systems and networking. Eighteen years of software engineering teaching has led him to co-invent (with Philippe Saliou) a new pedagogical paradigm called apprenticeship by immersion.

Merging Agile Methodologies The Case for DSDM and XP

*Andrew M Goulding and Rory V O'Connor
Dublin City University
Ireland*

Abstract

This paper presents the proposal that while the agile approach to developing software is proving to be a success, there is value in merging some of the best practices of more than one approach. In this paper, two specific agile approaches - Dynamic Systems Development Methodology (DSDM) and Extreme Programming (XP) - are examined. This paper focuses on the proposition that if an organisation has implemented one approach (DSDM) then is it possible to bring elements of good practices from another approach (XP) in order to add value to the overall process. The main objective of this research was to establish where common ground exists between these two approaches and to explore how they may best work together. A case study within an organisation was conducted to enable further investigation and to evaluate the underlying research proposal.

Keywords

Agile Methodologies, Extreme programming, DSDM, Process Improvement

1 Introduction

An organisation may be motivated to define and improve its software development processes due to a variety of forces affecting software development and maintenance. As a result there is a need for organisations to continuously explore new approaches to software development, which incorporate not just the development of new methodologies, but exploring new ways of using existing methodologies. DSDM and XP are both established agile methodologies. As a framework for business centred development, DSDM focuses very strongly on the full project lifecycle and describes an iterative and incremental process for software development in the context of new and established business processes, whereas XP has a narrower more rigorous software developer focus. At a simplistic level, DSDM can be viewed as a business focused systems development framework, where XP may be viewed as an approach to coding.

Although they are distinct approaches, DSDM and XP have much in common [1]. However, many see DSDM and XP as competitive methods for software development and therefore assess which methodology to adopt based upon the strengths and weaknesses as perceived from the firm's perspective, normally resulting in the selection of one method over the other.

This paper is focused on the proposition that if an organisation has implemented one agile approach then is it possible to bring elements of good practices from another approach in order to add value. Specifically, in this paper we will examine the case for two agile approaches – DSDM and XP. The main objective of this research being to establish where common ground exists between these two approaches and to explore how they may best work together. A case study within an organisation was conducted to enable further investigation and to evaluate the underlying research proposal.

1.1 Case Study Organisation

The organisation that participated in this research is a large Irish financial institution with an in-house software development department which employs approximately 250 software developers. The developed systems have been a major factor in the success of the organisation with the more traditional approaches to development projects being used. In the mid-nineties the need for change was recognised, driven by the fact that the existing IT approach was in danger of impeding rapid change. In answer to this an internal group composed of senior IT and business management along with an external expert advice conducted an investigation of industry best practices and the main outcome was the selection and implementation of DSDM. To date, DSDM has been a proven success within the company and has led to many implementations that have met business and IT requirements.

2 DSDM and XP – A Comparison

In general it appears that XP and DSDM share many of the same ideas on how projects should be run. XP attempts to simplify how teams work while also introducing some 'extreme' ways of coding and testing. Visibility and communication are also central to XP and these can only help ensure that the project is delivering what the customer expects. Many organisations have realised that XP and DSDM are not merely competitive methods for application development but can in fact complement each other [1]. The DSDM lifecycle puts more emphasis on the pre-project set-up in terms of feasibility studies and prioritisation. It also takes into account Implementation and Post-Project Reviews which are not apparent in XP. XP appears to focus more on the actual development phase once it is up and running but not really on what happens before or after this. XP has been described as being too light on project controls where this is a known strength for DSDM. It would appear that combining the two would give a controlled framework with robust programming practices [1].

Based on comparisons made [2] the following show how XP practices could be practically applied to a

DSDM project:

- User stories can be used for requirements gathering, estimation and for building test cases. This is a relatively easy concept to introduce with minimum preparation and training required. User stories to be publicly displayed within the project team for constant reference especially during stand-up meetings.
- Stand-up meetings can be used for discussions between business users and developers – user stories to be used as the main focus for these discussions.
- Simple design and re-factoring to be applied where possible. This can be employed regardless of whether DSDM or XP is used.
- Pair programming to be used as a means to improve code quality and enhance developer knowledge transfer. This will challenge traditional thinking but once the improved quality is proven then it could become an accepted development technique.
- Test-driven development to be used with suitable tools to be made available, where possible, to the developer. This will require a significant amount of training for the developer and an acceptance that improved quality is the output leading to less user acceptance testing time required. Test-first approach for user testing to be used as standard.
- Continuous integration can be implemented but at a cost to the business in terms of test hardware and time taken to re-build the test system and re-run tests on a regular basis.
- Collective code ownership has been around in different forms. The ability to make this work may depend heavily on the environment and the types of source code management tools in operation. The concept should be considered for each individual project.

3 Case Study

This project was the development of a new system to track the activities of the administration section of a business area where requests for action are taken on electronic forms and processed on a daily basis. The main driver for this new system was due to the fact that the business department lacked management information about the volumes and the time taken to process these requests as well as the information on the error rate during the processing of these requests.

Prior to starting the pilot project a set of three focus group sessions were conducted within the organization to present and discuss ideas on implementing XP practices within the DSDM environment and to gain views all stakeholders. These focus groups were chaired by one of the researchers and included senior management, development staff, a DSDM consultant (DSDM consortium) and an experienced XP developer. All stakeholders agreed that user stories, refactoring, test-first design, continuous integration and pair programming would be evaluated during a pilot project as additional practices to the DSDM approach. The idea of fully utilising automated testing was put aside, and the XP approach to testing using the normal execution of tests for both unit and acceptance testing.

The agreement was to use the following XP practices in addition to the normal DSDM process:

- The planning game - User stories were to be used during the requirements gathering phase. They were to be used for stand-up meetings and as a means of communication, feedback and build of test cases, with developers choosing user stories and estimate effort per story
- Test-First Design - Business testers to take the user stories and build acceptance test cases before development and developers to build unit test cases before coding.
- Refactoring - Developers were to use refactoring during project where possible or sensible, with continuous Integration and daily integration of new code.
- Pair Programming - Developers were to work as a pair.

3.1 Project Phases

- Design Phase - The project then began with the initial design workshop involving all stakeholders, where roles and responsibilities were defined as per DSDM. The main outputs of the were the story cards and these were the focus for discussion throughout the project. User stories were then used for requirements gathering and were used as part of a prioritisation session, with the MoS-Cow approach for prioritisation being used. Time-boxes were agreed- two time-boxes with three weeks per time-box with a live delivery at the end of each time-box.
- Development Phase - The design and coding phase began with agreement between the developers on ownership of story cards, which led to a clear set of objectives for both developers with the more experienced developers taking ownership of the more complex functions. As test-first development was to be used the programmers firstly wrote unit test cases for each user story or function to be written. Pair programming was the agreed format for coding and the developers set up with this in mind. All work was logged to the usual system for recording project time and this would be used to determine actual effort against planned effort for all phases of development. At the end of each working day all code developed was merged with previous latest version and the test system re-built.
- Test Phase - The test coordinator also took the story cards to create the user acceptance test cases in advance of the development. This led to discussion, particularly at the early stages, on the clarification of the requirements and the system functions. Traditionally user acceptance test documentation takes place independently of the development team and is based on the requirements specification which is usually contained in one single document. The significant change here was that the individual user story cards were used as opposed to the requirements specification and was carried out before any development began. As automated testing was not available for this project the user testing took place in a traditional manner.

4 Results and Analysis

The case study was completed with a post-implementation review. Key issues identified were:

- Project initiation and definition of roles and responsibilities took place as per normal for a DSDM project. Resources were assigned as required with full commitment from all stakeholders to the project plan.
- Time-boxes were decided up-front and delivered on time (two time-boxes with 3 weeks per time-box where a live delivery took place for each one).
- Priority needs were delivered early in first release with the reports coming from this phase being produced for the department some 6 weeks after project initiation.
- Requirements were fully captured – primarily during the initial workshop and a small number of changes were made subsequent to this. Subsequent prioritisation of requirements did not change significantly from the decisions at the workshop.
- Development and test phases ran to the expected time-scales with no significant overhead seen from the use of XP.
- User acceptance testing sign-off was achieved for both time-boxes in a shorter period to the planned time-frame.

The use of user stories during this project was considered successful. In particular it made the requirements gathering easier for both customer and developers and brought a personal touch to requirements, in that business users felt that they owned that story. Gaps and complexity were very clear in when stories were on a board and everyone could see them and all parties involved felt that this was an improvement from the DSDM approach where flip-charts of requirements were not as easy to view. Also prioritisation was easier to achieve as cards could be moved around without difficulty. User stories were used for stand-up meetings and as a means of communication, feedback and build of test cases, which was accepted by all involved as an improved way to discuss requirements

and issues compared to the normal DSDM approach where reference to requirements specifications was not so regular or easy. Accordingly user stories appear to work better than the DSDM approach. It is worth noting that in the pilot project there was no obvious improvement in requirements coverage as compared to previous DSDM projects.

Developers worked as a pair constantly throughout project as opposed to DSDM where developers work independently. The developers had no issues in working together though they commented that a strict time schedule needed to be enforced to ensure hand-over or there was potential for one person to dominate the coding. The cost of pair programming was calculated to be approximately 30% longer to code compared to the normal output. This was based on the estimate which was originally derived in the normal way as against the time recorded on the tracking system used for all IT projects. However, complexity becomes a factor here as complex issues were definitely resolved faster (developers estimate that solutions were found twice as quick as in the normal scenario). Given that approximately 20% of the code was complex then the actual increase in effort works out as 17% effort in total extra effort regardless of the size of program.

Test first design was split into two parts with the programmer writing unit tests before coding and the business tester writing acceptance tests as soon as possible after the requirements are identified. Test-first improved communication during the project, with business and IT staff working much closer than in traditional projects. The developers build unit test cases before coding and found that they could code with more confidence once this had happened and effort improvements would certainly be realised. It was evident that effort was not significantly higher for actual set-up of unit tests but there was a reduction in time taken to the overall development and unit test phase together compared to a traditional project of similar size. The overall reduction in defects through pair programming helped reduce user acceptance test effort significantly. Generally in the organisation experience has shown that 40% of all changes made after delivery to test are for bug fixing – a greater proportion by some 20% than for requirements changes. In this case a reduction in bug-fix testing was seen compared to a traditional project with 57% less defects identified after unit testing.

5 Conclusions and Discussion

The main finding of the research and pilot project is that the XP practices can truly add real additional quality assurance to the lifecycle of a project, regardless of what other methodology is being used. In this case, during the pilot project in the organisation, the benefits of DSDM and XP are realised with DSDM continuing to bring a strong framework and ensuring that the requirements are found and met. XP development and testing practices have helped achieve both improved quality and introduced some potential around effort savings when pair programming and automated testing are utilised.

The research presented in this paper is primarily based on the experience of a single case study experiment carried out within a financial institution. While the findings are positive and show good promise, the small scale of this experiment must be acknowledged. A larger scale investigation would need to be undertaken to validate the results of this early study.

With the recent launch of DSDM version 4.2 – which contains guidance for those wishing to use XP in conjunction with DSDM - the DSDM consortium have acknowledged the need to explore the potential for combining these two separate methodologies. However at this early stage there is a lack of any real experience of using XP and DSDM.

6 References

- [1] DSDM and Extreme Programming, Knowledge Sharing Space article, DSDM Consortium. 2002. Available from <<http://www.dsdm.org/kss>>
- [2] Goulding, A. Merging Agile Approaches: The case for DSDM and XP, Thesis, Dublin City University, 2004.

7 Author CVs

Andrew M Goulding

Andrew Goulding is an IT Manager with a large financial institution based in Ireland. He has over 15 years of experience in information systems development. Andrew received a B.Sc. in Computing and Mathematics from University College Dublin and an M.Sc. in Computer Applications from Dublin City University.

Rory V O'Connor

Dr. Rory O'Connor is a lecturer in the School of Computing at Dublin City University. He received a PhD. in Computer Science from City University (London) and an M.Sc. in Computer Applications from Dublin City University. He has previously held research positions at both the National Centre for Software Engineering and the Centre for Teaching Computing, and has also worked as a software engineer and consultant for several Irish and European technology organisations. His research interests are centred on the processes whereby software intensive systems are designed, implemented and managed, in particular, methods, techniques and tools for supporting the work of software project managers and software developers in relation to software process improvement, software project planning and management of software development projects. Dr. O'Connor is a member of the Irish Software Engineering Research Consortium (ISERC). He can be contacted by email at: roconnor@computing.dcu.ie

User Expectations for an SPI Knowledge Management System

Ilmari Saastamoinen⁽¹⁾, Markku Tukiainen⁽¹⁾, Timo Mäkinen⁽²⁾, Timo Varkoi⁽²⁾

⁽¹⁾ University of Joensuu, Finland

⁽²⁾ Tampere University of Technology, Pori, Finland

Abstract

Software process improvement (SPI) work has a positive effect on customer satisfaction, competitiveness and product quality of the software industry. To successfully achieve such goals SPI work requires appropriate knowledge and experience. As a part of SPI, the current state of the software process and a number of improvement needs are results of software process assessments. After an assessment improvement actions should be defined and prioritized. The improvement decisions should be based on the business goals and characteristics of the organization (e.g. clients, type, size, domain, organizational culture, current processes and their capability). However, it is difficult and time-consuming to choose appropriate improvement actions. The organization might not have earlier experience in the area of SPI and thus external support is needed. In many cases a proper software tool could assist organizations in their SPI decisions. We examined software enterprises' expectations for such a tool with the help of inquiries and a prototype of Gnosis, which is a software tool for SPI knowledge management.

Gnosis aims to support a software organization's improvement work in solving problems, analyzing and prioritizing the improvements, typically after the assessments. The purpose of the tool is to support the transfer of SPI knowledge and best practices into the software companies. Intended users of Gnosis are quality managers, project managers and consultants.

Two trials will be organized to examine Gnosis and the first one has already been completed. The goal is to evaluate whether the approach taken in Gnosis is understandable and usable to the envisioned industrial users and to test the relevance of the SPI knowledge available in the database. Eight Finnish companies from the regions of Pori and Joensuu participate in the trials. The participating companies are small and medium sized enterprises (SME) from different sectors of the software industry.

Usability of any software tool is essential for its success. A tool like Gnosis must support e.g. a SPI manager of a company to solve problems in prioritizing improvements of the company's software process. In this sense, the usability of the tool covers both the support for the SPI implementation and the tool's user interface. As a part of the first trial we gathered information on these areas with a semi-formal usability study. Before the second trial period both the knowledge content and the technical solution of the Gnosis prototype will be developed further based on the results of the first trial. The second trial will include independent use of the tool in the participating organizations and will provide practical evidence of the usability and usefulness of the Gnosis tool in solving SPI related problems in real software development environments. The project is planned to be finished by the end of year 2004.

In this paper we take a look at the Gnosis approach to SPI knowledge management, present the process for gathering information in the trials and, as the result of the first trial, describe the expectations of software SMEs for SPI knowledge management system.

Keywords: software process improvement (SPI), knowledge management

1 Introduction

Software process improvement (SPI) effects on customer satisfaction, competitiveness and product quality of a software organization. To success in achieving such goals, appropriate knowledge and experience is required in the SPI work. A part of SPI is the software process assessments, which result in determining the current state of the software process and a number of improvement needs. Next, improvement actions should be defined and prioritized. The decisions should be based on the business goals and characteristics of the organization (e.g. clients, type, size, domain, organizational culture, current processes and their capability). However, it is quite difficult and time-consuming to choose appropriate improvement actions. The organization might not have earlier experience in the area of SPI and thus external support is often necessary. In many cases a proper software tool could assist organizations in their SPI decisions. We examined software enterprises expectations for such a tool with help of inquiries and Gnosis, which is a recent software prototype for SPI knowledge management.

The paper is organized as follows: Chapter 2 introduces the Gnosis approach to SPI knowledge management. In Chapter 3 we explain the information gathering process used in the trial and in chapter 4 we present the results from the first trial. Finally, in chapter 5 we draw the conclusions and present future directions for the Gnosis project.

2 Gnosis Approach to SPI Knowledge Management

Gnosis aims to support an organization's improvement work in problem solving, analysis and prioritization, typically after the assessments [6]. The purpose of the tool is to support the transfer of SPI knowledge including best practices into the software companies. Intended users of Gnosis are quality managers, project managers and consultants. Gnosis has a database, which contains information about process reference models and best practice solutions applicable to actual improvement cases. A graphical user interface (GUI) allows a user to search and browse the database. A search results in SPI knowledge relevant to the context of the organization's software development.

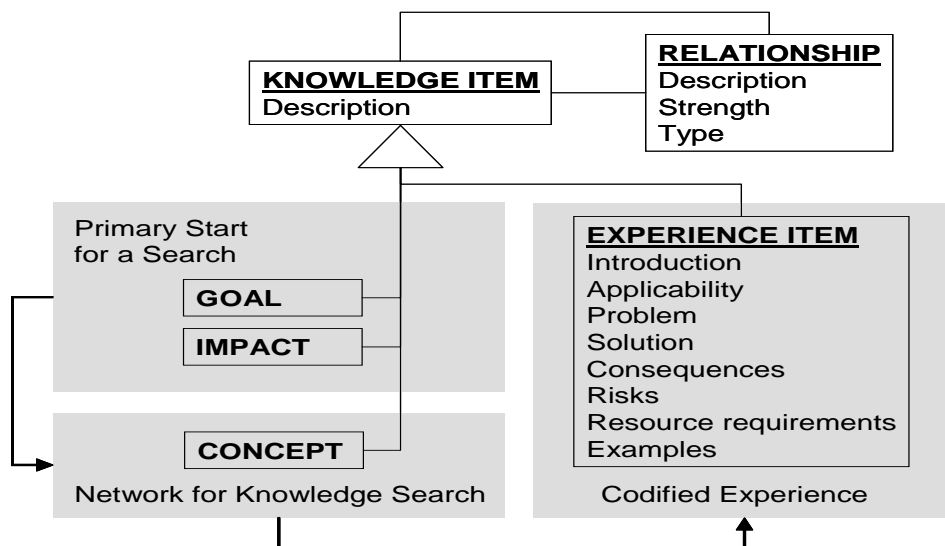


Fig. 1. The conceptual structure of Gnosis.

Gnosis knowledge base has four different types of knowledge items which relate to each other (Fig. 1): Goals, Impacts, Concepts and Experience Items [10]. *Goals* represent overall business goals of an organization. *Impacts* represent concrete results of the SPI work or the desired changes the company wants to implement in their organization. *Experience Items* represent codified expert knowledge giving

solutions to the recurring problems in the SPI work. These solutions are extracted from the practical experience of people working in the area of SPI or from the software engineering literature, e.g. extreme programming [1]. *Concepts* represent elements found in software process models like SPICE [4, 5] and CMMI [2], which represent generally accepted software process improvement knowledge today. The concepts with their relationships are based on conceptual analysis of the models [7, 8]. The structure of an experience item is derived from the patterns approach [3], which is broadly used as a device to share knowledge about software design. In Gnosis the concepts form a network, which is used to search experience items to assist in solving problems. In addition, the concepts can provide relevant information in problem solving.

3 Evaluation Process

Our approach for SPI knowledge management is empirical thus we have to carefully validate our work. The evaluation process takes place in two trials of which the first one has already been completed. The goal of the trials is to examine whether the approach taken in Gnosis is understandable and usable to the envisioned industrial users [10]. The purpose is also to test the relevance of the knowledge obtained as the result of the performed queries. Eight Finnish companies from the regions of Pori and Joensuu participate in the trials. The participating companies are small and medium sized enterprises (SME) from different sectors of the software industry.

The primary purposes of the first trial phase (Fig. 2) were to focus in studying appropriateness of the knowledge representation approach of Gnosis, to find out usefulness of Gnosis approach from the view point of an end user, and to test usability of the Gnosis system. The first phase consisted of two sessions. On the first visit to the company (Session 1) the focus was in gathering general background information of the company including its SPI activities, and to collect specific knowledge of the two processes selected for closer examination: customer requirements elicitation process and testing process.

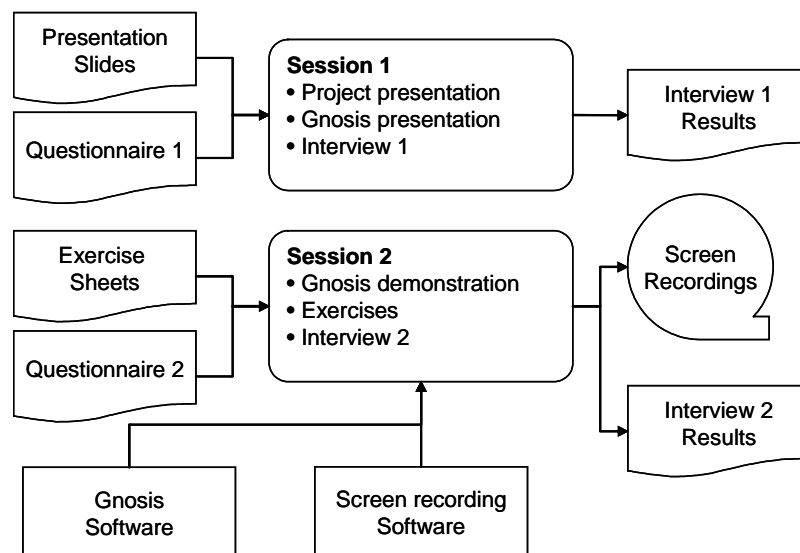


Fig. 2. The process for the first trial

On the second visit (Session 2), the focus was on the usability evaluation of the Gnosis tool. It consisted of a Gnosis tool demonstration, two problem solving exercises, and interviews of the participants. During the use of Gnosis we collected two protocols of usage for further analysis. First was the Gnosis tool output display and the input actions (keyboard and mouse clicks) of the participants and

the second was the voice protocol with microphone connected to a recorder. We used "think-aloud"-method to collect participant's utterances while using the Gnosis tool. After the exercises we interviewed the participants for their subjective experiences of using the Gnosis tool. The questions were directed to three different areas: first, the general approach of the Gnosis tool to help in SPI problem resolution; second, the relevance of the Gnosis knowledge base query results to the company's SPI situation; and third, the overall usability of the Gnosis tool and its user interface (Questionnaire 2).

4 First Trial Phase Results

This chapter presents the results from the first trial. The chapter is divided into three parts, the first presents the participants' opinions about Gnosis support for organization SPI work, the second discuss the relevance of the system's knowledge content for SPI and the third reveals the usability issues of the Gnosis tool. The results are mainly based on the interview done after the session 2. The questions are presented in the Appendix.

Gnosis support for the SPI work

In the first question of the interview we inquired, does the Gnosis approach support the organization's software process improvement work. The overall result was that the idea of Gnosis seems appropriate and it can support the SPI work of the company. Every interviewee answered affirmatively to this question. Especially Goal/Impact-search of the Gnosis system got much positive support. Trial users considered the possibility to start knowledge search from high level goals via impacts towards concrete process improvement actions very useful. It helped the users to see the big picture of their SPI situation, e.g. how experience items are related to the higher level business goals and impacts. Participants also think that Gnosis is an interesting tool which can accelerate SPI problem solution in certain cases. It was also noted that the tool is able to support organization's quality group/persons in the process improvement work, but it doesn't support for the software development work during a project. Our main intended end user group is quality personnel of the organizations.

Next we asked, could the participant ask appropriate questions and does the Gnosis tool support making questions. According to the participants' comments, it was difficult to find right keywords for the search input and they also considered that keywords had large impact on search results. The terminology of the system and its knowledge content were considered to be a little unfamiliar by couple of the participants. To solve this problem some participants used Goal and impact search-function to resolve the right keyword and after that they continued the search with keywords. Almost every participant considered the Goal/Impact-function very useful and natural.

The third question considered the relevance of the search results from the company's point of view. Generally the results were considered relevant although in some cases too general. According to the participants the order of the Gnosis search results is reasonable although some expostulations were expressed.

Relevance of the Gnosis knowledge base

In general, the knowledge content was considered to be relevant and useful for the SPI work. During the trial every participant stated that the knowledge content of the search results were relevant to the software process improvement target under examination. The experience items got mainly positive feedback and the level of the knowledge in most of these knowledge items were evaluated appropriate and useful. Also the structure of the items was considered good but according to the some of the participants some of the items were too generic. Feedback on process descriptions of the reference models varied more. For example, the knowledge content of the process descriptions and other concepts were considered in many cases to be too short and abstract. However, the knowledge content related to the international standards (e.g. SPICE and CMMI) was considered to be very useful. Also, the need for a cost/benefit –type attribute for the experience items was mentioned more than once. It was also noticed that in this phase of the system and knowledge development it is very difficult to assess relevance of the Gnosis knowledge because the amount of the knowledge is very limited. One of the participants also pointed out that in the future, when knowledge content will increase, it will be more and more difficult to find right solutions in the organization's context.

Overall usability of the Gnosis tool

According to the trial participants, the user interface of Gnosis works but the graphical outlook of the system is poor. We got plenty of ideas how to improve graphical design of our tool. The basic search-functions of the tool achieved mainly positive feedback. However, also alternative ways to search knowledge were presented. Participants hoped, for example, some kind of a visual browser for the knowledge content. With a visual browser it could be easier to navigate between knowledge items and to understand how the knowledge items are related to each other. We have already added this kind of a visual browser to our system and it will be tested and evaluated during the second trial. Some of the participants also hoped that questions could be presented in an interrogative form. This was tried in the Patterns project [9] and at the moment we don't support this feature. The basic keyword search and the idea of a goal/impact based search got support from the trial users. The trial users also wanted some on-line help function.

5 Conclusions

The feedback from the first trial was positive and encouraging. According to the participants the idea of Gnosis seems appropriate and it can support SPI work of the company. Gnosis can assist and accelerate SPI problem solution and help users to see the big picture of their SPI situation, e.g. how experience items are related to the higher level business goals and impacts. The knowledge content of the system was also considered relevant and useful for software process improvement, but in some cases too general. The user interface of Gnosis works, but the graphical outlook of the system needs improvement.

For the second trial period the knowledge content and technical solution of the Gnosis prototype is developed further based on the results of the first trial. The second trial will include independent use of the tool in the participating organizations as well as joint sessions with researchers to provide practical evidence of the usability and usefulness of the Gnosis tool in solving SPI related problems in real software development environment. The project is planned to be finished by the end of year 2004.

Literature

- [1] Beck, K. "Extreme programming explained: Embrace change," Addison Wesley Professional, 1999.
- [2] "CMMI for systems engineering, software engineering, integrated product and process development, and supplier sourcing (CMMI-SE/SW/PPD/SS) Version 1.1," Software Engineering Institute, Carnegie Mellon University, 2002.
- [3] Gamma, E., Helm, R., Johnson, R., Vlissides, J. "Design patterns: Elements of reusable object-oriented software," Addison Wesley Professional, 1994.
- [4] ISO/IEC CD Ballot 15504-5: Information technology – Process assessment – Part5: An exemplar assessment model, ISO/IEC, Jul 2003.
- [5] ISO/IEC IS 15504-2:2003, Software engineering - Process assessment - Part 2: Performing an assessment, ISO/IEC, 2003.
- [6] Haavikko, M., Johansson, M., Mäkinen, T., Nevalainen, R., Saastamoinen, I., Tukiainen, M. and Varkoi, T., "GNOSIS - An Approach for SPI Knowledge Management," Proceedings of the 4th International SPICE Conference on Process Assessment and Improvement (SPICE 2004), Lisbon, Portugal, pp. 91-99, 2004.
- [7] Lepasaar, M., Mäkinen, T., Varkoi, T. "Structural comparison of SPICE and CMMI," in *the proceedings of SPICE 2002*, Venice, Italy, March 2002.
- [8] Mäkinen, T., Varkoi, T., Jaakkola, H. "Database implementation for a software process assessment model," in *Papers presented at PICMET'99* [CD-ROM], eds.: D.F. Kocaoglu, T.R. Anderson, D.Z. Milosevic, K. Niwa and H. Tschirky, Portland, OR: PICMET, July 1999.

- [9] Patterns project, "Patterns to Adopt Knowledge Based Solutions to Software Management Problems, " IST-2000-25120, An EU Project within the Information Society Technologies programme under the RTD Spanning Key Action II (II.2.2 "Smart"Organisations) <http://www.esi.es/en/Projects/Patterns/patterns.html> (accessed Nov 2003).
- [10] Saastamoinen, I., Tukiainen, M., Mäkinen, T. and Varkoi, T., "Evaluation of a Software Tool for Knowledge Management in Software Process Improvement," Proceedings of the Portland International Conference on Management of Engineering and Technology (PICMET'04), Seoul, Korea, CD-ROM, 2004.

Appendix: Trial Questionnaire 2

Questionnaire 2

Relevance of the overall approach

1. Is the approach compatible with your company's SPI principles?
2. Are you able to do right queries?
3. Are the answers of Gnosis meaningful from your company's point of view?

Knowledge content

4. Is the knowledge necessary for SPI?
5. Is the knowledge correct?
6. Is the knowledge too specific or too generic?

User interface

7. How do you comment the user interface?
8. Is it easy to execute queries?

Interviewee's questions

9. Do you have anything to ask or any comments?

Gnosis use cases

10. How would you utilize Gnosis in your company?

Stages of Software Process Improvement Based on 10 Year Case Studies

Miklós Biró¹, Katalin Balla², János Ivanyos³, Richard Messnarz⁴

*¹Budapest University of Economic Sciences and Public Administration,
Veres Pálné u. 36. H-1053 Budapest, Hungary*

miklos.biro@informatika.bke.hu

²IOSYS, Csata utca 8, H-1134 Budapest, Hungary

³MEMOLUX, Thököly ut 137, H-1146 Budapest, Hungary

⁴ISCN, Florence House, 1 Florence Villas, Bray Co. Wicklow, Ireland

Abstract

This paper intends to give account on successes and difficulties encountered over the last decade by two Hungarian companies with strong commitment to software process improvement. The cases are positioned with respect to milestones of software process improvement initiatives world-wide and in Hungary, and the conclusions are globally valid despite of the differences in the current environments. The three distinguished stages of software process improvement are: "Awareness of process capability weaknesses", "SPI and ISO 9000 certification", and "Back to the business". And the latter one turns out to be one of the gateways to level 4 of the emerging ISO/IEC 15504 standard.

Keywords

business, case study, SPI, ISO 9000, Bootstrap, CMM, CMMI, ISO/IEC 15504, Balanced IT Scorecard, eSourcing Capability Model

1 Introduction

The software process movement was initiated in Hungary through the Hungarian membership in the Bootstrap Institute from 1993. The significance of this step marked by the publication of the paper describing the European Software Quality Network [5] is highlighted in both of the cases discussed in this paper. In addition to actual process assessments and improvement action planning, the message was conveyed in numerous presentations and publications in Hungarian language as well [6]. The BOOTSTRAP methodology was developed in the framework of the EU ESPRIT project 5441 and combined the following approaches: Software CMM, ISO 9001/9000-3, European Space Agency Software Engineering Standard PSS-05-0, and DoD-STD-2167A.

A parallel movement also rooted in TQM and taken into account in Bootstrap was the ISO 9000 series of standards first published in 1987 followed by Guidelines to the Development, Supply, and Maintenance of Software in 1990. ISO 9000:1987 focused on quality control via retroactive checking and corrective actions, ISO 9000:1994 emphasised quality assurance via preventive actions, while ISO 9000:2000 makes expectations of continuous process improvement and tracking customer satisfaction explicit returning to the common TQM roots and converging to CMM [7].

2 Awareness of Process Capability Weaknesses

In this section, the two companies providing the case studies are briefly introduced. Their core businesses are different, software development is however an integral part of their activities. Their common characteristic is the commitment to software process improvement which was manifested by their pioneering demand for the most advanced approaches corresponding to their maturity level and to the global state-of-the-art.

2.1 IQSOFT/ IQSYS

IQSOFT Ltd. (IQSOFT Intelligent Software. Ltd.), predecessor of the current IQSYS Ltd., was one of the main representatives of the software industry in Hungary. The company was formed in early 1990 from part of a large state organisation, the Theoretical Laboratory within the Computer Technology Co-ordination Institute (SZKI). IQSOFT joined the KFKI group in 1999, and, in March 2003 it became a constituting part of the newly established software development company, IQSYS. The company is a medium-sized one, having at present 150 employees. IQSYS kept the main software activity types of the forming companies: software development (mainly in a database environment, using 4GL development tools), software integration, and software implementation. The interest for software quality and software process improvement has also survived the reorganizations.

Since 1993, efforts have been made at IQSOFT to develop and introduce an internal Quality Management System (QMS). The basic reason of IQSOFT's management decision was the emerging request to be ISO-certified, formulated by a (potential) foreign customer. At that time, no Hungarian software company was registered according to ISO 9001, so it would have been a strategic advantage to be the first registered Hungarian software company.

In 1994, the overall company and two concrete projects were assessed according to the BOOTSTRAP methodology briefly described above. The results of this assessment pointed out that the whole organisation was situated on level 1.25 in the CMM model, while some of the activities (such as detailed planning, project management and usage of development model) turned out to be above level 2. Due to an insufficient understanding of the interconnections between CMM and ISO 9000, IQSOFT was unable to use the assessment results in an appropriate way. The company had to accept that the efforts made in building and introducing a QMS at IQSOFT [1], [2] did not produce the desired results. By the beginning of 1994, all basic quality documentation had been finalised, but remained almost unused, since management did not make their usage mandatory. Though the documentation set was ISO 9001-conform, there was a huge gap between ISO 9001 prescriptions used in the documentation and real life practices. IQSOFT believed at that time that the main reasons for this slow adaptation of new procedures were both the diversity of projects (IQSOFT's projects were difficult to standardise) and the human factors dimension (software developers were resistant to accept standard prescriptions).

2.2 MEMOLUX

Memolux, established in 1989, is a Hungarian private SME company with professional experience as a service provider in finance and public accountancy, management organization, software development and information system engineering. In Hungary, Memolux is ranked after the "Big Four", the four greater advisory firms in public accountancy. Memolux is a member of several economic chambers (AMCHAM, BCHH, CCCH). The payroll and accounting service lines represent the biggest one in the Hungarian market provided by an independent Hungarian SME with about 150 clients.

Since MemoLuX Ltd. was steadily growing and was managing larger and larger and a greater number of projects, MemoLuX had to achieve better control of the software development process, a quantitative view of the production process, and higher credibility among Hungarian and EU customers through an improved BOOTSTRAP maturity profile and an increased level of compliance with ISO

9001.

Memolux was the prime user and contractor of the PASS ESSI PIE (European Systems and Software Initiative Process Improvement Experiment) which was the first Central and Eastern European Framework Programme project directly and fully supported by the European Commission under the ESPRIT Programme EP21223 [8]. The maturity level aimed in the project was 3, fairly high on the international scale. The progress was monitored according to a well-planned quality measurement process and corresponding tasks were scheduled as separate workpackages.

A BOOTSTRAP assessment was performed in order to clarify the strengths and weaknesses in the software development process at MemoLuX. This gave the following conclusion. MemoLuX's maturity level was 2.5 in the SPU (Software Producing Unit) global environment and the same score was measured in the selected SPU project. Looking at the status of software engineering practices at MemoLuX, the quality management of IT projects needed enhancement. Improvement steps, to be carried out first with the guidance of the BOOTSTRAP consultant, were planned.

3 SPI and ISO 9000 Certification

The Software Process Improvement movement gained momentum in Hungary around 1996 when the Hungarian version of ISO 9001:1994 has been published under the name MSZ EN ISO 9001:1996. Significant SPI related events [9] were organized by the Software Quality Management Division of the Hungarian John von Neumann Computer Society which was established in this year (<http://members.iif.hu/birom/neumann/eng/index.html>), as well as by the Information Technology Foundation of the Hungarian Academy of Science which played an important role in the introduction of ISO 9000 certification requirements in public procurement bids. Many internationally recognized certification bodies established in Hungary, but only few of them were ready to audit software organizations in Hungarian language. However, they quickly trained their auditors [10], and by the end of 1996 there were 4 software organizations certified according to ISO 9001.

3.1 IQSOFT/ IQSYS

Experiencing the "failure" of the original quality-exercise, the management and the employees of IQSOFT became sceptic about the possibility to improve daily practices. The customer requiring ISO 9000 certification disappeared, and during 1995 there was no concrete idea about what the next step in quality management should be. Quality activities were not stopped, but became a kind of background activities limited to research in the domain.

IQSOFT *felt that* some important elements were lacking. It became obvious, that a QMS, even a structured system-development or project management methodology can not be introduced at once at IQSOFT. It also became obvious that first of all organisational and working-style changes were required at the company, as a first step towards quality management.

The company came to the idea that a software QMS will not be really operational and useful if we would take into account the ISO 9000-prescriptions alone.

So, also taking into account the results of the BOOTSTRAP assessment, IQSOFT's management decided to use the opportunity offered by an EU PHARE tender to introduce a project management methodology. Taking into account the former co-operation with Lucas Management Systems, the PM² methodology of this company, distributed by Metier Plancon (NL) had been chosen for supporting the project called IQPM² [3], [4].

The project started in February 1996 and finished in May 1997. The basic phases were: awareness and requirements gathering, development of project management standards and procedures, solution definition and implementation of a software tool, testing the developed standards and procedures, measuring and evaluating the results, improving the standards and procedures according to the results obtained in the baseline projects.

We can state that the project was successful: it reached its goal within the planned time and budget limits. A key factor in the success was the very strong support of management, which attached a high priority to the project. The employees were kept informed during the whole project about its evolution. Working together with many employees in defining the needs, in putting things on paper, in trying out the newly written prescriptions in real projects contributed to the decrease of the previously hostile or passive attitude towards all quality-issues at the company. People saw that it was possible to work out a set of prescriptions that were really helpful for their work [14, 15].

During this phase 3 it became obvious that there were serious communication problems within the company (e.g. employees were not aware of the management's requirements regarding their work). Therefore, the management decided to develop a handbook containing the description of the IQSOFT organisation, the possible positions, functions, roles and responsibilities. During the analysis made before developing this handbook, it was decided to start with a formal quality department.

According to a Bootstrap assessment carried out in 1997, the overall organisation had the maturity level 2, while the pilot projects reached 2.50 in CMM. Issues related to organisation were situated on level 3, life cycle independent functions and process related functions reached level 2.75. Compared to the assessment made in 1994, this Bootstrap assessment showed an increase especially on those fields related to the IQPM² project goals: organisation, description of standards and procedures, project management practices. This increase supported management's belief that we were moving in the right direction.

Besides the planned ones, the IQPM² project produced a series of side results, which, from the company's long-term perspective, were even more important than the planned ones.

IQSOFT understood that the processes of a software company could be - in their case: should be - divided into at least two distinct activity types: *project management processes* and *technical processes*. Project management processes are those concerned with the successful management of the project, regardless to what technical activities (e.g. system development, system integration, support etc.) are being done in the project. Technical processes are concerned with the technical work done in the project.

Projects can be modelled according to both activity types, so project management models and project type models can be situational configured. IQSOFT noticed that project management activities in the company were more stable than the technical ones, which justified again their separation. With the standardisation of project management activities –building the (*single!*) project management model of the company - IQSOFT made the first important step towards bringing order in the company. At that moment they consciously left the technical activities undefined, just grouping them into several "project types", which would be worked out later. The finding that there were just a few, well definable project types (development project, system integration project, implementation project, research project) strengthened again the belief that it was possible to structure the company's activities.

With the above positive experiences, IQSOFT's management decided to go again for ISO 9001 registration. This decision was a logic consequence of the results obtained in the previous phase. On the one hand, the CMM level close to 3 was said by literature to be more or less an ISO 9001-conform level. On the other hand, it was clear what had to be done: the project model developed in phase 3 had to be completed with the type models, the non-project- related activities had to be standardised as well. A very important fact was that the employees were accepting the need for an organised way of working.

The situation on the software market had also changed since IQSOFT's first attempt to register: by November 1997 a number of 6 Hungarian software companies had been registered according to ISO 9001. Being registered was not a novelty any more, while not being registered started to be a serious handicap.

On this basis a project was started to obtain registration. It can be called a "new approach" project because it had the scope of obtaining ISO 9001 certification using all former experience of IQSOFT in building a *customised QMS*. *The project was declared a top-priority one*. The project team was formed by a project owner - the technical director-, one of the authors was declared project leader, and 10 project members - one representative for each of IQSOFT's departments - spent part of their time working on the *type-models of their departments*. The estimated internal effort was 250 man-days. Official start date of the project was preceded by two training sessions: an internal auditor course was

organised for all project members, and a course attended by the top management and the leaders of the different departments.

The quality management system was fully operational beginning with February 1998. The final audit for the registration took place in April 1998, and it was successful. It is important to show the structure of IQSOFT's internal QMS: it followed the recommendations of the ISO standards, but it was built to fit the specific needs of the company itself. The basic idea underlying the QMS was *that project-related activity and non project-related activities existed at IQSOFT*. They grouped these into "project-related processes" and "non project-related processes". Describing the "non project-related processes" was easier: the processes related to sales, training, computer system administration and secretariat were relatively well defined already in 1993. Having the experience of the IQPM² project, *we separated the project management activities from the technical ones*. This way, a company-wide standardisation became possible: the project management standards - which were worked out earlier, within the IQPM² project - were valid for all projects. The differences were described in the so-called "project type models".

The ISO 9001 project has been evaluated to be a successful one. It reached its goals within the planned time and cost limits, and it produced the planned results. We can state that the success of the project was strongly influenced by the strong support of top management. The employees accepted to work in a standardised way. There was no evident opposition from any departments; passive opposition (characterised mainly by the absence of co-operation) was rarely observed.

The QMS built in IQSOFT is being used actively ever since, although reorganized in 2001 and 2003.¹ This fact is shown by the relatively high number of requests for changes in the prescriptions, templates, employees' propositions and problems raised in connection with the QMS. The procedures are updated to fit the modified practice, and, as new practice is being observed, new ideas are introduced to make things work better. According to the evaluations, working in conformance with the prescriptions of the quality management system causes about 30% of extra effort in small projects, and about 10-20% in the bigger projects.

3.2 MEMOLUX

As part of the PASS ESSI PIE project initiation, 6 main stages were set up covering the workpackage structure to provide checkpoints for project progress.

Due to the fact that there are strong connections between the baseline software development tasks and PIE tasks in measurement and quality monitoring issues, the set up and implementation of the Quality System were performed before completing the system planning phase of the PASS development, so only the project initiation of the baseline was completed at the first stage.

At the end stage review of the first stage performed in December, 1997 corrective actions were indicated in time, during the detailed planning and scheduling of the next stages:

- resources were re-allocated from the project management task to the use-steps of the scenario development to support actual try-out in the baseline project
- a software circle as a feedback mechanism was founded
- all intermediate results of each scenario and quality development were discussed in this circle.

Corrective actions succeeded. Two critical problems were solved. On the one hand the full ISO 9001 Quality System documentation was completed in compliance with the redefined process workflow, on the other hand internal dissemination connected with continuous improvement of the deliverables was working due to the new feedback mechanism.

After the completion of the ISO 9001 Quality System documentation based on the implemented quality

¹ The first reorganization was connected to registration according to ISO 9001:2000. The second reorganization was due to the organizational change. The details of the reorganizations are described in K. Balla: Software Process Improvement and Organizational Change. (50 jaar informatiesystemen 1978-2028. Liber Amicorum voor Theo Bemelmans. pp. 181-198. March 2004.)

scenarios and successful training and coaching period at the closure of the second stage, the decision was made to complete the implementation tasks of the Quality System in the third stage. This meant the overall introduction of the Quality System in the whole IT organization, which had been originally planned at later stage. This meant extra effort for the IT staff during the third stage so the baseline project was suspended for one and a half month period. The success of the performance of this stage was measured by the mid-term self-assessment, and after internal audits, a successful ISO 9001 certification was achieved.

On the basis of the results of the first three stages, the baseline project schedule was updated. The parallel activities of the baseline and the PIE projects regarding measurement and quality monitoring were performed in stage 4 and 5. The fifth stage ended by the final BOOTSTRAP assessment of the project. The measured maturity level of the IT department was 3. In stage 6 the external dissemination activities were completed.

Table 2 shows the dynamics of the experiment by comparing the first plan with the actual one. The main quantitative result of the project regarding scheduling is the two-month delay in baseline development compared to the one-month introduction period of the ISO 9001 Quality System. The software industry based expectation for the introduction was 6 months. This means that the formerly achieved process improvement maturity level 3 causes delay in the first project, but this delay is comparable to the introduction period of the ISO 9001 Quality System. The clear advantage is the extremely fast achievement of compliance with ISO 9001 requirements for the whole IT organization.

The main lessons learnt from the above experiment were the following:

- The approach of considering the improvement of the maturity level as the principal objective and the achievement of ISO 9001 certification as a side-effect was valid from the efficiency point of view.
- Even if ISO 9001 certification was not the principal objective of process improvement, it might be worth capitalizing on its high recognition by allocating appropriate resources to its achievement.
- According to international experiences, there is usually a significant decline of attention towards the quality system after the ISO 9001 certificate is granted. The approach of considering certification as a side-effect of overall process improvement helped avoiding this trap.

4 Back to the Business

One of the major criticisms of ISO 9000:1994 was that its introduction became a burden with the overwhelming ISO bureaucracy which was only meant to control the production and was not ready to adapt to the permanent change of processes, technology and customer demands.

These business issues were highly relevant in Central and Eastern Europe at the end of the 1990's since the efficient use of all resources became increasingly critical. Hungary played in general a major role in the involvement of Central and Eastern European companies in software process improvement initiatives and the creation of channels for their presentation [11], as well as contributed to the global understanding of business motivations for software process improvement [12], [13]. The publication of the basic concepts of SPI and of the business motivations in the form of book chapters accessible in Hungarian language was a major milestone as well [14], [15].

A further issue which is highly relevant in emerging countries is the consideration of the differences in cultural value systems when introducing new management processes. This issue is discussed in the context of SPI in [16].

The above global processes drove independently both of the discussed companies to orient their further process improvement initiatives towards their business needs. IQSOFT/ IQSYS worked out the structure of its quality goals model based on the Balanced IT Scorecard (BITS), while MEMOLUX, whose core business is payroll accounting service provision, introduced the newly published eSourcing Capability Maturity Model for IT-enabled Service Providers (e^{scm}). Both of the companies achieved of course ISO 9001:2000 certification as well.

4.1 IQSOFT/ IQSYS

While achieving ISO 9001:2000 certification in the spring of 2001, it became more obvious for IQSOFT - it was explicitly stated - that quality management was not a separate process but rather *an aspect of the management processes*. The QMS was not a separate system, but *the integration of all systems, processes, practices, documentation, rules, conventions used at the company*.

Being obliged to fulfill the standard requirement about setting quality goals and establish a metric program to measure them *guided the company towards connecting quality goals to business goals*.

It is interesting to analyze the changes in setting the quality goals of the company. If we look back to connecting quality goals to business goals while having in mind the Balanced IT Scorecard framework (BITS), we can notice that the quality goals of IQSOFT/ IQSYS have been grouped in fact according to the elements considered important in the BITS. The company established quality goals related to: financial issues, customers, people, processes, infrastructure and innovation elements.

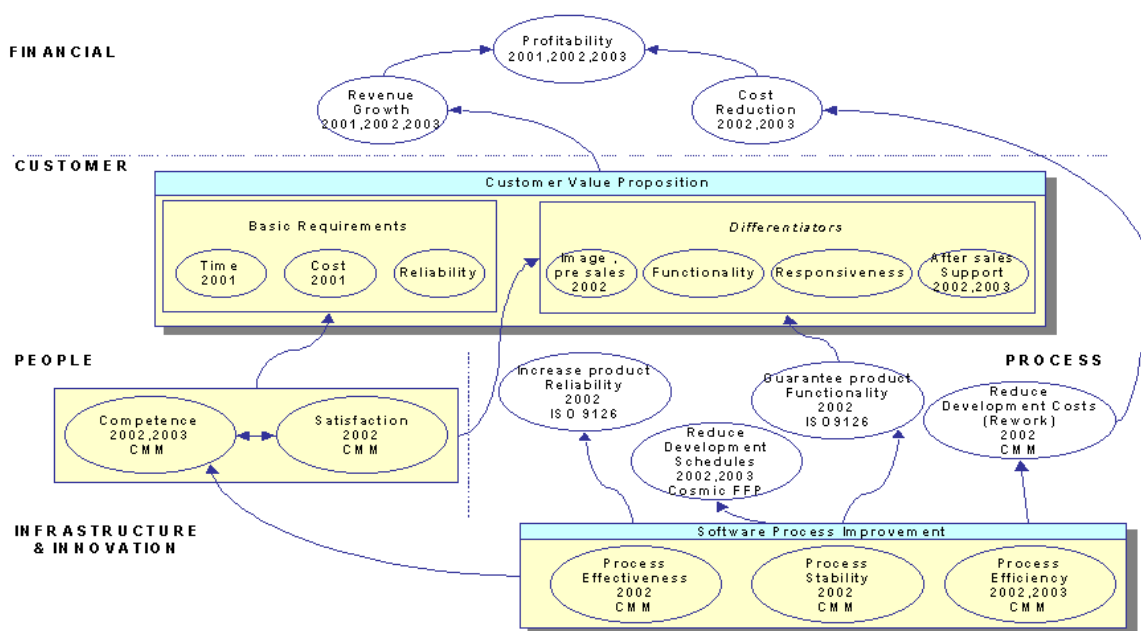


Figure 1: Connecting quality goals to business goals

In Figure 1 we show the elements contributing to the successful execution of a strategy using a representation suggested in [21], marking also the year (2001) when quality goals associated to each element appeared in the company. This year is also connected to the moments when the usage of models different from ISO 9001:2000 emerged.

Regarding the newly established goals, some important remarks can be made. First, it is obvious that quality goals in the first year (2001) were rather stereotypical (basically related to financial issues), while in the next years the company started to set quality goals more and more deriving from real business needs. Next, using further software quality models, besides ISO 9001:2000, appeared as a business driven quality goal.

Since 2001, there was no precise understanding about why and how measurement should be done, only project management- related data gathering was started (planned and actual time, cost and effort of projects were recorded). By 2002, measurement provided some data that, although not sufficiently accurate, guided the attention towards problematic areas of the company's activity.

The biggest problem was considered to be the huge difference between planned and actual effort of projects, forcing us to face that our estimates were not accurate enough. The wish to make them more accurate resulted in several quality goals for 2002, related basically to software product quality and software process improvement.

The understanding that IQSOFT was using in fact only one of the possible process oriented approaches – ISO 9001:2000 - towards software quality was there, while other possibilities in choosing appropriate models for different important software-quality-elements were considered. This way, the need to use further quality models for further important elements of software production came natural to the company. IQSYS continues the SPI started before 2003. Currently, the technological life cycle activities are being described, this is believed to bring IQSYS closer to L3 CMM or CMMI staged registration

4.2 MEMOLUX

The company built its success around the accounting and payroll outsourcing needs of Hungarian and foreign start-up companies following the social and economic transformation of the 1990's. The full time professional staff, the nimble organization, the innovative culture and their strong IT foundation enabled Memolux to maintain a stable growth and to adopt quickly to changing market requirements.

Due to a conscious and consistent integration of business and technology development efforts in the company's strategy, Memolux was able to build and maintain a competitive advantage in its markets. In 1999 the new project called PASS3Mill started based on the new definition of the company's Payroll business strategy. With the help of the ICT staff, the management of the Memolux Payroll Department developed a vision listing the major characteristics of the business Memolux wanted to run.

The effects of the PASS3Mill project on the capability of the outsourcing business environment were measured against the framework of the eSourcing Capability Model for IT-enabled Service Providers (e^{scm}) [19] in the beginning of 2003.

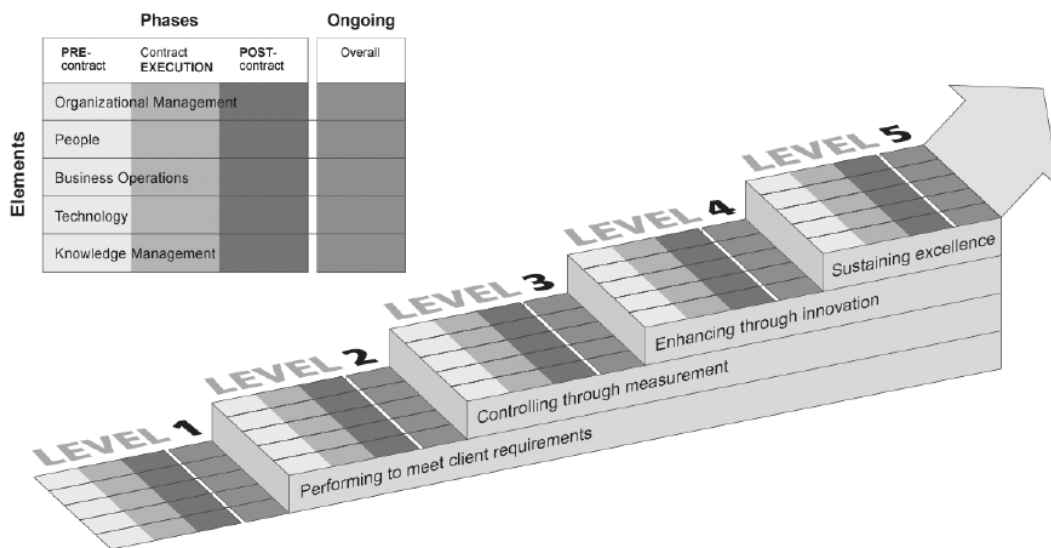


Figure 2: The e^{scm} framework

The e^{scm} framework contains nearly two hundred practices for IT enabled outsourcing providers that address the critical capabilities of the service. All the practices are organised into three outsourcing phases and five organisational elements and each practice is associated with a capability level.

The objective of the self assessment was to define the measurable effects of the previous PASS3Mill project. In 1999 when PASS3Mill project started no applicable capability models (such as e^{scm}) were available and could be applied to define the level and the necessary improvement steps of business related services such as the payroll outsourcing service.

In business terms, the PASS3Mill project achieved the main goal of providing enabled IT technology and process support for the redesigned outsourcing business process at predictable cost. During the migration period, the fast growing (40% per year) volume of business transactions were processed by a much lower increase (12% per year) of human resources.

There are four main lessons learnt from using e^{scm} model to assess Memolux outsourcing capability:

- High capability level (3) practices can't be achieved without external process improvement support.
- The high capability level can't be kept without running knowledge management system.
- The practices of the e^{scm} framework are well adaptable for any virtual organisation model as the high capability level outsourcing cooperation of service clients and providers implements a real knowledge-based virtual organisation.
- The verification and accounting of the transferred knowledge resources are critical issues for IT enabled outsourcing sustainability.

Based on the lessons learnt, Memolux is focusing on how its process improvement skills and the adapted technology having been developed for many years can be utilized as a professional new outsourcing service. One of the business intentions is to provide knowledge-based skill and technology support for SMEs and NGOs participating in public-private cooperation projects. The new outsourcing service organisation is naturally running on the same knowledge management platform, which is part of the service itself. The uniqueness of the proposed service is that it assumes cross European partnerships both in the service providing organisation and the users sides.

The implemented knowledge management platform provides:

- Teamwork tools and processes for inter- and intra-enterprise cooperation supporting quality management via integrated knowledge portal;
- Flexible organisational structures empowering knowledge communities and radical innovation;
- Product & service design solutions of supporting specific business models.

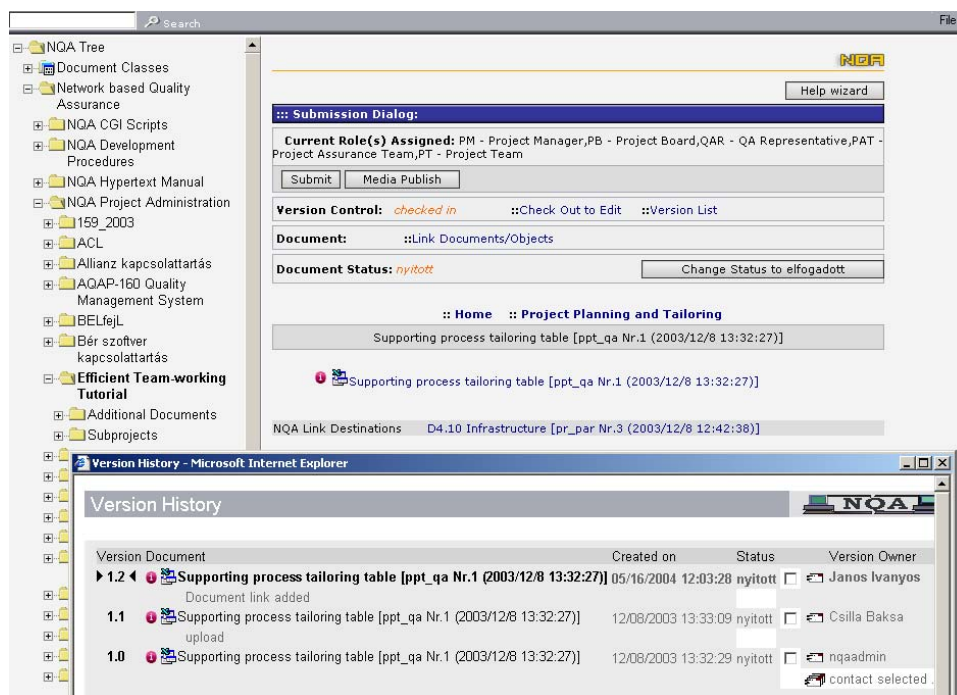


Figure 3: The e^{scm} implementation of Memolux

These methods and tools have not been available for the accession business communities. They still have been the advantage of the big multinational consulting network-organisations, which have no real solution to the local SME and NGO business environments in the enlargement countries. However successful local business process outsourcing firms have been growing in size and services in the accession countries as well, frequently joining to international groups to scope with the requests of globalisation. These local service providers are aiming at the SME and NGO communities as well, by offering value added services at reasonable prices via traditional outsourcing business models.

Based on the tailored fundamental quality requirements, the new knowledge management platform is following already proven valuable standards (e.g. ISO9001, ISO12207, AQAP160, AQAP170 and EFQM), while its basic content - knowledge and intellectual capital management for SMEs and NGOs - offers important contribution for the innovation strategy especially in virtual production, supply chain and life-cycle management, interactive decision-aid systems, development and rapid manufacturing.

5 Conclusion

The last couple of decades were rather turbulent in Central and Eastern Europe. Nevertheless, this paper shows that important advances have been made in the direction of process improvement and business efficiency, and despite the special nature of this emerging business environment, the experiences are globally valid. This means that a major initial motivation for SPI is the recognition of the existence of process capability weaknesses. Once the first successful steps are made, ISO 9000 certification is considered, since this is widely known by management and may provide marketing leverage to the firm. After possible disappointments, SPI is rediscovered in the context of its actual potential contribution to business success. And this is actually one of the gateways to level 4 in the emerging ISO/IEC 15504 standard.

6 References

1. Balla, K.: Attempts to introduce software quality management. In: Austrian-Hungarian Workshop on Software Engineering, pg. 6.1.-6.10, Klagenfurt, 8-9 April 1994.
2. Balla, K.: Software quality management : ISO 9000- but not only. Research Report. Technical University Eindhoven, March 1995. (EUT/BDK/68, ISBN 90-386-0055-0, ISSN 0929-8479)
3. Balla, K., Langer, T.: Beneficial effects of IQpm² on the quality oriented work. In: Proceedings of the Annual Meeting of European Artemis Users Association, T2, pp.1-10, Oslo, Norway, 1997. Sept. 18-19.
4. Balla K., Langer T.: Process Improvement for Better Software Quality. Proceedings of the "ESI & ISCN" International Conference. Budapest, 1997. November 11-12.
5. Biró,M., Feuer,É., Haase,V., Koch,G.R., Kugler,H.J., Messnarz,R., Remzsô,T.: BOOTSTRAP and ISCN a current look at the European Software Quality Network. In: The Challenge of Networking: Connecting Equipment, Humans, Institutions (ed. by D. Sima, G. Haring). (R.Oldenbourg, Wien, München, 1993) pp.97-106.
6. Biró,M.; Feuer,É.; Remzsô,T.: The Bootstrap Institute: International Organization for Software Quality Assurance in the European Union. In: Proceedings of the IV. Quality Week International Conference (ed. by M.Pákh, A.Róth, T.Turos, Á.Hargitai). (Hungarian Quality Society, 1996) pp.133-136. (in Hungarian)
7. Biró,M; Feuer,É.: Convergence of the Software and Systems Engineering Capability Maturity Models (CMM, CMMI) and ISO 9000:2000. In: Proceedings of the X. Quality Week International Conference (ed.by A.Róth, I.Jancsovcics). (Hungarian Quality Society, 2001) pp.136-141. (in Hungarian)
8. Biró,M.; Ivanyos,J.; Messnarz,R. Pioneering Process Improvement Experiment in Hungary. Software Process: Improvement and Practice (John Wiley & Sons, Ltd.) Volume 5, Issue 4, 2000. Pages: 213-229. (<http://www3.interscience.wiley.com/cgi-bin/abstract/76503384/START>)
9. Biró,M.; Remzsô,T.; Sz.Turchányi,P. Assessment and improvement of the quality of software processes from a decision maker's perspective. In: Proceedings of the VI. National Congress of the John von Neumann Society for Computer Sciences (ed. by I.Tóth). (John von Neumann Society for Computer Sciences, Siófok, Hungary, 1995) Vol.2. pp.671-678. (in Hungarian)
10. Biró,M. Quality Criteria in Software Development. In: Proceedings of the TÜV Rheinland Hungary Group Conference on Quality, Environmental Protection, Security, and Integrated Management Systems. (Balatonaliga, 1997). (in Hungarian)
11. Biró,M. (moderator); Gorski,J.; Stoyan,Yu.G.; Loyko,M.V.; Novozhilova,M.V.; Socol,I.; Bichir,D.; Vajde Horvat,R.; Rozman,I.; Györkös,J. Software Process Improvement in Central and Eastern Europe. Software Process Newsletter (IEEE Computer Society) no.12, Spring 1998, pp.19-21.

12. Biró,M.; Tully,C. The Software Process in the Context of Business Goals and Performance. Chapter in the book entitled Better Software Practice for Business Benefit (ed. by R. Messnarz, C. Tully). (IEEE Computer Society Press, Washington, Brussels, Tokyo, 1999) (ISBN 0-7695-0049-8).
13. Biró,M.; Messnarz,R. Key Success Factors for Business Based Improvement. Software Quality Professional (ASQ~American Society for Quality) Vol.2, Issue 2 (March 2000) pp.20-31. (http://www.asq.org/pub/sqp/past/vol2_issue2/hiro.html)
14. Biró,M. The Quality of Software Products and Software Processes. Chapter in the book entitled For Businessmen about Readiness (ed. by Dr. E.Szilvássy). (Qualika, Budapest, 1998) (ISBN 963 04 9517 1). (in Hungarian)
15. Biró,M. Software Quality. Chapter in the book entitled Quality Management and Informatics (ed. by T. Tóth). (Technical Publishing House~M?szaki könyvkiadó, Budapest, 1999) (ISBN 963 16 3047 1). (in Hungarian)
16. Biró,M; Messnarz,R; Davison,A.G. The Impact of National Cultural Factors on the Effectiveness of Process Improvement Methods: The Third Dimension. Software Quality Professional (ASQ~American Society for Quality) Vol.4, Issue 4 (September 2002) pp.34-41. (http://www.asq.org/pub/sqp/past/vol4_issue4/hiro.html)
17. Boehm, B.W.: Characteristics of software quality, North Holland, 1978.
18. Humphrey, W.S., Sweet, W.L.: A Method for Assessing the Software Engineering Capability of Contractors, Software Engineering Institute, CMU, Technical Report CMU/SEI-87-TR-23, 1987.
19. Hyder, E.B., Kumar, B., Mahendra, V., Sieges, J., Heston, K.M., Gupta, R., Mahaboob, H., Subramanian, P.: eSourcing Capability Model (escm) for IT-enabled Service Providers v1.1 21 October 2002; CMU-CS-02-155.
20. McCall, J. A.: The utility of software metrics in large scale software systems development, IEEE Second software life cycle management workshop, August 1978.
21. Reo, D.: Balanced IT Scorecard. ESI Commercial Network presentation. 2000.
22. Szentes, J.: SOMIKA: An automated system for measuring software quality. Software-Qualitätssicherung 1982, 188-202.

Author CVs

Miklós Biró

Dr. Miklós BIRÓ (miklos.biro@bkae.hu) is a professor at the Department of Information Systems of the Budapest University of Economic Sciences and Public Administration.with 27 years of software engineering and university teaching (including professorship in the USA), and 17 years of management experience. He has a Ph.D. in mathematics (operations research) from the Loránd Eötvös University in Budapest, an Executive MBA (Master of Business Administration) degree from ESC Rouen, France, and a Master of Science in Management degree from Purdue University, USA. He is fluent in Hungarian, English, and French.

He is a SPICE (Software Process Improvement and Capability dEtermination - ISO/IEC 15504) assessor. He gives Ph.D. courses and company training courses on software quality management, and on the Capability Maturity Model - Integrated (CMMI - service mark of Carnegie Mellon University, USA). He designs and delivers software technology courses tailored to the requirements of the program in business informatics of the Budapest University of Economic Sciences and Public Administration.

He initiated and managed the Hungarian participation in numerous European multinational projects and organisations committed to software process improvement (European Software Institute, Bootstrap Institute). He was the initiator and head of the Information Society Technologies Liaison Office in Hungary for the European Union's 5th Framework Programme. He is invited as expert consultant by Hungarian and international organizations (European Commission; Irish National Policy and Advisory Board for Enterprise, Trade, Science, Technology & Innovation~Forfás; Communications Authority of Hungary; Hungarian Committee for Technological Development; Investment and Trade Development

Agency of Hungary; Hungarian Airlines; United Nations Industrial Development Organization~UNIDO; International Software Consulting Network;...).

He has numerous publications in international scientific and professional journals ([Software Process Improvement and Practice](#), [Software Quality Professional](#) (1, 2), [Software Process Newsletter](#), [European Journal of Operational Research](#), [Zeitschrift für Angewandte Mathematik und Mechanik](#), [Optimization](#), [Information Processing Letters](#), [Discrete Mathematics](#), [Journal of Advanced Transportation](#), [Acta Cybernetica](#)) and conference proceedings. He is the co-author of Hungarian and English language books on operations research models, software engineering, software process improvement and business motivations.

He is member of the [Editorial board of the journal Software Process Improvement and Practice](#) published by John Wiley & Sons, and founding president of the professional division for Software Quality Management of the John von Neumann Computer Society. He is the Hungarian member of [Technical Committee 2 \(TC-2\) Software: Theory and practice of the International Federation for Information Processing \(IFIP\)](#) . He is member of several other professional bodies and societies.

Evaluation of using a UML methodology in a large embedded development project

Hanne Kristin Thorsen, Bente Anda, Kai Hansen and Ingolf Gullesten

This presentation gives the result from an examination of using a UML based development methodology in a large, complex system development project. The development project has introduced and used UML for developing software and hardware for the industrial market. The evaluation shows which part of the method that has improved the development and what have been more difficult parts in the process of introducing a new method. There have been gained positive effects from use of the method, and most developers have found the method to be suitable for the kind development performed in the project. However, some problem aspects are identified, which may be considered potential areas for improvement.

Keywords

[Click here to insert key words]

1 Introduction

ABB is an international company with the main business areas development of power and automation technologies, including production of electronics, software and hardware for automating process control systems. ABB has a large number of development projects, the majority of them are embedded and system development where development of special hardware is included. A large number of different methods, programming languages and software tools are used, but generally there is little common streamlining of the system development [2].

To meet market requirements, ABB decided to get their safety products certified by TÜV to the international standard for safety equipment, IEC 61508 [8]. This safety standard is becoming the dominating generic standard for hardware and software development [7]. The company also identified that adapting "state of the art" methodology in analysis and design might lead to improvement of their system development projects. Consequently, a Software Process Improvement (SPI) effort was started, involving development, introduction and use of a new system development method, the ABB UML method, adapted to the company's specific product development.

1.1 The ABB SPIKE project

ABB has become contributor to the national SPIKE (Software Process Improvement based on Knowledge and Experience) project [12]. In cooperation with Simula Research laboratory, located in Oslo, Norway, ABB's SPI project was extended, and preceded as an ABB SPIKE improvement project. The focus of the project is to evaluate the use of the ABB UML method as it has been applied for development of safety products in a large-scaled system development project, the Pluto program.

The research motivation is related to the use of UML for system development. To meet IEC 61508 requirements of use of semi-formal methods for development of safety equipment, ABB based their new development method on use of the graphical Unified Modelling Language (UML). UML has become a de facto standard diagramming notation for object-oriented system modelling in industry [9, 10], primarily for software. Numerous software development methods are based on the use of UML [9]. However, based on literature search, it seems like little material on evaluation of costs and benefits from use of UML and associated method has been published.

The main research goals are:

- Identify prerequisites for successful use, particular challenges and most immediate effects of UML-based development when the team members are newcomers to UML.
- Evaluate the suitability of UML-based development when developing safety-critical embedded systems.
- Identify possible improvements in the current UML-based development method applied in ABB, for example with regards to how to handle large and complex UML models, how to apply UML in development based on an existing system, and how to assure traceability.

The research methods have involved gathering experiences from the use of the ABB UML method by depth interviewing some the project participants at the different sites. The results from the interview analysis was used to focus a questionnaire, aiming at having as many as possible of the project participants answering it, which in fact is ongoing work. A comparison of the ABB UML method with alternative UML-based development methods (i.e. RUP) is another work in progress. The comparison is done to position the method and evaluate how general the results from the evaluation are, and inspire to method improvements. Another planned activity to be performed are analysis of project documents to triangulate results from interviews and questionnaires, as well as identify best practices that can be used to improve the ABB UML method.

The ABB UML method [13] is applied within the ABB V-life-cycle model. It is a semi-formal variant which involves seven phases organized in an iterative V-model, and equal to the life cycle model, each of the analysis and design phases has a corresponding test counterpart. It pays specific attention to the requirement analysis and design phases, specifying the use of UML diagrams and Rational Rose as tool, and the system documentation to be produced; the Description of Functions (DoF) and Detailed Design (DD) documents. It is applied for both software and hardware (VHDL and standard) development, but hardware development is only required to follow the requirements analysis part of the method. The phases are described on component and module level. It is document-driven in the sense that series of documents based on predefined templates is produced, constituting important milestones in the development process.

2 Introduction of the method

The depth interviews were performed in the middle of the project, and the interview analysis was not only important to focus the questionnaire, but constituted a spot test of the project situation at the time, resulting in a preliminary report [3]. Some of the results are presented in this article.

2.1 Expectations

Most developers focused mainly on that they had to use new method to meet the new safety requirements. Some expected positive effects to be obtained from applying the new method though, like improving the development process and the system product quality in terms of lower fault rate and ensuring product and safety requirements to be met. Some also had self-interest in learning to use UML for design and documentation. Others in fact were rather skeptic and reluctant to the usability of method to this kind of development, especially the hardware developers. Even though varying attitudes towards the ABB UML method existed, most of the interviewees seem to have had a neutral or positive approach to the method.

2.2 Experiences

The introduction of a new development process with associated methods and tools have represented an effort. The change process has influenced several organizations, and required human barriers to be conquered to achieve changes in attitudes, implying a cultural change as well as technology shift.

A change process requires involvement and commitment on all organization levels, to be successfully accomplished [6, 9]. This was considered only partly successful. The ABB UML method was introduced at project level. This might have affected how the management at the different work places handled it. The management was accused of neither knowing the method very well, nor understanding how demanding it was for the developers to take the ABB UML method into use, as it represented adapting to an object-oriented way of thinking while most of them were used to functional programming.

Some criticism were given for introducing the ABB UML method after the project had started, and the developers already had began working, basing documentation on old templates. Introduction of new guidelines and templates resulted in some rework to be done.

The UML training was criticized for not being sufficiently tailored to the kind of development (real-time, embedded) to be performed in the project. This was not considered easy to be done though, as it would be difficult to find people with experience from both use of UML and this kind of development.

Some felt it should have been considered more thoroughly how to apply the method, and likewise, the method introduction could preferably have been better planned. The new method was introduced to a large number of people at the same time, without having any previous reference projects to learn from or a sufficient number of mentors to provide guidance in how to apply the method. Not much support

could be gained from comparable similar previous experiences either. As a consequence, very many people simultaneously had to figure out mostly themselves how to do things. This resulted in different ways to perform UML analysis and design, and likewise different appearances of the documentation documents, between the individual project participants and between teams.

The introduction problems seem to be due to the specific introduction circumstances. A new quality system structure was established and a new development process introduced, while the development work was already in progress. With a project that size, involving cooperation between different cities in different countries, this necessarily involved some challenges. An important factor is that the project did not start out this large, but grew in complexity and number of people along the way, as more knowledge of the development products to be produced was acquired.

3 Effects from use of the method

It seemed like most interviewees considered the ABB UML method to be a suitable system development method for real-time, embedded development. Some of the positive effects resulting from use of the method are presented below. Some of the problem aspects identified also deserves a closer look and may be potential areas for improvement.

3.1 Positive effects

The main impression seems to be that the documentation documents, the DoF's and the DD's, have improved from previous documentation in elder ABB projects. They have become more comprehensive and cover more aspects. The use of the UML ensures a more uniform documentation, compared to previous textual descriptions in English and use of various types of diagrams and pictures not having unified notation or syntax. It was also claimed to be easier to learn people good UML than good English. The use of UML is considered to be intuitive when correctly used, and has increased the readability of the documentation, which is especially appreciated by the reviewers.

The traceability between requirements and code has improved. It was by most interviewees found reasonable and useful to trace the requirements from use cases, through the different diagrams, to implementation. It was not always easy though, to ensure this traceability consistency, as the size of the requirements could vary from a small design requirement to a communication protocol.

The ABB UML method was considered having enforced better design through making people take a more top-down approach to the development, which by the more experienced developers was considered the best way to take full advantage of UML. They could not go straight ahead prototyping. The design became more thoroughly considered, and constituted a framework very useful for implementation purposes. Together with a comprehensive documentation this contributed to lower the risk.

Use of UML increased communication between the project participants. The developers with system parts/functions having interfaces against each other had to discuss design matters early in the process. People were enabled to present to each other what they had understood in a unified way, expressing themselves in a more consistent and understandable manner, through the UML diagrams. For instance, sequence diagrams were frequently used as communication means, in terms of developers discussing design matters while drawing diagrams at white boards. Sequence diagrams were regarded especially useful when to describe matters that is not easy explainable by text. Both use cases and sequence diagrams were considered intuitive and useful, especially when to explain things to an outsider, not familiar with this particular development. The early discussions increased the possibilities to capture problems early in the project, facilitating reduction in the number of faults.

Having already made the DoF and DD descriptions, including the UML diagrams, writing test cases represented less effort. It was easier to write test specifications from the use cases and sequence diagrams because the diagrams cover more aspects than did textual descriptions in previous system documentation. Controlling that all aspects specified in the diagrams are covered, more complete tests were ensured.

Most interviewees felt it would be easier to make further development of this particular system, as it now has become thoroughly documented. Equally, people were positive to use the method in other future projects. The experiences from the Pluto Program were expected to make it easier to take full advantage of the method in future projects, with greater opportunities to achieve possible gains.

3.2 Problem aspects

An important aspect is the suitability of the method to integration with an already existing system compared to development from scratch, as both types were contained in Pluto project. Integrating with the old system meant considerable more problems. The existing code was poorly documented, and no architectural overview document existed at the time the interviews were conducted (later it has become available), making it difficult to figure out how to fit the new parts into the elder system. In addition, the old system was non-safety and developed by functional programming, while the new parts to be added were safety related, requiring use of UML and Rational Rose, with associated object-oriented method.

For the new system to be able to reuse interfaces from the old system, it was necessary to make models of the old system by reverse engineering. There were no ambitions though, to reverse engineer the entire system, or combining the models resulting from this process. This was regarded as both too time-consuming and expensive.

The developers had to start out modelling right in the middle of something, making it difficult to take a top-down-approach. Starting out with a top level model of the entire system, followed by a breaking-down process into subsystems, components and further classes and so on, the method better could have helped structuring and dividing the system into parts, and ensured consistency control between the interfaces. However, it was not considered easy to achieve such an approach in a project this size and complexity though. Many teams of developers have to start development work in parallel within a short time, making it difficult to reach identifying the interfaces as fast as required. This breaking down process was not considered particularly useful in hardware development as most of the components usually already were defined.

Both the new system parts/functions to be added and the old part it had interfaces against were required to be documented by UML. This resulted in rather huge DoF's and DD's, up to an amount of about 2-300 pages each document, and perhaps only 50 of those pages covered the new part, by developers considered the most relevant. Most interviewees complained no one either could make use of or had they time for reading DoF's and/or DD's that size. Some examples of modelling adaptations due to the integration with the old system were pretending the old code to be a black box and having to make fictive high-level classes. It was also considered difficult to model the interfaces. There had been difficulties in deciding the appropriate interface description detail-level, and according to the reviewers the interface descriptions typically were insufficient.

The method description provided by the guideline and the templates seem insufficient in some areas, and rather rigid in others. They did not sufficiently support a uniform use of UML. More guidance was requested on how to properly differ between the versions of the diagrams with regard to both abstraction level and detail level. More guidance on detail level in the DoF's and DD's in general were requested, to make it easier to distinguish between analysis and design activities. The DoF's typically have become too detailed, and this is reflected in the DD's, causing problems getting the right balance between those documents.

More relevant examples on appropriate use of the method were also requested. The templates did not allow UML models to be replaced by textual descriptions or non-UML models, or those features to be added. However, this seems to have been done to some extent, when considered necessary for viewing aspects not covered by UML, particularly for hardware development. Too little guidance was given on what to be contained in the system description in the DoF's and DD's, and the developers tended only to describe how the smaller new package was placed into another larger function, not its place in the entire system. The reviewers did not appreciate this, as it was considerably more useful to them if it was possible to picture the connections between the system parts described by the DoF's and DD's.

The reviews were criticized for not paying enough attention to functional aspects, but wasting time and

money on long-lasting discussions on UML subtleties, which resulted in much document rework. For instance, if the models themselves were subject for review instead of the DoF and DD documents, the update process would have been easier and felt less demanding. The first reviews perhaps had too many participants to be sufficiently efficient, but this had improved during the project.

The ABB UML method was experienced more as a waterfall approach than iterative, something which could increase the risk of not uncovering major faults until late in the project. Some interviewees argued the problem was not the method itself, but how it was applied. It was both possible and preferable to use the method both in a more iterative and incremental manner. This was preferable due to that it would make it easier to take full advantage of the use of UML [5, 9, 10]. The method is also characterised as document-driven, strongly emphasizing the production of the system documentation, which constitutes important milestones in the development process. The UML models are developed in a way that make them fit into the specific DoF and DD templates. This means the templates drive the development process. Some of the integration problems also aroused because the templates required certain models at certain places.

4 Conclusions

A successful introduction of a new ABB UML method is dependant on management commitment, investment of both time and money in training and having available experts who know the method involved in the design and reviews. The external safety requirements have been effective enforcement for method introduction. The method facilitates communication between developers, and the system documentation has improved considerably compared to previous documentation, but still not sufficiently. Most interviewees seem to consider the ABB UML method suitable to this kind of development, and to be positive to use of the method in future projects. However, there seem to be an improvement potential for development involving integrating with an already existing system. Some of the challenges have been difficulties in learning some of the UML concepts, modelling interfaces, and finding an appropriate abstraction and detail level of the diagrams influencing finding right balance between the analysis and design activities and the DoF and DD documentation.

5 Literature

1. ABB Gate Model for Product Development 1.1, tech.report 9AAD102113, ABB/GP-PMI, Västerås, Sweden 2001.
2. Anda, B. & K. Hansen, 2003. Prosjektplan for UML metode. Innføring og evaluering i ABB. SPIKE Intern.
3. Anda, B, Hansen, K., Gullesen, I. and Hanne K. Thorsen, 2004. A Preliminary Evaluation of Costs and Benefits of UML-based Development Methodology in ABB, V0.1 17/2 – 04.
4. Boehm, B. W., 1979. Guidelines for verifying and validating software requirements and design specifications. EURO IFIP 79, P.A. Samet (editor), North- Holland Publishing Company.
5. Booch, G. Rumbaugh, J. and I. Jacobson, 2000. The Unified Modeling Language User Guide. Addison Wesley Longman, Inc.USA.
6. Dybå, T., Fingsøyr, T. and N.B. Moe, 2002. Praktisk prosessforbedring. En håndbok for IT-bedrifter. Fagbokforlaget Vigmostad og Bjørke AS,, Bergen, Norway.
7. Faller, R. , 2001. Project experience with IEC 61508 and its consequences. Computer, Safety, Reliability and Security: 20th International Conference, SAFECOMP 2001. LNCS 2187, p200.
8. IEC 61508: Functional safety of electrical/electronic/programmable electronic safety-related systems. 1998 (<http://www.iec.ch/>)
9. Kruchten, P., 2004. The Rational Unified Process. An introduction. 3rd Ed., Pearson Education, inc. c2004, Boston, USA.

10. Larman, C., 2001. Applying UML and patterns. An introduction to Object-Oriented Analysis and design and the Unified process. c2002 Craig Larman. Prentice Hall PTR, Prentice Hall, Inc. Upper Saddle River, USA.
11. Sommerville, I., 2001. Software Engineering. 6th Ed. c2001 Person Education Limited. Addison Wesley. USA.
12. SPIKE (Software process improvement based on knowledge and experience) prosjekt beskrivelse. 15. oktober 2002.
13. The ABB Guideline "Guideline for use of semi-formal methods in SW and HW design", 2003.
14. The ABB Instruction "Software and Hardware development", 2001.
15. Wallin, C., Ekdahl, F. and S. Larson, 2002. Integrating Business and software Development models. IEEE Software, November/December 2002.

Building a learning organisation: experience with establishing communities of practice

Hege Dreiem, Rune Myrdal, Kjetil Jørgensen-Dahl

Abstract

In order to enhance the collective learning and enable sharing of experience and knowledge between employees, Objectnet, which is a small to medium sized Norwegian consulting company, set out to form several communities of practice. Although well founded in strategic considerations and encouraged by management, the communities that were established were not equally successful in their accomplishments. However, executives and community members report that the initiative has had noticeable beneficial effects on the organisation.

When initiating the communities, management and key personnel grouped topics suggested by members of the engineering staff into related topics. All technical members of the organisation were assigned to a topic. A full day workshop gathering all potential group members was then arranged to speed up community formation. After this initial activity, the interest groups have lived on without much interference from management.

The interest group initiative has shown that with a minimum of effort it is possible to initiate communities that enable sharing of insight and experience. Encouraging the formation of such communities enhances the possibility to learn from colleagues across temporary project structures. In addition to acquiring new technical knowledge, the community members also report establishing valuable social relations with their colleagues. A number of factors can be identified to facilitate successful initialisation of communities of practice.

Keywords

Community of Practice

1 Introduction

In a software consulting firm that aims for the ability to develop and deliver software systems based on cutting edge technology, there is a constant need for keeping up with new technologies, tools and practices to make development efforts effective and accurate. As different projects are assembled and dissolved, it is crucial that experience is shared across the entire company, building a collective intelligence. In order to enhance the collective learning and enable sharing of experience and knowledge between employees, the Norwegian consultancy Objectnet set out to form communities of practice [1], [2].

Communities of practice often form rather spontaneously, but under certain conditions the formation of such communities face obstacles like physical distance, lack of knowing each other and time pressure. This paper reports some of our experiences on guiding the formation of communities, aiding and speeding up the process of community formation.

2 Background

Objectnet is a small- to medium sized Norwegian consulting company, developing business critical information systems for its customers. The company has offices in several locations, Oslo, Arendal and also personnel based in Porsgunn.

Although employees are knowledgeable experts in their field, the company recognizes that one of our greatest challenges is to share knowledge across our organisation. A core challenge is to track; who knows what in the organisation. Written records soon become outdated as employees are assigned to new projects, develop new skills and become familiar with new technologies. Facilitating the sharing of knowledge across the organisation, thereby enabling to share tacit knowledge may be one of the critical factors for success in a highly competitive market. We must learn how to capture and transfer knowledge more effectively.

The main work form for the employees is projects, typical project time span varying between a few months to several years. Most often, the different projects are located in different physical locations and involve different customers and technologies, which is not the best basis for cross project knowledge sharing. There have been limited joint efforts to share knowledge across employees participating in different projects. We don't find this unnatural, as the project members are the closest and most natural discussion partners and the project participants are always pressed for time. In other settings, community groups may arise in an informal way, but encouraging the formation of such groups enhances the possibility to learn from colleagues across temporary project structures.

A company like Objectnet needs knowledge on cutting-edge technology and best practices to win interesting projects and to ensure our capability to see these projects through in a good way. The projects we have been working on over the last years, demands that our developers continuously keep themselves up to date on the latest innovations and technologies.

The organisation has always valued its member's efforts in acquiring new skills – this value set has been clearly communicated and rewarded for years. Before establishing community of practice, the company arranged community meetings with focus on development with presentations of different aspects. For the last year, interest groups have been the main arenas for sharing knowledge within the company.

3 Creating the communities

The formation of the communities started in the wake of a strategy seminar where, when focusing on improvement potential in the organisation, all employees were asked to define an area of interest in

which they would like to focus. Although encouraged to join others and work in teams, almost no one came up with team projects for the improvement efforts. The management therefore sought other means of achieving more cooperation, and invited key personnel to group the proposed areas of interest into fewer, related topics. The topics were all technology focused, and eventually narrowed down to the following four: Java architecture, Object-Relational mapping, the Maven project management tool, and Rich Internet Applications. All the technical members of the organisation were assigned to the topic closest to their initial suggestion, this resulting in 4 groups, each with 4-8 members. To jump-start the effort of the groups a full day workshop gathering all employees were then arranged. The workshop focused on the desired effects of the initiative for the company with regards to developing and sharing knowledge. All groups were required to elaborate on the topic of the group, define their preferred way of working, sketch out goals and desired results for the group and select a leader. The groups were told to present their mandate in public sessions summarizing the workshop, and were required to present management with an estimate of the expected cost of their activities. After this initial activity, the interest groups have lived on without interference from management.

The initial plans and organisation of the groups differed in several ways. Apart from dissimilar topics they had different meeting schedules, different planned deliverables, different ways to organise their meetings, different number of participants, and differed also in location of members. Two groups had only participants from the Oslo office, while the other two had participants spread between two of the company's offices. Still, most employees have been working together for years and knew each other fairly well.

After 10 months, three of the original groups are still active, while one group seemed to die away not long after the initial workshop. The initiative taker in this group left the company shortly after the initial workshop. This group did not succeed in establishing any common activities and has also not succeeded in reporting results of their individual work back to the organisation.

One group focused their effort on theoretical knowledge as well as sharing experience and creating discussions around selected themes. Two other groups have had a more practical approach, involving product tryouts and evaluating existing projects.

As the normal workday of a consultant is characterized by heavy workloads, not leaving much spare time for 'out-of-project' activities, some of the groups focused from the start on how they could overcome this obstacle and gain momentum as a group. Two of the groups decided to arrange meetings or workshops at regular intervals. As the organisation already had an established practice for arranging status meetings once a month, bringing together all employees, both groups chose to schedule their meetings after the regular status meetings, when all groups members would already be gathered.

The two other groups did not plan their meetings to this degree. Separated by physical distance, they opted for more individual work forms until bringing themselves up to speed on their topics, which they would then come together and discuss.

The two communities that planned their initial meetings ahead also focused on how they could further reduce the workload on their participants by dividing the responsibility for preparations and the facilitating of each meeting across group members. In one group the members all took their separate turn in filling a coordinating role. The coordinator planned the meeting, looking for more specific themes and encouraging each member of the group to prepare and present parts of the total agenda, drawing on their experiences and new knowledge. The other group assigned the task of providing background material and leading the group activities to half the members of the group, the two halves alternating in preparing the sessions.

One of the groups planned their agenda in rather detail straight from the beginning, agreeing on the theme of their meetings in advance. Not all themes were considered equally exiting, some themes were mostly included because they were perceived as useful, but not very interesting. Both the meetings and the documented results of the groups clearly show which themes the members found more interesting. The meetings that triggered most enthusiasm were better prepared, lasted longer and sometimes ended up on a close by pub for even further discussions.

The three communities that were established also used other forms of communication than face-to-face meetings. The company has been using different tools for sharing information within the company intranet for a long period of time. Among the most useful tools is the wiki-web. A *Wiki* is a website or other hypertext document collection that allows any user to add content, as on an Internet forum, but

also allows that content to be edited by others [3]. Three of the interest groups created wiki-pages, documenting their charter and immediate plans. Two of these groups used their wiki pages for keeping track of their activities, and to some extent also as a means of publishing results and facilitating discussions. Only two of the initial four groups established internal mailing lists.

After the initial activities the groups have been self-organising, and has a large degree of freedom regarding subjects and participation. One of the groups realized quite soon after the initial workshop that they had defined their charter too narrowly, and changed their focus from one specific product to a much broader interest in a whole field. The other groups also changed their focus over time, though not as explicitly as the fore mentioned group. Although the changes are reflected in the wiki sites and project plans, there is no formal decision involving other parts of the organisation. The groups are not static, they change their set of goals and objectives as they learn and explore, and the number of members in each group varies.

The groups that arranged meetings have had rather good attendance from their members. Still, a change of attitude seems to have taken place. Although all employees were assigned to a group initially, participation in the communities is now perceived as voluntary – participation is acknowledged, but is no longer perceived as required.

There have not been many activities that span several of the different communities. A few employees have been able to direct questions to other communities, asking for help on specific topics. Although the communities that based their activities announced their meetings on their wiki sites, only a few employees took opportunity and joined in on events arranged by another community. Still, when asked, the employees respond that the themes of the other groups are perceived as interesting.

During the company's latest common seminar, a one-hour session was devoted to the communities. The communities were asked if they could identify areas where they as communities could contribute to the company's latest strategy. The employees that did not identify themselves with any of the three remaining groups were allowed to form a new group or join one of the existing groups. There were no obligations to change the communities' charter based on the suggestions that may result. All groups easily came up with a lot of very relevant ideas, and all groups stated that they would to some extent like to change their agenda to focus on some of the issues that were raised.

Through the SPIKE research project [4] there has been conducted interviews with all employees and arranged workshops evaluating the effort of the communities. The results will be published in a research paper next year.

4 Discussion and Results

4.1 Effects of community of practice

Introducing acknowledged communities into the organisation has had several noticeable effects. The employees report the value of sharing experiences with others and the ease of starting discussions as the most valuable effects. Generally, the commitment to participate on meetings and the fact that most members are very active at the meetings are also appreciated. One of the employees state it this way: "In my opinion the initiative of the company is important, as the best way to learn is through the communication and work with others – to avoid learning being a solitary exercise."

Most employees state that they find the communities useful, both professionally and by increasing consciousness regarding the process of common learning. The employees get to know each other better and knows which problems others have faces in their previous projects. Through the communities the members build an overview of who knows what in the organisation. The employees perceive the use for the company to be concentrated around the communities work on innovative technologies, developing skills that provides for better quality and effectiveness in development projects, and documenting an innovative environment to potential customers and for recruiting purposes.

A senior manager says the restructuring of the company educational effort from open meetings every

fortnight to interest groups have resulted in more commitment and excitement from the community members. The employees now meet focusing on themes they have a special interest in, and the level of skills and actual work performed are better adjusted to the individual participant. The communities gather people with common interests in a different structure than projects or organisational structures, and to some extent also across physical distances.

In addition to acquiring new technical knowledge, the group members also report establishing valuable social relations with their colleagues. The communities have a cohesive and relation building effect among the members of the organisation. The resulting knowledge network that grows between the employees is of great value to achieve better communication and cooperation within the company. The communities have clearly contributed to building human relations and social networks between the employees. Knowing each other makes it easier to contact others, to look for support and help to find good solutions in projects.

One quite unexpected effect relates to community activities as a good way of documenting company effort within interesting and innovative areas. Being able to document and demonstrate these intentions and activities to our customers and external partners provides marketing advantages. Increasingly, our customers and partners wish to be included in our professional skill building activities. Often our customers lack both the tradition and the professional environment to follow the latest technical developments. The activities and resulting artefacts of our communities enable us to invite our customers to share our knowledge and participate in our activities, conveying their experiences and needs to us.

4.2 Discussion

Although there are a lot of factors that greatly influence communities within an organisation, we believe there are some issues that are of importance when reviewing how our communities came into life and how well they perform.

Even in our organisation, where employees are technology focused and used to communicating through various online tools, a key to success seems to lie in whether the group's preferred way of working actually assembles the members of the group. The groups that over time have proved to be the most active focused on meetings or workshops bringing the group members together on a more or less regular basis. Even if the members of the communities know each other from the start, meeting face to face seems to be a necessity to get the discussion started and trigger group activity.

A major obstacle for all groups is time starvation. Most of the work done in the communities has been performed after working hours. Performing educational and social activities after ordinary working hours is part of the company culture, and is to a certain degree agreed and expected both by employees and the organisation. Still, as the projects often call for overtime, it is often difficult for community members to devote time to the communities.

Many of the employees are also concerned with how the management appreciates the effort of the communities. Even if the communities appreciate being self-controlling, they would like to see management concerned with their activities, as this would confirm the status of the communities as important to the organisation. Further, most employees seem to agree that although they would like management to notice their activities, they don't want specific demands that they must meet. There is a subtle but important difference between seeing results and demanding results. The employees are also concerned with the management noticing their individual efforts within the community, as the organisation's reward systems – both acknowledgement and salary – are individual based.

Further along this line of thought, we also think it is important for the communities to be able to clearly relate their efforts to company strategy or specific organisational values. Perceiving the efforts of the community as useful for the organisation is a major motivation factor for community members. We believe the attention from management as well as relating the communities' efforts to company strategy to be the main reasons why all communities experienced a boost in activity after the latest company seminar.

Although community members wish for management appreciation, the initial request from manage-

ment to specify the expected cost of community activities has been referred to as an obstacle to community formation. Members of two of the initial groups specifically stated that they felt it necessary to get their estimates approved before they could start any activities. This delayed the start of these groups significantly. The two other groups decided to start their activities right away, betting that it would be easier to beg forgiveness later than permission in advance. Although the initiative to form the communities originated from the management, again this demonstrates that management can easily, without knowing, kill its own initiative by imposing restrictions on the communities.

4.3 Keys for success

The following keys to success summarize our recommendations based on the experiences thus far in the community initiative at Objectnet:

- Management support and acknowledgement – management support is essential in establishing communities. Management must work continuously to raise and keep the status of the communities, and be very careful not to restrict the communities.
- Self-authority in communities – the communities must be allowed to evolve and change rather freely. Each community must have the chance to proceed tentatively and experiment.
- Face to face communication - meetings or workshops is essential to establish a well functioning community (at least in the initial phase)
- The role of a community coordinator – the responsibility of this role is to facilitate meetings and workshops and drive the community forward in the initial phase. The members of the community may take separate turns in filling this role.
- State relevance to company strategy and added value for the organisation – seeing their efforts contribute to added value is a major motivation factor for the community members.
- Reward system – state if and how participation in the communities will be rewarded to avoid uncertainties.

5 Conclusion

The way the communities formed was not spontaneous. We believe our experiences indicate that our initial activities were a nice approach to encourage the formation of the communities and to draw attention to cooperative learning and knowledge sharing. The initiative to form communities has shown that with a small amount of effort it is possible to initiate groups that enable sharing of insight and experience. The groups continue their work with no dismissal dates, and now change and grow on their own initiative.

The varying degree of activity within the communities also shows that they are easily influenced by a lot of factors naturally present in an organisational environment. The company must continue by looking for ways to support and facilitate the groups even better.

6 Literature

- [1] Communities of Practice Learning as a Social System, Etienne Wenger, <http://www.co-i-l.com/coil/knowledge-garden/cop/lss.shtml> (published in the "Systems Thinker," June 1998)
- [2] Communities of Practice An Introduction, Michael McMaster, <http://www.co-i-l.com/coil/knowledge-garden/cop/mmintro.shtml>
- [3] Wikipedia: <http://en.wikipedia.org/wiki/Wiki>

[4] SPIKE (Software Process Improvement based on Knowledge and Experience) home page: <http://www.abelia-innovasjon.no/>

7 Author CVs

Hege Dreiem

Since graduating from University of Oslo in 1998, Hege Dreiem has been working as a consultant in the IT industry, developing object-oriented systems. Hege has worked on projects within finance, telecom and public sectors. Hege has worked with different software improvement initiatives over the last years, especially with regards to development processes, process guides and estimation techniques.

Rune Myrdal

Rune Myrdal is a senior consultant in Objectnet. He has been involved in Java and J2EE application development for six years, focusing on development productivity, quality assurance and process improvement. Currently Rune is working for the Research Council of Norway, developing Internet based research administration including portal, Content Management and electronic application handling.

Kjetil Jørgensen-Dahl

Kjetil Jørgensen-Dahl has been working with systems development for seven years (six years with Accenture). His interest fields are architecture and methodology. Kjetil has experience from finance, telecom and public domain. Kjetil has a msc/siv.ing degree from NTNU (Norwegian University of Science and Technology).

Process Improvement is Not Enough – Do not Forget Innovation Processes

Richard MESSNARZ¹, Romana Vajde Horvat², Katja Harej², Eva Feuer³

¹*ISCN GesmbH, Schieszstattgasse 4, A-8010 Graz, Austria*

Tel: +43 316 811198, Fax: + 43 316 811312, Email: rmess@iscn.com

²*CIT, Gosposvetska cesta 84, 2000 Maribor, Slovenia*

Tel: +386 2 228 2351, Fax: +386 2 35 5134, Email: cit@cit.si

³*MTA SZTAKI, Kende U. 13-17, H-1111 Budapest, Hungary*

Tel: + 36 1 279 6285, Fax: +36 1 279 5269, Email: eva.feuer@sztaki.hu

Abstract

This paper discusses different studies about innovation management which have been performed between 1996 and 2003 and describes key success factors of innovative organisations. Innovative organisations rule the market. It is of vital importance to install strategies which allow both, encourage innovation and pursue process improvement and standardisation at the same time.

Success factors of innovative organisations are taken into account for the design of an innovation manager skill card in the EU Leonardo project ORGANIC (2003 – 2006). This skill card is currently being configured into a skills portal and allows managers to browse required skills, do a self assessment, and receive guidance.

The paper will present case studies (a) where process improvement would have failed if innovation would not have happened on the product or technology side, and (b) where innovation would not have happened without a process improvement strategy.

Finally it summarises a list of key criteria which organisations have to consider when learning, continuously improving and innovating in nowadays information society

Keywords

Process Improvement, Innovation, Networking

1 *European Innovation Management Studies*

1.1 **Best Regional Innovation Transfer Study**

A study in 1998 (EU Leonardo da Vinci Project BESTREGIT – Best Regional Innovation Transfer, 1996 – 1999) analysed how innovative organisations operate and compared 200 organisations in Europe ([1], [5],[7]).

The study outlined that innovative organisations

- invest time and money into the understanding of the fundamentals of the forces of change,
- understand the different cultures through personal contacts using networking at a personal level,
- study trends and they are always up-to-date,
- concentrate their energy in areas where they excel or where no one else can operate,
- outsource all other non-core activities,
- are practical users of information technologies and have information technology strategies in place

This study illustrates the development in the mid 90s where organisations became more networked, the market became more global [10], and the learning and innovation is largely influenced by a networked learning society.

1.2 Team-working and Team-Learning Study

In a project TEAMWORK ([5], [6]) with 13 partners from 7 countries (IST-2000-28162 TEAMWORK, 2001 - 2003) a generic platform has been developed that shall support these networked team-working and team-learning and tested this platform with teams from 59 organisations in 13 countries of Europe. The working behaviour of the users (team-working and team-learning members of the networked platform) has been analysed and a study with key success factors for social team-learning and team-working has been produced as a project deliverable. There were 42 different projects running through the system using the defined environment and managed by a virtual team leader. The team size of each project varied between 13 and down to 2 different organisations involved.

This study based on the situation at the end of the 90s where developments were not done any more by single operations but a supply chain or a partnership of different companies integrating joint solutions. And these networked teams were multinational, multicultural, and consisted of members of different organisations. So the question was "How to manage the complexity of the distributed, multinational team-working and team-learning partnerships?".

The social experts team in the TEAMWORK project applied the social patterns analysis methodology [4] and also performed a set of interviews which were analysed to draw a number of conclusions concerning the social factors influencing team-working. This methodology analyses and selects typical patterns of behaviour.

Some selected results of the 2003 study about organisations were:

Management Style Related

- It was significant (statistical analysis) that people were happy with the more democratic distributed team management and team learning style of a TEAMWORK environment. Actually people from university, public service, and research centres were happier, than from large enterprises.
- We have found, that there was a significant difference how people from university, public service, and research centres feel the effect of such this tool to enhance teamwork spirit, for managers from large enterprises it was felt more that it gives negative affect to their influence.

The difference between research and large traditional companies was due to the fact that if you set up a team and you involve a large organisation it is not sufficient to just define your agreed team-working processes, but also to integrate properly the existing hierarchies inside the collaborating large organisation.

Innovation Related

- It was significant that people from university, public service, and research centres believed that a TEAMWORK environment encouraged an innovative approach to solving problems together, and creating new ideas together, people from traditional large enterprises believe it might interfere with existing procedures.
- It was significant that people from university, public service, and research centres believed that e-working helped them to improve the understanding of team roles.

It was an observation that larger organisations want to have more impact and decision power on projects and innovations, while the distributed team-working concepts (role based) only gave them equal weight among all team members (independent from their size).

Some selected results of the 2003 study about individuals were:

Community of Trust

„If you are building a human networked organisation you must have a foundation of trust and respect for effective communication at levels deep enough to sustain growth.“

Work-Flows Inward

„If you want information to flow to the innovative producing roles in an organisation then put the developer at the centre and see that information flows toward the centre, not from the centre.“

This means that innovation is carried out by a selected set of highly intelligent people (producers) who should not be hindered by too many hierarchies above them but become a central role in a networked organisation, to unleash the innovation power.

Altogether the study produced some 200 criteria and proposed a list of 17 criteria which shall be considered by managers [3],[6].

1.3 Innovation Management Study at 124 Multinational Large Companies

While these first two studies were carried out with involvement of research centres, SMEs and to some extent by large companies, the third study has only been performed at 124 very large multinational companies. Thus it largely represents now the viewpoint on innovation by large co-operations [8].

Large co-operations had the following expectations from learning organisations:

- Learn faster than competitors
- Balance teamwork with personal performance
- Reward high performers
- Develop ambidextrous thinking
- Transformational rather than incremental change
- Create and exploit new knowledge
- Ensure customer & company-wide diffusion of ideas
- Inherently entrepreneurial with a high risk tolerance

The favourite approach to implement innovation in a large corporation were

- Pilot programme with review 30%
- Company wide process with review 18%

- Systematic implementation 15%
- Project by project 14%
- Area specific 9%
- Benchmarking with competitors 5%
- Don't know 9%

And the drivers that influenced the success of an innovation were

- Executive management commitment 26%
- Supportive organisational structure 17%
- Positive culture with global reach 15%
- Young technically literate employees 13%
- Realistic milestones - fewer layers 6%
- Don't knows 13%

The overall model for the implementation of an innovation management programme contained seven phases :

1. Understanding the learning antecedents (nature of global business, anthropomorphism, dissatisfaction with traditional paradigm, customer responsive culture, intellectual capital)
2. Creating an environment of innovation
3. Analysing the perceived need and the required learning mechanisms
4. Executive challenge and implementing a learning processes
5. Including cultural and human factors and joint mission building
6. Organisational wide learning
7. Sustaining a learning organisation

And already in the first phase a number of parameters appeared which seemed to deal with factors that influenced the difference in opinion between large organisations and SMEs, research centres and universities in the study mentioned in section 2.2 of this paper.

A large co-operation very much focuses on the protection of the intellectual capital and global business opportunities and thus a networked learning and working across organisations needs additional controlling parameters to ensure the commitment from top managers. And as we see in the study the top managers by 26% have the largest influence on the success of an innovation in a large co-operation. (compare with results in previous chapters).

2 Process Improvement Case Studies

2.1 SPI and Product Innovation Example

All software developers in an organisation were originally hardware engineers and started to develop software without using efficient software specific design methods. They even invested effort in software requirements analysis, testing, configuration management, and maintenance, but neglected architectural software design. This did not create a problem as long as the program they developed (and

needed in the embedded systems) was quite small. However, during 10 years the size of the software grew and due to missing architectural design the product had more and more a monolithic structure. If you changed parts of the system it was not visible which and how many other parts were affected. Based on this situation the continuously incoming customer wishes and maintenance activities nearly always caused a number of additional unexpected change activities. All parts of the system were so dependent on each other that any change led to a re-compilation of most parts of the entire product. (case study published in [11]).

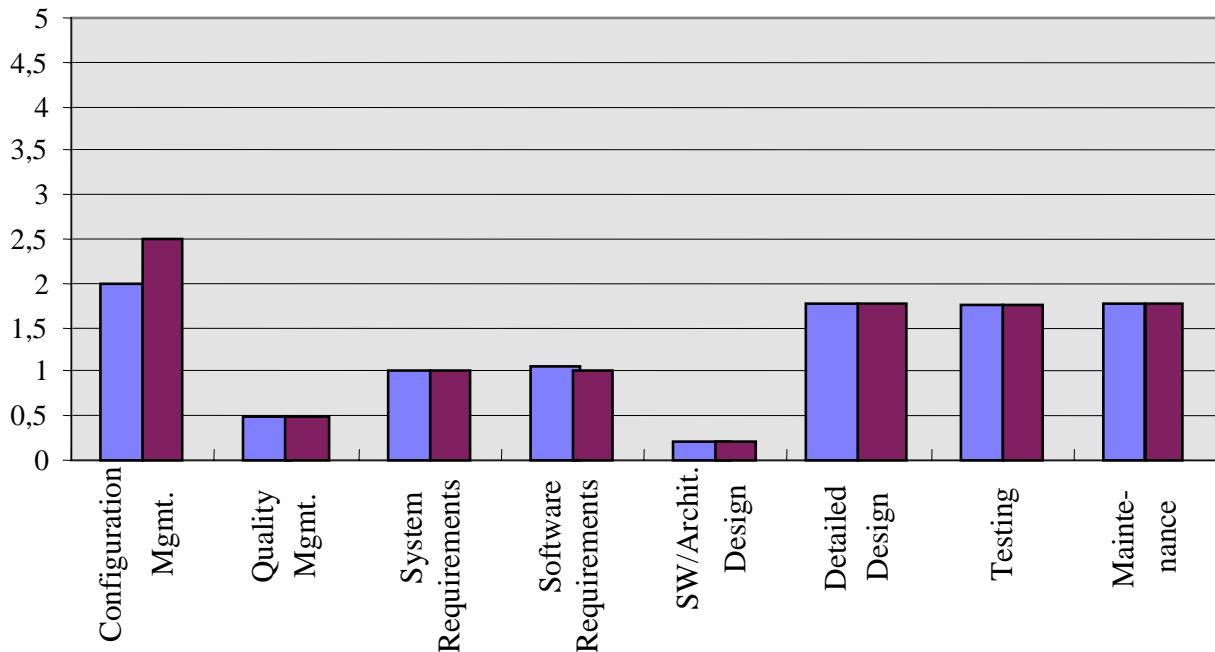


Figure 1: A typical assessment result from the case study organisation

A critical problem was the fact that the current technology was expected to be out of the market in about 2 years, and for a new technology development all resources were eaten up by the maintenance problem.

To define in this situation to more document quality management or to define more procedures would have requested for further effort which would not directly attack the root cause problem. So in reality the problem was solved in the following stages:

1. Analysis of the system and re-design to understand better the existing software, which led to 2 staff less needed for the maintenance due to large reduction of time needed to fix errors.
2. Starting a pilot developing a new open architecture based system which incorporated a central database system with standardised interfaces to control programs. Two of three major components of the former software could be re-used.
3. Design of a role (teamwork) models focussed quality system.
4. Accreditation against ISO 9001 and re-assessment after 2 years with new system and new quality models.

If process improvement would have kept the old system (running out in 2 years) and would have demanded just additional control and documentation effort, the division/product would have vanished from the market.

2.2 SPI and Networking of Knowledge Example

Some areas in SPI are more complex than others and cannot be solved without large system engi-

neering support. In a German initiative SOQRATES (www.soqrates.de, 2003 – initial phase, 2004 – 2005 multi-company task forces) a group of leading engineering firms joined forces to exchange capability profiles and to analyse synergies (where one can learn from the other one) and areas of common interest. In synergy and common interest areas task forces are build to enable a system-based and innovative approach to reach a capability level 3 in ISO 15504 using an innovative approach which brings competitive advantage against firms outside the group.

This is based on principles , such as

- Involvement of synergy team leaders (those who are leaders in this field)
- Involvement of a key method provider
- Involvement of quality experts
- Applying win-win (if you benefit from a team, you must be adding knowledge required by the others in another task force)

• Innovation Task Forces

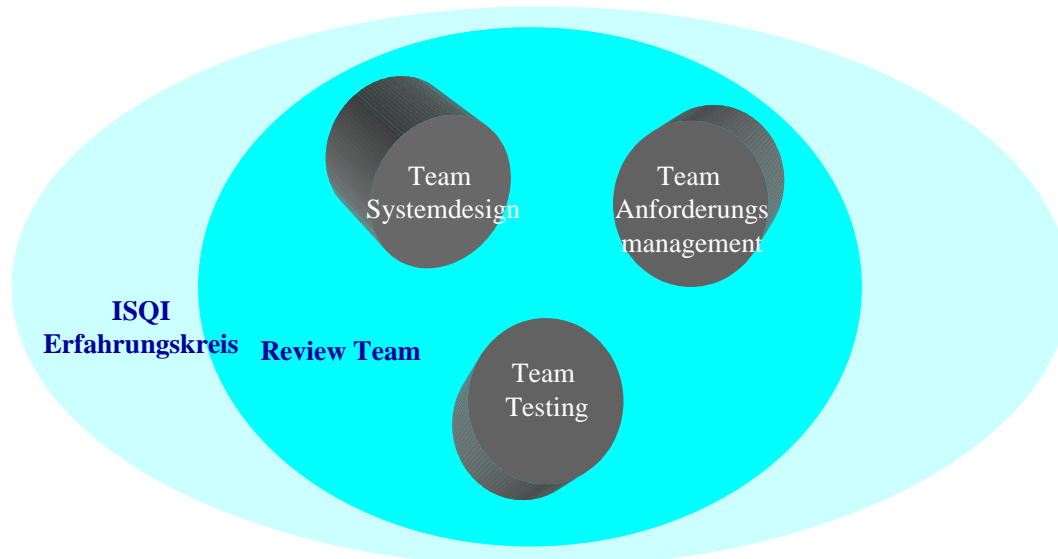


Figure 2: SOQRATES Task Forces (Knowledge Sharing) Concept

This cross-organisational learning approach is applying also the product innovation strategies (see section 2.1). For instance, the system design team introduces new methods to automatically re-engineer design structures for the old legacy code. The design structures can then be included in a new platform based design (re-use on design and not code level).

2.3 SPI and Process Innovation Example

A typical situation in embedded engineering at the end of the 90s was that a waterfall model would not work. Too many requirements come in by the manufacturers (and with big pressure with suppliers acting in a supply chain) on a continuous basis and it is a problem to manage these requirements and trace them. Studies showed that only approx. 50% are known at project start, around 80% are known after the first months, and the rest of requirements come in on a continuous scale. This is not caused by a non-complete requirements situation but by the field competition. E.g. an executive of firm X tries the equipment of manufacturer Y , finds out a competitive advantage of Y , and then immediately requests to implement that for X. A project at the end is successful not only by technical means but if it gains a market share. So from a manufacturer viewpoint a technical correct system that misses parts which lead to a failure on the market will be non-acceptable. Thus driven by the competition this dy-

dynamic requirements behaviour of manufacturers will most probably never change.

Thus at the start of 2000 many leading embedded firms invested into change management systems, where quarter annual software releases are packaged and traced (leading to a SPIRAL model) where the software releases are growing into a system release. Then a requirements document is a query in the system concerning a specific release and the print out of a bound report of the query result.

So it would have been wise for assessors in the beginning of 2000 to accept system based arguments and not search just for a specific document.

3 The Innovation Manager Skills (Eu Project)

The job role of an innovation manager (see Figures 2, 3) is a specific position in an organisation who understands all these factors: networking, learning, organisational and social factors and is able to encourage, promote, manage, control, and disseminate innovation strategies in an organisation.

For structuring a skills set the EU leonardo da Vinci project ORGANIC followed the EU standards for skills cards [2].

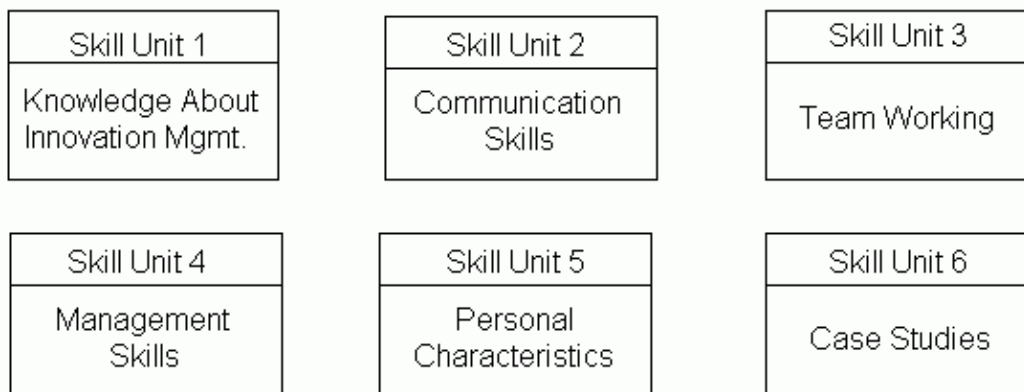


Figure 3: Learning Elements of the Innovation Manager Skill Card

For each learning element ORGANIC develops a training module. A skills portal has been configured with the skills card and supports the steps of browsing required skills (www.innovationmanager.org), self assessment, formal assessment, evidence collection, generation of skills profiles, and learning recommendations.

4 Conclusion

To make the right decisions for process improvement it is not sufficient to just know all the standards. It requires a profound understanding of the goals and business aspects of the organisation and the relationships between innovation potentials and process improvement demands.

We acknowledge the support for this paper from the following project partners from ORGANIC: D. Ekert (ISCN), B. Wöran (Danube), K. Ennsfellner (IMC Krems), A. Gvishiani (EDNES), N. O'Shea (Tecnet), G. Velasco (FGUVA). We acknowledge the support for this paper from the following project partners from EPI: Miklos Biro (BUESPA) and Gearoid O Suilleabhain and Ray Coughlan (CIT-IRL).

5 Literature

- [1] M. Biro, R. Messnarz, A. Davison (2002) The Impact of National Cultures on the Effectiveness of Improvement methods - The Third Dimension, in Software Quality Professional, Volume Four, Issue Four, American Society for Quality, September 2002
- [2] Feuer E., Messnarz R., Best Practices in E-Commerce: Strategies, Skills, and Processes, in: Proceedings of the E2002 Conference, E-Business and E-Work, Novel solutions for a global networked economy, eds. Brian Stanford Smith, Enrica Chiozza, IOS Press, Amsterdam, Berlin, Oxford, Tokyo, Washington, 2002
- [3] Feuer E., Messnarz R., Wittenbrink H., Experiences With Managing Social Patterns in Defined Distributed Working Processes, in: Proceedings of the EuroSPI 2003 Conference, 10-12 December 2003, FTI Verlag, ISBN 3-901351-84-1
- [4] Project EASYCOMP (IST Project 1999-14191, homepage: <http://www.easycomp.org/>)
- [5] Messnarz R., Stubenrauch R., Melcher M., Bernhard R., Network Based Quality Assurance, in: Proceedings of the 6th European Conference on Quality Assurance, 10-12 April 1999, Vienna, Austria
- [6] Messnarz R., Nadasi G., O'Leary E., Foley B., Experience with Teamwork in Distributed Work Environments, in: Proceedings of the E2001 Conference, E-Work and E-commerce, Novel solutions for a global networked economy, eds. Brian Stanford Smith, Enrica Chiozza, IOS Press, Amsterdam, Berlin, Oxford, Tokyo, Washington, 2001
- [7] A Learning Organisation Approach for Process Improvement in the Service Sector, R. Messnarz, C. Stöckler, G. Velasco, G. O'Suilleabhain, A Learning Organisation Approach for Process Improvement in the Service Sector, in: Proceedings of the EuroSPI 1999 Conference, 25-27 October 1999, Pori, Finland
- [8] O'Keefe, T., & D. Harrington, 2001. Learning to Learn: An Examination of Organisational Learning in Selected Irish Multinationals. Journal of European Industrial Training, MCB University Press, Vol. 25: Number 2/3/4
- [9] DTI - Department of Trade and Industry UK, British Standards for Occupational Qualification, National Vocational Qualification Standards and Levels
- [10] Gemünden H.G., T. Ritter, Inter-organisational Relationships and Networks, Journal of Business Research, 2001
- [11] Messnarz R., Tully C. (eds.), The PICO - Book: Better Software Practice for Business Benefit - Principles and Experience, IEEE Computer Society Press, in publication

6 Author CVs

Dr Richard Messnarz

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

- PICO - Process Improvement Combined Approach 1995 - 1998,
- Bestregit - Best Regional Technology Transfer, 1996 - 1999,
- TEAMWORK - Strategic Eworking Platform Development and Trial, 2001-2002,
- MedialSF - Eworking of media organisation for strategic collaboration on EU integration, 2001-2002

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

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

He is a lead ISO 15504 assessor. He has worked as a consultant for many automotive firms, such as BOSCH, ZF TE, ZF N, Continental TEMIC, Audi/VW, etc. He is a member of the ITACS accreditation board, he is an initiator of the German MDISQ (www.mdissq.de) initiative, and he is the technical moderator of the SOQRATES initiative (www.soqrates.de).

Prioritizing SPI actions based on business needs and impacts

Matti Haavikko⁽¹⁾, Mika Johansson⁽²⁾, Risto Nevalainen⁽²⁾

⁽¹⁾ Tampere University of Technology, Finland

⁽²⁾ Finnish Software Measurement Association FiSMA

Abstract

In this article we introduce an integrated approach for professional software process improvement. It is called GNOSIS, meaning "explicit knowledge". During years 1998 – 2004 Finnish Software Measurement Association FiSMA has developed several models for process assessment and improvement. In GNOSIS they are integrated as one family of methods, models and tools. Main elements of GNOSIS are 1) Process repository and Model Editor, implementing SPICE and CMMI elements, 2) Process Assessment System, implementing a full-scale tool to support SPICE and CMMI assessments and their combinations and 3) a knowledge-based SPI database and reasoning tool for professional SPI initiation and planning.

FiSMA has about 40 member companies in Finland. GNOSIS system has been used for process assessment already for some years. During 2004 also the other elements have been released for use. In this article we explain ...

Keywords: SPICE, CMMI, Business goals, Business impacts, SPI Initiation

1 GNOSIS Approach to SPI

Many companies are “true believers” of models. Some are SPICE fans, some respect more CMMI. Some of them like to have only their own models, and want to invent them based on their real or thought special needs. In contrary with model believers, some other companies do not believe on any models at all. They may be interested on “agile methods” or “competence-driven software development” or what so ever.

In FiSMA we have all such kinds of members. So, we cannot support only one approach and neglect the others. As a starting point, we respect ISO based models and approaches, because we have long traditions in ISO9000 and ISO15504 related work. ISO-based approaches leave lot of space for interpretation and further method development. Current de-facto standard CMMI has also some benefits, and it is more commercial and packaged than ISO standards. Based on those reasons, we have developed a set of methods, models and tools to support several approaches. As a result, GNOSIS is a kind of “model portfolio” in which you can use only some models or all of them.

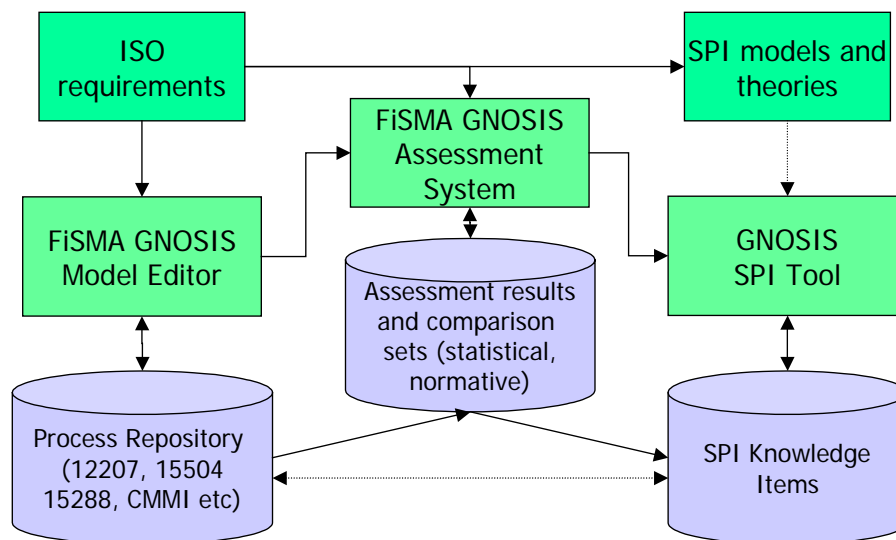


Fig. 1. Main elements of GNOSIS

The main elements of GNOSIS are:

- ❑ **Model Editor**, which implements all details of any SPICE or CMMI like model. Current content in Process Repository is SPICE and CMMI, but for example ITIL, ISO15288 (Systems Engineering Processes), ISO61508 based models (safety-critical systems) can be also easily implemented. Totally we have now about 100 processes. We have implemented also SPICE-based capability model, CMMI Staged and Continuous presentations and can implement any other hierarchy-based model.
- ❑ **Assessment System**, which implements SPICE- and CMMI-based assessment process, including scope definition, data collection (interview notes, document record etc), process rating and deriving of several high level capability or maturity based result compositions. We can also compare any assessment with statistical or normative reference sets, for example we could express a goal level for some company and then identify gaps between current and goal levels.
- ❑ **SPI tool**, which includes a search mechanism to Process Repository and additional knowledge items, to help in SPI initiation and planning. Assessment results can be an input, but also company goals, expected SPI benefits etc. For example SPICE and CMMI models are expressed in SPI database as conceptual networks, including their all internal and mutual interdependencies. SPI tool can prioritize and find most relevant SPI actions, based on those inputs. Typical result is a candidate set of improvement ideas and actions.

As can be easily seen, GNOSIS system is quite ambiguous. It will never be fully ready, and needs intensive development and maintenance all the time. FISMA community in Finland can do most of it, because member companies have already long traditions in using models and other SPI approaches. Also some national R&D projects are helping to improve GNOSIS system.

2 Goals and Impacts as Drivers of SPI

Business goals and development needs are the only sound starting point for SPI investments, for sure. That self-evident statement is not so easy to implement. You should be able to combine tacit personal and organisational knowledge with well-expressed models like SPICE and CMMI. You should also know quite well mechanisms how the improvement of each process impacts on business goals. Such mechanisms can be short or long term, and their effect can be only weak or strong.

We implemented SPI database as a conceptual network. Network nodes can be business goals and SPI impacts, SPICE or CMMI model elements and additional knowledge items. The relationships between nodes are expressed with a type and intensity. Each relationship type has a direction (one-directional, two-directional) and weight. As a result, we have a full conceptual network of all model elements, additional knowledge items and their links to business goals and impacts. The number of such elements is about 400 plus about 300 practice-level model elements.

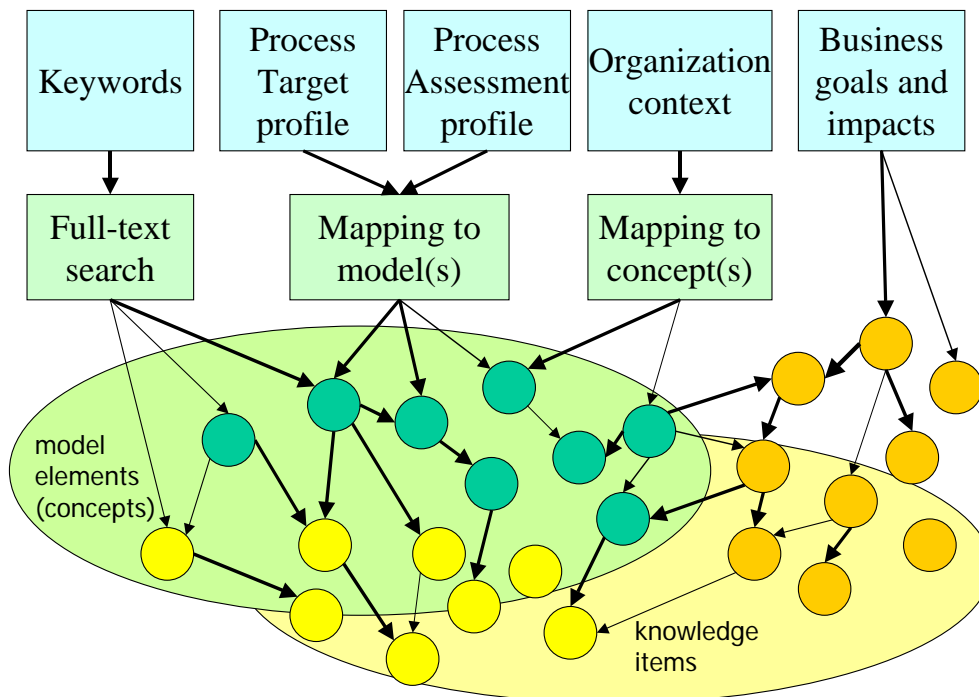


Fig. 2. User access mechanisms to SPI knowledge in GNOSIS

Each user has several ways to find and read content of our process repository and SPI database. It behaves like an “electronic book” which you can read chapter by chapter or by using keyword search. More structured approaches are for example gap between current and goal level of each process. For real end users maybe the easiest ways are access using organisational context or business goal/impact lists and ready-made links.

3 Modeling of each concept and knowledge item

The principal idea of conceptual modeling in GNOSIS is to express each knowledge item by using a design pattern type of template. Business goals and impacts are just defined and their mutual interdependencies are expressed by using relationship type idea. Goals and impacts are grouped by using BSC type classification, because most FiSMA companies are using BSC as an essential part of strategic planning and business management.

Assessments and other close cooperation in and between FiSMA companies have brought a lot of insider information about different business concepts, strategies and needs. During summer 2004 we wanted to deepen our understanding and did a survey of current business driven SPI needs and potential areas of improvement. We asked four different questions:

1. What business goals are most relevant from SPI point of view?
2. What benefits we should achieve through process improvement?
3. What are the most important areas of SPI at this moment?
4. What best practices you have, what could be beneficial for other FiSMA companies?

Some snapshots of this survey are in annex 1 of this article. Based on earlier experiences and survey results, we modeled most relevant business goals and impacts as one network. Because FiSMA companies have very different business strategies, we had to implement all of them in GNOSIS! So, at least product-based business, competence-based business, product delivery business and partnership businesses may find something predefined as a starting point for SPI. So far we have modeled 17 business goals and their place in our conceptual network.

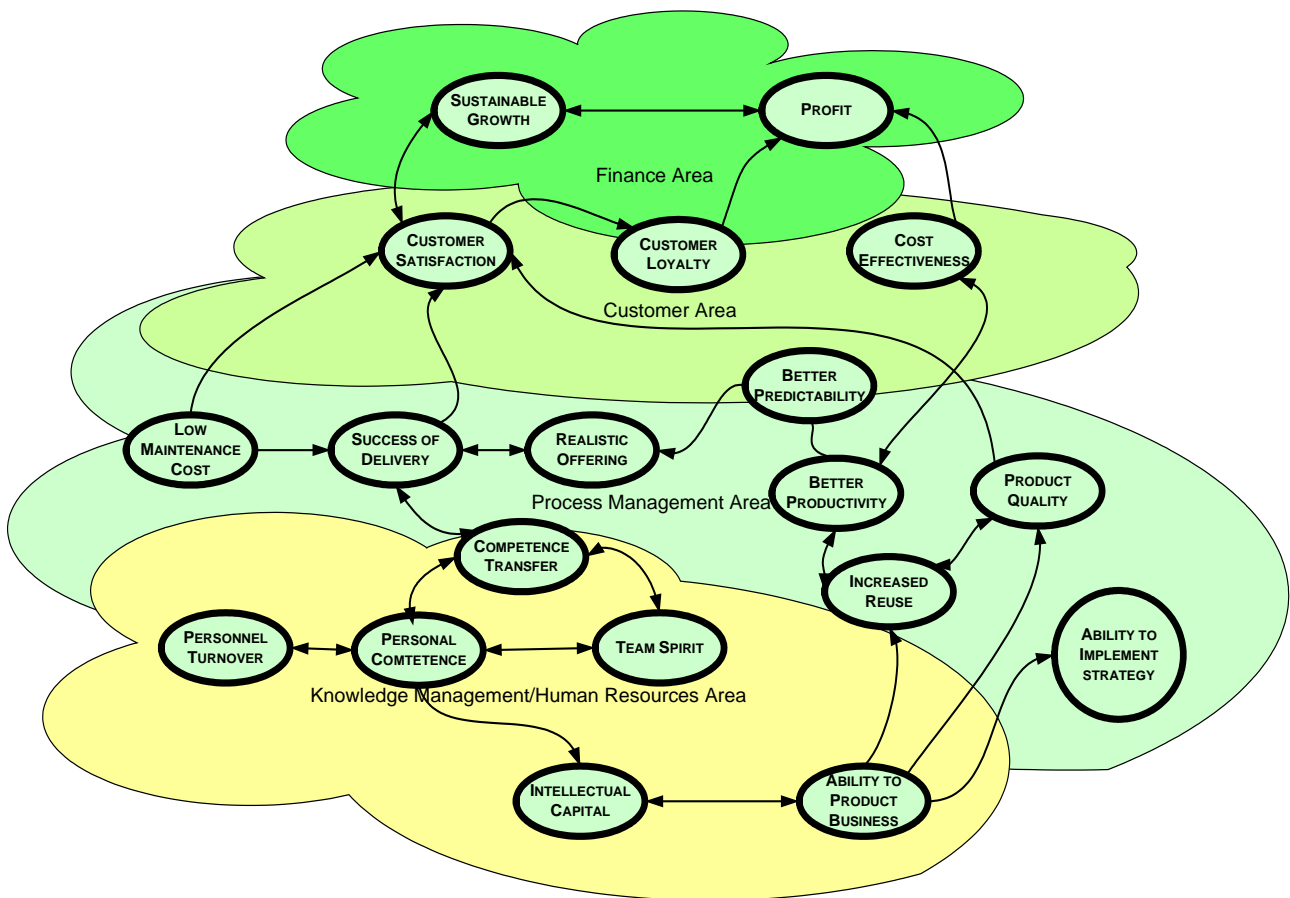


Fig. 3. Predefined goals and their most important dependencies in GNOSIS system

Relationships between business goals and SPI impacts are modeled similarly as business goals, see figure 4 as an example. We have predefined about 20 different impact types, and each of them is further linked with model elements and knowledge items, as needed.

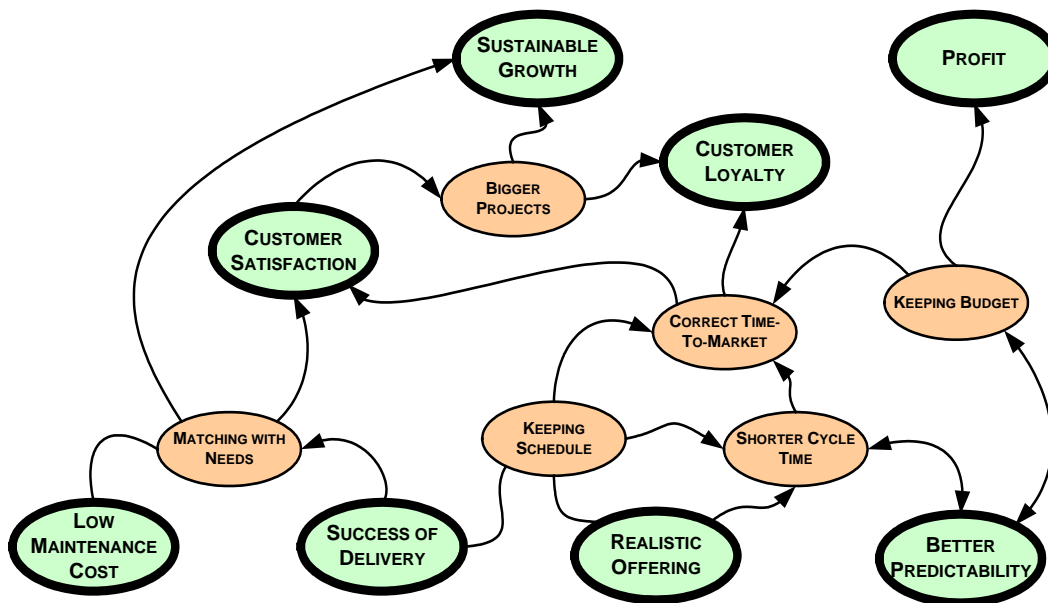


Fig. 4. A small sample of business goal vs. SPI impacts conceptual network

Conceptual networking continues similarly towards more detailed model elements and knowledge items. The main problem is to know impact mechanisms of each model element. For example, what is the impact to achieve capability level 4 of testing processes? Such relationships are modeled using best available knowledge, and a specific knowledge engineer user interface is developed for modeling and knowledge expression needs. This feature of GNOSIS is still under development, and needs also lot of further research.

4 Some experiences of GNOSIS

So far FiSMA companies have used quite lot of Process Repository and Assessment System. Process Editor is used almost only to create Process Repository. The idea is that each FiSMA member company can tailor model processes (SPICE and CMMI, for example) for their internal use. Also any additional processes can be created, for example sales, human resource management, software operation and software acquisition could be such topics.

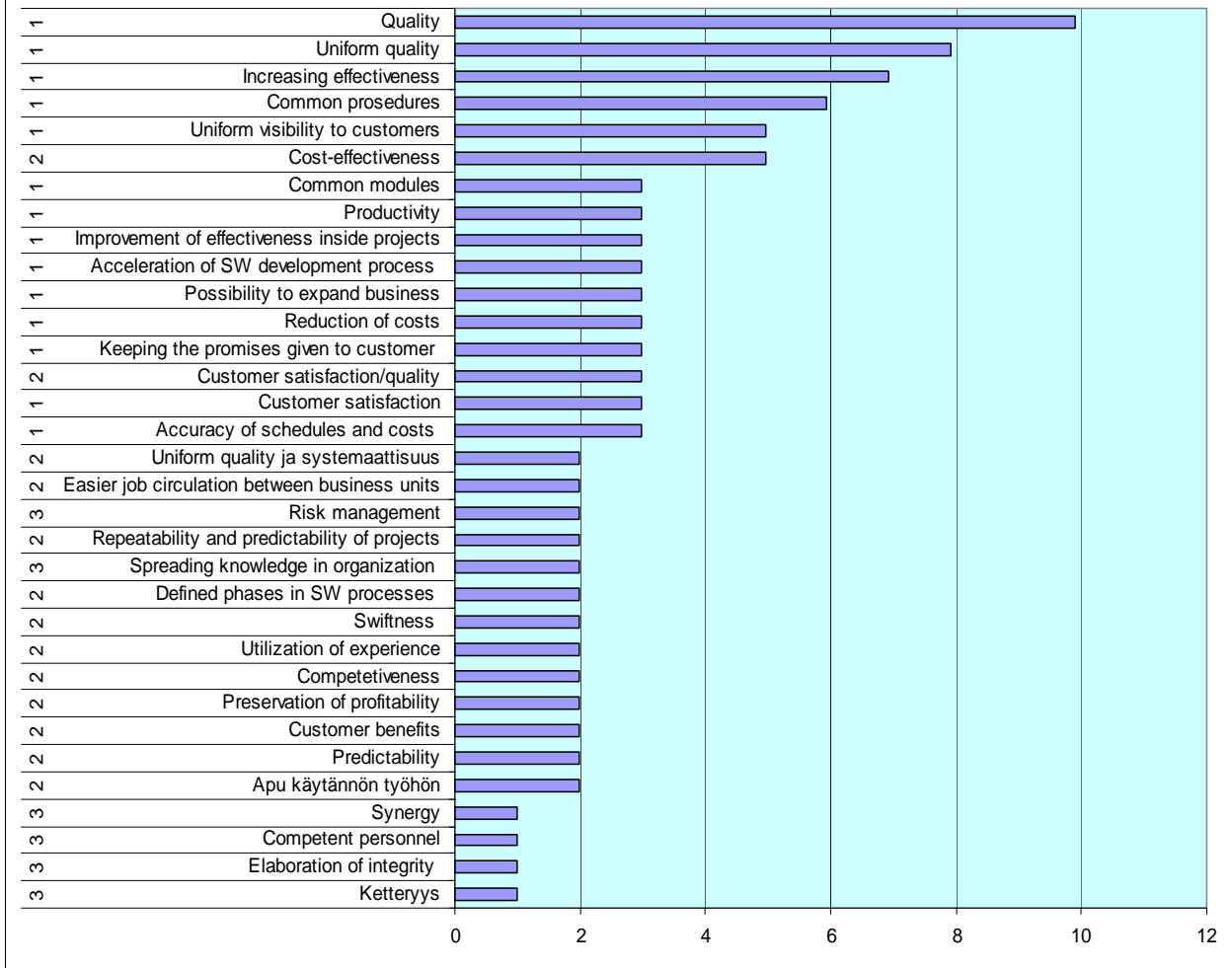
SPI tool is still in prototype and piloting phase. Several companies have piloted it and the general feedback is "exciting". Of course, the ultimate success criteria for our SPI tool is it's answering capability and that depends heavily on content. We are just running structured pilots to validate answering capability from algorithmic point of view. Content-based validation takes more time and will never be complete, because new needs and new experiences always evolve.

5 Conclusions

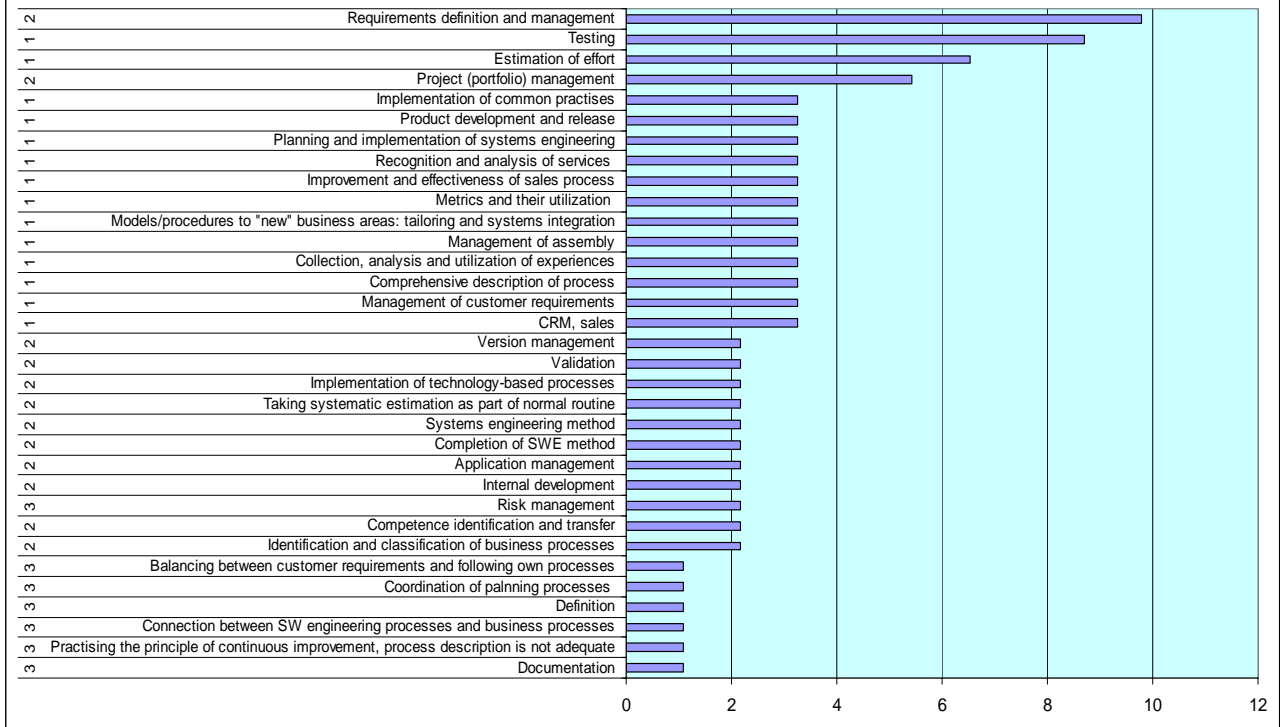
Annex 1. Some snapshots of FiSMA survey about SPI needs



1b. Benefits to be (or should be) achieved through development of processes



3. What things should be improved in your organization at the moment?



Software Productivity estimation based on Association Rules

S. Bibi, I. Stamelos, L. Angelis

Abstract

Software process improvement models have as a target to help software organizations produce successfully, under the expected quality and within time and budget constraints their projects. For this purpose various steps are suggested by these models for the improvement and measurement of the processes followed. One of them involves project planning which is strongly connected with software cost or productivity estimation. Target of this study is to provide a method that could be adopted by an organization in order to estimate the required productivity for the completion of a software development project. Association Rules (A.R) is a suitable technique capable of discovering knowledge concerning productivity. The proposed method is applied and evaluated on two different data sets, namely the COCOMO81 dataset and the Maxwell dataset. The evaluation shows that A.R is a promising method whose results can be confirmed intuitively.

Keywords

Cost estimation, Intervals, Association Rules, COCOMO.

1 Introduction

Software process improvement models include a framework for planning, managing, controlling, and improving the development, operation and support of software. Their target is to identify key process areas and practices that may comprise successful and within time and budget constraints projects. Common features of these models include ability to perform and measurement and analysis, two features that involve project planning and cost estimation. Analyzing and summarizing data for cost and schedule estimation and using collected metrics to calibrate and update software estimate models are key practices that could improve a part of the software process.

An estimation of productivity, effort or duration needed to complete a project would help organize and distribute the required work. The above estimation is characterized by difficulties such as poorly defined requirements, frequent staff turnover and volatile software platforms and therefore should take into consideration uncertainty and risk.

A technique that takes into consideration the above issues is Association Rules (A.R) [3]. A.R. can be applied on past historical data in order to discover knowledge concerning productivity and deals with uncertainty and risk in two different ways. The first one is presented by two probabilities that accompany each rule. These probabilities express the validity of the rule and its frequency in the dataset, providing a measure of appropriateness of each rule. The second way refers to the estimation of productivity intervals. Productivity values are quantified into categories having as a result the estimation of productivity intervals. Intervals give a pessimistic estimate and an optimistic estimate between which the actual productivity of a project may fall in.

Purpose of this study is to extract useful patterns from cost estimation data with the help of Association Rules and to provide some evidence of the prediction accuracy of this technique. Also some conclusions will be drawn concerning the factors that tend to affect productivity directly. The method is applied on two different data sets, namely the widely known COCOMO81 dataset and the Maxwell dataset.

Various studies have been conducted so far concerning the comparison and evaluation of different cost estimation techniques [4], [6], [8], [10], [11]. In particular, some of them suggest the estimation of intervals [1], [5], [12]. Effort estimation with the help of A.R is presented in [8] where a limited number of rules are extracted from decision trees. The results of rule induction are not encouraging as rules are used as representation method, not as a modelling technique. In our approach we propose that rules be used both as a modelling and representation method allowing the extraction of many useful patterns.

2 Modelling technique and results

2.1 Modelling technique and methodology

The modeling technique is Association Rules. A.R [3] belong to descriptive modeling and have as a target to describe the data and their underlying relationships with a set of rules that jointly define the target variables. An association rule is a simple probabilistic statement about the co-occurrence of certain events in a database. A simple association rule has the following form: IF A1=X AND A2=Y THEN A3=Z with probability p (Confidence). There is also one more probability that accompanies each rule: Support. Support is a measure that expresses the frequency of the rule in the whole data set. As a consequence, the rules that are extracted from a dataset are ranged hierarchically according to their confidence at first and then by their support.

The data sets used in order to extract A.R with the help of an open source tool [7] are COCOMO81 data set and the Maxwell data set. The data sets are analytically presented in [6], [9].

Before extracting A.R, productivity that is a continuous variable, had to be quantified into discrete categories. We preferred to consider intervals that may be appealing to software managers: relatively few intervals were chosen (because of the low number of projects in the datasets) with rounded lower and upper limits, so as to be easily identifiable by a human. Productivity intervals for both data sets are presented in [1].

2.2 Results

The accuracy metrics that will be used in order to evaluate the results of each model are the Mean Magnitude Relative (MMRE), the PRED(Y) and the hitrate.
$$MMRE = \frac{100}{n} \sum_{i=1}^n \frac{|P_i - E_i|}{P_i}$$
 where P_i is the actual productivity and E_i is the estimate and n is the number of projects. PRED(Y) is the percentage of projects (k) for which the prediction falls within the Y% of the actual value. Relative errors are calculated by considering the mean of the interval in order to derive a point estimate from an interval estimate. Finally, hitrate will be used [5] i.e. the percentage of projects for which the correct interval has been successfully estimated. Usually the validation of a model is done by removing one data point at a time from the data set, recalculating the model and estimating the value of the project that was left out (a method known as JackKnifing).

While extracting A.R the following issues arose: The most representative and powerful, rules should

be selected. In addition, the selected rules should be able to provide estimates for all possible projects. In order to achieve this, the rules had to be as general as possible, with few constraints in their rule body, so as, given the attributes of a new unknown project, to be able to provide an estimate. For that purpose, rules with high confidence and as high support as possible have been preferred. In order to satisfy the second constraint, rules with few attributes in the rule head were selected so as to avoid over specialization. Eventually, for COCOMO81 data set 36 rules were selected with support threshold 4.7% (3 projects), which were used for the evaluation of the model and confidence threshold 50%. For instance, two rules concerning two categories of productivity are the following in priority order (PROD_i corresponds to productivity interval *i*, The intervals are presented in [1]):

Support	Confidence	Rule Body	Rule Head
6.3	80.0	NOM +H_VH_EH_SH_RVOL+ H_VH_TURN	==>PROD_3]
6.3	66.6	ACAP_H + DATA_N	==> PROD_4]

As an example, the second rule can be interpreted as follows: When the programmers' analysis capability is high and the database size is nominal then the productivity is likely to be in the fourth category ($100 < \text{PROD} < 160$). This pattern is presented in 6.3% of the dataset projects (4 projects) and 66.7% of the projects that present ACAP high and DATA nominal fall into the fourth category of productivity (4 out of 6 projects). The rule implies that, in the given cost database, the combination of these two conditions is sufficient to suggest a plausible productivity level, with a certain probability. The estimate is probabilistic in nature because in various cases other project factors may also rise or lower productivity. Notice that a new project with attributes that satisfy both rules will actually receive two estimate intervals from which the one is more likely to appear than the other. For both datasets the evaluation results are presented in table1.

Table 1: Results of the model for both datasets.

	COCOMO81 Jackknifing method	Maxwell data set Projects with starting dates 1992,1993 are esti- mated	Maxwell data set JackKnifing method
HITRATE	76.1	66.6	63.33
PRED(25) %	63.4	75	60
MMRE %	29.5	23.5	42.5

While observing carefully the extracted rules it appears that RELY, MODE, CPLX, PCAP and LEXP are the attributes that appear in the majority of the rules defining the productivity of a project. PRED(20)= 55.5 is also calculated in order to compare the results with other studies. The original Intermediate COCOMO81 model, estimating project effort, has a PRED(20) equal to 68 percent and a MMRE equal to 18.4%. However, this model is constructed in an *ad hoc* way not easily repeated in other databases. In [6], where Forward Pass Residual Analysis was performed the results indicate a MMRE of 36%,and a PRED(20) of 49% .

In the Maxwell dataset, 36 rules are extracted with support threshold 5.0% (3 rules) and confidence threshold 40%. The same criteria used in the COCOMO81 dataset are used in order to prune the rules. The are presented in table 1.

Frequently met attributes in the rules are CPLX, Installation Requirements, Efficiency Requirements, Staff Tool Skills and Staff Team Skills.

In [9], where the dataset was published, regression was applied on the projects with starting dates

before 1992 and the projects with starting dates 1992,1993 were evaluated. For this model, the dependent variable was project effort. The model is within 25 percent of the actual effort 58 percent of the time and in that case has a MMRE equal to 32%.

It should be mentioned that, when leaving out of the study the most often observed variables, the evaluation results were disappointing indicating that some times the rules suggest causal relationships.

3 Conclusions and future work

In this paper Association Rules are proposed for creating a model predicting productivity. A.R have been implemented and evaluated on two public datasets, testing the accuracy of the method and its suitability.

In both data sets the results are competitive with those obtained through conventional techniques and promising for further evaluation. Because the models proposed in [2], [6] and [9] predict effort and not productivity, only tentative comparisons may be made. In the case of the COCOMO81 dataset the prediction accuracy of A.R. is slightly lower than that of the original COCOMO81 intermediate model. However, it must be stressed that the latter is an ad hoc model, and cannot be validated through JackKnifing. Regarding the model presented in [6] A.R produce competitive results with a noticeable improvement in the estimation accuracy. In the case of the Maxwell dataset, A.R. JackKnifing accuracy is relatively low, but is acceptable for the 1992, 1993 projects and much better than that of Maxwell's model. In addition, when Hitrate is considered, which is a more appropriate and fair method to evaluate interval estimations, the results show that in the majority of cases, A.R predictions fall into the correct interval of productivity and are able to guide software managers in staffing their project and defining the time schedule of their project.

Regarding the advantages of A.R, it should be pointed out that they are one of the most expressive and human readable representations for learned hypotheses in sets of if-then rules, expressing uncertainty in many ways. First, by considering productivity intervals, and secondly, by characterizing each rule with two probability values. Also A.R's performance can be improved easily by using expert judgment as a support for pruning of the final rule set and for the initial selection of productivity intervals.

An issue that deserves further attention is the fact that the pruned A.R cannot cover all the cases that may appear. In addition, A.R present the usual drawback of all machine learning techniques, i.e. the possibility of over specialization of the training data. Future research also needs to focus on confirming and enriching the results of A.R in larger, multi-organizational datasets, such as those coming from ISBSG .

4 Literature

1. Bibi, S., Stamelos, I. Angelis, L.: Software Cost Prediction with Predefined Interval Estimates, 1st Software Measurement European Forum, Rome, Italy, January 2004.
2. Boehm, B.: Software Engineering Economics, Prentice-Hall, NJ, (1981)
3. Hand, D., Mannila, H., Smyth, P.: Principles of Data Mining, MIT Press, US, (2001)
4. Jeffery, R., Ruhe, M., Wiczorek, I.: Using Public Domain Metrics to Estimate Software Development Effort, in IEEE 7th International Software Metrics Symposium proceedings, London UK (2001) 16-27
5. Jorgensen, M.: An effort prediction interval approach based on the empirical distribution of previous estimation accuracy, Information and Software Technology 45, (2003) 123-126
6. Kitchenham, B.: A procedure for analyzing unbalanced datasets, IEEE Transactions on Software Engineering 24 (4), 1998 278-301
7. Machine learning software in Java, <http://www.cs.waikato.ac.nz/ml/weka/>.

8. Mair, C., Kadoda, G., Lefley, M., Phalp, K., Schofield, C., Shepperd, M., Webster, S., "An investigation of machine learning based prediction systems", *Journal of Systems and Software* 53, (2000) 23–29
9. Maxwell, K. *Applied Statistics for Software Managers*, Prentice-Hall, N. J., (2002)
10. Briand L., Emam, K., Surmann, D., Wiecek, I. "An Assessment and Comparison of Common Software Cost Estimation Modeling Techniques", in *21st International Conference on Software Engineering proceedings*, Los Angeles, California, United States (1999) 313-322
11. Srinivisan, K., Fisher, D., "Machine learning approaches to estimating software development effort", *IEEE Transactions on Software Engineering* 21(2), 1995, pp. 126-137.
12. Stamelos, I., Angelis, L., Dimou, P., Sakellaris, E., "On the use of Bayesian belief networks for the prediction of software productivity", *Information and Software Technology* 45, (2003) 51-60.

5 Author CVs

Stamatia Bibi

Bibi Stamatia received her BsC from the department of Informatics at the Aristotle University of Thessaloniki, Greece. Currently, she is a PhD student at the Software Engineering Laboratory of the same department. Her research interests involve software metrics and software process improvement. In particular she is comparing estimation methods and is pursuing novel estimation techniques for software cost prediction. In that direction, she has published a study on the application of Bayesian Belief Networks for cost estimation and another one on the estimation of the development cost for an open source ERP system.

Ioannis Stamelos

Ioannis G. Stamelos is an Assistant Professor of computer science at the Aristotle University of Thessaloniki, Dept. of Informatics. He received a degree in Electrical Engineering from the Polytechnic School of Thessaloniki (1983) and the Ph. D. degree in computer science from the Aristotle University of Thessaloniki (1988). He teaches language theory, object-oriented programming, software engineering, software project management and enterprise information systems at the graduate and postgraduate level. His research interests include empirical software evaluation and management, software education and open source software engineering. He is author of 50 scientific papers and member of the IEEE Computer Society.

Lefteris Angelis

Lefteris Angelis received his BSc and Ph.D. degree in Mathematics from Aristotle University of Thessaloniki (A.U.Th.). He works currently as an Assistant Professor at the Department of Informatics of A.U.Th.. His research interests involve statistical methods with applications in software engineering and information systems, computational methods in mathematics and statistics, planning of experiments and simulation techniques.

Navigating the Minefield -- Estimates Before Requirements

Carol A. Dekkers, CMC, CFPS
President, Quality Plus Technologies, Inc.
Seminole FL USA

Abstract: Sophisticated parametric-based project estimating models are becoming the norm in IT (information technology) industries providing increasingly realistic estimates of cost and schedules. At the same time, however, agile and other eXtreme development methods challenge the traditional waterfall assumptions of estimating models by pushing back the requirement for project estimates much earlier in the software life cycle. What happens when project estimating moves back a full phase from after to before requirements? How can acquisition managers, contractors, auditors and financial analysts develop and support estimates based on yet to be named projects with virtually unknown requirements? This paper outlines how one can create auditable estimates based on identifying and documenting assumptions. The process outlined creates a logical and traceable project floor plan and maps potential “landmine” locations (calculated risks) that further substantiate the preliminary estimates. Whether you are an experienced estimating professional or contract manager, this paper provides a basis to support your work and provides the basis for dialog and discussion of early estimates among project participants.

Introduction:

Estimating the effort, duration or cost of a software project is often a daunting task, especially considering the myriad of estimating models and tools available. Complicating matters are up to 200 input variables depending on the model chosen, and these cover the entire spectrum of functional, quality, design, and technical drivers for an estimate. Despite these complexities, software project managers and cost estimators do a fairly good job of estimating development projects, when the requirements are done well.

This paper addresses the following topics:

- Cost Estimating Challenges (*After* requirements)
- Project Requirements - Demystified
- Estimating *Before* Requirements
- Recommendations and what you can do when estimating *Before* requirements

As one of the first authors to recognize that software engineering differs from traditional engineering, David Card stated, “Engineering projects usually can wait until after design to provide an estimate, while software engineering requires an estimate before design.” In the author’s experience, software projects can be even worse -- some projects need estimates before *requirements!*

The folly of management demands for project estimates, sometimes even fixed price estimates, often disables a project before it is even named. To illustrate, consider these fictional dialogues:

Dialogue 1: Home Construction Example

- **Potential homeowner:** *How much would it cost to build me a house with three bedrooms and two bathrooms?*
- **Builder:** *Well that depends.... Hmmmm.*
- **Potential homeowner:** *Depends on what?*
- **Builder:** *On the size of the house, style (single or two stories), amenities and desired rooms, and what you want to have included.*
- **Potential homeowner:** *Ok, I don't have time for this – give me a ballpark figure.*
- **Builder:** *Ok, anywhere from \$100K to \$500K assuming that it will be built during the summer in NYC.*
- **Potential homeowner:** *Can't you be more specific than that?*
- **Builder:** *Not without a floor plan! (i.e., requirements and design!)*

Dialogue #2: Software Project

- **User Manager:** *How much would a new software system cost me – it has to produce a pile of financial reports and do some up-to-the-minute financial processing? (I won't hold you to the estimate...)*
- **Software Project Manager:** *Well that really depends on a bunch of factors like the language we use, the skills, and how much functionality you need in the first release.*
- **User Manager:** *Just give me a ballpark figure – you should be able to do that quickly. After all, you've built lots of software.*
- **Software Project Manager:** *Without requirements or any understanding about what we will build, I can't really give you an estimate – I can barely give you a guesstimate.*
- **User Manager:** *Ok. I'll take that.*

One of the most daunting challenges of software projects is arriving at reasonable and defensible estimates of project duration, work effort and cost – once the requirements are known.

Cost Estimating Challenges (After requirements)

While software estimating is easier once the requirements are known, the following challenges still prevail:

- **Accuracy:** (How accurate are the productivity and cost driver inputs? Estimates are only as accurate as +/- 50% of the least accurate input variable)
- **Availability:** (Can all input variables even be provided to any reasonable level?)
- **Applicability of historical data** (If historical data is available, how applicable are the data points? How can one be sure to ensure an apples-to-apples comparison?)
- **Completeness:** (How complete are the requirements? Do project costs include or exclude hardware acquisition, software tool purchases, and other costs – or simply the software development labor costs?)
- **Risks:** (Have risk factors been considered and evaluated? Are they even included in the project? Risks must be calculated and predictive.)
- **What:** (What work effort tasks are included / excluded, and whose time is included?)

In spite of these challenges, cost estimators do produce estimates of duration, cost and work effort which becomes the input for project schedules. Estimates made early in the software development life cycle are subject to large variations due to factors mentioned above. While it should be obvious that estimates based on guessed values of input variables are unreliable, many managers and users give them undue weight and treat the estimates as predictive project forecasts. In the US, part of our American culture is that if something is too good to be true, it probably is – yet we have an insatiable optimism that maybe, “just this once” it might come true. Time and time again over-optimistic estimates are self-fulfilling as dates slip, functionality is reduced and project budgets surpassed. We would all do well to remember several key points related to project estimates:

- An estimate is only as good as its least reliable input variable
- Garbage entered into an estimating tool produces a garbage estimate
- Even if the corporate desire is for faster, better, and cheaper software development – an overoptimistic estimate won't make it happen
- Just because a cost estimator provides an estimate does not mean that it is realistic or achievable, When estimates are based on flawed data, they generate a false sense of security that the estimate is reality. This in turn gives rise to pressure on the cost estimator and project team to “prove” the goodness of the estimate by working desperately to meet it – even if it is grossly in error. and to make it come true.

The following table provides a few easy estimating equations that can be used as part of the estimating process:

Project Cost Ratio (completed projects)	\$ / FP (or SLOC)	$\frac{(\text{Total Hours} * \text{Hourly Cost}) + \text{Other Costs}}{\text{Project Functional Size}}$
Support Cost Ratio	\$ / 1000 FP (or FTE / app)	$\frac{(\text{Support Hours} * \text{Hourly Cost}) + \text{Other Costs}}{\text{Application Functional Size}}$
Repair Cost Ratio	\$ / FP (or per fix)	$\frac{(\text{Repair hours} * \text{Hourly Cost})}{\text{Functional Size of Repair}}$

Project Requirements – Demystified

Given that project requirements are the source of project rework (up to 45% according to the Software and Systems Technology Conference of the US Department of Defense, 2002), it makes sense to examine what can be improved in the requirements process – and thus improve the estimating process(es). In many organizations, project requirements are as elusive as icebergs – consisting of known-known requirements (the obvious functions of a type of software, such as the requirement to store account information for a banking system), known unknown requirements (those functions that always arise during a project based on corporate history with similar projects), and unknown-unknown requirements (the unpredictable requirements that cannot be anticipated yet lurk dangerously below the project surface like the submerged part of an iceberg).

The requirements discovery and articulation processes strive to maximize the known requirements while managing to minimize the unknowns (risks).

Project requirements can be easily broken down into three distinct types to increase understandability with users and the project team. The three types of requirements are:

- ❑ **Functional requirements.** These represent the business processes performed by or supported by the software, (e.g., record ambient temperature) and include what functions the software must do. These requirements are part of the users' / customers' responsibility to define. Functional requirements can be thought of similar to the software's Floor Plan. Functional requirements can be documented with Use Cases, sized by FP, and Costed by \$/FP
- ❑ **Non-functional requirements.** These represent how the software must perform once it is built. These include the "ILITIES" : (*Suitability, Accuracy, Interoperability, Compliance, Security, Reliability, Efficiency, Maintainability, Portability, and Quality in Use*) as described by ISO standards ISO/IEC9126 series. *While these requirements are also the responsibility of users / customers to ensure proper definition, they are often not articulated explicitly (or at all) but rather are "sprinkled" throughout requirements documents in dribs and drabs. Using a construction analogy, the non-functional requirements are similar to the contracted specifications for software. Non-functional requirements are **NOT** part of Use Cases (they are documented as part of 'Supplementary Specifications').*
- ❑ **Technical (build) requirements.** These types of requirements address how the software will be developed or "built" and includes tools, methods, work breakdown structure, type of project, etc. This is where most software developers have the most affinity with the requirements as the overall project requirements (the two above) are combined and a sort of software "blueprint" is delivered.

Modern software development approaches such as use cases and agile development attempt to keep these three types of requirements distinct and separate—when used correctly. Unfortunately in a manner similar to the contractor who only has a hammer and everything looks like a nail, some software developers cannot overcome the need to insert technical requirements into modern method deliverables such as use cases and agile user stories.

Estimating *Before* Requirements

When a cost estimator or project manager is asked for estimates BEFORE they have access to solid requirements, guesses take the place of solid information. What types of challenges arise when doing this type of "estimating":

- ❑ **Pre-functional requirements:** Often cost estimators or project managers may be asked for an estimate based on functional requirements scrawled on paper napkins or other informal, non-standard presentation media. One can only guess at what the software really will do based on assumptions such as: Kind of sort of like... (another system); or rough "ideas" that are conceptual at best.
- ❑ **Pre-nonfunctional requirements:** Assessment of the "ilities" of how the software will be required to perform can be done based on comparing them to other projects already completed for the same project department or business area. Unless there are major influences across the gamut of these non-functional requirements, they are not often seen as required input parameters needed to increase estimate accuracy. To overcome this flawed

assumption, cost estimators and project managers often underestimate the complexities that the project will bring.

- ❑ **Pre-technical requirements:** In many software development shops today, IT project teams use a standard suite of software development toolsets and technology aids. The technical requirements area is the least dangerous of the three areas in project estimating. Assumptions are generally made that the development platform is pre-selected along with other technical aspects of the project.

When faced with the daunting task of producing an estimate for a project for which there is little or no known input variables, what are the options for an ethical cost estimator? He/she could attempt to do one of the following, however, the response from management would likely be negative and could cost the estimator his/her job:

- ❑ Refuse to do an estimate (too early)
- ❑ Delay estimate repeatedly until requirements are at least partially done
- ❑ SWAG
- ❑ Use “Kind of sort of” actuals as estimate
- ❑ Cite “professional” ethics and hide out ... OR...(and this is the preferred method)

Document assumptions and use them together with the estimate (guesstimate) to substantiate the estimation results. When addressed and documenting the unknown aspects of functional requirements, they can still be sized using Function Points by indicating that each maintained entity in the software will typically follow the Add, Update, Delete, Inquire and Output (the AUDIO) profile given that the functionality is updating / maintaining the particular data.

For non-functional requirements, documented assumptions are critical here too. Overlooking or underestimating the complexity of the non-functional project requirements can cause major problems in keeping to the original estimates produced – because they would have emerged without documented and defensible assumptions.

For technical requirements, it is important to document novelty items (methods, tools, lack of skills, etc) for the technical aspects of the project. The impact of good or bad personnel skills on project teams can increase or decrease morale and project productivity (thereby altering the delivery speed and project completion either positively or negatively).

Recommendations for Estimating BEFORE Requirements

The following list re-emphasizes the advice provided in this paper when a cost estimator or project manager is faced with having to produce and estimate before the requirements have been fully fleshed out:

- **Document all of your assumptions (and then don't forget to validate them again later on the project for future releases and associated pre-requirements estimates.)**
- Separate and document project requirements as the three distinct requirements types
- Create a range of “guesstimates” when there isn't enough information to generate an informed and reliable estimate. (AND, ensure that the implied accuracy of an estimate is not misconstrued – just because your project estimating software includes numbers with decimal places in the results, use common sense and realize that the estimate is only as good as its least input
- Ensure you use standard estimating models that are proven for your environment

- Label results as “Preliminary”
- Level with your Customers – there is no magic estimating wizard who will direct the project (no “Magic”) -- ensure that they understand that an “estimate too early in the life cycle” cannot remain fixed throughout the project, nor can it be accurate.
- Despite your best wishes, Cost Estimators cannot create estimates out of “ether”
- However, (gu)esstimates based on documented assumptions are a step forward

What Can You Do to Improve Project Estimation at YOUR company?

In summary, project estimating can get better if you follow some simple advice:

- DOCUMENT as many of your assumptions about the project as you can, and revise them and the estimate according to the same assumptions to establish estimate traceability
- Document the requirements clearly and objectively:
 - Divide into 3 types
 - Consider recommendations
 - Consider FP as a measure of functional requirements size
 - Practice, document, follow-up, learn from prior “guesstimates”
- Join Quality Plus Measurement Forum:
www.groups.yahoo.com/group/quality_plus_measurement_forum

Best wishes with estimating and discovering and documenting great requirements.

BIOGRAPHY

Carol A.Dekkers, CMC, CFPS
President, Quality Plus Technologies, Inc.
Seminole FL USA

Professional Expertise

- Currently, and since 1994, Carol has been the president of Quality Plus Technologies, Inc., an industry leading software measurement training and consulting firm based in Seminole, FL.
- Recognized world expert in the area of functional size measurement (also known as Function Point Analysis), and represents the US to the International Organization for Standardization (ISO).
- Speaker at International Software Project Management and Software quality and measurement conferences since 1992 including the 7ECSQ in Helsinki in June 2002, and the 2WCSQ in Yokohama, Japan Sept 2001.
- Project Editor for ISO’s functional domain standard. (ISO/IEC JTC1 SC7 WG12)
- Former Host of the 26 shows of the weekly IT talk radio show: Quality Plus e-Talk with Carol Dekkers (see radio show homepage at www.qualityplustech.com)

Leadership Positions

- Chair of the PMI Metrics SIG 2003-2004
- In 2000, honored by the American Society for Quality (ASQ) as one of the 21 New Voices of Quality for the 21st Century.
- ASQ Software Division Track Chair for the Software Division Track at AQC 2003, (and AQC 2002)
- Moderator of Yahoo! Group: Quality_Plus_Measurement_Forum since August 2001 (over 300 invited members)

- Past-president of the International Function Point Users Group (IFPUG)
- Member of PMI, ASQ, APEGGA, ICMCI, CIPS, IFPUG

Publications

- Publications include *CrossTalk*, *Software Quality Professional*, *IT Metrics Strategies*, *Cutter IT Journal*, *MetricViews*, *the Voice*, *Application Development Trends*, others—**over 55 published articles**

Education / Certifications

- Bachelor of Science degree in Mechanical Engineering (Calgary, Canada),
- Certified Management Consultant (CMC), a Certified Function Point Specialist (CFPS) and a Professional Engineer (Canada).

Contact Information

Carol A. Dekkers
President
Quality Plus Technologies, Inc
8430 Egret Lane
Seminole, FL 33776

Voice: 727-393-6048
Fax: 727-393-8732
Email: Dekkers@qualityplustech.com

Quantitative Project Management

Integration of Balanced Scorecard and CMMI Maturity Level 4

*Carsten Højmosse Kristensen and Peter Voldby Petersen
Systematic Software Engineering A/S
Søren Frichs Vej 39, DK-8000 Århus C, Denmark*

For four years Systematic Software Engineering A/S has used the Balanced Scorecard (BSC) as a framework to define and communicate strategy in a consistent way throughout the organization. Furthermore, we have over seven years of experience in implementing the Software-related Capability Maturity Model (CMM) and Capability Maturity Model Integration (CMMI). At April 2004 we were externally appraised to CMMI maturity level 4. This presentation describes how the BSC and CMMI level 4 processes were integrated in a coherent framework for strategy that directs and justifies project management emphasizing the strengths and reducing the weaknesses of both approaches. We have named our approach Quantitative Project Management.

The benefits experienced after having implemented project BSCs and CMMI level 4 are substantial. The project objectives are better defined and therefore easier to control and fulfill. The objectives are defined with a balanced focus emphasizing not only the traditional effort, quality and schedule but also e.g. increased customer satisfaction and increased employee competences. The business responsible and project members all explicitly commit to the project objectives. The project BSC makes it easier to communicate the strategy on a day-to-day basis in the projects and aligns the projects with the strategy. The methodology of the BSC when defining balanced objectives fits nicely together with the CMMI level 4 processes focusing on implementing defined objectives. The somewhat internal and production-oriented emphasis in the CMMI model is broadened to include e.g. financial and customer views.

We believe that the approach taken substantially improves the value of both the BSC and the CMMI level 4. Organizations striving their way to higher maturity levels as well as organizations improving existing high maturity implementations can benefit from the approach. Furthermore, it can be used to improve traditional project management with a more quantitative understanding.

Keywords

CMMI Maturity Level 4, Quantitative Project Management, Balanced Scorecard, Strategy Implementation, Model synergy

1 Introduction

Clarifying, communicating, and implementing strategy is among the most difficult tasks in project-based organizations where projects dynamically appear and disappear and where employees play various roles over time. Participating in implementing such organizational strategy is a project task that extends beyond delivering at agreed cost, schedule and quality. For example building up organizational competencies and ensuring future business are usually not in the critical path of the projects' plans. One way of handling this problem is to encode the organizational strategy as quantitatively defined goals to which the projects must align. This however, proposes a new challenge in the projects which then must define and follow a path to make the goals successful.

Managing projects is often envisioned as applying systematic and objective methods. Reality, however, shows that typically neither of these attributes dominates the effort. Lack of objective insight into the project management process obstructs systematic behavior and lack of insight into current status blurs the decision making. Quantitative understanding strongly benefits the quality of project management decisions.

In this paper we describe our implementation of a marriage between two well-known and estimated methodologies; the Balanced Scorecard (BSC) [1] and the Capability Maturity Model Integration (CMMI) [2]. We entitle our approach Quantitative Project Management (QPM). For four years we have used the BSC as a framework to define and communicate strategy in a consistent way throughout the organization. We have over seven years of experience in implementing the software-related Capability Maturity Models. Systematic Software Engineering is appraised at CMMI maturity level 4.

Both BSC and CMMI have proven to provide value for modern, empowered organizations but typically companies have focused on one of the two. The BSC is often considered a management tool operating at the strategic level while CMMI is seen as production-oriented. We believe that this is a consequence of restricted and traditional thinking where disciplines are implemented independently because of organizational stove-pipe thinking. Furthermore, the more business oriented approach in the CMMI as opposed to the CMM [3] seems unrecognized by many companies. We consider mixing the BSC and CMMI to be a novel, natural, and effective approach.

In Sections 2 and 3 we briefly present the BSC and CMMI frameworks. An overview of our history to current practice is described in Section 4 and key concepts are explained in Section 5. Experiences gained are stated in Section 6 and concluded in Section 7.

2 Balanced Scorecard

The BSC was introduced by Kaplan and Norton in 1992. It is a recognized and widespread management tool which enables communicating strategic goals and implementation of these. This includes effort and results within four dimensions: Finance, Customers, Internal processes, and Employees (the latter also known as "Learning and Growth"). This is illustrated on Figure 1.

Finance	Customers	Internal Processes	Employees
Describes the expectations to the financial results including growth, profit and turnover	Describes important drivers of the financial results – i.e. market and customer segments, customer satisfaction objectives	Identifies the internal processes which are productive to the customers. The company must be very competent to carry out these processes which requires focus on continuous improvement.	Focuses on the company's future ability to fulfill the objectives for internal processes, customers and financial performance by means of attracting, developing and utilizing competencies.

Figure 1: The four BSC dimensions.

Alignment to strategy is accomplished by goal flow down, e.g. from top management to projects and/or from projects to individual employees. Important aspects of the goal flow down are definition of objectives and communication. When the projects understand the objectives of the organization, they can align their activities and resources with these objectives. In this way, the BSC acts as a management tool for implementing the strategy.

At Systematic, the BSC is used on two levels – at company level and at project level. Based on the company level BSC, the individual projects relate to the strategy and interprets it into project related objectives. Initiatives are planned to achieve the objectives. Ownership and commitment is created by actively involving the project manager and relevant project members.

The BSC is strong in defining balanced goals sustaining the organizational strategy but has little guidance in how to achieve such goals.

3 Capability Maturity Model Integration

The Capability Maturity Model (CMM), which was officially published in 1991, is a model based on best practices for software development. It describes an evolutionary method for improving an organization from one that is ad hoc and immature to one that is disciplined and mature. The CMM is internationally recognized and has been developed by the Software Engineering Institute at the Carnegie Mellon University, Pittsburgh, USA.

In 2000 a new and significantly extended version called CMMI was announced, where the 'I' stands for 'Integration'. This model addresses the problems relating to the many derived models from the original CMM framework, a practice which has proved to be inefficient and expensive [4].

According to CMMI, maturity in software organizations is measured in terms of quality, predictability and efficiency. The model prescribes a staged way to improve processes at five levels, each level describing sets of recommended practices organized in a number of process areas that enhance software capability. A software organization must master all process areas at a particular level in order to move onto the next level. Characteristics of the five levels are described in Figure 2.

Level 1: Initial	Level 2: Managed	Level 3: Defined	Level 4: Quantitatively Managed	Level 5: Optimizing
Processes are unpredictable, ad hoc and often chaotic. Performance is difficult to predict. The projects depend on 'heroes'. High quality relies on assigning the best people. Schedule and budget are usually overrun. Much time is spent on fire-fighting.	Processes are planned and executed in accordance with policy and procedures. Focus is on disciplined project management, and plans are produced and followed up. Tools are implemented to control quality and engineering. Employees are trained to undertake different functions.	A standard set of processes is defined and tailored to each project's needs. Engineering processes are implemented. Quantitative measurements are used to monitor project progress. Organizational process improvement is constantly implemented. Employee training needs are identified and met.	Projects are managed on the basis of quantitative measurements. Statistical methods to control product quality and processes are implemented. Models are used to predict the development of central control parameters in order to take preventive measures and to adjust plans.	The processes are continually improved based on return-on-investment considerations. Process improvement has become a natural, daily activity in which most employees actively participate. The organization optimizes its ability to deliver software products and systems better, faster and cheaper.

Figure 2: Characteristics of the five maturity levels.

CMMI prescribes an increase in the use of quantitative measurements as software organizations move up the maturity levels. The idea is that the more and the better the measurements, the more

control and predictability there will be with the development process and the faster corrective actions can be taken, if necessary.

CMMI level 4 is about managing quantitatively i.e. managing using numbers. The work in the organization is focused on the processes that contribute the most to the organization's quality and process performance objectives. These processes are improved to be statistically stable and therefore predictable. This is achieved by identifying outliers using control charts, analyzing for root causes and taking actions to remove those and to prevent reoccurrence. Based on historical data on statistically stable processes, process performance models are defined and used to predict progress in achieving the project's quality and process performance objectives.

The focus in the CMMI level 4 is on implementing project objectives but the model has little guidance in how to define the right goals.

4 Implementation Timeline

Our BSC and CMMI work was developed along two loosely connected tracks which were integrated conceptually in 2003 and fully integrated in practice the year after. A timeline of the major activities from year 2000 is defined in Figure 3.

2000	2001	2002	2003	2004
Engaged in BSC research project	Organizational BSC prototypes Strategy defined in BSC format	Organizational BSC reported regularly (monthly)	BSC project pilots Standard project BSC	BSC goal flow down to projects QPM planning QPM tracking
Diagnosis and gap analysis for level 3 Plan for level 3	Measurement systems Training Process sanitation	Institutionalization Assessment for SW-CMM level 3	Data sanitation Process Performance Baselines Process Performance Model	Institutionalization Assessment of the CMMI level 4

Figure 3: Key activities in implementation. BSC related activities are shown on the top track and CMMI activities on the bottom track.

Our BSC activities were started in 2000 as we entered a national research project on implementing the BSC in practice. This led to an initial BSC prototype which was refined and used to define the company strategy at the end of 2001. For a year, monthly reporting on the BSC effort and results showed the value of this approach, which includes e.g. increased focus on customer satisfaction.

In 2003 three pilots on the project level BSC was started. The resultant BSCs were acceptable although the novelty of the method was observable in the goal flow down process. Some objectives were not easily tailored by the projects. Another consequence of the novelty of the method was the high effort required to define the project level BSC. To avoid excess effort being spent and improve the quality of the project objectives, a standard project BSC was defined which left the projects with primarily quite simple tasks such as defining the concrete target value of the objectives. Some tailoring was still required for a much reduced number of objectives, which was supported by strongly improved guidance that emphasized the intention of the objectives.

In 2000, we effectively operated at CMM level 2 with a number of level 3 practices implemented. We were evaluated against CMM level 3 and the gap analysis led to a plan for level 3. The three major points in the plan was 1) implementation of a measurement system to efficiently collect and store measurements, 2) an elaborated training capability expanded to cover all roles in the organization, and 3) process sanitation to fully cover the model. This effort was institutionalized in 2001 and 2002

and completed with a successful assessment.

Conversion from CMM to CMMI and incrementing the maturity level was the focus for the next 25 months. The effort concentrated on:

- Sanitization of data in the measurement system. Additional requirements to data quality and coverage were introduced at level 4. This subject was addressed as the initial effort.
- Creation of Process Performance Baselines (PPB). PPBs document the historical results achieved by following a process and are used as benchmarks for comparing actual performance against expected.
- Creation of Process Performance Models (PPM). PPMs describe the relationship between attributes of a process and its products and are used to estimate or predict a critical value that cannot be measured until later in a project's lifecycle, e.g. predicting remaining number of defects in the product. The PPMs are based on PPBs.
- Improving existing practice to be compliant with changes in existing processes as well as new processes at levels two and three.

The effort was completed and the processes institutionalized by the end of 2003. Successful CMMI level 4 assessment was achieved in 2004.

5 Current Practice

Below we characterize the current practice as shown in Figure 3 (excluding institutionalization and assessment).

5.1 BSC Goal Flow Down to Projects

The starting point of the goal flow down is the organizational BSC, which defines the overall, organizational objectives. In the process of defining a BSC the project must know the answers to questions such as: How do we contribute the most to the goals of the organization? How do we align our project activities with the company strategy? Some of the answers to these questions are quantified and communicated more or less directly in the organizational BSC but in general the project goals can be fairly complex to derive. The major challenge being that the organizational objectives are not defined in a project context and therefore they are simply hard to understand and interpret when having a specific project focus.

In order to ease the interpretation and adoption of the objectives defined in the organizational BSC we have developed a Standard Project BSC that suits a norm project. The Standard Project BSC is used as a boilerplate by the projects when defining their project BSCs and is augmented to reflect the most important objectives of the project's stakeholders.

Depending on the project, different levels of tailoring is expected for different parts of the Standard Project BSC. For each of the four dimensions in the organizational BSC we have defined strategic focus areas and objectives including specific targets. The Standard Project BSC extends this in two ways: First each of the organizational objectives is evaluated against the context of a norm project in Systematic. Secondly, expected type of tailoring is defined as given in Figure 4.

Type of tailoring	Description
Include	Use the defined project level objective including target as is
Change	Change the target value of the project level objective
Redefine	Define a new objective and target that is aligned with the strategic focus area
Omit	Do not define a project level objective in the given strategic focus area

Figure 4: The defined types of tailoring of the objectives in the Standard Project BSC.

At the beginning of a project, the project manager and leading roles in the project tailor the Standard Project BSC according to the defined tailoring types to fit their specific project. To assure alignment and insight, this is done in cooperation with the relevant business responsible. The business responsible is well-informed about the organizational strategy, the project scope and the customer of the project and therefore he is capable of deciding the necessary tradeoffs in the definition of the project BSC. This includes deciding on the appropriate tailoring of the Standard Project BSC (see Figure 4) and extending this with objectives from the customer or other relevant stakeholders. Typically, these objectives add to the Internal Processes dimension of the BSC. The internal processes part of the project BSC are used as the project's quality and process performance objectives as this term is defined in the CMMI.

Commitment from top management is ensured by reviewing the BSC with top management for approval.

5.2 QPM Planning

Different objectives require different levels of tracking from just monitoring status over planning and tracking and control using thresholds to statistically control of subprocesses and prediction of progress. This raises the question of which level of tracking is the right one for a given objective? We find that this depends on data quality, frequency of sampling and the dynamics of the parameter. Furthermore, the expected benefits from the given level of tracking compared to the cost of collecting, presenting and analyzing the indicator are necessary to take into the planning considerations. Today we use the highest level of tracking for the objectives in the Internal Processes dimension of the BSC including statistical control and prediction, where as the "soft" objectives in the customer and employee dimensions are tracked using the lower level, traditional methods. As a part of the QPM Plan the level of tracking is defined for each objective. We consider this a very important practical aspect.

5.3 QPM Tracking

The progress against achieving project objectives defined in the project BSC are tracked as defined in the QPM Plan. This includes keeping the processes contributing the most to the objectives under statistical control using control charts and predicting progress in achieving the objectives using Process Performance Models. Extensive analysis is performed in order to stabilize and improve overall project performance. Analysis guidelines are integrated into the organizational definition of measurements. Project decisions are taken based on a quantitative basis and corrective actions are issued.

Project status is reported on objectives defined in the project's BSC as well as on project parameters as required by the CMMI. The unified reporting includes all status information and analysis which are aggregated at the appropriate management level. The status reports are used to communicate status to managers and project members and to propose required actions.

Indicators matching the objectives in the organizational BSC are produced every month and an organizational BSC status report is issued monitoring the performance of strategic goals over time.

6 Experiences Gained

6.1 BSC Goal Flow Down to Projects

Quantitative management is based on the organizational strategy and must be integrated in the management strategy process. The communication of strategy in the organization has improved greatly since we started working with QPM. At the organizational level, defining the objectives quantitatively makes the strategy more concrete and understandable. The Standard Project BSC interprets the organizational strategy in a project context and makes it possible for the projects to tailor and define relevant goals in a cost-effective way. The project BSC is an excellent mechanism for communicating project objectives to project members. QPM improves the presence of strategy at all levels in the organization.

Decisions are taken based on available knowledge. If the project manager understands the strategy of the organization and has goals set for his project that interpret how his project contributes the most to implementation of the strategy, then the decision process of the project manager is directed to include what is the best for the organization. This extends the view of the project manager and is beneficial for the organization. The alignment between the projects and the organization improves and the strategy is better implemented. Future project management decision can be aligned to the strategy.

The goal flow down from the organizational BSC, defining the company strategy, to project BSC defines objectives better than we have seen before since objectives are easier to control and fulfill. The balanced focus emphasizing not only effort, quality and schedule but also e.g. increased customer satisfaction and increased employee competencies is definitely a strongpoint in the real world.

Tailoring the project BSC from the organizational BSC is a non-trivial process, and we have seen this improved each time it is done. Through initial experiences, it appears as if the process of tailoring get more attention than the value of the objectives. However, this has changed to a focus on quantitative objectives as the process has institutionalized and matured.

6.2 QPM Planning

Clear objectives are necessary for effective and efficient planning. The BSC methodology improves the definition of project objectives as it includes both financial, customer and employee aspects which are easily dominated by a production oriented view only. When applying this four-dimensional focus, we have seen more balance in the definition of project objectives than previously experienced.

One example of this is the use of the project BSCs to establish project focus on long-term employee competences. In the Standard Project BSC we defined certification of employees as a project objective. The project managers thus realized that the objective was important for the organization and therefore effort for preparing certification was allocated in project plans. Another way of phrasing it is that the focus of the project managers changed from what is good for the projects to also include what is good for the organization. The other good thing was that the employees reacted to the defined objective and started taking certifications.

6.3 QPM Tracking

Quantitative measurements provide objective insight in project status. Objective insight improves the probability of making the best decisions. Focus is taken away from feelings, pride and personal involvement, which makes it possible to discuss which decision is the right one based on the facts.

Prediction of critical parameters makes it possible to take corrective actions earlier than if no prediction was made. Our experience so far is that predicting critical parameters of a project with a useable

accuracy is very difficult, but working with this continuously improves the basis of decisions and thereby eventually also the quality of the products.

Effective planning ensures that the status on the defined objectives is measured. Effective control ensures that the indicators are analyzed, relevant actions are proposed and real project management decisions are taken.

7 Conclusion

In this article we have described how the BSC and CMMI level 4 processes have been integrated into a coherent framework emphasizing the strengths and reducing the weaknesses of both the BSC and the CMMI level 4. We recommend that organizations implementing CMMI consider using the BSC in the implementation of maturity level 4 because this provides value and links to business goals. Furthermore, we suggest that software organizations implementing the BSC consider the CMMI framework or parts of the framework to help implement strategy e.g. the practices from the Measurement and Analysis process area.

We consider the most important findings of our approach to be:

- The BSC can be utilized to express strategy in quantifiable terms within four relevant and related dimensions. One of these (Internal Processes) is closely related to process improvement but goal flow down from the other dimensions to the project is also valuable.
- CMMI sets a number of production oriented requirements to parameters which the project is to be measured and controlled against. The BSC approach extends this set with important metrics quantifying the strategy and ensuring that the performance of strategic goals is monitored.
- BSC and CMMI have strong synergies because they complement each other very well. The BSC is strong in defining goals while CMMI is strong in implementing known and defined objectives. Both models do, when used separately, have weak points in terms of assumptions and lack of guidance in relation to a practical implementation, but the weak points are minimized when using a combined approach.
- When developing guidelines describing how to interpret and analyze indicators, the BSC strategy map helped a lot because it describes the causality between strategic goals which could be used as analysis guidance.

Our assessment for CMMI level 4 acknowledged a strong infrastructure to support QPM. This included documented objectives in the BSC, standardized and mandatory indicators and measures to be used on the projects, and efficient and effective PPBs and PPMs. Likewise, the practice of QPM was recognized including the establishment of objectives, collecting and analyzing data, keeping defined processes under statistical control, and identifying corrective actions as appropriate.

The assessment also suggested improvements. First of all, recommendations for further institutionalization were to put additional processes and subprocesses under statistical control, creating additional PPBs and PPMs, and expanding the quantitative management disciplines to more processes. Another suggestion was to strengthen the documentation of the criteria used by the project to compose their defined process based on process performance requirements.

Despite the many positive features of the QPM approach recognized above, some caution should be taken before implementation. The BSC and CMMI level 4 together is a substantial change to any organization. Implementing level 4 requires a lot in terms of time and effort as known from SEI studies [5]. Although we do believe that integration of the BSC into a level 4 improvement program is very beneficial, it also adds additional complexity – especially if the BSC methodology is new to the organization. We have experienced that it took us 25 months and required 6% of the productive hours in the organization. This appears comparable to the SEI studies of changes from maturity level 3 to maturity level 4.

Although we are a software engineering company with a technical educational level among the highest, quantitative management is a new discipline that is not easily accepted. The lower maturity levels of the CMMI are intuitively acceptable in our organization. People understand that project management, organizational learning, training etc are useful and necessary, but in QPM a lot of energy can easily be wasted on discussing the statistical methods and other technical details. Focus must be kept on the business value of QPM.

Our goal is to reach CMMI level 5 in year 2005. On this journey, we aim to consolidate our QPM on the following areas:

- Gain more experience with the goal flow down process, especially regarding implied requirements to the definition of organizational goals.
- Enhance the level of statistical control in relation to the goals.
- Strengthen the importance of using QPM to make real management decisions.

8 Literature

- [1] – Kaplan and Norton, The Strategy-Focused Organization, HBS Press, 2000
- [2] – Chrissis et al, CMMI – Guidelines for Process Integration and Product Improvement, Addison-Wesley, 2003
- [3] – Paulk et al, The Capability Maturity Model – Guidelines for Improving the Software Process, Addison-Wesley, 1995
- [4] – SEI, Capability Maturity Model Integration (CMMI), SEI CMMI tutorial, 2001
- [5] – SEI, Process Maturity Profile – Software CMM 2003 Year End Update, SEI slide number 25, 2004

Author's CVs

Carsten Højmosse Kristensen, MSc EE, PhD

Carsten is Business Unit Director at Systematic Software Engineering in Denmark. His fields of responsibility include business development, process improvement, quality assurance, competence development, and research and development. Carsten has had the business responsibility for process improvement efforts at the company level since 1998.

Peter Voldby Petersen, MSc EE, PMP

Peter is Chief Program Manager at Systematic Software Engineering in Denmark. Peter has been project manager on both industrial, defense and process improvement projects in Systematic. Peter has lead the Systematic software process improvement effort since the end of 2002 including the work leading to CMMI Maturity Level 4 certification.

Systematic Software Engineering, Denmark

- Established in **1985** and now Denmark's **largest privately-owned** software and systems company
- **370+ employees**; 70% hold a MSc or PhD in software engineering
- High **employee satisfaction** – attractive **workplace** for **ambitious** software engineers
- CMMI **Level 4** and ISO 9001:2000 and AQAP 110 + 150
- Supplier of products and projects to more than **27 countries**
- 97% of our **customers would recommend Systematic** to other customers
- For further information – see **www.systematic.dk**

Developing a RE Process Improvement Model

Sarah Beecham, Tracy Hall and Austen Rainer

University of Hertfordshire

s.beecham@herts.ac.uk

Abstract

This paper presents a partially validated model of requirements process improvement, based on the Software Capability Maturity Model (SW-CMM) and Goal/Question/Metric framework. The development of the model is motivated by findings from our three year empirical study that examined how practitioners impact software process improvement. The twelve companies in our study all showed a lack of control over their RE processes. Our observation that all the companies were using some variant of the SW-CMM led us to investigate the strengths and weaknesses of the SW-CMM in relation to requirements engineering (RE) support. From this investigation our model emerged. In this paper we explain our model development process and give an overview of how we partially validated the RE model through an expert panel of twenty people from academia and industry.

1. Introduction

This paper presents a newly developed validated requirements capability maturity model (R-CMM¹) that aims to support practitioners in their software process improvement activities. This requirements engineering (RE) process model provides a pathway to improved processes by prompting practitioners to examine RE processes within a five-stage maturity framework based on SW-CMM principles. It is a work in progress, where initial maturity stages are built in sufficient depth to allow for an external validation to be performed by a group of experts. The study draws on the findings of the EPSRC funded 'Managing Practitioners' Impact on Process and Product' (PPP) project². Analysis of the PPP empirical data showed RE process issues to be a recurring problem for software practitioners in the UK today [1, 2]. Although all companies in our study were using the Software Capability Maturity Model (SW-CMM) [3] to guide them in their software process improvement activities, they all showed a lack of control over their requirements engineering process [4].

The R-CMM differentiates itself from other work in the field by placing RE processes in context with the SW-CMM. We are aware of the many weaknesses of the SW-CMM that include an over complex, incomplete, assumption based and inflexible presentation (see for example, [5-9]). However, basing the R-CMM on a known software improvement framework offers the user many advantages. The R-CMM taps into the SW-CMM to form a strengthened, specialised best practice model that is familiar, integrates with related software processes, and has a tried and tested methodology. The framework offered by the SW-CMM pulls together disparate work in the field of the RE process and presents solutions in a way that is accessible to both practitioners and researchers. The R-CMM includes an assessment method that guides the user to identify strengths and weaknesses in their current requirements process with a view to prioritizing process implementation against maturity goals.

The term 'requirements engineering process' or 'RE process' as used in this study, refers to activities performed in the requirements phase that culminate in producing a document containing the software requirements specification [10]. More specifically, the RE process is the set of activities required to gather, specify, validate and engineer a set of requirements [11]. Our view of the RE process takes a complementary approach to existing work where multiple factors are found to affect the production of the requirements specification and an important class of factors are those internal to the development organisation (see, for example, Procaccino et al [12]). Empirical research confirms that RE problems are indeed multifarious, yet inter-dependent, with each individual problem influencing project success [13]. Moreover, the impact of each individual problem is likely to differ in terms of severity. It would therefore follow that there is a need for an empirical evaluation as to how RE process problems are recognised, evaluated and prioritised. Further work is also needed in considering how the different solutions to RE process problems might be combined into one framework. Therefore the focus of this research is on RE process problem recognition, evaluation and assessment.

¹ ®CMM is registered in the U.S. Patent and Trademark Office. CMU has not participated in this publication.

² Grant number GR/L91962.

Researchers and practitioners understand that the earlier RE process problems or requirements defects are detected, the easier and cheaper they are to repair [14]. Lauesen and Vinter [15] build on this observation noting that, “detection as well as prevention [of requirements defects] requires some effort in addition to usual development.” This need for additional effort is also a major theme in the early empirical study of Bell and Thayer who state that, “the requirements for a system do not arise naturally; instead, they need to be engineered and have continuing review and revision” [16]. The experience report of a high maturity organisation, the NASA space shuttle project, confirms that moving resources to the front end of software development can contribute to the reduction of delivered software defects [17].

This is not the first attempt to represent the RE process in a maturity framework, for example, Sommerville and Sawyer [8] also recognised this need. Their RE good practice guide, although based on maturity levels, does not link directly to existing maturity models. Other models of the RE process presented by requirements experts such as [18-24] all offer different approaches to solving the RE problem. Their properties for quality requirements, although using mainstream terminology, offer a confused message. Definitions of desirable qualities are often found to be vague, complicated and lacking in detail. Lists can be unstructured and overlapping, and some goals are unrealistic and even impossible to reach [25]. The practitioner is therefore left not only wondering which advice to take, but having opted for a method, may not be given the means to achieve their objectives.

This paper is organised as follows: Section 2 gives an overview of the model development phases. Section 3 introduces a top down view of the R-CMM, while section 4 presents a bottom-up view of the R-CMM based on a goal question metric paradigm. Section 5 gives an example of the R-CMM at a level 2 process maturity. Section 6 gives a summary of the results of the validation exercise that leads to many of the ideas for future work given in the following section. Section 8 summarises the work presented in this paper.

2. Modelling the RE process

We examine the RE process in order to model the factors that contribute towards the gathering, specifying, validating and engineering of a set of requirements. According to Panedo and Shu [26] formalising the lifecycle process is key to software improvement. However, although software lifecycle models play an important role in software engineering as a general guide to software developers, they lack the detail required for an analysis of a specific process [27, 28]. Houdek and Pohl observe that RE activities such as elicitation and validation are heavily intertwined and question whether differentiation between these phases yields benefit [29]. Indeed, taking a purely process view of requirements breaks away from a partitioned lifecycle view.

The R-CMM includes a traditional process view of the activities involved in the production of requirements as described in general software engineering texts such as Dorfman and Thayer [30] and Pressman [31]. The five phases represented in the R-CMM (requirements management, elicitation, analysis, documentation and verification) bridge the gap between a conventional/structured ‘lifecycle’ view and a process view of RE. This traditional view has the advantage of using familiar vocabulary and creating an intellectual tool that allows the user to focus on different areas of the RE process.

The R-CMM focuses on company practices involved in developing bespoke software systems. It is necessary to make this distinction as the needs of companies involved in other forms of development such as ‘commercial-off-the-shelf’ development are likely to differ, as the role of the customer and system constraints have a different emphasis [32]. However, despite this specialisation, the more detailed and refined the model becomes, the less generic it is [33].

The RE process is integral to software development. Indeed, the resulting requirements specification may be revisited and changed throughout the project. Brooks captures the iterative nature of the RE process in his seminal paper where he states “..the most important function that the software builder performs for the client is the iterative extraction and refinement of the product requirements process” [34]. Yet in order to attempt to improve current representations of this front end of development the R-CMM, it is necessary to capture only those activities that relate directly to the RE process.

2.1 Four stages of model development

In developing the requirements process maturity framework, we follow set rules of model building as suggested in the literature [35, 36]. Model development involved four key stages as shown in Figure 1 that comprise: specifying the criteria for building the model; abstracting RE processes from three sources; creating a specialised R-CMM model that adheres to the criteria; and finally, validating the model. This is the first cycle of model development and does not constitute a finished model.

Model development is initiated by creating and agreeing model criteria. These criteria serve as objectives that clarify the purpose of the model and outline what the model is expected to describe. They specify that the model should: Adhere to SW-CMM characteristics; be limited in scope; be consistent; be understandable; be easy to use; be tailorable; and be verifiable. These seven criteria steer development and are later used to help evaluate the model in stage four of the development cycle. Detailed studies of each of these stages are given in [37-40].

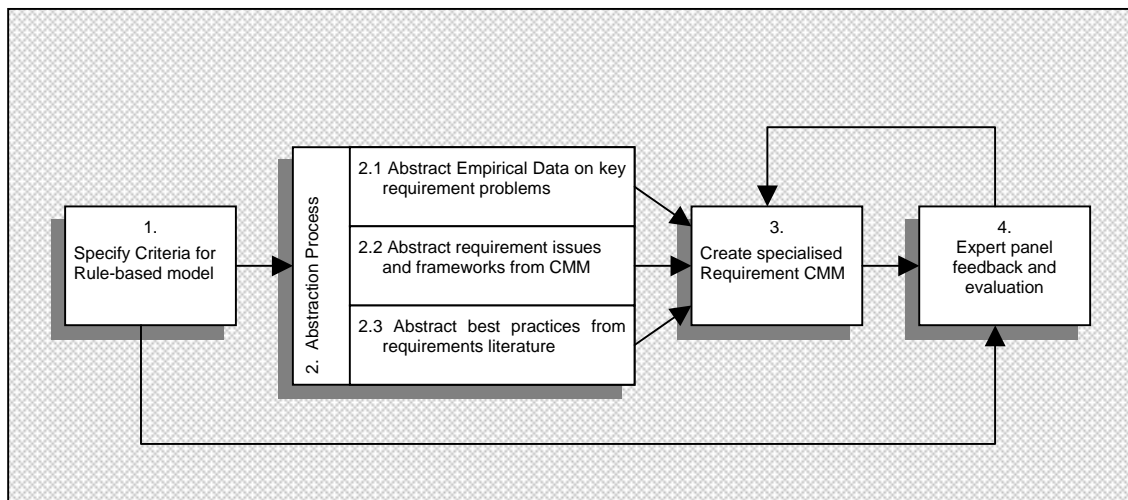


Figure 1: Activities involved in building the R-CMM

3. The R-CMM

The R-CMM aims to present complex activities in a way that can be easily understood and prioritised. As shown in Figure 2, the SW-CMM (Paulk et al, 1995) provides a broad maturity framework which the R-CMM uses to describe requirements activities. The R-CMM is sympathetic to the existing culture surrounding companies who are using the SW-CMM as their mechanism for SPI.

Taking a top-down view of the improvement cycle complements this familiar model and should help practitioners as they do not need to learn yet another improvement methodology. Applying the SW-CMM characteristic of capturing the ‘repeatability’ of best practices is particularly appropriate to the requirements process, as “ ‘Requirements’ ... are captured not only within specific projects, but also carry across between projects, embedded in cultures, organisations and communities ” [33, page 7].

4. A goal/question/process metric approach to RE process improvement

The R-CMM adapts the Goal/Question/Metric (GQM) paradigm developed by Basili and Rombach [41] in order to guide practitioners to reflect on their improvement goals prior to implementing an improvement plan. As shown in Figure 3, the four main areas covered in the R-CMM are the identification of goals for improvement, the questions to ask to address the goals, the processes that need to be considered, and how to measure the strength of the existing processes. Users of the model are guided through three components, ‘goal’, ‘question’ and ‘process’ prior to ‘measuring’ the strength of individual processes. Figure 3 also shows how the R-CMM supports continuous improvement as advocated by [42] and [43]. All 5 levels of process capability will follow the improvement cycle presented in Figure 3, where the SW-CMM maturity characteristic (or goal) noted in Figure 2 is decomposed to relate to five requirements phases. Figure 3 shows how a ‘process’ element has been added to the GQM template to ensure that the required focus is given each of the activities listed.

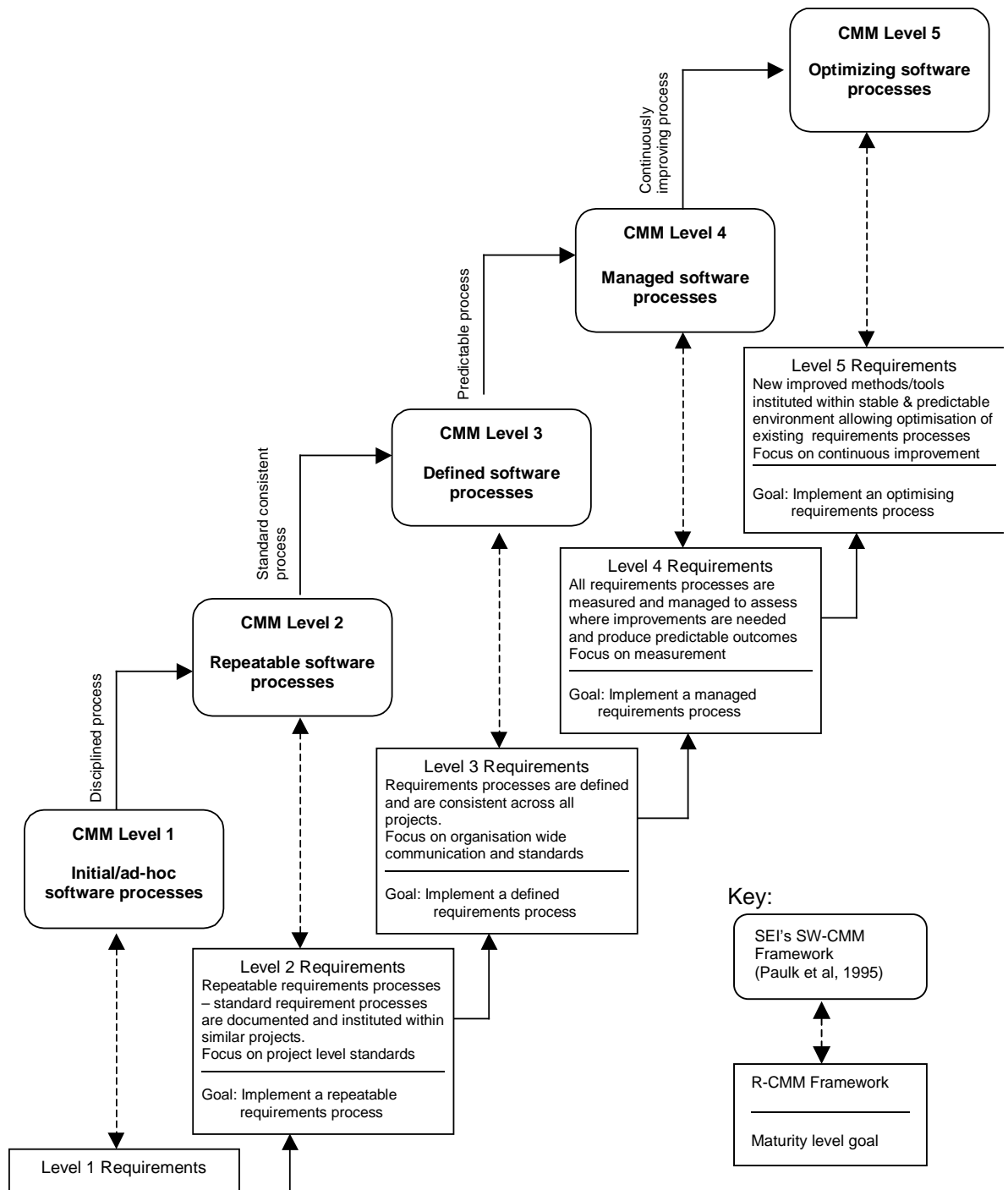


Figure 2: The R-CMM 5 level framework

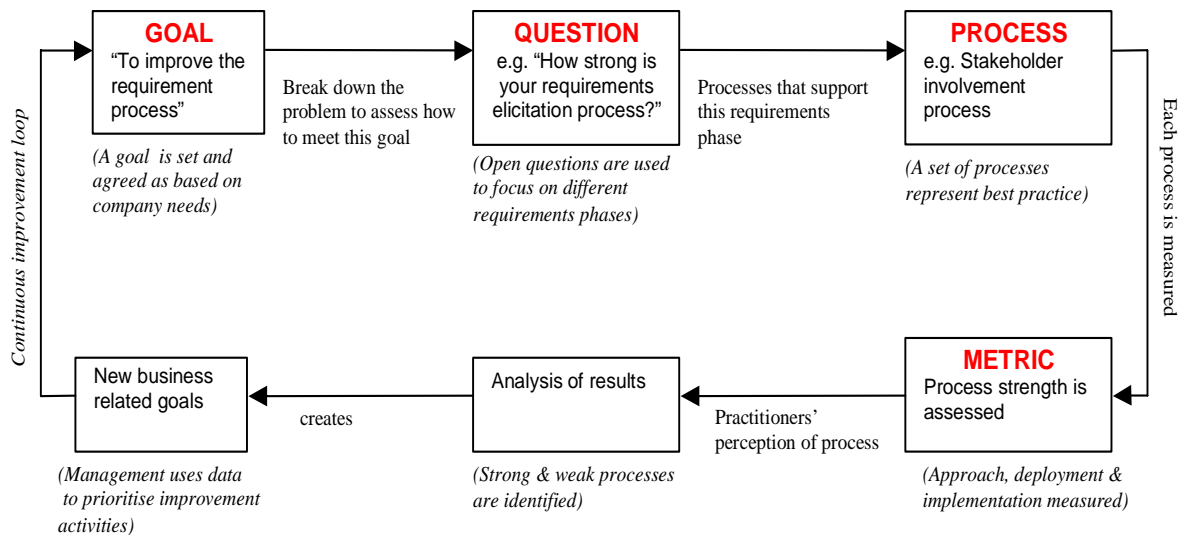


Figure 3: Example of continuous process improvement in the R-CMM

5. Tailoring the R-CMM

An example of candidate base-line processes as abstracted from the literature and the SW-CMM is modelled in Figure 4. All these processes are included in response to practitioner needs as identified in our empirical studies. The R-CMM contains both project management processes, or 'organisational' processes and technical requirements processes as a lack of organisational control is a major source of requirements process problems [2, 44]. Figure 4 also shows how the level 2 goal (given in Figure 2) provides a focus for this more detailed level 2 model. Processes are modelled separately as identifying and defining processes within a maturity framework is the essence of the R-CMM. This approach is also an ideal way to bring together both technical and organisational needs of a company. The process element directly addresses the needs of a business. For example, the process focus gives a visibility into what is required to improve RE quality (e.g. improved traceability) and can combine these technical processes with those that relate to organisational processes such as resourcing, time-scales and cost that are equally important to business [45]. Figure 4 gives an example of how the organisation sets an improvement goal and how this goal is decomposed through a series of questions that relate to given processes.

The R-CMM also includes a finer grained description, or guideline, of each individual process that will assist process implementation. This follows a similar goal, question, process, metric framework. An example of a guideline derived from P4 in Figure 4 is given in Figure 5.

5.1 Process assessment – the metric element of the model

Organisations cannot always rely on external researchers or wait two years for the lengthy, external process assessment, to identify where their weaknesses are. And although companies can perform their own internal SW-CMM assessments, current methods combine the RE process improvement with the whole of software development. The R-CMM therefore includes an internal assessment component (or metric) that allows practitioners to analyse their current RE activities with a view to prioritising where best to focus their improvement efforts. The assessment can be self administered where individual processes are analysed to measure how they are deployed, approached and how effective they are.

The assessment method is based on a tried and tested evaluation scheme used by Motorola who developed it to be compatible with the SW-CMM [46]. For a full explanation of how the R-CMM adapts this assessment method please refer to [38].

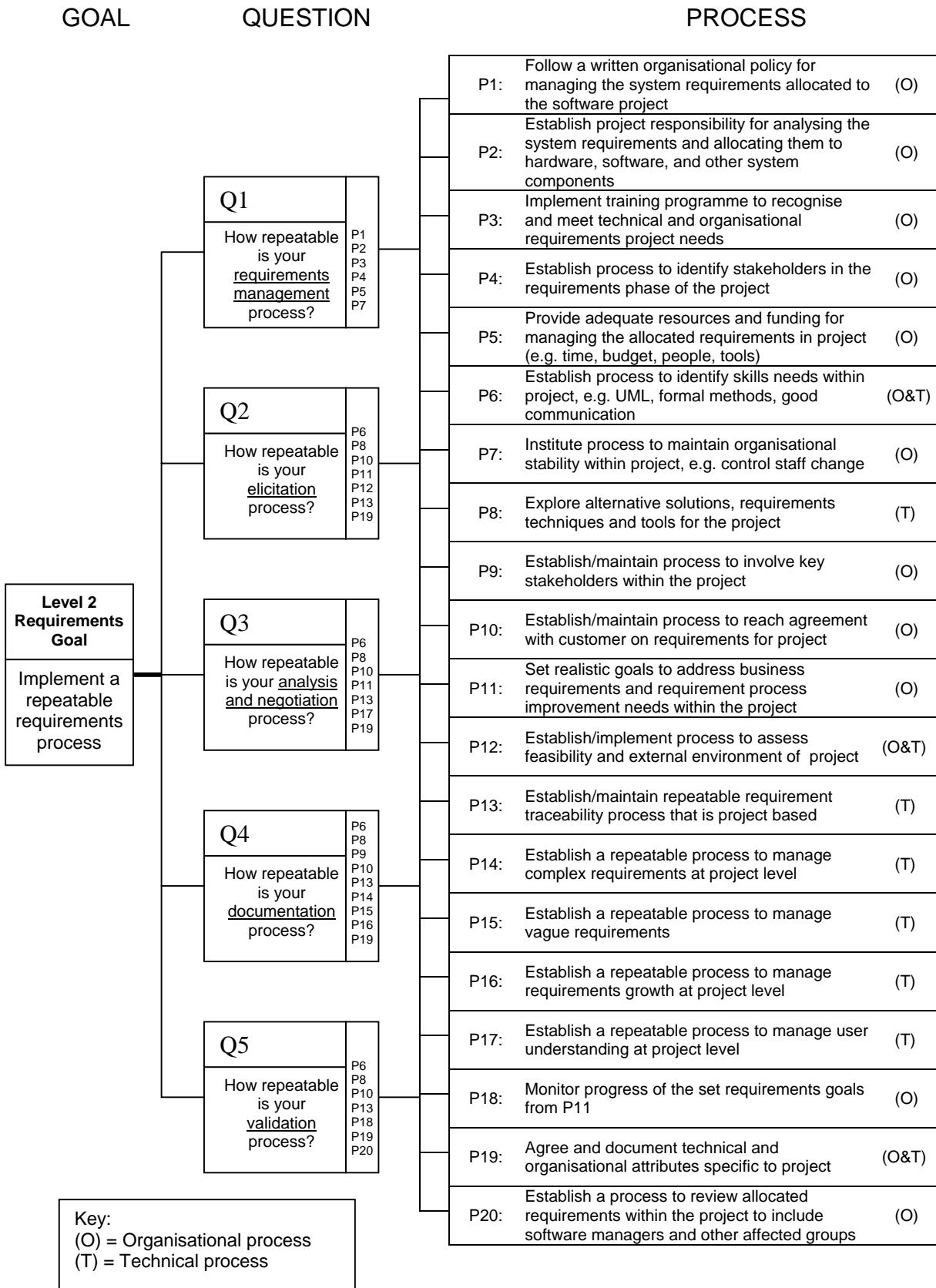


Figure 4: R-CMM candidate processes at level 2 maturity (base-line processes)

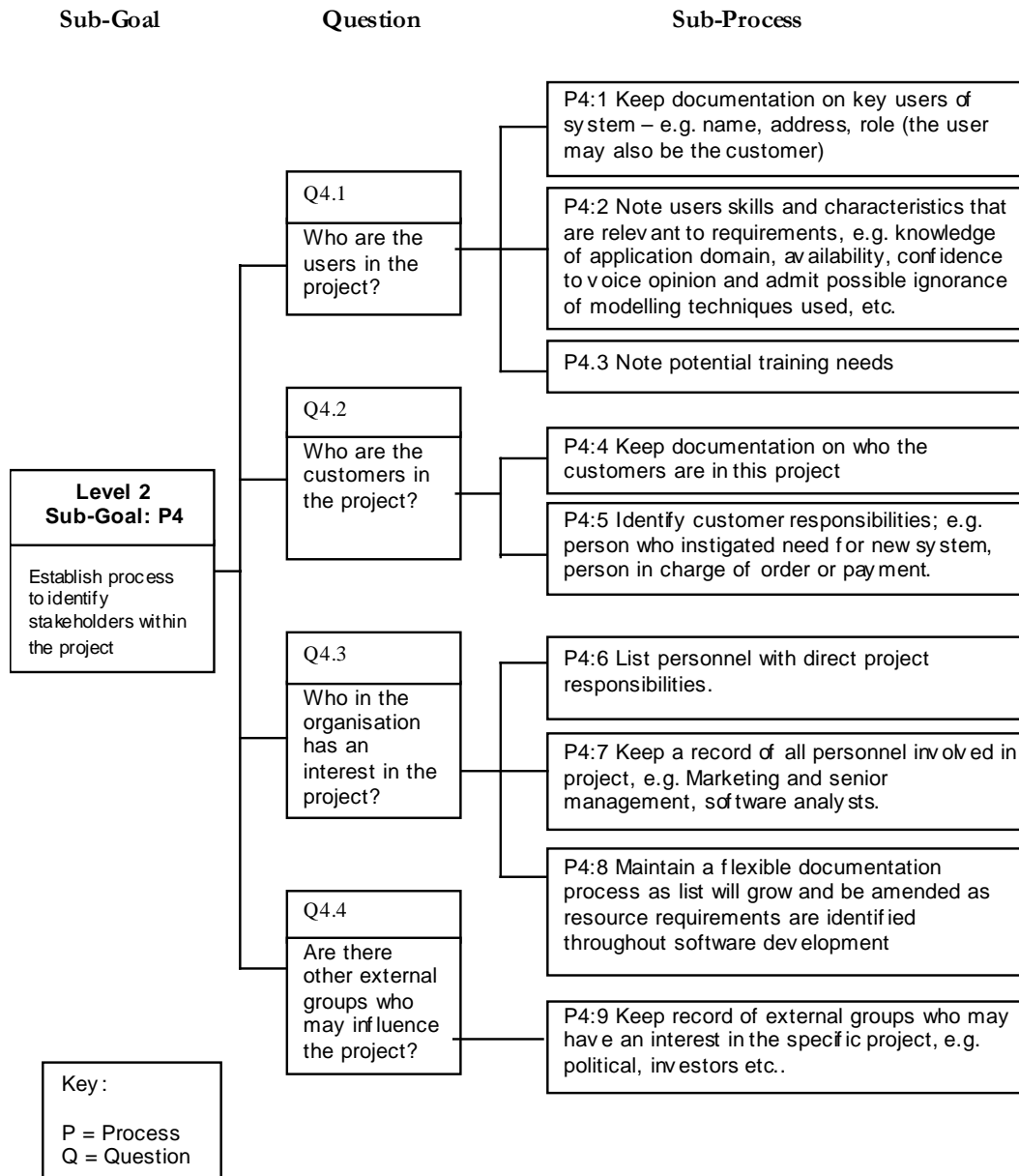


Figure 5: Guideline example of a level 2 process

6. Model validation

The final stage of the first cycle of model development involved validating the R-CMM components presented in this paper. A questionnaire³ was sent to group of experts in the fields of SPI and RE. The feedback gained from experts with such diverse and proven knowledge was of great value as was found in [13, 47][[48]. Twenty experts, representing a response rate of over 80%, looked at model components and responded to questions that directly relate to the success criteria detailed in section 2.1. As shown in figure 6, multiple items are combined in the questionnaire to test each success criteria and help average out possible errors in single item responses [47].

To confirm the association between questions and model criteria two separate researchers performed an inter-rater reliability test with a kappa result of .85 which indicated an almost perfect agreement [49]. To limit research bias in interpreting the questionnaire data based on a relatively small sample we use a 95% confidence interval formula that Agresti and Coull describe as giving the desired level of confidence for virtually all sample

³ Please contact the first author for a copy of the questionnaire and the accompanying documentation

sizes [50]. Analysis of the 84 model related questions, based on a this validation exercise highlighted areas in the model that should be retained, as well as areas that could be improved as shown in Table 1.

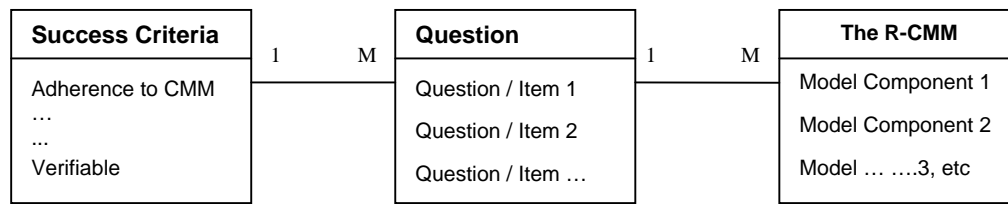


Figure 6: Relationship between Success Criteria, Questionnaire Design and R-CMM

The range of responses elicited from this group of experts provided an objective view of the model that, otherwise, could easily become unrelated to the needs of the community. For further details of the methodology used in validating the model please refer to [40].

Table 1: Summary of questionnaire results

R-CMM Strengths	R-CMM Weaknesses
The concept, ‘to support the requirements process’	The model appears incomplete (intentional at this level of abstraction)
Retaining a SW-CMM concept (although some experts do not view this as a strength)	Ambiguous process definitions (looking at high level definition alone is insufficient)
High level consistency of detail	Unrealistic structured view (phases don’t relate to requirements practices)
A strong structure	Structure may appear inflexible and detached
Taking a process view of requirements	Inconsistent level of abstraction (a problem of combining multiple sources)
Decomposing activities	Missing key baseline processes
General adaptability/tailorability	Wrong key baseline processes
Assessment component appears a good way to recognise requirements process weaknesses	The assessment component is not self-explanatory

7. Future Work

For real value to be gained from this research follow-up work is essential as the results of the empirical investigation are not ends in themselves [51]. The validation of the R-CMM and the development transparency provides a foundation for future development in the area of RE process improvement. Suggestions for future work include:

- RE process definitions: The R-CMM requires more precise and accessible process definitions as depending on high level definitions alone leads to ambiguity.
- Conduct further research on key RE processes. Work is needed to see whether RE processes are associated with, for example, company size, application area, process maturity, culture and structure.
- Practical application trials. The R-CMM is at a stage where processes could be tested in the workplace. Although the assessment component of the R-CMM was well received by the experts, there was some concern that more information and possibly training would be required in order for it to be used successfully.
- Further testing: A tool is in the process of being developed that allows users to immediately test their RE process. Future work therefore would involve a group of expert practitioners applying the R-CMM in the workplace.
- Development of guidelines. All the processes listed at a high level could be developed into guidelines that relate more closely to how the processes are implemented.
- Development of process maturity levels 3, 4 and 5 should be presented in the same detail as the level 2 example.

8. Conclusion

Basing the R-CMM on the SW-CMM enables practitioners to build a quality RE process that supports and integrates with software engineering activities. The validation of the base-line processes shows that the R-CMM has the potential to help practitioners reach an understanding of how to tailor RE processes to meet their own needs, how to set realistic quality RE process goals and provide a means to achieve their goals. The R-CMM presents RE processes, or best practices, within a maturity framework that guides users towards an integrated view of the RE process, where maturity goals are set to help with process prioritisation and implementation.

The R-CMM directs practitioners to examine their RE process in a systematic and detailed way. The R-CMM combines some SW-CMM best practices together with additional RE processes that are outside the scope of the SW-CMM. The R-CMM abstracts processes from the RE literature in order to meet the needs of practitioners in our empirical study.

The work undertaken in developing the R-CMM narrows the gap between RE process research (suggesting principles, techniques, languages and tools to help analysts understand a problem or describe a potential product's external behaviour) and the practice "where software customers understandably wonder if anyone is listening" [52]. We believe that the R-CMM pulls together disparate work in the field of software process improvement and RE process solutions in a way that is accessible to both practitioners and researchers.

Acknowledgements

We are very grateful to the twenty experts who participated in this validation. We thank Dr C. Britton and Dr M. Cottee who helped in piloting the study and provided statistical and technical support. We are also grateful to all the companies who collaborated in the EPSRC project who remain anonymous. Finally thanks to Dr N. Baddoo for assisting in the inter-rater testing.

References

1. Hall, T., S. Beecham, and A. Rainer, 2002. *Requirements Problems in Twelve Companies: An Empirical Analysis in EASE 2002, 6th International Conference on Empirical Assessment and Evaluation in Software Engineering*. Keele University, Staffordshire, UK. April 8 -10th
2. Hall, T., S. Beecham, and A. Rainer, 2002. *Requirements Problems in Twelve Companies: An Empirical Analysis in IEE Proceedings for Software*. October
3. Paulk, M.C., et al., 1995. *The Capability Maturity Model: Guidelines for Improving the Software Process*. The SEI Series in Software Engineering, ed. C.M.U.S.E. Institute, Reading, Massachusetts: Addison Wesley Longman, Inc.
4. Beecham, S., T. Hall, and A. Rainer, 2003. *Software Process Improvement Problems in 12 Software Companies: An Empirical Analysis*. *Empirical Software Engineering*. 8 (1): p. 7-42.
5. Baumert, J., 1994. *Process Assessment with a Project Focus*. *IEEE Software*. 11 (2): p. 89-91.
6. Brodman, J.G. and D.L. Johnson, 1994. *What Small Businesses and Small Organisations Say About the CMM in 16th International Conference on Software Engineering*. Sorrento, Italy: IEEE. May 16-21
7. Hayes, W. and D. Zubrow, *Moving on Up: Data and Experience Doing CMM-Based Process Improvement*. 1995, Carnegie Mellon University: Pittsburgh CMU/SEI-95-TR-008.
8. Sommerville, I. and P. Sawyer, 1997. *Requirements Engineering A good practice guide*, Chichester: John Wiley & Sons Ltd.
9. Ngwenyama, O. and P. Neilsen, 2003. *Competing Values in Software Process Improvement: An Assumption Analysis of CMM From an Organizational Culture Perspective*. *IEEE Transactions on Engineering Management*. 50 (1): p. 100-112.
10. Jalote, P., 1997. *An Integrated Approach to Software Engineering: Second Edition*. Undergraduate Texts in Computer Science, ed. D. Gries and F. Schneider, New York.
11. Britton, C., 2000. *Object-oriented system development : a gentle introduction*, New York; London: McGraw-Hill.
12. Procaccino, J.D., et al., 2002. *Case Study: factors for early prediction of software development success*. *Information and Software Technology*. 22: p. 53-62.
13. El Emam, K. and N.H. Madhavji, 1996. *An Instrument for Measuring the Success of the Requirements Engineering Process in Information Systems Development*. *Empirical Software Engineering*. 1: p. 201-240.
14. Boehm, B.W., 1981. *Software Engineering Economics*. advances in computing science and technology series, ed. Prentice-Hall, Englewood Cliffs: Prentice-Hall, Inc.
15. Lauesen, S. and O. Vinter, 2001. *Preventing Requirement Defects: An Experiment in Process Improvement*. *Requirements Engineering Journal*. 6 (1): p. 37-50.
16. Bell, T.E. and T.A. Thayer, 1976. *Software Requirements: Are They Really a Problem?* in *Proc. ICSE-2: 2nd International Conference on Software Engineering*. San Francisco
17. Krasner, H.K.C., J.S. Pyles, and H.S. Wohlwend, *A Case History of the Space Shuttle Onboard Systems Project*. 1994, SEMATECH Technology Transfer: Texas Technical Report Technology Transfer # 94093551A-TR. p. 1-22.
18. Davis, A., 1988. *A Taxonomy for the Early Stages of the Software Development Life Cycle*. *Journal of Systems and Software*. p. 297-311.

19. Arisholm, E. and D.I.K. Sjöberg, 2000. *Towards a framework for empirical assessment of changeability decay*. Journal of Systems and Software. (53): p. 3-14.
20. Boehm, B., 2001. *Using Win-Win for Requirements Negotiation: a mini tutorial*. Imperial College: London.
21. Jackson, M., 1995. *Software Requirements & Specifications: a lexicon of practice, principles and prejudices*, Wokingham, England: Addison Wesley Publishing Company.
22. Kotonya, G. and I. Sommerville, 1998. *Requirements Engineering : processes and techniques*. Worldwide series in Computer Science, ed. D. Barron and P. Wegner, Chichester: John Wiley & Sons.
23. Leffingwell, D. and D. Widrig, 2000. *Managing software requirements: a unified approach*. First ed. The Addison-Wesley object technology series, ed. J.C. Shanklin, Reading, Massachusetts: Addison-Wesley. 491.
24. Thayer, R.H. and M. Dorfman, 1990. *System and Software Requirements Engineering*. IEEE Computer Society Press Tutorial, Los Alamitos, CA: IEEE Computer Society Press.
25. Lindland, O.V., G. Sindre, and A. Solvberg, 1994. *Understanding Quality in Conceptual Modeling*. IEEE Software. 11 (2): p. 42-49.
26. Panedo, M.H. and C. Shu, 1991. *Acquiring experiences with the modelling and implementation of the project life-cycle process: the PMDB work*. Software Engineering Journal. 6 (5): p. 259-274.
27. Madhavji, N.H., 1991. *The process cycle*. Software Engineering Journal. 6 (5): p. 234-242.
28. Penedo, M. and C. Shu, 1991. *Acquiring experiences with the modelling and implementation of the project life-cycle process: the PMDB work*. Software Engineering Journal. p. 259-274.
29. Houdek, F. and K. Pohl, 2000. *Analysing Requirements Engineering Processes: A Case Study in Proceedings of 11th International Workshop on Database and Expert Systems Applications*. London, UK: IEEE. 4 - 8 September
30. Dorfman, M. and R.H. Thayer, 1997. *Software Engineering*, Los Alamitos, CA: IEEE Computer Society Press.
31. Pressman, R., 2001. *Software Engineering 5th Edition*: McGraw Hill.
32. Fairley, R. and P. Rook, 1990. *Risk Management for Software Development*, in *System and Software Requirements Engineering*, R.H. Thayer and M. Dorfman, Editors IEEE Computer Society Press: Los Alamitos, CA. p. 387-400.
33. Loomes, M.J. and S. Jones, 1998. *Requirements Engineering: A Perspective Through Theory-Building in Third International Conference on Requirements Engineering*. Colorado Springs, Colorado, USA: IEEE Computer Society
34. Brooks Jr., F.P., 1987. *No Silver Bullet: Essence and accidents of Software Engineering*. IEEE Computer. 20 (4): p. 10-19.
35. Koubarakis, M. and D. Plexousakis, 2002. *A formal framework for business process modelling and design*. Information Systems. 27: p. 299-319.
36. Rossi, S., 1999. *Moving Towards Modelling Oriented Software Process Engineering: A Shift from Descriptive to Prescriptive Process Modelling in International Conference on Product Focused Software Process Improvement*. Oulu, Finland: Technical Research Centre of Finland ESPOO 1999. June 22-24
37. Beecham, S., T. Hall, and A. Rainer, *Building a Requirements Process Improvement Model*. 2003, University of Hertfordshire: Hatfield Technical Report TR 378.
38. Beecham, S., T. Hall, and A. Rainer, *Assessing the strength of a requirements process*. 2003, University of Hertfordshire: Hatfield Technical Report TR 381.
39. Beecham, S., T. Hall, and A. Rainer, *Defining a Requirements Process Improvement Model*. 2003, University of Hertfordshire: Hatfield Technical Report TR 379.
40. Beecham, S., Hall, T., Britton, C., Cottee, M. and Rainer, A., 2004. *Using an expert panel to validate a requirements process improvement model*. Journal of Systems and Software, Accepted and In Press.
41. Basili, V.R. and H.D. Rombach, 1988. *The Tame Project: Towards Improvement-oriented software environments*. IEEE Transactions on Software Engineering. 14 (6): p. 758-773.
42. Deming, W.E., *Quality, Productivity, and Competitive Position*. 1982, Massachusetts Institute of Technology Center for Advanced Engineering Study: Cambridge, MA.
43. Humphrey, W.S., 1989. *Managing the Software Process*. The SEI Series in Software Engineering, ed. S.E. Institute, Reading, Massachusetts, USA: Addison-Wesley Publishing Company, Inc. 494.
44. Lubars, M., C. Potts, and C. Richter, 1993. *A Review of the State of the Practice in Requirements Modeling in In proceedings of the IEEE International Symposium on Requirements Engineering*. San Diego, CA. January 4-6
45. Solingen, R.v. and E. Berghout, 1999. *The Goal/Question/Metric Method: a practical guide for quality improvement of software development*, Maidenhead, UK: McGraw-Hill Publishing Company.
46. Daskalantonakis, M.K., 1994. *Achieving Higher SEI Levels*. IEEE Software. 11 (4).
47. Dybå, T., 2000. *An Instrument For Measuring The Key Factors Of Success in Software Process Improvement*. Empirical Software Engineering. 5 (4): p. 357-390.
48. Rosqvist, T., M. Koskelo, and H. Harju, 2003. *Software Quality Evaluation Based on Expert Judgement*. Software Quality Journal. 11 (1): p. 39-55.
49. Landis, J. and G. Koch, 1977. *Measurement of Observer Agreement for Categorical Data*. Biometrics. 33: p. 159-174.
50. Agresti, A. and B. Coull, 1998. *Approximate is Better than "Exact" for Interval Estimation of Binomial Proportions*. The American Statistician. 52 (2): p. 119-126.
51. Potts, C., 1993. *Software Engineering Research Revisited*. IEEE Software. 10 (5): p. 19 -28.
52. Davis, A., 1994. *Giving Voice to Requirements Engineering*. IEEE Software. 11 (2): p. 12-16.

How Evaluation Techniques Influence the RE-tool Evaluation: An Experiment

Raimundas Matulevičius, Lena Karlsson, Guttorm Sindre

Abstract

Requirements engineering (RE) tools could be compared by analyzing documentation and performing explorations, but the evaluation process is complex and time consuming. This work reports on a two-session experiment which compares evaluation techniques for RE-tools. First a theoretical framework and test scenarios are used, next tool tutorials and a feature-based questionnaire are applied. The findings show that the framework provides a detailed evaluation, but it is more time consuming than the questionnaire-based technique. The results also support the use of test scenarios instead of tool tutorials. Although the scenario preparation is relatively long and requires expert involvement, the scenarios provide comparison of the same tool functions in the same problem domain.

Keywords

Requirements engineering (RE), RE-tools, RE process improvement.

1 Introduction

The requirements engineering (RE) process is described as a sequence of actions, during which the requirements for a new software system are elicited, analyzed, validated and documented [2] in a formal, complete and agreed specification [8]. For a complex activity like requirements engineering (RE), a powerful tool support is useful and could lead to the RE process improvement, which comprise quality improvement, RE scheduling enhancement and resource reduction. The need for RE-tool support may vary in different RE activities and project size; e.g. if a company does not have a mature RE process, the automation of the RE process won't necessarily help. But if the company deals with big projects, and requirements specifications, containing many requirements, which need to evolve over time, the RE-tool support could clearly be very helpful. The usefulness of automated support for the software development process is recognized in literature [1, 2] and indicated in a case study [6]. However, the mainstream of RE practice relies on generic office and modelling tools rather than dedicated RE-tools. The organizations are not aware of the gains in taking up more advanced tool support. A part of the reason might be that it is difficult to evaluate the tools available.

This work considers a two-session experiment of RE-tool evaluation. In the first session an evaluation framework [7] and test scenarios are applied. The second session is executed with tool tutorials and a feature-based questionnaire. The work purpose is to analyze the RE-tool evaluation techniques. The paper is structured as follows. Section 2 describes the research method. Results are analysed in section 3. Section 4 describes validity threats. Finally, section 5 provides conclusions and future work.

2 Research method

The experiment analyzes two questions: 1) how different evaluation techniques influence the RE-tool evaluation quality; and 2) which RE-tool features are captured during the evaluation process using different evaluation techniques.

Participants. Two experiment sessions were executed at the Norwegian University of Science and Technology (NTNU) and Lund University (LU). 6 students from NTNU and 30 students from LU participated in the experiment. The students from NTNU were in their graduate course. They had knowledge of IS engineering, conceptual modelling, and the semantic Web. The LU students were from the third or fourth year and had knowledge from courses in RE, and software development methodologies.

At the NTNU an evaluation framework [7] (figure 1) and RE-tool test scenarios were applied to evaluate RDT¹, RequisitePro², and CORE³. At the LU participants were using a feature-based questionnaire (figure 2) and the RE-tool tutorials to evaluate DOORS⁴, CaliberRM⁵ and RequisitePro.

The evaluation framework [7] describes three RE dimensions [8] – representation, agreement and specification. The functional features are expanded with the lists of activities [3, 4, 5]. The *representation* dimension deals with the degree of formality, where requirements are described using informal, semiformal and formal languages and traceability between them. The *agreement* dimension deals with the degree of stakeholders’ agreement and support of the rationale and communication among them. The *specification* dimension deals with the degree of understandability, supported by the documentations. The framework was adapted to the *evaluation form*. In addition the questionnaire was prepared to evaluate RE-tool usability and understandability, and understandability of the evaluation framework.

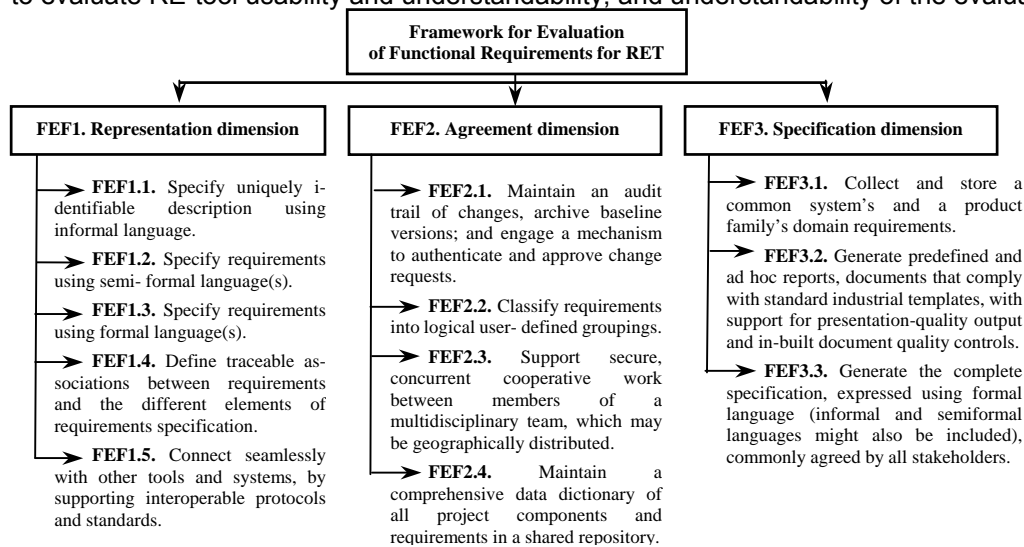


Figure 1: Framework for Evaluation of Functional Requirements for RE-tools

The testing scenarios describe the IS to support the electricity fault handling. The scenarios include requirement import, maintenance of traceability, requirements models, and requirements specification.

The feature-based questionnaire used at the LU, consists of 13 questions. Questions consider students’ experience with RE-tools, information about RE-tools on the Internet, requirements representation, traceability, prioritization, collaboration, reports, usability, understandability, tool integration with other tools. The final question asks which of the RE-tools respondents would prefer for their projects.

The requirements tutorials are provided by the tool vendors. The tutorials are meant to make the potential customers interested in a tool purchase and teaching the tool functionality.

RE-tools. Both CORE and RDT were downloaded from the Internet. NTNU has a teaching and research licence of RequisitePro. The LU has a teaching and research licence of DOORS. The other two tools – RequisitePro and CaliberRM - were downloaded from the Internet.

Evaluation process. The NTNU students performed the RE-tools test scenarios in different order: 1) RDT, RequisitePro, and CORE; 2) RequisitePro and CORE (because of software failure, RDT was not

¹ <http://www.igatech.com/>
² <http://www-306.ibm.com/software/rational/>
³ <http://www.vtcorp.com/>
⁴ <http://www.telelogic.com/products/doorsers/>
⁵ <http://www.borland.com/caliber/index.html/>

tested); 3) CORE, RDT and RequisitePro. The framework forms which contained 40 activities were filled in after each tool test. At the LU the tools were tested according to the tutorials in the same order: DOORS, CaliberRM and RequisitePro. The feature questionnaires were filled in after all tool tests.

2. Have you used any of the tools before?

	NO		Yes, a little			Yes, a lot	
		1	2	3	4	5	
DOORS	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
CaliberRM	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
RequisitePro	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

Please motivate your answer.

Functional features

4. Please evaluate the functionality to represent/ specify requirements.

	Not-	Bad			Good	
	tested	1	2	3	4	5
DOORS	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
CaliberRM	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
RequisitePro	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Please motivate your answer.

Non-functional features

9. How difficult was it to understand the functionality of the RE-tool?

	Not-		Difficult			Easy	
		1	2	3	4	5	
DOORS	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
CaliberRM	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
RequisitePro	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

Please motivate your answer.

13. Which of the RE-tools would you prefer to use, for example in your "KRAM-project"?

- DOORS
- CaliberRM
- RequisitePro
- None

Please explain your answer.

Figure 2: Fragment of the feature-based questionnaire

3 Results

Experiment results are shown in tables 1 and 2. A *feature evaluation* is the average of the framework activity (in table 1) and feature (in table 2) evaluations for the particular RE-tool. The *weight* is the average of all feature evaluations. The *not-tested* ratio is the feature percentage, which got evaluation '0'. The *weighted evaluation* is the multiplication between the weight and the feature evaluation.

Table 1: Results at the NTNU

FEATURES (The feature deal with ...)	WEIGHT	NOT TESTED	FEATURE EVALUATION			WEIGHTED EVALUATION		
			RequisitePro	RDT	CORE	RequisitePro	RDT	CORE
FEF1.1. (... informal representation)	4,25	0,0%	4,78	4,00	3,89	20,32	17,00	16,53
FEF1.4. (... requirements traceability)	2,91	27,5%	3,17	3,05	2,57	9,22	8,88	7,48
FEF1.5. (... integration with other tools)	2,68	29,2%	2,85	2,33	2,75	7,64	6,24	7,37
FEF3.2. (... report printing)	2,66	40,6%	3,17	2,00	2,58	8,43	5,32	6,86
FEF1.2. (... semi formal representation)	2,39	37,5%	3,42	1,00	2,28	8,17	2,39	5,45
FEF2.3. (... cooperative work)	2,00	58,3%	1,67	2,17	2,22	3,34	4,34	4,44
FEF2.1. (... user collaboration)	1,94	50,0%	2,63	2,45	0,90	5,10	4,75	1,75
FEF3.3. (... standard specification)	1,92	54,2%	1,44	1,83	2,44	2,76	3,51	4,68
FEF3.1. (... repository activities)	1,86	56,3%	1,71	2,00	1,92	3,18	3,72	3,57
FEF2.2. (... requirements specification)	1,70	50,3%	1,60	1,90	1,67	2,72	3,23	2,84
FEF2.4. (... user support)	1,54	62,5%	1,11	2,17	1,56	1,71	3,34	2,40
FEF1.3. (... formal representation)	1,06	75,0%	2,83	0,00	0,00	3,00	0,00	0,00
OVERALL:		45,1%	30,38	24,90	24,78	75,60	62,73	63,38

Evaluation results. Well evaluated RE-tool features are traceability, RE-tool integration with other software and informal requirements representation. Weak evaluated feature is requirements prioritization. Report printing and collaboration got opposite evaluations. Report printing was not available at the LU. The collaboration is not guided in the tutorials, but was described in the test scenarios.

Table 2: Results at the LU

FEATURES	WEIGHT	NOT TESTED	FEATURE EVALUATION			WEIGHTED EVALUATION		
			CaliberRM	RequisitePro	DOORS	CaliberRM	RequisitePro	DOORS
Traceability	3,60	1,1%	4,20	3,57	3,03	15,12	12,84	10,92
Internet information	3,44	3,3%	3,63	3,13	3,57	12,51	10,79	12,29
Representation	3,28	2,2%	3,77	3,17	2,90	12,35	10,38	9,51
Understandability	3,26	0,0%	3,90	3,17	2,70	12,70	10,31	8,79
Usability	3,11	1,1%	3,87	3,03	2,43	12,03	9,44	7,57
Collaboration	2,16	37,8%	2,37	1,30	2,70	5,10	2,80	5,82
Reports	1,90	44,4%	2,03	1,80	1,87	3,86	3,42	3,55
Prioritization	1,79	20,0%	1,70	1,83	1,83	3,04	3,28	3,28
OVERALL:		13,5%	25,47	21,00	21,03	73,67	59,98	58,44

There could be two reasons for not-tested features (overall 45,1% in table 1 against 13,5% in table 2). First, the tool does not support the feature. Second, the respondents did not understand the feature. To mitigate the second reason, the participants could always ask about the questionnaires.

RE-tool usability. At the NTNU the participants had to answer: “which of the RE-tools helps to prepare “the best” specification?” 4 answers are RequisitePro, 2 – CORE. All the respondents indicated that RDT is the most complex tool, but table 1 shows that both CORE and RDT are evaluated equally. At the LU question was “which tool would you prefer to use in your own project?” 15 respondents would prefer CaliberRM, 7 - DOORS, 4 – RequisitePro, 4 – both CaliberRM and RequisitePro, and 2 respondents would not use any RE-tool.

Satisfaction of the techniques. The respondents at the NTNU think positively about the framework. Some problems are related to feature understandability, which could be a reason for the high not-tested rate. At the LU ease of use and satisfaction of the evaluation techniques were not analysed. Nevertheless, most participants found the experiment instructive, although, some participants complained about headache due to the small tutorial fonts. The NTNU participants commented that they felt a lack of time: the experiment took 4 hours - the first tool test was performed in 70 minutes (the participants had to learn the evaluation framework), other two in 40 minutes. At the LU the participants used 2 hours for tutorial execution and a half hour for questionnaire filling.

4 Threats to Validity

The experiment involved students as subjects, and it was executed for educational purposes. The participant number was only 36, which may reduce the external validity [9]. The participants had basic knowledge but limited experience with RE in practice. This situation may be similar to the one experienced by practitioners, who perform the evaluation in industry. The fact that a tool acquisition was not performed, may influence the level of commitment in the experiment.

Time limits and fatigue may influence the results, especially at the NTNU where less time was available and a larger feature number was evaluated. All the tools at the LU were evaluated in the same order, so there may be a learning effect or boredom effect which could influence the responses.

Some participants had experience with RE-tools, and this may have influenced their opinions. But since the experiment provides two independent sessions with correlating results, the participants' prior experience has most likely not affected the results. The experiment at the LU is based on tutorials; therefore, it is possible that the evaluation is affected by the differences between the tutorials.

The evaluations are influenced by limited tool functionality. At the NTNU, CORE and RDT were limited by the database size, and the participants had reliability problem with RequisitePro. At the LU the laboratory computers had no MS Word™ installed, and RequisitePro and CaliberRM tutorials had to be adjusted so that it only included tasks that do not require integration with MS Word™.

The internal validity regards whether the treatment actually cause the outcome [9]. In this experiment, the two sessions differ from each other in two ways: one investigates a framework-based approach using test scenarios, while the another explores a feature-based questionnaire and uses tutorials. Therefore it may be difficult to determine which treatment that actually causes the results.

5 Conclusions and Future Work

This work reports on a two-session experiment, where the RE-tool evaluation techniques are tested. The study shows that the framework-based evaluation is more time consuming than using the questionnaire. But the framework [7] prepares a more detailed evaluation. It is recommended to use the framework, when the purpose is to acquire a tool for targeted RE activities. The questionnaire-based technique evaluates the RE-tools on an abstract level, but the questionnaire construction is time consuming in comparison to the framework application. The scenarios allow testing of the same RE-tool functions and they could be created for any problem domain, which is familiar to the evaluators. However, the scenario preparation is relatively time-consuming and involves RE-tool experts. The tutorials are designed for commercial purposes and are therefore difficult to compare.

The findings suggest a combination of both techniques could be interesting. First, the tool candidate number is reduced with a feature questionnaire. Next, the detailed tool comparison is performed using

the framework. The ideas for future work involve the construction of an RE-tool evaluation approach, where comparison of functional and non-functional requirements would guide to the tool selection. The evaluation techniques should be considered for various purposes in academic and industrial settings.

Acknowledgment. The authors would like to thank Björn Regnell for contributing ideas to this study.

Literature

1. Kaindl H., Brinkkemper S., Bubenko jr. J., Farbey B., Greenspan S. J., Heitmeyer C. L., do Prado Leite J. C. S., Mead N. R., Mylopoulos J., Siddiqi: Requirements Engineering and Technology Transfer: Obstacles, Incentives, and Improvement Agenda, *Requirements Engineering*, 7 (2002), pp. 113-123.
2. Kotonya G., Sommerville I.: *Requirements Engineering: Process and Techniques*. Wiley (1998).
3. Krogstie J.: Integrating the understanding of Quality in requirements Specification and Conceptual Modeling. *Software Engineering Notes* 23 (1) (1998), pp. 89-91.
4. Lang M., Duggan J.: A Tool to Support Collaborative Software Requirements Management. *Requirement Engineering* 6 (2001), pp. 161-172.
5. Lindland O.I., Sindre G., Sølvsberg A.: Understanding Quality in Conceptual Modelling. *IEEE Software* 11 (2) (1994), pp. 42-49.
6. Matulevičius R., How Requirements Specification Quality Depends on Tools: A Case Study. In Persson A., Stirna J. (eds.) *Proceedings of the 16th International Conference on Advanced Information System Engineering (CAiSE'2004)*, Riga, Latvia, (2004), Springer-Verlag Berlin Heidelberg 2004, pp. 353-367.
7. Matulevičius R., Validating an Evaluation Framework for Requirement Engineering Tools. In Krogstie J., Halpin T. and Siau K., (eds.) *Information Modeling Methods and Methodologies (Adv. Topics of Database Research)*, 2004, Idea Group Publishing, pp. 148-174.
8. Pohl K.: The Three Dimensions of Requirements Engineering: a Framework and its Applications. *Information systems*, 19 (3) (1994), pp. 243-258.
9. Robson, C., *Real World Research*, Blackwell (1997).

Author CVs

Raimundas Matulevičius

Dept. of Computer and Information Science, Norwegian Univ. of Science and Technology. Raimundas Matulevičius received his Master of Science in computer science by Vytautas Magnus University (VMU) in Lithuania, in 2000. Currently he is a PhD student at Norwegian University of Science and Technology (NTNU). His research interests include process improvement in requirements engineering (RE), RE-tools, conceptual models of information and information processes.

Lena Karlsson

Department of Communication Systems, Lund University, Sweden. Lena Karlsson received her Master of Science in Electrical Engineering with Industrial Economy by Lund Institute of Technology in 2001. Currently she is a PhD Student at Lund Institute of Technology and received a Degree of Licentiate in Engineering in 2003. Her research interests are in the area of market-driven requirements engineering and management and include release-planning and decision-making in product management.

Guttorm Sindre

Dept. of Computer and Information Science, Norwegian Univ. of Science and Technology. Guttorm Sindre holds a Ph.D. in Computer Science, Norwegian Institute of Technology (NTH), University of Trondheim, in 1990 and a *Master of Engineering* in Computer Science, Norwegian Institute of Technology (NTH), University of Trondheim, in 1988. From 1992-95 he was associate professor in Software Engineering and from 1999 onwards associate professor in Information Systems at Norwegian University of Science and Technology (NTNU). He has 10 papers in journals and books and more than 40 papers in international conferences and workshops since 1987. He is currently manager of the research project WISEMOD (2004-2007), funded by the Norwegian Research Council, and is also involved in the EU IST 6FP NoE project INTEROP.

A Business-Driven Requirement Process

Rolf Nergaard, ConsultIT AS

Abstract

Various studies conclude that requirement errors are the most common and most expensive errors to fix in later life cycle phases.

There is strong evidence that controlling requirements is the single most effective action a project team can take to improve project outcomes and assist in our goal of delivering quality software, on time and on budget.

We need to change from chaos to order with some sort of a methodical requirement process. Hence, an improved requirement management process is a vital part of software process improvement (SPI) in a software company.

A Common problem is that the requirement process is too technically focused, alienating the users. ConsultIT has adopted a RUP centric (Rational Unified Process) approach focusing on the business and user problems, through work with business models and stakeholder analysis. This paper describes our experience with this approach.

Keywords

Requirement engineering, software process, rational unified process, requirement management, needs, features, use case, business model

1. Introduction

Several times in the last year we have seen several articles written about budget overruns and failure in software projects. This is not a new phenomenon in the software industry. In a well known study by the Standish Group there was made a survey among 352 companies reporting in over 8000 projects. The results of this survey were published in a report called "The Chaos Report" [1]. According to the report:

- 31% of all software projects are canceled before completion
- 53% of all projects will cost 189% of estimates
- 9% are on time and on budget (large companies)
- 16% are on time and on budget (small companies)

Further they asked the respondents to identify the causes of these failures. The respondents identified several factors that cause to be challenged. The top three factors are identified in Table 1 below.

Project Challenged Factors	% of Responses
1. Lack of User Input	12.8%
2. Incomplete Requirements & Specifications	12.3%
3. Changing Requirements & Specifications	11.8%

Table 1. The Standish Group Project Challenged Factors

The fact that top three (of more than ten) factors are related to requirements tells us that the industry is not able to effectively work with users to understand and describe their requirements.

Even if this report was published in 1994, we have to acknowledge that we still have not been able to make order out of chaos.

After a case study of selected projects they identified 10 success criteria where the top one was "User Involvement".

These results clearly indicate that the industry must focus improve the process of requirement engineering.

In this paper we will illustrate the importance of a methodical requirements process and practical use of such a process. In ConsultIT we have adopted a process which is based on Rational Unified Process. Through several projects, both internally and externally, we have gained considerable experience in the use of a requirement management process. Through our experience with such a process we have adapted the process in a practical and realistic way. We will share some of these experiences in this paper.

It is not in the scope of this paper to discuss in detail the different techniques used in the process.

2. Requirement Chaos

Many projects suffer from requirements chaos, where chaos, as defined in Merriam-Webster's dictionary, is "a state of utter confusion". It follows that the requirements chaos, results in confusions between the project's stakeholders. At the end this will lead to:

- Cost and time overruns
- Poor product quality

- The project fall short of the user's or business' expectations

Even if a large part of the software industry acknowledge the fact that better requirement management is an important success criterion for successful software projects, we still se that projects fail. There are many reasons for this:

- Despite acknowledgement of the importance of quality requirements, we still lack ability and willingness to prioritize and carry out the necessary steps in practical use.
- We live in a dynamic world where things change. This is also true for the user's needs and understanding of the system to be developed. This inability to manage change as part of the requirement process often leads to budget overruns and time delays.
- Business goals are missing or lose focus during the development process. Less critical factors or tasks overshadow the business goals and are the reasons for the overruns.
- Overoptimistic project estimates. Our limited ability to foresee the future must be compensated for more systematic estimation approach. We must identify the risk elements and there will always be unforeseen factors we must consider. We often squeeze the estimates to enhance our probability to win the contract.
- Lack of requirement focus from developers. A developer may not think much about requirement management, compared to design and implementation techniques. Many developers believe that requirements are the responsibility of the analyst and that the developer's involvement is reduced to simply receiving the requirements specifications.

Fortunately the diagnosis and treatment of the requirement chaos doesn't need to be very complex. We need to change from chaos to order with some sort of a methodical requirement process. Hence, an improved requirement management process is a vital part of software process improvement (SPI) in a software company.

3. Requirement Management

“The Chaos Report” [1] is strong evidence that controlling requirements is the single most effective action a project team can take to improve project outcomes and assist in our goal of delivering quality software, on time and on budget.

A requirement process must take into consideration the root causes to the requirement chaos described in chapter 2, and it must be practical and realistic.

Most important is a focus on the business goals and the stakeholders' needs in the requirement process. The business goals must “drive” the process, ensuring that we develop the right system which the business and the stakeholders really needs.

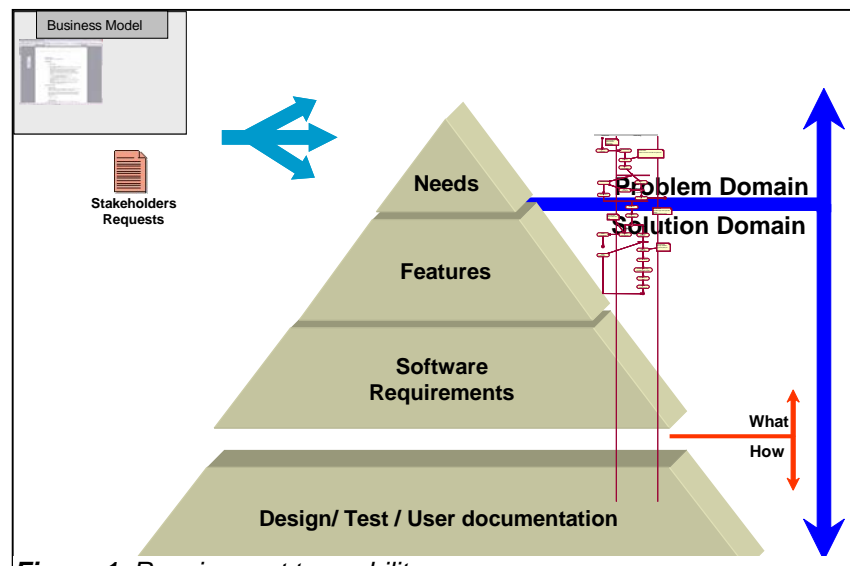


Figure 1. Requirement traceability

Figure 1 shows the different elements in the development process, where the business model and the stakeholders requests are important input to the process. First we must focus on the problem domain, identifying the elements that look like problems waiting to be resolved.

After we have defined the problem domain we are ready to form a solution to the problem. In the solution domain we consider the set of activities, the “things” we must build to solve the problem, and the artifacts we must create in the process.

4. Stakeholder Needs

Before we build the new system, we need to make sure that we understand the real needs of the business and the stakeholders in that problem domain. Without this understanding we face the risk of building a system not needed.

In Rational Unified Process® (RUP) a stakeholder is defined as:

“An individual who is materially affected by the outcome of the system.”

In other words, not all stakeholders are users and we have to understand the needs of both communities. A manager would typically have an interest in the system. We must consider this interest when building the system. The manager is not necessary a user of the system.

It is important to understand that in order to understand the real needs; we also must consider the business’ real needs. The business is often represented by a business model defining the business goals and business processes.

In Rational Unified Process® (RUP) a stakeholder need is defined as:

“The business or operational problem (opportunity) that must be fulfilled in order to justify purchase or use.”

The medical doctor seems really frustrated when trying to gather the necessary patient information before an operation. Maybe the information must be collected from different sources from different hospital, resulting in a time costly delay. What the doctor needs is a user friendly system that provides all the necessary information needed in an efficient way.

A security manager, on the other hand, is concerned about the juridical aspect of making patient information available.

The Process

Start identifying the business and stakeholders needs by inviting some key people who already have been involved in the project to a workshop. Maybe they were involved in the project participating in pre-project activity, business modeling or domain modeling.

In the workshop we identify the stakeholders in the project, and describe each stakeholder’s profile in a table. We then identify which individuals will represent each stakeholder role and we decide which individual we want to talk to.

We then invite each of the selected stakeholders to an interview where the goal is to identify and understand their needs. It is very important not to loose focus on the problem domain, because in such meetings there is a tendency that the stakeholders want to describe his or hers opinion on what the solution should be.

Also remember that the system should support the business goals. Use a business model, if one exists, as an aid in the interviews. If a business model doesn’t exist it will often be useful to create a simplified business model illustrating the business processes related to the problem domain.

A business model describes the business processes performed by different stakeholders. In each business process the stakeholder performs some activities. Some of these activities could be performed more efficiently if supported by a new system, and the business and activity models help us identifying potential improvement in the business processes.

Document the needs in short terms in a table (se example in *Table 2*) and don’t let the interview last too long (maximum 2 hours). It is also important not to involve too many people in the interview, just the interviewee, the interviewer and maybe one additional project member which also can act as a scribe.

No	Need	Priority	Concerns	Current Solution	Proposed Solution
N101	List of relevant patients	High	Lot of manual work if one would find patient with the specific disease (heart disease).	Receives different list from different sources (patient systems).	Integrate the system against different patient system and receive the patient list directly into the new system.

Table 2. Example of list of stakeholders needs

At first the needs that are collected in the interviews are actually just stakeholder requests, until these requests are analyzed and filtered. Some requests are not in the scope of the problem domain, some are just unrealistic wishes and some are obviously a matter for the future. The resulting list is the real needs and the definition of the problem domain.

Thorough work with the business and stakeholder needs will ensure that we describe the system needed by the business and users. Involving the key stakeholders early in the process reduce the possibility for unpleasant surprises later in the process. Many development projects have experienced problems when deploying the software, because of fatal security issues. Talking to the security manager and operations department early in the requirement process would prevent such problems.

5. Features

After defining and describing the problem domain we need to describe the system to be built. This means forming a solution to the problem.

The features are a first outline of the system, describing the system in an abstract way.

In Rational Unified Process® (RUP) a feature is defined as:

“An externally observable service provided by the system which directly fulfills a stakeholder need.”

A feature is, in other words, a service that the system must provide to fulfill one or more stakeholder needs.

Features are expressed in natural language using terms that are familiar to the stakeholders. They are not refinements of the stakeholders' needs, but the top level solutions to the problems being solved. Typically you should be able to describe the system by defining 25-50 features that characterize the behavior of the system.

In the doctor example, a typical feature could be that the system must be integrated with the X-ray proprietary system.

The Process

The features are best defined by the system analysts; in cooperation with key stakeholders if needed. Each need is analyzed and the features are identified and described based on those needs. The features must not explain “how” the system should do it; only what the system should do as a response to the problem being solved.

Again, it is important not to lose the business focus. The features will shape the system being built, and we want to make sure we build a system the business and the users want and needs. It is no rooms for fancy non-business related features here. All features must be traced back to the needs.

Document the features in a list, describe them briefly, and give them attributes to help define their general state and priority.

No	Feature	Description	Need
F1	Fetch patients	The system should fetch patients with specific disease from different patient systems in the region.	B101

Table 3. Example of a feature list. Notice the reference to a specific need.

6. Requirements

The features could be characterized as abstract requirements, so we need more detailed requirements to describe the system behavior. A requirement is defined as “a condition or capability to which a system must conform”.

We could separate requirements into functional and non-functional. Functional requirements specify actions that a system must be able to perform, without taking physical considerations into consideration. A use case model and use cases are useful techniques for describing functional requirements. Non-functional requirements describe only attributes of the system or attributes of the system environment. Examples of non-functional requirements are Usability, Reliability, Performance and Supportability [2].

The Process

There are several different approaches to identifying and describing the requirements. One approach is to sit down and refine and elaborate the features defined earlier. Each feature is refined by one or more requirements and each requirement serves one or more features. The advantage of this approach is that we avoid unnecessary creativity in the requirement process, which often results in unnecessary requirements. The disadvantage is that we are dependent of a complete and well defined set of features, or else we may miss some important requirements.

As mention above, use case techniques are useful when describing functional requirements. This approach is a more complex process than the approach described above, but will hopefully give us better requirements.

First, the use case model is defined by interviewing one or more users. Different creative techniques could be applied in the interviews. We have good experience with drawing the use case model “live” using a software tool (such as IBM Rational Rose®).

When the use case model is defined, it is time to elaborate the use cases and developing additional use cases as we gain new knowledge to fully elaborate the system. The use case elaboration is done in user interviews. The steps in the use cases are refined into more and more detailed system interactions.

The interview should be carried out with a limited number of people. It is important not to involve too many people in the interview, just the interviewee, the interviewer and maybe one additional project member which also can act as a scribe. Too many participants make it more difficult to achieve the objectives. Many people have many different opinions, which are more difficult to govern, and less critical factors overshadow the important ones. A good facilitator can interrupt participants when they don't follow the agenda, but participants often take offence of what they think is an obstruction of their creativity. This can quickly turn the participant hostile, and more difficult to govern.

The interviewee must not loose focus in the interviews as there is a tendency that the participants would like to talk about the solution or details about the user interface. The user interface should not be part of the use cases. The user interface should rather be defined in a prototype. It is also useful to involve one or more developers in the use case elaborations, because the use cases can be considered as the contract between the users and the developers.

The features are useful inputs when defining the non-functional requirement. Hopefully we have enough information from the interviews with the stakeholder to describe the non-functional requirements. Otherwise we could carry out additional interviews to obtain the necessary information. Non-functional requirements are documented in natural language like: “The system should manage 100 simultaneous users.”

The features, defined earlier, should also be incorporated in the use cases and the use cases should service one or more features. A feature could be serviced by one or more use cases.

It is important to continuously refine the requirements during the development process. A “freeze” of the requirements could lead to unsatisfied users and stakeholder. Nevertheless it is important to strive to complete as much as possible of the use cases in early phases.

7. The Artifacts

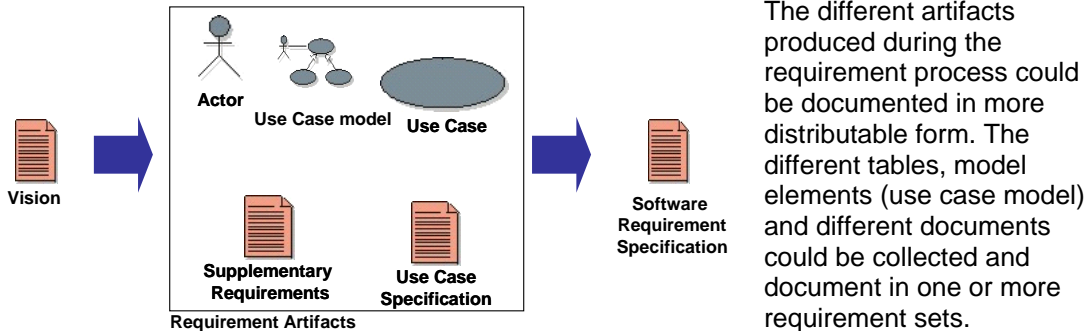


Figure 2. Requirement Artifacts

One of the most important artifacts in the requirement discipline is the Vision document. The Vision document contains an outline of the envisioned core requirements, and provides the contractual basis for the more detailed requirements.

The Vision document contains an overall problem statement, key stakeholder profiles, stakeholder needs, a list of features and perhaps some example use cases. The Vision document may be called by other names, such as Project Charter, Product Requirements Document, Marketing Requirement Document, and so forth. No matter what the name is, the Vision highlights the overall intent and purpose of the system being built.

The functional and non-functional detailed technical requirements could be described in a more traditional way in natural language or in a more structured way as a model. In either case, both are artifacts containing the requirements. With requirements expressed in natural language, these are described in a document. If the requirements are expressed as use cases, we may have a model and maybe some other documents. This is often satisfactory, but in some cases there is a need to document all these different elements in a single document. This document could be called Software Requirement Specification. It may contain the use case model, use case specifications and the supplementary specification. The supplementary specification contains non-functional requirements and functional requirements which apply to several use cases.

8. Traceability

If we turn our attention back to Figure 1, we see the different elements in requirement management and that there is some kind of relationship which may exist between the elements.

In requirement management it is important to be able to elicit, organize and document the requirements. An important quality attribute is the ability to trace the different elements through the development process (specification, design, implementation, and test). A good traceability mechanism shows the impact changes may have on the different elements. For example, if there is some change to a feature, we may need to update several use cases, some design elements, some code, and some test scripts. We also must verify that the changes to the feature comply to the business and stakeholder needs.

Traceability could be implemented by creating different tables with a relationship to the different related elements. For example, if we have a table with features, we could reference each of them to one or more needs. There also exists different tools that help with the tracing.

9. Summary

According to the "Chaos Report" [1] controlling requirements is the single most effective action a project team can take to improve projects outcomes and assist in our goal to delivering quality software, on time and on budget.

A well defined practical and realistic requirement management process has proved to be a good technique to control the requirements. The process must be business and user oriented in order to ensure that we define and build the right system.

The requirement management process must attack the requirements in the right order. We define the problem domain before the solution domain. Start describing the solution on an abstract level, then in more and more details.

Don't fall into the detailing trap, ending up with a description of the solution instead of the problem or endless feature list. Describe the detail where it belongs, namely in the requirement specification. Remember that, when detailing requirements, the users should understand the specifications and that the developer receives all the necessary information to develop the system.

The requirement process described in this paper has change the way we carry out projects in ConsultIT. We are focusing much more on requirements in the development project, resulting in better project estimates and provide more accurate fixed price offers. Also the requirement process helps us develop more quality software, with less frustrated users, less rework and fewer budgets overruns.

We have also adopted a subset of the process in other non-developing consulting services, such as advisory services like purchasing, security, architecture, and so on.

It is difficult to obtain absolute proof that the requirement process is the single most important factor for project success. Still, based on our experience with the process, we can conclude that it is a prerequisite for success. In our first project adopting the new requirement process several years ago, we came to a point half way into the requirement process where we experienced some doubt in the project team. This was due to more hours spent on requirements than earlier. Someone got cold feet and went back to a more traditionally requirement process. When the development started, we could clearly see the difference between the two. The requirement part created with the traditionally requirement process gave us much more frustrated developers due to unclear requirements. Most of the requirements had to be refined or rewritten, resulting in more hours spent than initially estimated for the requirements work. The requirement part that was created with the new requirement process did not need to be refined or rewritten in such an extent.

10. References

- [1] The Standish Group, "The Chaos Report (1994)", 1994
- [2] Robert Grady 1992. *Practical Software Metrics for Project Management and Process Improvement*. Prentice-Hall.
- [3] IBM Rational Unified Process ®

11. Author CV

Rolf Nergaard

received his M.Sc. degree in Computer Sciences at the University of Trondheim in 1993. He has since been a consultant, and is currently working at ConsultIT. He has been a mentor in software process improvement and architecture in several projects for several years. He has also helped several companies to improve their requirement management. He has also been a mentor with several companies in implementing a new software process, including Rational Unified Process®.

Risk-driven Software Process Improvement - a Case Study

Jakub Miler¹, Janusz Górski

*Gdansk University of Technology, Narutowicza 11/12, 80-952 Gdansk, Poland
{jakubm, jango}@eti.pg.gda.pl*

Abstract

The paper reports on a case study of practical application of a risk-driven software process improvement framework in a real-life software project. The framework assumes explicit modelling of the process and its deficiencies (risk factors) as well as provides for process evolution. It also includes dedicated techniques to identify process risks and to derive from them suggestions for process improvement. The techniques are embedded in a recurring procedure involving process modelling, risk identification and process improvement steps. The paper presents the case study objectives and reports on the results of two phases aiming at process improvement. Finally, it discusses the case study results as well as presents the issues open for further research.

Keywords

process modelling, risk identification, process improvement, case study

¹ Research partially supported by the grant no. 4 T11 C 008 25 from Polish Research Council (KBN)

1 Introduction to Risk-driven Software Process Improvement

The aim of any software project is to provide the stakeholders with a satisfactory software-based solution of their problem within the schedule and budget limits. The risk of poor product quality and schedule or budget overruns is high which is confirmed by a number of cancelled, delayed or overpaid projects. Effective management of those risks is presently perceived as one of the most important areas of project management [1, 10]. Still, current software processes leave a considerable space for improvement. As process improvement aims at maximizing process quality and effectiveness while minimizing its risks, therefore the support for identification of the most risky process areas and their potential improvement is especially worthwhile.

Current risk identification practices adopt primarily two techniques: checklists and group effort (e.g. brainstorming). Checklists such as [3, 5, 13] help to control the identification scope and protect from overlooking significant risks but they are often too general and do not relate well to actual software processes. Group effort studied e.g. by J. Kontio [4] benefits from synergetic use of human intuition and experience but it exhibits problems with scope focusing and control. Consequently, both existing approaches provide limited output aimed at the process improvement.

The paper proposes a framework for the risk-driven software process improvement. The following features characterize the framework:

- explicit process modelling [6, 9] as well as providing for model evolution [2],
- interpretation of model deficiencies as process risks and areas for potential improvement,
- modelling risks with risk patterns [6],
- supporting risk identification by referring to model metrics and consulting referential models [7],
- deriving suggestions for process improvement from identified deficiencies,
- running the improvement implementation and the modelling as a continuous process.

The recurring procedure of continuous risk-driven process improvement comprises the following steps:

1. initial modelling of the original process,
2. discovering the process risks by applying selected risk identification techniques,
3. (risk-driven) selection of process improvements,
4. systematic transformation of the original process model into the improved process model.

The procedure of continuous risk-driven process improvement is shown in Figure 1.

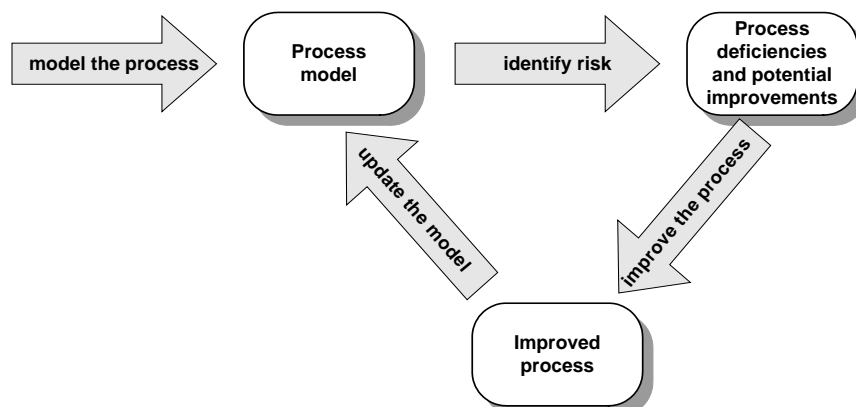


Figure 1: The risk-driven process improvement procedure

2 Case Study

2.1 Methodology

The case study aimed at repeated identification of risks in a real-life software project and providing suggestions on possible process improvements.

A software project involving participants from several countries and scheduled for over a dozen months has been chosen for the case study. The project objective was to build a complex, distributed information system based on a novel architecture and business model. During the case study the project remained in the initiation phase. The project description and plans were used in the case study.

The case study comprised 2 phases (cycles of risk-driven process improvement):

- Preliminary risk identification carried out in January 2004 based on formal project description,
- Second risk identification with respect to the improved process carried out in April 2004 based on the Quality Plan, partial Development Plan and the same project description as in Phase 1.

The case study was evaluated by:

- subjecting the risk identification results to the judgement of the project managers based on their intuition and individual experience,
- examining the ratings and priorities the project managers assigned to the identified risks at a risk analysis session,
- assessing the scope of improvement initiated by the project managers after the risk identification.

2.2 Results of Phase 1

To begin with, a process model was built based on the project description. Due to the initial phase of the project, the development process was planned at a rather general level. The most detailed activities covered several months. The project description did not define any qualitative features of the activities, artefacts and roles as those elements were left to be defined later in the plans of the particular project areas. The final model comprised 55 activities at 3 levels of detail, 49 artefacts and 45 roles.

The process risks were first identified using model metrics [7] that were applicable to the model. The metrics indicated two activities and five artefacts for further investigation which resulted in identification of four significant risks. For all those risks, their scenarios have been developed with the help of risk patterns [6]. The example of an identified risk is given below together with the corresponding scenario (first expressed with the help of risk patterns, then expressed as a natural language statement).

Risk factor: *Distributed, multinational development of business models*

Risk scenario (in terms of risk patterns): If **New Business Modelling**<activity> loses **Consider regional differences in reality**<practice> then **System Requirements Specification (Vision)**<artefact> loses **Conformity to target reality**<feature> and **Use Case Design**<artefact> loses **Conformity to target reality**<feature> and then **Pilot One**<artefact> loses **Conformity to target reality**<feature>.

Risk scenario (natural language): *Business modelling is skewed by local viewpoints and results in missed target reality of the pilot implementation of the system.*

The risks were further identified by comparing the analyzed model with the Rational Unified Process (RUP) [11] taken as a referential model. RUP was particularly chosen as being well structured, defined in detail yet generally applicable and finally compatible with the development process of the studied project. Due to the limited resources for the case study, a complete mapping of the analyzed model on the RUP referential model was not developed. Instead, the most evident differences were taken into consideration. This way, three additional risk factors were identified. One of them is given below together with its exemplary scenario.

Risk factor: *Configuration & Change Management is not explicitly defined*

Risk scenario (in terms of risk patterns): If **Configuration & Change Management**<activity> is not performed then **System Integration**<activity> loses **Keep the set of integrated subsystems coherent**<practice> and then **Pilot Deployment**<activity> takes more time than expected.

Risk scenario (natural language): *Without explicitly defined change management process the pilot may not be integrated and deployed on time.*

The risk identification step was completed by comparing the analyzed model with the referential model derived from the Steve McConnell's 'Complete List of Schedule Risks' [5]. Following the same procedure as in the previous step, four additional risk factors were identified. One of them is given below together with the exemplary scenario.

Risk factor: *Long duration of the project and mainly part-time employment of the staff.*

Risk scenario (in terms of risk patterns): If **Project**<activity> loses **Maintain personnel continuity**<practice> then **Project**<activity> loses **Personnel**<role> and then **Project**<activity> loses **Avoid excessive schedule pressure**<practice> and **Pilot One**<artefact> loses **Completeness**<feature>.

Risk scenario (natural language): *The project can have difficulties with keeping part-time employed staff resulting in workforce shortages, more effort for available staff and limited scope of the pilot.*

In total, 11 risk factors were identified in Phase 1 of the case study. The results were then compared with the 9 risk factors indicated in the project description (identified by the project management). 6 out of the 11 risk factors identified in Phase 1 of our case study were also indicated in the project description. Still 5 of them were new regarding the project description and resulted in important suggestions for the process improvement which otherwise would have been missing.

Partial correlation of the detected risk factors with the factors indicated earlier by the project management confirms that the proposed method is consistent with the intuition and experience of the managers of software projects. The 5 new risks were then communicated to the Project Management Board who judged them important and initiated activities aiming at the process improvement.

The process was improved by defining its deficient areas in detail in the newly issued documents and initiating new activities related to the redefined process. The improvements covered in particular configuration management tools and practices, and procedures for quality assurance. All of the 5 deficient process areas identified with the help of the proposed framework were subject to the process improvements.

2.3 Results of Phase 2

The second risk identification and improvement attempt focused on the managerial issues such as operational management, quality management, communication management, software management and so on. A partial model of those areas was built from the available data. The model comprised 85 activities at 3 levels of detail, 16 roles and 37 artefacts.

Due to incompleteness of the process model the risk identification technique using model metrics could not be effective. Instead, the technique of comparison with a referential model was applied. For the same reasons as in Phase 1, the Rational Unified Process (RUP) [11] was selected as a primary referential model.

As a result, some 26 risks were identified suggesting possible process deficiencies. The examples of identified risk factors are given below together with possible scenarios.

- Quality management – establishing success criteria and metrics

Risk factor: *Immeasurable success criteria, weak metrics*

Risk scenario (in terms of risk patterns): If **Measurement Plan**<activity> loses **Use quantifiable and objective metrics**<practice> then **Pilot One**<artefact> loses **Defined scope**<feature> and then **Pilot One**<artefact> loses **Completeness**<feature>.

Risk scenario (natural language): *Product scope is arbitrarily redefined and intended product scope is not covered.*

- Document Management – general risks

Risk factor: *No explicitly defined procedures for maintaining traceability of key business and design decisions*

Risk scenario (in terms of risk patterns): If **Document Management**<activity> loses **Maintain traceability of key decisions**<practice> then **Documentation**<artefact> loses **Consistency**<feature> and then **Subsystem**<artefact> loses **Compatibility**<feature>.

Risk scenario (natural language): *Key business and design decisions are vague, inconsistent or contradictory in different documents resulting in incompatibility of partial commitments.*

- Communication Management – communication by project portal

Risk factor: *Poor performance of hardware and software platform of project portal*

Risk scenario (in terms of risk patterns): If **Project portal**<artefact> loses **Platform performance**<feature> then **Personnel**<role> loses **Motivation**<capability> and **Communication**<activity> loses **Follow defined communication paths**<feature> and then **Communication**<activity> takes more time than expected.

Risk scenario (natural language): *Discomfort in portal usage causes users' rejection and hasty construction of alternative communication means that impacts communication.*

The identified risk factors were taken as input to the risk analysis session carried out by the Project Management Board - PMB (some risk factors were merged together which resulted in total of 20 risks). The list of risks was also extended with the risks identified independently by PMB members. The risk analysis session involved rating and prioritizing the identified risks by some 15 members of the PMB. As a result, a list of project top 10 most important risks was elaborated. Of those 10 risks 7 were identified with the help of the proposed framework.

The risks found in this phase indicated more detailed deficiencies than those identified in Phase 1 and suggested areas for further process improvement. To save project resources only the areas indicated by 10 topmost risks were taken care of. For instance, some new actions were defined and launched in the area of technology management including tools selection and design decisions and communication means were defined in response to risks related to poor sharing of scientific know-how.

2.4 Summary

All the analytical tasks in the case study were carried out by one of the authors. However, the key part (risk identification) was performed according to precisely defined procedures of the proprietary risk identification techniques. Thus it may be presumed that every risk analyst would obtain comparable results. However, a complete correlation (reproduction) of the results is difficult to achieve, because while being considerably formalized the proposed method still engages the analytical capabilities and the individual perception of the analyst.

An effort on every phase of the case study was carefully measured. Phase 1 required 30 man-hours, 22 of which were spent on initial building of the process model while risk identification took 2-3 hours for each of the techniques and referential models. Phase 2 took 8 hours, 3 of them used to model the managerial area of the project and the remaining 5 to identify risks and potential improvements. The effort of the risk identification steps may be reduced by taking advantage of the already built dedicated supporting tool [8], which is currently adapted to the proposed framework.

3 Conclusions

The paper presented a two phase case study aimed at assessing the feasibility, effectiveness and efficiency of the risk based approach to improvement of a real-life software project. In the first phase,

11 risk factors were identified with the help of two different risk identification techniques. 5 of the risks were not known to the project management. All identified risks provided valuable suggestions for process improvement. In effect the process was significantly redefined. The repeated application of the approach in the second phase led to identification of some 26 more detailed risks and resulted in further process improvements.

The results of the case study demonstrate that the proposed framework is able to reveal new, previously undetected risks that provide important (in the opinion of project managers) input for process improvement. The framework requires that the process is defined in the level of detail sufficient to build its model which is then used during risk identification. A high-quality relevant referential model is also necessary if the model comparison technique is to be used. As the proposed approach is based on models and their metrics we expect the delivered results to be highly independent of the analyst's intuition and experience. It also provides a good base for automatic tool support.

Further research is planned in the following areas:

- further case studies with industrial partners,
- investigation of new metrics to improve risk identification,
- building the proposed framework into a supporting tool [8],
- using the framework in different domains such as e-health and/or e-commerce.

The results presented in this paper originate from a wider context of research towards a holistic approach to risk identification and process improvement in software projects supported by dedicated tools. The description and the results of the research are available at [12].

4 Literature

1. CMMI for Systems Engineering/Software Engineering/Integrated Product and Process Development, V1.1, Tech. Rep. CMU/SEI-2002-TR-004, Carnegie Mellon University, Software Engineering Institute, Pittsburgh, PA, 2002
2. Jaccheri M.L., Conradi R., Techniques for Process Model Evolution in EPOS, IEEE Trans. on Soft. Eng., Vol. 19, No. 12, 1993, pp. 1145-1156.
3. Jones C., Assessment and Control of Software Risks, Yourdon Press, New Jersey, 1994.
4. Kontio J., Software Engineering Risk Management: A Method, Improvement Framework, and Empirical Evaluation, PhD Thesis, Helsinki University of Technology, Finland, 2001.
5. McConnell S., Rapid Development, Microsoft Press, 1996.
6. Miler J., Górski J., Risk Identification Patterns for Software Projects, Found. of Comp. and Dec. Sci., Vol. 29, No. 1-2, 2004, pp. 115-131
7. Miler J., Górski J., Identifying Software Project Risks with the Process Model, proc. of 17th International Conference "Software & Systems Engineering and their Applications", Paris, France, 2004 (accepted for publication)
8. Miler J., Górski J., An environment supporting risk management in software projects, proc. of I National Conference on Information Technologies, Gdansk, Poland, 2003 (in Polish)
9. Software Process Engineering Metamodel Specification Version 1.0, Object Management Group, November 2002.
10. PMBOK Guide, 2000 Edition, Project Management Institute, 2000.
11. Rational Unified Process version 2002.05.00.25, <http://www.rational.com/rup>
12. RiskGuide project website, <http://mkzlwaj.eti.pg.gda.pl/RiskGuide>
13. Sisti F. J., Joseph S., Software Risk Evaluation Method, Tech. Rep. CMU/SEI-94-TR-19, Carnegie Mellon University, Software Engineering Institute, Pittsburgh, PA, 1994.

5 Author CVs

Jakub Miler

He is a PhD student at the Department of Software Engineering, Gdansk University of Technology. His primary research area is software risk management. Under the supervision of Janusz Górski he prepares the PhD thesis titled "A method of identification and analysis of risk in software projects". Recognizing the risk management as an effective method for software process improvement, he works on useful techniques for risk identification and analysis together with a model of risk representation and classification. He also investigates the methods of communication of risk-related information within a project team and develops the RiskGuide software tool. In 2000 he presented the master's dissertation titled "Computer system for supporting risk management in a software engineering project".

Janusz Górski

He is a professor and the Head of the Department of Software Engineering at the Gdansk University of Technology. Authored and co-authored 6 books and more than 150 research papers. Led several software development projects in the domains of process control, telecommunications and information management. His present interests include software and system engineering and risk and trust management. In those areas he actively carries research (through a number of EU funded and national projects) as well as cooperates with industry through common projects and consulting. In 1999 he initiated the National Conference on Software Engineering which is since then organized every year.

Increased Value through Reduced Risk in Joint Industry /External Researcher Projects – Towards Guidelines

*Lars Bratthall, Torbjørn Skramstad - DNV Research – www.dnv.com
{lars.bratthall, torbjorn.skramstad} at dnv com +47 6757 7127
Edda Mikkelsen - DNV Software
edda.mikkelsen at dnv com
Email spam stop: Change "at" to "@" and use "dnv.com" for server*

Abstract

Two research projects with external researchers that study some work processes of each an industrial organization are compared for the purpose of providing higher value of the research for the industrial partner. Both research projects studied spanned approximately six months. In case A, a researcher has visited the studied organization 2-3 days per week during six months, and has had access to virtually all material produced, all people, etc. The researcher had a strictly non-participant role. In case B the researchers were in much less contact with the industrial partner. This paper compares the two projects with respect to a number of characteristics resulting in a number of questions that can be asked in advance for an industry considering working with a researcher are suggested. A threat to the validity of this paper is that the analysis made in this paper was not anticipated in neither case A nor in case B in advance. This suggests that some of the information available during the original research projects might have been forgotten before this analysis has been made. On the other hand, had a coming analysis like this been anticipated in advance, it is not unlikely that the research setup would have been different. Based on experiences from our own first trial, it is believed that these questions can help enhance value of external research for the industrial partner, through better known and thus possible to manage project risk.

Keywords

Research value, joint research, external researcher, risk reduction

1 Introduction

Industry partners are often approached by external researchers, or looking for external researchers, to perform some kind of research with an outsider's eyes. In the process of selecting what to engage in, an essential question for the industrial partner is: Is it likely that the research conducted will eventually have value enough to merit the investments made in the research?

To answer such a question, both the study target, and the research methodology applied in the industrial context must be understood. If the industrial partner is represented by someone trained in research methodology, sufficient skills are probably available to judge in advance. However, in practice this is often not the case. Thus, there is need for some decision aids. The objective of this paper is to contribute to the industrial partner's decision process regarding whether to engage external researchers or not.

In this paper, we further focus on the attempt to study larger work processes performed in an industrial

organization over longer periods of time, because larger work processes represent larger investments in companies. Hence, there is often an increased financial motivation for evaluating them compared to smaller issues.

We have chosen two particular projects to compare. The reason for this is that they have similar duration, they work in organizations of similar “maturity”, and the size of their software products can both be described as “very large, transactional oriented”. Thus, convenience sampling is used, but the merit of this is that the effects of the *object* studied are reduced, and instead findings related to *how* the research is conducted are derived. The cases are referred to as “A” and “B”.

In total, the focus of this paper is set to *Analyze two projects, for the purpose of minimizing the risk for the industrial partner, in the context of a) academic research on industrial organization processes; b) half-year duration of study; and c) large scale object for study, such as set of interconnected work processes.* The purpose is to present some control questions that can be used before engaging in similar endeavours with the objective of increasing the effectiveness and efficiency of joint research, as seen from the perspective of the industrial partner.

To pursue the focus, the two research projects studied are compared with respect to a set of characteristics. These are: **General characteristics:** The studied organization, the studied set of processes. **Initial expectations** from researcher and industrial partner. **Research method characteristics:** Degree of contact researcher/industrial partner, data collection methods, data analysis methods, presentation of results. **Interest characteristics:** Academic researcher involvement, industrial organization interest. **Result characteristics:** Outcome of project usefulness. **Overall evaluation:** What should have been done otherwise?

2 Cases

In **organization A**, a 15 person team developed a large financial software module. The research focus was on how the organization could work faster. Organization A had little – if any expectations on the outcome from the beginning of the research, while the researcher expected to see a capable development process being acted in practice. In this project, a single researcher took a non-participant observer role two or three days a week, during six months. All written material was available for analysis, including previous versions of documents. All meetings were open for listening to. The study is considered methodologically advanced, in that multiple sources of information were used for triangulating findings. Feedback sessions, interviews, etc., also ensured multi method triangulation. Analysis includes both statistical analysis of some data, as well as subjective analysis grounded in “tagged evidence”, i.e., individual pieces of evidence with a reference number.

It is interesting that in this project, the perceived interest from both parties grew over time. The researcher learned a lot, and as confidence in the researcher grew from the industrial organization’s employees, they also voluntarily shared more information without being asked about it. It is believed that the presence of the researcher contributed strongly to sustained interest in the research project. The resulting research report was presented multiple times to organization A, and it resulted in lasting changes to the development process – and practice.

In **organization B**, about 100 people develop a variety of technical software products for engineering disciplines. The research focus was on how one could increase the use of a knowledge management system, developed in previous research and development efforts. The initial set of expectations from the industrial organizations was high; several high level managers approved of the research project in advance, which is more than in case A. There was a perceived match between researcher competency, and the current needs of the organization. The researchers were happy to pursue a research focus that appeared to be an area with few studies: They could study how increased support could increase the use of a knowledge management system.

The research process in case B is very different from that in case A. At the beginning of a six-month period, the external researchers visited organization B. A questionnaire constructed together with the industrial organization, and almost 100 (~“all relevant”) persons answered. 20-minute interviews were held with a number of persons in the industrial organization. During the six month project, organization B was supposed to change the way it worked by using more of a knowledge management tool.

Largely, this did not take place as anticipated, due to organizational changes. After six months, the researcher were supposed to returned for data collection: The same questionnaires should be used, and in addition, 20 minute per person interviews were to be held. Instead, as the anticipated change did not take place, one made a set of new hypotheses grounded in the first survey, and validated these from other data.

The outcome of the research project was less than planned, but the reduced outcome is valuable to a smaller subgroup of the full organization.

3 Analysis

Though there are many factors that might have led to the relative success case A, and similarly the limited success of case B, we here attempt to subjectively pick out what are believed to be key success factors.

First of all, in case B, the research project did have more management support. This is usually considered a key success factor, but obviously, this is not enough in itself. Secondly, the amount of researcher presence stands out as a key differentiator. In case A, the researcher was present almost every week during six months, while in case B, the researcher was much more away, than present at the industrial organization. But what was the effect of this? First, the industrial organization had little “emotional incentive” to focus on the research project. Secondly, the researchers themselves have had little chance to change the direction of the research project, or follow up potentially interesting leads. It is possible that the studied organization saw this during the study, and thus also lost interest in the research project: The external primus motor may have been seen as not really interested in the research project.

Another key difference is that case A wanted to study how things were, while in case B, one wanted to study the results of a change. And this change never took place to the extent anticipated. This suggests that the risk of organizational change in the industrial partner must be balanced in the choice of research questions, as well as methodology.

4 Guidelines

In this section, we give some guidelines for the industrial partner, based on earlier analysis. The guidance is given in the form of questions that an organizational partner can use while considering the risk in engaging in cooperation with an external research partner.

Strategy fit: In this paper, we have seen how the research resulting from case B in practice will have lower value than anticipated, while case A had good value. The following two questions would be among the first to ask:

Q1: Given that the objective of the research is fulfilled, can the expected new knowledge improve the industrial organization enough to merit the research cost?

Q2: If the research objective itself is not enough to merit investment in the research, can other benefits motivate the investment?

Feasibility: In case A, the research reached its initial objective, while in case B, the industrial partner did not perform all activities as necessary. This resulted in the researcher adjusting objective of research under way, into something that has much lower value for the industrial partner than initially anticipated.

Q3: If the industrial partner is supposed to do something in the course of the research, is there any risk that this will not be performed during the required time interval?

Q4: If the research project spans some non-trivial time, will there be a driving force with enough presence in the industrial organization to maintain drive in the research project throughout its duration?

Quality of results: In case A, the research method ensured trustworthy results. In particular, various kinds of triangulation ensured method robustness. In case B, one was forced to redefine hypotheses under way to get any usable results.

Q5: Does the industrial organization itself have enough research methodology knowledge to evaluate the potential trustworthiness of the results?

Q6: Will results be considered trustworthy? Are the threats to validity acceptable?

Q7: If research methodology experts are required in the industrial organization, are there such with enough time budgeted for the research?

5 Validation

To validate the suggested questions, they have been applied on three cases. The validation from cases A and B may be of limited value due to self-validation (“X” was learned from cases A&B, if “X” works on cases A and B it must be good), but case C is a third case that did not help construct the suggested questions. In case C, DNV has been asked to participate in a similar-duration research project, where some work processes of ours constitute the research object.

In below table, we show the tentative effects of applying the questions on the three projects. Note that “risk” can be seen as a calculation of probability * impact.

Table 1 Application of questions to three cases - validation

Q	Case A	Case B	Case C
Q1	Yes	Yes	Yes – but some changes to research questions was made because of this question.
Q2	No (thus a risk indicator)	Yes	Yes
Q3	No risk	High risk	Medium risk. This has lead to the identification of backup solutions prior to engaging in project. Without those, DNV might have declined project participation.
Q4	Yes	High risk	High risk. This has led to the adjustment of the time budget for some key employees.
Q5	No (thus a risk indicator)	No (thus a risk indicator)	Yes, but because of some time budget adjustments.
Q6	Yes, yes	Yes, Maybe	To be seen – thus a risk. Risk is balanced by keeping an eye on further not-yet-made research methodology choices.
Q7	No (thus a risk indicator)	No (thus a risk indicator)	Yes, but because of time budget adjustments.

In the table, it is no surprise that cases A and B show lower and higher total risk levels, as the questions were derived from these. However, the value of the questions is, at least as a pilot, validated when applied to case C. In this case, the table shows how several adjustments made to the setup of the research project, both to research questions, and to the industrial organization. It is expected that these changes will lead to lower cost of risk, hence higher value for the industrial organization.

6 Summary

This paper has presented a number of questions that can help an industrial organization judge the risk in engaging in research with external researchers studying some phenomenon in the industrial organization. The questions are based on two six-months research projects investigating some work-processes. The questions suggested are few, and can only be considered as a start in assessing the risk in engaging in research with external researchers.

Our first trial of the questions on a new project resulted in a) Some changes to the research questions from the external researchers; and b) Changes to the time budgeted for the research by our selves. It should be noted that the changes made in this case is also expected to have value for the external researcher. The expected value of the changes is such that these questions will become a part of our formal process library for assessing the risks when engaging in research on our own processes with external researchers.

7 Author CVs

Lars Bratthall

Mr. Bratthall received his MSc from Lund University in 1996 and his PhD in software engineering from Oslo University in 2001. He has been active as consultant in industrial software process improvement at Q-Labs, as researcher in the software research program at ABB Corporate Research, and is currently performing research related to the management of risks related to software-intensive systems at DNV Research.

Torbjørn Skramstad

Mr. Skramstad has more than 30 years of experience from the software engineering and software development field in various roles. He has also been quality manager in one of Norway's largest software consulting companies. His main areas of experience and expertise include software development methodologies, software quality assurance, software metrics, software project risk analysis and project management, and independent software verification/validation. His current interests are in the area of assessment of safety and reliability critical software and systems. Mr. Skramstad is also a professor in computer science and informatics at The Norwegian University of Science and Technology (NTNU). He is currently performing research related to the management of risks related to software-intensive systems at DNV Research.

Edda Mikkelsen

Mrs. Mikkelsen's work is concentrated on Software Development; mainly administrative applications implemented by relational database technology and object-oriented development tools. From a number of software development projects the experience covers project management, quality management, business process modelling, requirements analyses, data modelling, system design, test design, programming, testing, maintenance and support. In the latest years the focus has been on backing up ongoing software development projects with efficient methods and tools for their work. This includes suggesting external researcher to work with.

Software Quality Models: Decisions by Indian companies

Brendan Keane & Ita Richardson

Abstract

Many organisations look to improve the quality of software through improving their software process. What actually influences organisations to do this? Furthermore given the variety of models available, how do organisations decide which software process improvement (SPI) model should they choose for their purposes? Models such as the Capability Maturity Model (CMM), Capability Maturity Model Integration (CMMI), ISO 15504 Software Process Improvement and Capability determination (SPICE) and ISO 9000:2002 address similar aspects of software processes within the organisation. These aspects include: development, maintenance, products, security, services, each with specific focus and coverage.

To investigate the answers to these questions, a case study approach has been taken, whereby the authors have conducted research into software development organisations in India over a 3-month period. The analysis of this research is based on data collected using qualitative research methodologies such as interviewing and surveying. This paper presents results drawn from this research.

Keywords

Software quality models, selection, SPI, CMM, CMMI, ISO, SPICE

1 Introduction

Many organisations look to improve the quality of software through improving their software process (Richardson and Varkoi, 2003). Why are organisations choosing to implement software process improvement (SPI) models and/or techniques? Why would organisations choose to spend their time and money on implementing these models? Increasing global competition and rising customer expectations make product quality an important strategic priority (Hutt and Speh, 1998). SPI is essentially about continually improving the software process in an organisation, the quality of a software product is largely determined by the quality of the software development and maintenance processes used to build it (Paulk, Weber et al., 1995). Therefore by deciding to improve their processes, organisations are hoping this will have the knock on effect of improving the overall quality of their product.

The decision to implement a software quality model should not be taken lightly. A quick fix to a production problem, or "everyone else is doing it" should not be reason enough to attempt to implement one of the many models that currently saturate the market place. West (2004) warns that the race to achieve a specific maturity level (the perceived fix) may cause widespread cynicism, which in turn may lead to a grass-roots resistance to the process improvement initiative. Management must consider the primary business needs and goals of the organisation. They must attempt to discover which quality model or even combination of quality models would best suit their organisation. They also need to ensure that they have enough knowledge, expertise and commitment to follow the implementation process to its conclusion.

Once an organisation decides to implement SPI, they face the decision of which model to follow. There exists a multitude of quality models, standards and frameworks to choose from in the market place. The CMM, CMMI and their variants, developed by the Software Engineering Institute (SEI) from Carnegie Mellon University (CMU). The International Organisation for Standardisation (ISO) have developed the ISO 9000 series of standards as well as ISO/IEC 15504, which is sometimes referred to as Software Process Improvement and Capability dEtermination (SPICE). Other quality initiatives include: Trillium, ITIL and Bootstrap.

1.1 Use of Quality Models

Due to the nature of the CMM those seeking to gain lucrative US DoD contracts have to have achieved a specific CMM level within their organization. In these circumstances CMM may not have been implemented from a deep desire to improve process, but to satisfy a particular customer demand. However, Paulk (1996) cautions against using quality models for the wrong reasons and warns that "standards and models such as the Software-CMM can help organisations improve their software processes, but focusing on achieving maturity level without really improving the underlying process is a danger". West (2004) agrees with this stating "sometimes, process improvement — or worse, the model we are using, i.e., The CMM or CMMI — takes on a life of its own, and we end up doing process improvement for its own sake". Our research is specifically concerned with why people choose to implement process improvement models in the first place.

In a special report focusing on the impact and benefits of CMMI, Dennis Goldenson and Diane Gibson presented compelling evidence on the noted benefits that quality models such as the CMMI can bring to organisations (Goldenson and Gibson, 2003). Such results clearly show the benefit of implementing SPI initiatives in the organisations that were surveyed. However, what was not presented in this report were examples of implementation attempts failing in organisations for whatever reason. Understandably companies where process improvement initiatives failed are unwilling to come forward and report the hows and whys of their failure. But until such a time as credible data is available to document

these failures and why they failed, reports of the benefits and return on investment of implementing process improvement initiatives may be passed over by sceptics as picking the best from a bad bunch!

In the literature several criticisms of the various quality models were noted, including the impression that models are overly bureaucratic, or that while they may describe what it is to be an effective software development organisation they do not describe how to achieve this. A justifiable criticism levelled at the CMM is, when it comes to assessment, the CMM ignores the investment in technology an organisation has made (Bollinger and McGowan, 1991). Management in smaller companies had to be convinced that models could be tailored and applied on small projects as well (Pierce, 2000), fearing that it would be too costly for small companies to implement, even if they wanted to.

The original ISO standards were developed for use within the manufacturing industry. In spite of ISO 9000-3 which catered to the software community, software professionals shied away from the ISO standards, viewing them as being more hardware oriented than anything else. Richardson (1999) agrees with this, saying that the main difficulty experienced when applying ISO 9001 guidelines to software development is that the standard was not developed exclusively for software. Though the latest version of the standards ISO 9000:2000 address much of the criticisms of the earlier versions, there are still some concerns. In a survey of quality professionals by Qualityworld and SGS Yarsley, it was found that 33% of respondents were worried that the new standard would not be easy to apply to their companies, 29% believed that the standard would benefit from greater use of 'plain English' and 18% felt that there was still too much emphasis on document control (Daniel, 2001).

2 Research Methodology

Two main research questions have to be answered. First, given the concerns and criticisms of the various models, what prompts an organisation to change the way they do things and to implement an SPI initiative? Second, given the wide range of software quality models in the market, what does an organisation need to consider when making the decision on which model to adopt? To investigate the answers to these questions the authors conducted research within a software consulting organisation and its customers in India over a three month period. This type of research is not aimed at proving or disproving some theory, it is simply aimed at finding the answers to the research questions.

A qualitative approach was initially taken whereby small samples of data were collected and analysed. Interviews were conducted with software process consultants and with some process improvement personnel within customer organisations. Interview questions were open ended and semi-structured, allowing the interviewee to discuss issues at length while allowing the interviewer to collect as much information as possible. Questions focused on gathering information relating to the business concerns that prompted change, the criteria used in determining the best suited quality model and the desired achievements the management hoped the implementation would bring to the organisation. Once the interview stage was completed, interviews were transcribed, coded, analysed and conclusions and results were drawn from the information received.

Based on the information obtained from the interviews a questionnaire was developed. The questionnaires were distributed to various software and non-software development organisations in both India and the Republic of Ireland. Due to the sensitivity of some of the customer information required, the consulting organisation in India undertook to deliver and return the survey information to the authors. Results from these were used to support and validate the conclusions drawn from the interviews as well as to fill in any gaps in the information that the interviews highlighted.

3 Findings

From our initial meetings with software consultants it was evident that for software development organisations there is a clear business need to demonstrate mature software processes. This need does not arise exclusively from any internal pressure to improve processes or services, but the external pressure to stay alive in a competitive market place. These organisations must implement a quality model or framework, and more often than not the SPI model chosen is dictated by one of two things – either a direct customer requirement or the requirements of the market place.

The research then focused on software development departments within organisations whose main business was not software, but have significant IT departments. Interviews were carried out with customers who have implemented various SPI models such as those mentioned earlier. The software quality consultants who advise such organisations on their software process were also interviewed.

3.1 What prompts change?

When the research was carried out with Indian companies, it was evident that the three major factors influencing change were:

- Competitive advantage
- The bottom line
- Customer complaints

The need to keep up with competitors and the advantage within the market place to an organisation showing that they are, for example, at a given CMM level, is considered important. Prospective customers see the ability to demonstrate mature processes reflected by certification or maturity level as an influencing factor in selecting to whom they should give their business. The higher a maturity level an organisation has, the better chance they have of obtaining a valued customer contract. It was also evident that a business goal of breaking into a new market dominated by organisations with specific certification or maturity levels was an influencing factor. Organisations stood a better chance of competing if they could at least match up to their competitors' demonstrated quality processes. A direct customer request for the organisation to have a particular model implemented at a specific level was a common factor in the decision to implement an SPI model.

Bottom lines – that's what its all about. Everyone from top level management to quality consultants that were spoken to said that the desire to improve the "bottom line" was seen to be a major influencing factor in the decision to implement an SPI model. Organisations do not implement a quality model just for the sake of it, they do it because they see the benefits to their bottom line. Whether the desired bottom line is improved project scheduling, reduction in costs, increase in productivity, higher rate of return on investment, all play their part in convincing management of the need to implement an SPI initiative.

The third major factor influencing the way an organisation's processes work, was customer complaints or production problems. When either of these has been traced back to the IT division, ultimately leading to loss of business, SPI was implemented. In an increasingly demanding market place, there is a need for the IT division to be leaner, meaner and more agile in order to meet the business demands not only of its external customers, but also of its internal customers who rely upon them for product support and maintenance.

The incentive of improved process performance was not found to be the primary goal of implementing SPI strategies. One quality manager said “actual process improvement was not considered relevant during the implementation process and the fact that it was noted after implementation was considered an added bonus”!

3.2 Why choose one model over another?

Organizations must be very careful when considering the daunting number of quality models and frameworks, as the field is truly a quagmire in which process improvement efforts can be bogged down (Sheard, 1997). In deciding which SPI model to implement top level management must clearly understand what it is they want to change and why they want to change it. There must be a clear understanding of the core business needs and goals of the organisation. Casey and Richardson (2002) state that “without a clear understanding of why an improvement initiative is undertaken, it will in most circumstances be doomed to failure, given the level of commitment required by all concerned for its successful conclusion”.

The main reasons given as to why certain models were considered for selection, but not actually implemented were:

- There was no tangible benefit for implementing the model
- Practices were implemented just to satisfy the model and not for business reasons.

Managers need to be convinced of the value of process improvement. Process improvement takes time to institutionalise and requires a commitment from management in order to succeed (King, 2002). During our research it was evident that for management to be fully committed to any SPI initiative, they require clear and tangible benefits (return on investment) that a particular SPI model would bring their organisation before deciding whether or not to implement. “When push comes to shove, what really matters to management are the numbers” and the answers to questions such as: will this investment save us money and what is the projected payback period and return on investment? (Reifer, 2002).

In choosing a particular quality model over another the organisation needs to ensure that the model can fulfil most if not all of the organisation’s primary business goals. “Your objective is not simply to satisfy the model’s expectations” (Wieggers, 1999), but to ensure that your organisation’s expectations are satisfied by the model. Our evidence suggests that with this in mind, organisations must ensure that there is a clear business reason for each practice or key process area being implemented and that practices are not being implemented to satisfy a particular model.

3.3 What needs to be considered?

Once an organisation has chosen which model it will implement a number of factors need to be considered. These include:

- What change is required?
- Where and how to start?
- What is the timeframe and budget?

The size of an organisation, the current state of the process and practices as well as how the organisation has been performing over the last number of years affect how easy the implementation process would be and how much change is required within the organisation. If the organisation's existing process and practices are in poor shape it goes without saying that they would require a higher level of change than an organisation with structured and stable practices. The organisation must be prepared to ensure that a high level of change does not result in failed implementation. This necessitates having the right people in the right place with enough knowledge and expertise to carry out the SPI initiative.

Organisations must not aim too high too fast as disillusionment may set in. Some of the organisations seen during this research have chosen to start off small, implementing improvements on a set of processes or projects, noting the results and subsequently rolling out the model on an organisation wide basis. Weigers (1999) agrees with this approach, advising organisations to "select a small set of appropriate practices that will address your current process shortcomings and move toward your goals. Identify a project or two on which to pilot new processes and make adjustments before officially rolling them out". This has the effect of showing both management and staff the benefits of such improvements and keeping them fully focused and committed on continued improvement throughout the organisation. Process improvement is a continually improving cycle, not a static achievable plateau.

Our investigation within Indian companies, leads to a warning for management. They must set both a realistic timeframe for the implementation as well as a realistic budget. The desire to improve must not result in situations where process improvements are rushed to meet unrealistic deadlines. As this could lead to the people involved doing what they are doing without knowing why they are doing it. Management must appreciate that sometimes software process improvement initiatives may be costly, but in time if managed correctly they will pay for themselves. It must be determined if external certification/ assessment is required, how the organisation would get certified/ assessed and how much this would cost.

4 Conclusion

In this paper we have discussed the reasons that organisations have given for changing the way they do things to incorporate a software process improvement initiative. We have looked at what influences the selection of one model over another and we have examined the factors that organisations need to consider when implementing an SPI. Given the information that has been gathered here, it may be possible to help organisations with their model selection process.

What can European organisations learn from this? Many models do exist to choose from, but intelligent choices based on the business needs of the organisation have to be made. We have not seen these choices break down by, for example, sector, but rather the choice is driven by a market requirement i.e. a direct customer request for a particular model. Indian companies are prepared to change to meet this demand, but are we? Business issues arise here, such as the need for organisations to be flexible to adapt and cater for any customer need, the ability of customers to shop around to get the best deal, and organisation's dependence on customers. Indian companies have adapted to the new global marketplace and are reaping the rewards. Do we follow their giant leaps for quality, or continue with small steps?

5 Literature

- Bollinger, T. B. and C. McGowan (1991). "A Critical Look at Software Capability Evaluations." IEEE 8(2): 25-41.
- Casey, V. B. and I. Richardson (2002). A Practical Application of the IDEAL Model. Profes 2002, Rovaniemi, Finland.
- Daniel, F. J. (2001). "ISO 9001:2000 - direct hit?" Quality World(October 2001).
- Goldenson, D. R. and D. L. Gibson, (2003). Demonstrating the Impact and Benefits of CMMI: An Update and Preliminary Results, Software Engineering Institute.
- Hutt, M. D. and T. W. Speh (1998). Business Marketing Management: A strategic view of industrial and organizational markets, The Dryden Press.
- King, J. (2002). "How CMM Impacts Quality, Productivity, Rework, and the Bottom Line." Crosstalk: The Journal of Defence Software Engineering(Mar 2002): 9-14.
- Paulk, M. C. (1996). Effective CMM-Based Process Improvement. 6th International Conference on Software Quality, Ottawa, Canada.
- Paulk, M. C., C. V. Weber, et al. (1995). The capability maturity model: guidelines for improving the software process. Reading, Mass, Addison-Wesley Pub. Co.
- Pierce, B. (2000). "Is CMMI Ready for Prime Time?" Crosstalk: The Journal of Defence Software Engineering(Jul 2000): 21-24.
- Reifer, D. J. (2002). "Let the numbers Do the Talking." Crosstalk: The Journal of Defence Software Engineering(Mar 2002): 4-8.
- Richardson, I., (1999). Improving the Software Process in Small Indigenous Software Development Companies using a model based on Quality Function Deployment. CSIS. Limerick, University of Limerick.
- Richardson, I. and T. Varkoi (2003). Managing for change when implementing software process improvement initiatives. European Software Process Improvement Conference, EuroSPI, University of Music and Dramatic Arts, Graz, Austria.
- Sheard, S. A. (1997). "The Frameworks Quagmire, a Brief Look." Crosstalk: The Journal of Defence Software Engineering(Sep 1997).
- West, M. (2004). "Applying Systems Thinking to Process Improvement." Crosstalk: The Journal of Defence Software Engineering(Mar 2004).
- Wieggers, K. (1999). "Process Improvement that Works." Software Development(October 1999).

6 Author CVs

Brendan Keane

Department of Computer Science and Information Systems
Computer Science Building, University of Limerick, Limerick, Ireland
Brendan.Keane@ul.ie

Brendan Keane is currently a postgraduate student in the Department of Computer Science and Information Systems at the University of Limerick, where he is undertaking a masters degree by research in the area of Software Process Improvement, with a specific focus on SPI models and how they are selected. As part of his research, he has spent time in India investigating decisions by companies on their selection and use of software quality models. Prior to undertaking his postgraduate studies, Brendan was a graduate engineer with Logica Mobile Networks.

Dr. Ita Richardson

Department of Computer Science and Information Systems
Computer Science Building, University of Limerick, Limerick, Ireland
Ita.Richardson@ul.ie

Dr. Ita Richardson is a senior lecturer in the Department of Computer Science and Information Systems at the University of Limerick where she lectures to undergraduate and postgraduate students. Her main research interests are in Software Process Improvement with a specific focus on small to medium sized enterprises. This involves qualitative research with companies, and some of her postgraduate students are in full-time employment with software development companies. She is involved in two SFI funded projects - as a researcher on the B4-STEP Principal Investigator project and project leader of the Global Software Development for SMEs project within the Irish Software Engineering Research Consortium.

NOEMI , a collaborative management for ICT process improvement in SME: experience report

Bernard DI RENZO, Christophe FELTUS, Sylvie PRIME

Abstract

This paper relates to an R&D project – called NOEMI¹ – aiming to propose to SME's a collaborative management of their respective information systems in terms of quality, reliability and cost. The targeted SME are those without IT dedicated internal staff.

The model developed in the project focuses on the usual IT activities of the SME's, which are classified in five domains: infrastructure, support, management, security, and documentation. The NOEMI model has been successfully experimented with PME partners for validation purpose.

This paper insists on the experimentation of the NOEMI model. The results are discussed and compared with other classical solutions usually encountered within IT practices of SME's. The last part of the paper draws the perspectives opened with the model according to the results of its experimentation.

Keywords

IT Service Management, collaborative management, SME, process assessment, process improvement.

¹ Nouvelle Organisation de la Maintenance et de l'Exploitation Informatiques

1 Introduction

It sounds obvious that SME's competitiveness is more and more linked to the quality and the reliability of their information systems. Nevertheless hiring an IT dedicated staff is often discouraged due to the small size of the IT infrastructure of the SME's. The NOEMI project addresses this critical issue for the SME's and has developed a relevant solution to bridge the gap between the mandatory need of quality in IT infrastructure of the SME's and the dramatic lack of internal skills. The solution is based on the creation of clusters of SME's aiming to increase IT infrastructure volume. The global IT activities within the clusters is then manageable in a collaborative way, which can lead to higher quality and reliability as larger IT departments would provide.

This paper reviews briefly the NOEMI collaborative model developed by the Centre Henri Tudor, its validation by a case study, its positioning regarding alternative solutions and at last, as conclusion and perspective, the transfer of the research results to the market.

2 The NOEMI model

The NOEMI model relies on partners, gathered in clusters. A common dedicated operational team performs their IT activities and is managed on a regular basis by an IT Coordination Committee (the CCI).

Figure 1 illustrates the organisational frame of the model.

This chapter explains the references used to create the model, the openness of the cluster and its management.

2.1 The baseline and the references

The NOEMI model is built on 5 areas:

1. Management,
2. Service Support,
3. Infrastructure,
4. Security,
5. Documentation.

These five areas include 20 processes tailored for use a SME environment [1]. For each area, a set of activities are defined, performed by the operational team and controlled by the CCI.

Development has been partially inspired by ISO/IEC TR 15504 [2] and ITIL [3, 4, 5 and 6].

2.2 Constitution of SME's clusters

The NOEMI model allows the increase or decrease of the number of SME partners in a cluster. Before a new SME joins the cluster, a capability profile of each area is defined according to a specific as-

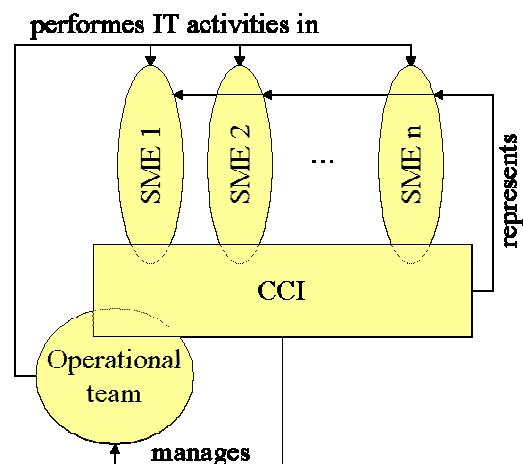


Figure 1: Organisational frame of the NOEMI model

assessment method developed in the NOEMI project [7].

The NOEMI assessment method is directly inspired by ISO/IEC TR 15504 [2].

This assessment analyses in depth the weaknesses and the strengths of the 20 processes defined in the model and defines a capability profile of the 5 domains. A list of pragmatic improvement actions to take according to urgency and impact is proposed.

The results of the assessment are used as a reference for the IT operational team for the first steps when the new SME is entering the cluster.

2.3 Evolution of the IT Infrastructure

Following the recommendations, a lot of changes and manpower are required to implement the most urgent and important activities. This phase, called “Up To Date Phase” will take from 6 to 9 months.

The stabilization phase will then start.

It is to notice that, in this phase, the number of incidents will decrease in opposition with the number of requests, which will get higher.

Figure 2 illustrates the main phases in the model implementation.

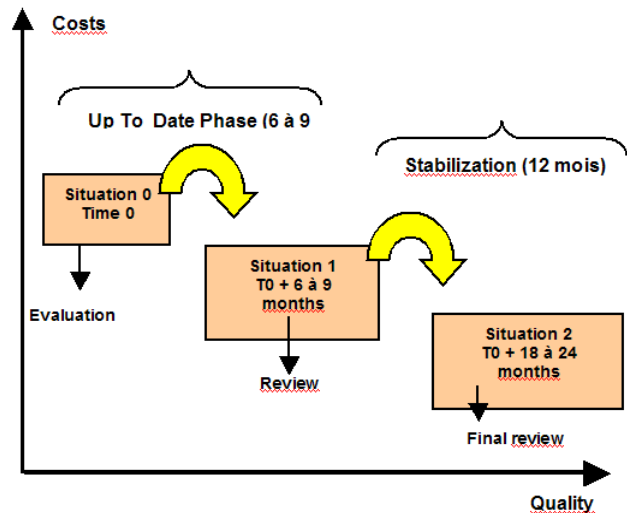


Figure 2: Implementation phases of the NOEMI model

2.4 Management of the SME Clusters

After assessment and depending on the improvement actions agreed with the SME, the operational team will handle all activities and the 20 defined processes. A particular focus is put on incident management, change management and configuration management assumed as the most critical IT processes in SME's [8].

The operational team provides manpower for the activities needed to improve the capability level of each area. The size of the team is linked to the global IT infrastructure of the cluster and includes both technical staff and project leader. Team members are qualified (IT or project management), can act autonomously, and have strong relationship skills.

The operational team is managed the CCI where a representative manager from each SME stands. The main objectives of this committee is to ensure:

- the coherence between the cluster and the own objectives and strategy of each SME,
- the convergence of IT improvements,
- the evaluation of preventive and corrective actions.

The committee bases its management action on a balanced set of indicators within financial, partner's satisfaction, activity efficiency, and people skills [9].

Moreover, the capability profile of each SME is followed up on the basis of the assessment and reviewed on a regular basis (6-12 months).

3 Case study and validation

The model NOEMI, as shortly described above, has been validated through a case study. This chapter focuses on this experimental validation according to three perspectives: economic health of the experimental cluster, its management and the satisfaction of the SME's of the cluster.

3.1 Economic health of the cluster

The financial management of the IT activities within the cluster is based on the balance between the costs and the charges. The total cost is the sum of the IT staff charges, travel costs of the IT staff, training, equipments (laptop, mobiles, etc.), and miscellaneous charges (phone call, Internet access, etc.)

In the case study, no benefit is down because the cluster is the ownership of the SME.

This total cost is transparently financed by each SME of the cluster and is apportioned to their IT infrastructure, according to an empirical calculation. Each server has a 10 points value, each PCs has a 1 point value on which 20 points are added to enter the project. The point represents a certain financial value. With this simple and clear calculation, the SME's know exactly how much their IT service will cost on a monthly basis and on an annual basis. Here under, the main advantages related to the economic health of the cluster:

- budgeting becomes very easy,
- hardware and software investment gets also easier as the inventory helps efficiently in making the decision,
- the workload related to IT activities has been reduced for the staff of the SME's, they can focus more efficiently on their business,
- business productivity of the SME's increases according to the better resolution of end-user incidents,
- some of the SME's of the cluster invest more in the IT activities than before.

3.2 Management of the cluster

All SME partners meet together – through the CCI – with the project team, on a monthly basis, to discuss the results of the past month, projects in the coming weeks and months. This monthly meeting is a core activity in the project. This tight relation, the confidentiality of the activities and the complete visibility on the cost (who is paying what) is a very strong point in the success of the project.

A tool is used to follow the incidents during their whole life cycle. Changes are tracked as well as assets; reports on the activities are available on-line. The planning is also published so any one knows where operates the IT team. This openness positively drives the trust among the partners.

On the same area, for all the software's, email system and configurations, it is tended to use common procedures in order not to reinvent the wheel with new products, therefore, to avoid new incidents and new problems or bugs to handle.

On another hand, after sharing IT skills, sharing IT equipments has been mentioned. The issue is that SME are not ready to have data out of their offices due to a fear of stolen or misused data. Evermore the price of the leased lines is still very expensive and the SME's of the cluster can't afford such an extra cost.

3.3 Satisfaction evaluation

A satisfaction survey is performed monthly (during the CCI) and a close relation is kept with the

SME's.

Monthly, 40 questions about the quality of the intervention, the speed, the feedback, the overall satisfaction from the users, the feeling from the IT contact in the company, the effectiveness and efficiency in interventions are asked to the partners.

Activities are quoted "Not", "Partially", "Largely" and "Fully" performed. The results from the survey on the 9 past months give an average for each 14 questions between "Largely" and "Fully". This overall satisfaction is very good indicator for driving the experimentation. This indicator has a direct impact on the number of SME's in the cluster.

4 NOEMI positioning among other solutions

The positioning of the NOEMI model regarding different way of IT exploitation on the market is made by a SWOR analysis.

This analysis is based on experience returns within 68 SME's contacted through different projects managed by the Centre Henri Tudor.

Solution	Strengths	Weaknesses	Opportunities	Risks
Sourcing of the IT activities of a SME's cluster to a common IT team, with a collaborative management (NOEMI model)	Cost directly bound with the IT staff charges and under control. Priorities are managed regarding business needs. Focus on results. No direct dependency of IT service providers.	Issue on being always up to date with the new technologies. Tooling is mandatory and is an extra cost.	Other enterprises resources can be shared regarding new business activities (manpower, tools, equipment sharing)	Overload: time spent by the IT staff can't exceed the time allowed for each partner Overcost: time effectively needed for IT activities is under the available manpower
Sharing of a common IT staff in an organisational structure owned by a group of SME (time sharing) (Synergie project²)	Cost directly bound with the IT staff charges and under control. No direct dependency of IT service providers.	Issue on being always up to date with the new technologies. Medium quality of the service. Tooling is mandatory and can get very expensive (maintenance fees, evolution, hardware..). Focus on means, not on results.	Other enterprises resources can be shared regarding new business activities (manpower, tools, equipment sharing)	Overload: time spent by the IT staff can't exceed the time allowed for each partner Overcost: time effectively needed for IT activities is under the available manpower.
Buying a pool of hours from an external IT service supplier	Cost limited and controlled	Strong dependencies with the IT company. SME's generally wait the last minute to call and it is often too late or the con-		IT company not paid regarding the quality of its intervention but accordingly to the time spend to solve the problem.

² "Synergie" is a project of the CRP Henri Tudor which has lead to a time-sharing of an IT person within three companies in order to improve efficiency, cost control of their IT infrastructure and management.

		<p>sequences are worse!</p> <p>No focus on result, only on means.</p>		
Contracting with an external IT service supplier	<p>Cost limited for pre-defined activities.</p>	<p>Strong dependencies with the IT company.</p> <p>SLA are mandatory in order to follow the efficiency – Penalties should be added if the service is a core business one.</p>		<p>Irreversibility of the dependency of the supplier.</p> <p>Overcost: time effectively needed for IT activities is under the available manpower.</p> <p>Lost of IT control</p>
Hiring a part-time IT dedicated person	<p>Cost directly bound with the IT staff charges and under control.</p> <p>No direct dependency of IT service providers.</p>	<p>Issue on being always up to date with the new technologies.</p> <p>Lack of employee stability.</p> <p>Quality bound with the IT person competences.</p> <p>No focus on result, only on means.</p>		<p>Conflicts between the different employers of the IT person.</p> <p>Important problem occurs while staff isn't present in the company.</p> <p>One-person dependency.</p>
Hiring a full-time IT dedicated person	<p>Cost directly bound with the IT staff charges and under control.</p> <p>No direct dependency of IT service providers.</p>	<p>Issue on being always up to date with the new technologies.</p> <p>The staff is not enough exploited.</p> <p>Most of the SME's can't afford such a full time IT people.</p> <p>Overcost: time effectively needed for IT activities is under the available manpower.</p>	<p>Possibility to develop new business activities based on IT innovation.</p>	<p>Job not enough attracting for It staff.</p> <p>One-person dependency.</p> <p>Overcost: time effectively needed for IT activities is under the available manpower</p>
Hiring an IT dedicated person for IT management and for an other activity	<p>Cost directly bound with the IT staff charges and under control.</p> <p>No IT company involved, no dependency</p>	<p>Issue on being always up to date with the new technologies.</p> <p>Quality depending on the time assigned to the different activities</p> <p>No focus on result, only on means.</p>		<p>Depending of the staff, doing two different jobs can be interesting or difficult to manage.</p> <p>One-person dependency.</p>

The more significant differences between the NOEMI model and the other solutions rely on:

- the financial aspects,
- the openness of the activity performance,
- the focus on results.

The sourcing approach and the defined financial rules of the NOEMI model links the objectives and the motivation of the clients (the SME's) and the supplier (the operational team), whereas other solutions propose a time-based charging.

5 Status on the development and perspectives

The NOEMI project leads nowadays to an 8 partners cluster.

The project under the cover of the Centre Henri Tudor reaches its end and is ready to be transferred to the market. The model has been estimated as a success by the SME's participating in the project; they have decided to go on with the initiative. So, it is time for the SME cluster to choose their way for the future of their IT-sourcing service.

Many solutions are possible amongst which the transfer to an IT service company, the creation of a dedicated common company, the creation of a spin-off, employment of the IT staff with a back charging.

A call for tender aiming to take over the "NOEMI cluster #1" operational activities is being written and not less than 12 large or medium IT companies have shown their interest to be involved in the mailing for the call for tender.

The success of the model will be directly related to its appropriation by the market. Some indicators to measure it could be:

- the number of partners still working together in the horizon 2006,
- the number of new clusters created through IT companies or through a GIE (Economic grouping from the companies),
- the number of IT companies interested in the NOEMI label we work on (model requirements to respect...) and the annual renewing of the label,
- the satisfaction survey performed regularly in the clusters,
- the number of amendments to the model.

6 Literature

- [1] Di Renzo B., Feltus C., Development and experimentation of a model of IT collaborative management for use in SME, in: Proceedings of the International Conference AIM 2003, Grenoble, France, 2003
- [2] ISO/IEC JTC 1/SC7, ISO/IEC TR 15504, 1998
- [3] IT Infrastructure Library – Service Support, Office of Government Commerce, United Kingdom, 2000 – 2002
- [4] IT Infrastructure Library – Service Delivery, Office of Government Commerce, United Kingdom, 2000 – 2002
- [5] IT Infrastructure Library – ICT Infrastructure Management, Office of Government Commerce, United Kingdom, 2000 – 2002
- [6] IT Infrastructure Library – Security management, Office of Government Commerce, United Kingdom, 2002
- [7] Di Renzo B., Feltus C., Process assessment for use in very small enterprise: the NOEMI assessment methodology, in: Proceedings of the International Conference Eurospi 2003, Graz, Austria, 2003.
- [8] IT Infrastructure Library – Practices in small IT unit, Central Computer and Telecommunication Agency, 1995.
- [9] Robert S. Kaplan and David P. Norton, The Balanced Scorecard – Translating Strategy into Action, 1996.

7 Author CVs

Bernard DI RENZO

Bernard Di Renzo graduated as a "Civil Engineer" in electronics and computer science from the University of Liege (Belgium) in 1989. He first worked as a project engineer in the engineering company of an international industry group. There he managed several projects re-

lated to the IT discipline. From 1993 to 1999, he was the head of the IT-affiliate of a banking group in Luxembourg. In this function he developed an accurate focus on quality service provision. Then he joined an international clearinghouse for wireless operators as the Head of Worldwide Client Support. Since 2001, he has worked in the CITI (Centre d'Innovation par les Technologies de l'Information) department of the Centre de Recherche Public Henri Tudor (Luxembourg). He has managed several projects related to quality in the IT discipline with a special focus on process assessment and improvement, security management, risk management, IT clustering... He is a SPICE Qualified Assessor, ITIL Foundation Certified, and Quality System Auditor certified. He is the coordinator of two project portfolios related to IT quality and certification and to IT security. He also the head of a research unit related to process assessment and improvement, service management, operational risk management, IT-business alignment, and new organisation framing.

Christophe FELTUS

Christophe Feltus is graduated as an Electromechanics Engineer from the "Institut Supérieur Industriel des Art et Métiers Pierrard in Virton, Belgium. He worked for several years in private companies as: Production Head at Pfizer SA in Jette, Belgium, Project Coordinator at Nizet Entreprise in Louvain-la-Neuve, Belgium, and Assessor for the Civil Belgium Aviation Administration in Brussels, Belgium. He joined the Centre de Recherche Public Henri Tudor in the Grand-Duchy of Luxembourg in 1999 to work in the Centre d'Innovation par les Technologies de l'Information (CITI). There he has taken part in a project called Prisme Tremplin for the accompaniment of SMEs towards the information society. In 2002, he integrated the project Noemi (Nouvelles Organisation de l'exploitation et de la Maintenance Informatiques) as Assessor and he now leads projects IT Cluster and SecurePME.

Sylvie PRIME

Sylvie Prime is graduated as an IT Engineer from The university of Nancy II, France. After a year of development for a pharmaceutical group, Sylvie began to work in Luxembourg in 1995 at the EIB, then at RTL Group in charge of the IT production, sites abroad included. She began to work on organization of IT Teams when she joined Arthur Andersen in 1999 and, even deeper when working at Quint Wellington Redwood. She then passed the ITIL foundation exam, the Service Manager one. She has been a Service Management consultant for 2 years and joined the CRP Henri Tudor in 2003 to develop the Service Management activities (ITIL, training, IT Service Management Forum). Sylvie is also SPICE Assessor.

Practical Experience of Virtual Team Software Development

Valentine Casey & Ita Richardson

Abstract

Given the current popularity of Global Software Development (GSD) increasing numbers of organisations are taking the opportunity offered to set up virtual software teams. This approach allows companies to partner experienced engineers located in high cost economic areas with less experienced team members based in low cost centres allowing both locations to be leveraged to their best advantage. A number of different approaches are currently being undertaken. Some organisations have set up subsidiaries in low cost economies and are developing virtual teams in this environment. Others are partnering with third party outsourcing organisations.

The focus of this research is an Irish based company partnered with an organisation located in the US to develop and maintain bespoke financial software. A number of virtual teams were established, each team had members based in the US and Ireland. Given the locations involved it was initially believed that the organisations were culturally near shore. While that proved correct to a point, it soon became clear that communication, motivation and cultural differences existed and needed to be addressed. It was also realised that a new approach to process development and improvement had to be undertaken to ensure the successful development and operation of these virtual teams. Both organisations had good single site development and maintenance processes, but they proved inadequate for a multi-site GSD environment and needed to be extensively reengineered.

Initially mistakes were made and areas of conflict arose. Rather than view these events as negative, each was leveraged to insure that the process was improved to address and minimise any reoccurrence. As a result over a four-year period a very successful common virtual team development process evolved between both organisations. This paper outlines some of the key problem areas encountered and offers practical advice and experience on how they were successfully addressed. Given the importance of GSD to the software industry this experience should be of value to anyone interested in outsourcing and virtual team software development and maintenance.

Keywords

Global Software Development, GSD, Virtual Teams, Process Improvement, Process Re-engineering, Communication, Motivation, Culture, Outsourcing

1 Introduction

Organisations are gaining competitive advantage from globalization due to labour arbitrage, which allows reduced costs; this facilitates competitive pricing which helps organisations to increase market share. The move to globalize software development has also been facilitated by the availability, in large numbers, of well educated and technically competent software engineers in low cost centres [1]. Given the economic benefits offered, the current trend to globalize software development continues to expand [2]. This paper outlines some of the experience gained while establishing and operating virtual software development and maintenance teams by an Irish based company, Software Future Technologies partnered with a large US financial organisation Stock Exchange Trading Inc. (both pseudonyms). Stock Exchange Trading Inc. had an on going requirement for the development and maintenance of bespoke financial software. Initially all development and maintenance activities were carried out in-house. As a result of continued expansion Stock Exchange Trading Inc. had an increasing de-

mand for software development and maintenance, but their in-house IT strategy was becoming too expensive. The solution was to find an efficient alternative, which would leverage the experience of their existing IT department while maintaining the level of quality and support required at a cost effective price.

Initially a number of near shore outsourcing options were considered, but were rejected due to cost. Stock Exchange Trading Inc. had previously successfully outsourced their Y2K legacy code renovation to an Irish based company Software Future Technologies. The possibility of expanding this relationship was identified and explored. After extensive negotiations a four-year contract was agreed. The terms outlined that Stock Exchange Trading Inc. would partner with the Irish based Software Future Technologies and establish virtual teams to undertake the development and maintenance of all its software applications.

The rationale for adopting this approach included:

- Proven track record of co-operation between the organisations
- Availability in Ireland of highly educated and technically competent staff
- Low salary levels in Ireland compared to the US.

Furthermore development and maintenance time could be decreased, more extensive out of hours support provided and the use of hardware maximised by leveraging time zone differences.

The Irish based Software Future Technologies, while being owned by a US multinational operated as an independent profit centre and was wholly Irish managed. In GSD terms Ireland, while being geographically off shore, was considered near shore, because of language and cultural similarities to the US [3]. Given the continued success of the Y2K collaboration and the near shore status attributed to Ireland, the transition effort required for the establishment of virtual team software development and maintenance was under-estimated. In fact the management of both organisations considered it a straightforward task.

At the time the activities outlined in this paper were taking place neither organisation had knowledge of any research or published literature in the GSD field. The only experience Stock Exchange Trading Inc. had in this area was outsourcing their Y2K work. Software Future Technologies experience of being outsourced to was confined to the successful renovation of ten US based clients Y2K code. This experience, while relevant was limited and it proved inadequate for the development and effective operation of virtual teams. As a result both organisations had to embark on a steep learning curve, which required the re-evaluation and modification of how they both operated.

2 Establishing Virtual Teams

Initially a common sense approach was employed by both organisations based on their limited experience of outsourcing. A team structure was agreed and four separate cross-site teams were established. Each team consisted of twenty cross-site members; tasks were shared among team members regardless of location. The teams normally operated as separate units and each took responsibility for different development and maintenance projects. It was decided that a US based team leader and project manager would manage each team.

The need for an effective configuration management system was realised [4]. As a result one of the first activities undertaken was the identification and selection of an effective configuration management tool and a documented operating procedure was developed for its implementation.

Forty staff were selected from the existing IT personnel at each location based on their technical ability and levels of experience. Stock Exchange Trading Inc. had a well-defined and documented process and it was agreed that this would provide the basis for the virtual teams operation. Initially very little modification was made to the process to facilitate a virtual working environment. An extranet [5] was established to facilitate remote access to process documentation. Conference calls and e-mail were

selected as the main methods of communication. It was also agreed that direct telephone calls would take place with and between team members when required.

The Irish team members were accustomed to operating with a well-defined process and an initial task was to familiarize themselves with the US organisation's process and documentation. To this end a basic orientation course was developed and undertaken by the Irish team members. The focus of this course was process centric and ignored such issues as cross-site cultural differences and possible communication problems. The US based team members were not offered any virtual team orientation or training.

Once basic orientation had taken place there was an unexpected demand, which required seventy percent of the Irish team members to spend six to twelve months working onsite in the US organisation. This was an unplanned emergency strategy and arose due to the need for Stock Exchange Trading Inc. to develop complicated bespoke software within a short timeframe following the winning of a large contract. When this arose the virtual teams were just being set up and the infrastructure for their operation was not yet in place. Moving as many Irish team members as possible to work on site with their US based team members on a temporary basis was the only solution in these circumstances.

This proved to be a good opportunity in a number of ways. It enabled seventy percent of the Irish team members first hand experience of following Stock Exchange Trading Inc. development and maintenance process. More importantly it allowed them to meet and develop working relationships with their US based Project Manager, Team Leader and fellow team members. This was initially successful and it provided a good base on which to build [4]. However it did not prevent the breakdown of those relationships when the full impact of GSD related factors were encountered.

Research has shown that it is difficult to integrate separate groups into one coherent team when they are remotely located [6]. The need for the development of trust in a GSD environment and particularly with virtual teams is paramount [7]. This is best achieved by face-to-face contact and interaction between as many team members as possible. The goal is the development of what has been described as "teamness" [4], which is defined as the ability of people to work together as a team. Effective team relationships are based on the development of trust and respect [8], which are very difficult to establish and maintain when team members and management are located on opposite sides of the Atlantic.

Given the near shore cultural status ascribed to Ireland [3] it was interesting to see just how differently both groups worked in a one to one development environment. There were clear cultural differences between the US and Irish team members. This was reflected in their behaviour, the way work was carried out and their attitude to authority and process. These differences became apparent to both groups and did not result in any problems being encountered while they were co-located. The experience was beneficial to a point as it allowed them develop a limited understanding and appreciation of each others culture and first hand experience of how the other group worked [9]. While both groups worked very effectively when they were co-located the experience postponed rather than prevented future problems.

3 Virtual Teams in Operation Lessons Learned

Once the urgent projects were complete the Irish team members returned home. Meanwhile the infrastructure had been put in place and the team members who remained in Ireland had successfully provided support to their Irish and US colleagues. Now the full virtual teams were established and work commenced. Initially everything seemed to be going well, but soon problems started to arise.

Research in the area has identified distance as being a major factor impacting GSD [9]. Distance introduces complexity which arises due to its impact on communication and co-ordination [10]. These factors are further compounded by culture [11], which all negatively impact on co-operation, motivation and trust [12]. Our experience would concur with these findings and within a short period of time their impact and effects were obvious. After what could be describe as a "honeymoon period" where people in both locations endeavoured to work as single teams, working relationship started to break down. This directly impacted productivity and resulted in increased project time and costs. This threa-

threatened the partnership between Software Future Technologies and Stock Exchange Trading Inc. and urgent action had to be taken.

Following an extensive investigation it became clear people who worked together very successfully while co-located were now actively obstructing and blaming each other for all the problems that arose during projects. It was obvious that team members were now aligned by geographical location and there was a very clear “we verses they” culture [8]. This was a totally unexpected outcome given the level of harmony achieved in the earlier co-site projects. Questions had to be asked - how had this happened? how could it be addressed? how could it be stopped from reoccurring?

3.1 Communication Issues

The misuse of e-mail was identified as a major contributing factor to the conflict, alienation, mistrust and lack of co-operation between locations. While e-mail was used to communicate, it was also being used as a weapon to publicly attack fellow team members. The practice of copying senior management on minor problems, which were caused by team members at the other location, was widespread. Both groups were equally guilty of employing this tactic. This activity had the desired effect of highlighting the problem, but it alienated the individual it was directed toward. It also had a negative impact on fellow team members at that location, who saw it as an attack on the group as a whole. This situation was further compounded by management reacting to these e-mails and getting involved with minor issues that the relevant team leader should have addressed. It was noted that management in their response normally took the side of individuals where they were located. This further alienated and added to the mistrust felt by people at the other site. It was clear that the “we verses they” culture [8] was not restricted to team members and was prevalent between some levels of management as well.

This issue had to be addressed and a documented e-mail procedure was the solution. Clear guidelines were agreed stating when, how and to whom problems should be highlighted. This procedure was clearly outlined to team members at both locations. All minor issues were to be raised directly with the team leader and only with those directly involved. If and when it was necessary it was the responsibility of the team leader to inform the relevant project manager of issues that could not be addressed within the team. If the project manager was unable to resolve the matter it was their responsibility to raise the issue with senior management at both locations who would formulate a joint response. Interestingly, following the change in procedure very few minor issues were raised with the project managers and none, had to go to senior management for resolution.

Having addressed the immediate manifestation of the problem it was important to determine why it had arisen. It was realised that a number of factors were involved. Distance and the lack of the opportunity for informal communication played a part. A five-hour time zone difference between sites meant the opportunity for direct contact was limited to three hours a day. No informal method of communication was available. At the time instant messaging was not seen as a business tool.

Further communication problems were also identified between sites. Both locations spoke different dialects of English. The English spoken in Ireland is based on the British dialect, which is in a number of areas different to that used in North America. This is reflected in the difference of spelling and in the use and meaning of words. One of the more serious problems encountered was when code written by a US based team member was peer reviewed in Ireland. The Irish reviewer commented in response that everything was fine and all the “full stops” were in the correct positions. On receipt of this review an irate response was received which stated “my code does not full stop!!” It took numerous e-mails and a telephone call to explain that what in Irish/British English is called a “full stop” in the North American dialect is referred to as a “period”. This highlighted the need for training for both groups on their use of English. As a result it was also stressed that thought should be given to all methods of communication to insure clarity and limit ambiguity. Local accent was also identified as a problem in telephone and conference calls. The need for people to speak slowly and clearly was stressed. If someone was unsure of what had been said they were encouraged to ask to have the statement repeated. This approach proved very successful.

3.2 Cultural Issues

Surprisingly, given their previous experience of working together cultural differences came into play. While the Irish and US culture appeared similar [3], distance highlighted their dissimilarity. The Irish attitude to authority and respect required that it must be earned rather than imposed. This manifested itself in the Irish tendency toward frankness, to question procedures, use of humour and work ethic. These attitudes were construed by the US staff as confrontational. The US based team members belief in their technical superiority, their view that the Irish team members were working for them, not with them and their sole ownership of the process were seen as naïve and arrogant by the Irish staff. These conflicting perceptions added to the mistrust and alienation felt by both groups and needed to be addressed.

3.3 Process Reengineering

The process while effective for single site development and maintenance proved inadequate for a virtual team environment. It was seen, as being imposed and the sole property of the staff of Stock Exchange Trading Inc. Team members based in Software Future Technologies while having relevant suggestions for process improvement were not consulted and any suggestions they made were ignored. This added to the alienation being experienced by the Irish staff. To address the process issues, the need for establishing common goals, objectives and rewards were identified. The process had to be totally reengineered to incorporate these issues. A common vocabulary with clear definitions of artefacts, deliverables and milestones was jointly formulated. These were incorporated into the development of a shared and agreed process, which specifically addressed the needs of the virtual environment in which it operated. The input of staff at both locations was encouraged and valued.

3.4 Major Contributing Factor Identified

These measures helped to facilitate team co-operation, built trust and relationships. While these addressed some of the problems experienced they did not fully explain the underlying cause. How had teams that worked effectively when they were co-located, deteriorated into opposing groups? This question needed to be answered to ensure that similar problems did not arise again. After extensive interviews motivation was identified as a major contributing factor. While the majority of the teams were co-located in the US the American team members did not comprehend the full implications of the virtual team strategy. Once the virtual teams were established the full impact on their day-to-day work, promotion and future employment prospects became clear. Management reinforced these negative aspects by utilising the strategy to justify maintaining salaries at their current levels. They also stressed the additional cost of US based staff and the need for them to be value adding to justify the extra expense. The effect was unmotivated people who directed their hostility toward their fellow team members in Ireland who they saw as a threat to their careers. This manifested itself in a lack of co-operation, alienation and on occasions outright obstruction when the opportunity arose. This was met with a similar negative reaction from the Irish side, who felt that if this was how the Americans worked it would be more effective to move the whole operation to Ireland. These attitudes were identified as a major contributing factor to the problems experienced and were compounded by the other issues outlined in this paper.

Once it was realised, the only effective strategy available was to make it clear to staff at both locations, if utilising virtual teams was not an option the only feasible economic alternative was outsourcing the whole operation to the Far East. The options had to be made clear, work as a team regardless of location, or find new positions elsewhere. This was a drastic approach, but the seriousness of the situation warranted it and there was no alternative if the virtual team strategy was to continue. This, along with the other measures outlined, helped to establish a productive working relationship between sites and facilitated the completion of projects on time and within budget for the lifetime of the contract.

4 Further Research

On completion, the four-year contract between Stock Exchange Trading Inc. and Software Future Technologies was not renewed. It was agreed by both parties that the virtual teams had operated successfully, but in the renegotiations cost proved to be the deciding factor. The Irish Punt to Dollar rate had substantially increased making Ireland a less attractive location. The availability of more cost effective operations in the Far East meant that Software Future Technologies could not compete on price and the contract was lost. It is interesting to note that Stock Exchange Trading Inc. has continued to successfully utilise the virtual team strategy with its new outsourcing partners. Software Future Technologies was unable to attract similar type business due to cost and after a period closed.

Following the closure of Software Future Technologies one of the authors undertook full time PhD research in the area of GSD at the University of Limerick. During the literature review undertaken as part of this effort it was discovered that the experience gained in Software Future Technologies is mirrored in the current published literature in the GSD field. This experience has been analysed and it has formed the basis for further research [13]. Both authors have focused on investigating the establishment and operation of virtual teams in the Far East, Europe and the US. The authors agree with the view that distance introduces complexity as a result of the impact it has on communication and co-ordination. They also have identified a number of additional and related variables that need to be addressed to facilitate successful virtual team establishment and operation. This has led to the development of a framework, which outlines these key variables and a proposed methodology to effectively address and leverage these variables to facilitate successful virtual team development and operation. The research associated with this work is still currently underway.

5 Literature

1. Toaff Shermaria S. Toaff, *Don't Play with "Mouths of Fire" and Other Lessons of Global Software Development*. Cutter IT Journal, 2002. **15**(11): p. 23 - 28.
2. Laplante, P.A., et al., *The Who, What, Why, Where and When of IT Outsourcing*. IT Professional, 2004. **6**(1): p. 19 - 23.
3. Hayes Ian S., *Ready or Not: Global Sourcing is in Your IT Future*. Cutter IT Journal, 2002. **15**(11): p. 5 - 11.
4. Battin, R.D., et al., *Leveraging Resources in Global Software Development*. IEEE Software, 2001. **1**(2): p. 70 - 77.
5. O'Brien James A., *Management Information Systems Managing Information Technology in the Business Enterprise*. 6 ed. 2002: Mc Graw Hill Irwin.
6. Carmel, E., *Global Software Teams: Collaboration Across Borders and Time Zones*. 1999, Saddle River, NJ: Prentice Hall.
7. Handy, C., *Trust and the Virtual Organization*. Harvard Business Review, 1995. **73**(3): p. 40 -40.
8. Oppenheimer, H.L. *Project Management Issues in Globally Distributed Development in ICSE Workshop on Global Software Development*, 2002.
9. Carmel, E. and R. Agarwal, *Tactical Approaches for Alleviating Distance in Global Software Development*. IEEE Software, 2001. **1**(2): p. 22 - 29.
10. Herbsleb, J.D. and D. Moitra, *Global Software Development*. IEEE Software, 2001. **1**(2): p. 16 - 20.
11. Ebert, C. and P. De Neve, *Surviving Global Software Development*. IEEE Software, 2001. **1**(2): p. 62 - 69.
12. Pyysiäinen, J. *Building Trust in Global Inter-Organisational Software Development Projects: Problems and Practices*. in *International Workshop on Global Software Development*. 2003. Portland, Oregon, USA.
13. Casey, V. *Global Software Development Testing Infrastructure*. in *European Software Engineering Conference*. 2003. Helsinki, Finland.

6 Authors CV

Valentine Casey

Department of Computer Science and Information Systems
Computer Science Building, University of Limerick, Limerick, Ireland
Val.Casey@ul.ie

Valentine Casey has over 20 years experience in the IT industry, four of which were spent as Software Quality manager in a virtual team software development and maintenance organisation. Previous positions held include Software Engineer, Team Leader, Project Manager, and Senior Software Test Consultant. He is a SEI trained CMM assessor and holds a MSc. in Software Re-Engineering and a BSc. in Economics and Organisational Theory. He is currently a researcher on B4-STEP, Science Foundation Ireland (SFI) Investigator Program which is examining processes, methods and tools in software development. He is completing his PhD at the University of Limerick where he has also lectured. His PhD research is focused in the area of global software development and the establishment and operation of virtual teams.

Ita Richardson, Ph.D.

Department of Computer Science and Information Systems
Computer Science Building, University of Limerick, Limerick, Ireland
Ita.Richardson@ul.ie

Dr. Ita Richardson is a senior lecturer in the Department of Computer Science and Information Systems at the University of Limerick where she lectures to undergraduate and postgraduate students. Her main research interests are in Software Process Improvement with a specific focus on small to medium sized enterprises. This involves qualitative research with companies, and some of her postgraduate students are in full-time employment with software development companies. She is involved in two SFI funded projects - as a researcher on the B4-STEP Principal Investigator project and project leader of the Global Software Development for SMEs project within the Irish Software Engineering Research Consortium.

Process improvement through development of an extended electronic process guide

From electronic process guide to integrated work tool

*Ken Rune Nilsen, Kongsberg Spacetec
kenrn@spacetec.no*

Abstract

Process guides are meant to give a more intuitive and easy to understand description of the processes taking place in a company, than traditional procedures and standards descriptions. Combined with the acceptance of process oriented quality systems it has become increasingly popular to add an electronic front-end to the process guide, turning the process guide into an electronic process guide.

As a part of developing a process guide tool, we wanted to rethink all our vital processes by transferring our existing handbooks, standards, and procedures into a new process oriented quality assurance system. A driving philosophy has been to anchor the processes to the organisation by involving the target users of the processes.

After introducing the tool into operational use, a strategy of rapid response to user feedback was implemented, expanding the functionality of the tool as fast as possible. The rationale behind this was to keep up the positive attitude amongst the users by allowing them to have real influence on the tool.

The presentation will describe our experiences and how we came to realise the need for further expansion of the process guide. A central part of this is the question of keeping up the interest for the guide.

Our "project web" is now well on its way to becoming an integrated workbench for not only project managers but for the whole project team. It is the central project planning tool and it has become the central tool for experience collection, status reporting as well as generation of statistics.

Keywords

Process Guide, Process Improvement, Project Web, project management, project planning, project status, organisational implementation, development, tools.

1 Background

1.1 The company

Kongsberg Spacetek AS of Norway is one of the leading producers of receiving stations for data from meteorological and Earth observation satellites (optical and SAR). Since Kongsberg Spacetek was founded in 1984 its products, including the MEOS™ (Multi-mission Earth Observation System), has been delivered to a number of clients around the world. Our export share is currently 85%.

Our speciality is operational systems for remote sensing, and associated services. Kongsberg Spacetek has expertise in electronics-, software development and applications. Currently Kongsberg Spacetek has 62 employees, 80 % of the staff have master degree in physics or computer science.

1.2 From engineering to product development

From the start in 1984 the main tasks of the company was engineering through customer specific projects. More specifically the main customer was ESA. The methodology and the processes was instructed and supervised by the customer. Kongsberg Spacetek adapted and followed the ESA PSS-05 engineering standards for all projects even when the customer did not have any requirements.

During the 1990s the market situation changed for Kongsberg Spacetek. A new kind of customers became increasingly important. The new kind of customer was not interested in how the product was developed or how we performed quality assurance. Instead of providing detailed requirements specifications they expected off-the-shelf products that could be delivered on a short notice. These customers were not interested in buying a unique product, on the contrary they preferred products that had a history and a widespread use. In return for lack of uniqueness the customer demanded a much lower price, it became impossible to demand enough for a product to cover the complete development costs.

The obvious consequence for the company was that it became necessary to develop generic products through internally financed and managed projects.

1.3 The need for process improvement

Having launched R&D programs to develop the generic modules needed in order to meet the requirements of the market, we found that using the old engineering standards was neither easy nor ideal. Our methodology was focused on having one strong customer, and was suited for large projects. The R&D projects had many customers, with different requirements and were relative small. Typically a R&D project can be to develop only a small part of a system or even to improve an existing module.

Kongsberg Spacetek soon developed a "lightweight" system engineering standard and methodology. Initially our internal standard was an extract of the more complete ESA PSS-05 standard. At the time this standard was very much the traditional "waterfall approach".

In order to further strengthen our quality assurance focus we became ISO-9001 certified in 1998.

During the late 90's and into the new century, the trend on lower prices and faster delivery combined with more customers continued. Our paper based, document heavy and highly manual quality system came under increasing pressure. It became impossible to follow the standards and even more impossible to do effective QA on all projects.

The need for improvements became obvious. In fact what we needed was a revolution.

As we all know the new ISO-9001:2000 standard demands a process oriented quality system. With the requirement that a process oriented system had to be implemented before December 2003 in mind, Kongsberg Spacetec started a program to define a whole new system for the entire company.

1.4 Getting started

Being part of the Norwegian SPIKE project, we were lucky to have a whole community of companies and research organisations available for getting new ideas.

The first that came obvious was the need for defining what processes were important in the company. Mapping processes and documenting them seemed relative straightforward – with a little help from our friends [1]. The next problem was how to publish the processes. It seemed obvious that creating a new set of paper tigers, so that all employees could leave them to dust in their shelves, was a hopeless idea.

It early became clear the standards and processes had to be electronically published and maintained. However this was not new. Documents on the Intranet are only marginally better than documents on the shelves. Along came the idea of a graphical interactive electronic process description tool, with boxes showing the processes, and flows showing the interaction between them.

In the mean time a company part of the SPIKE project, Firm, had came up with a more advanced tool for documenting the processes [3]. The genius with this tool was that it tailored the process to each project and integrated templates and all other information needed by the project into the project guide.

It is said that seeing is believing, after seeing the Firm solution Kongsberg Spacetec immediately decided that we had to do something similar.

2 *Developing the project web*

2.1 Setting up the Requirements

Having decided the form of our new electronic process guide, we started off setting up the most important requirements for all the different stakeholders. Some of the initial key requirements and stakeholders identified were:

Management and QA

- The tool shall present an overview of all the running projects in one page.
- The tool shall show key status for each process on one line in the form of text and symbols.
- Status shall be immediately accessible and understandable and up-to-date at all times.
- Each project shall have an associated front-page documenting key project data at the project blastoff (kick off).
- Each project shall have an associated end-page documenting key project data and statistics at the time of project termination.

Project managers

- A project summary page per project shall provide access to all processes applicable for the specific project.
- Each process shall have a checklist through which the project can track progress.

- The checklists shall be possible to adapt to the project by removing checkpoints that are not applicable, by adding additional checkpoints and by defining multiple iterations of a process.
- The status of each process (progress) shall be presented visually and a milestone on phase completion shall be possible to define.
- The project main page shall provide direct access to vital project management tools, documents and information. At least:
 - Project action tracking.
 - Project work packet planning.
 - Risk analysis and management tool.
 - Accounting data.
 - Monthly status reporting.

Developers

- The processes must reflect the „best practices“ currently used within the company.
- The processes shall be consistent with modern methodology and compliant with Unified Process and Component Based Development.
- The guide shall give access to additional documentation such as procedures and instructions via links in the processes.
- The guide shall integrate important tools for development, at least:
 - Requirements definition.
 - Use-case description.
- Each project shall have assigned a copy of development documentation templates.

These initial requirements were defined by the QA department, based upon separate meetings with the different stakeholders as well as own research.

2.2 Development

Our strategy for development was divided in two:

2.2.1 Documenting the processes

Consulting with our research partners in SPIKE we decided that instead of adapting a ready made third party processes we wanted to develop our own processes using best practices within the company [1, 2].

The methodology we chose was to arrange a set of workshops, summoning key personnel within the company with extensive experience on the relevant processes. We set out a plan targeting the most important processes first. The priority processes were identified as:

- Software development, from initiation to delivery.
- Deployment, delivery and installation.
- Sale process, from writing bids until signing a contract.
- Maintenance, handling warranty and maintenance.

For each set of processes a workshop were scheduled. The first workshop was held autumn 2002 and addresses software development processes.

Prior to the workshop the QA department prepared a list of basic requirements for the processes. In

the case of development we put forward the following requirements:

- The development process is started with a hand-over of a contract to the development department.
- The development processes shall be iterative and incremental.

Beside this, we left it up to the workshop to decide what sub-processes were needed and what contents each sub-process should have.

The workshop was lead by our research contacts. It started by brainstorming and discussing what sub processes were needed. In the case of Development we soon agreed on the following list of sub processes:

- Blastoff: Hand-over and initial planning of the project.
- Spesification: Requirements analysis.
- Elaboration: Architectural Design.
- Construction: Detailed Design and Implementation.
- Integration: Integration and Testing.

Project KRTEST (1115 - KRN) making "PWEB"					
Processes and Project Status					
Blastoff	Spesification	Elaboration	Component construction	System integration	Project closedown
Process (20/20 closed)	Process (0/5 closed)	Process (0/10 closed)	Process (0/10 closed)	Process (0/9 closed)	Process (0/2 closed)
	TOA (0/3 closed)	TOA (0/4 closed)	TOA (0/4 closed)	TOA (0/3 closed)	
	PM (0/6 closed)	PM (0/3 closed)	PM (0/3 closed)	PM (0/5 closed)	
100% Completed	0% 14 of 14 remaining	0% 17 of 17 remaining	0% 17 of 17 remaining	0% 17 of 17 remaining	0% 2 of 2 remaining
Milestone: 20040212	Milestone: 20040311	Milestone: 20040401	Milestone: 20040501	Milestone: 20040601	Milestone: 20040615
Project end-date according to front-page: 20040529					

Figure 1: The development processes in the project web.

Each of these processes was analysed separately. A brainstorming session using “yellow notes” were held identifying:

- Input required to the process.
- Roles participating in the process.
- Output from the process.
- Work tasks in the process.

After the brainstorming the activities were grouped, sequenced and agreed.

The experience using this method was only positive. All participants (up to 8 persons + researchers) were active and engaged. In spite of the large number of people participating, we managed to use only one day agreeing on the contents of all development processes. Furthermore the process descriptions became surprisingly complete, it took one man only one day to transcribe and finish the processes.

2.2.2 Developing the project web

The name of our electronic process guide was set to “The Project Web”. A screening of the market looking for suitable tools using our SPIKE contacts gave no results. At the time we could find no suitable off-the-shelves product to use. We also evaluated the tool developed by Firm but found that we had requirements that made it hard for us to use their solution.

Once having decided to develop our own solution we decided for a simple and iterative strategy:

1. Develop a prototype based upon the key requirements using technology that was well known in the company and easy to maintain.
2. Present the prototype to representatives for the stakeholders and collect responses.
3. Improve the prototype according to feedback as far as possible.
4. Back to step 2 until all parties agree that the prototype is ready for real-life test.
5. Once implemented in the organization a strategy of rapid response to real user feedback should be applied to develop the tool into a finished product.

The theory was that giving a lot of people direct influence on the tool would create a sense of ownership to it and hence inspire the stakeholder to use it.

Management of the project was kept with the Quality Assurance department. The implementation was executed by QA personnel. The first prototype was ready after only a few weeks. After few iterations the prototype were in such a condition that it was possible to start real-life testing with a few projects.

Development Milestones:

- November 2002: Start of implementation.
- December 2002: Prototypes were tested.
- January 2003: Test with real projects.
- March 2003: Introduced as a mandatory tool for all **new** projects.
- May 2003: Deployment processes added to the guide.

3 Running Projects

Status	Project Name	Project Nr.	Started	End	P. Report	Economy	Alert	Mail PM	Team	Customer
84%	IPAS	8342	20030620	20031212					SH (PM), KAS, KI. <i>TQA: KRN</i>	CSIR
58%	QM Avdeling 2004	5004	20031203	20041230					KRN (PM), NJV, BIN.	KSPT
22%	KRTEST	1115	20040209	20040529					KRN (PM), . <i>TQA: NOT GIVEN!</i>	KSPT

Create New Project

Show Old Projects

[QA Actions](#) [Search actions](#) [Show statistics](#)

Figure 2: The project web main page (early version).

3 Organisational implementation

3.1 Introduction of the project web

As mentioned before, development was carried out during a relative short time in tight co-operation with many of the stakeholders. Involving the management, middle management, QA and developers generated expectations within the organisation. We experienced a pressure to release the project web for use in the real-life projects.

The introduction was done in March 2003, at the time we chose to only allow new projects to use the tool. Running projects were to follow the existing processes and standards. The reason for this was mostly practical; we wanted the project web to grow in a controlled manner in order to handle the expected need for support.

The reception of the tool was good, project managers had a positive attitude and showed a genuine interest for using the project web. As expected, real life use generated a flow of feedback. Bugs were found and suggestions for improvement were made. We chose to prioritise correcting bugs, believing this was important in order to keep up the positive attitude. After a short time we had a stable product and loads of improvement suggestions.

3.2 Continuous improvement

Having gained experience with real life usage, we found that a tool like the project web generates a momentum for process improvement by itself. The very nature of the project web is that all the processes are reviewed with every project, the project team immediately responds to the process descriptions if they are unclear, uncompleted or unusable. The challenge is only to gather this feedback and handle it properly. The QA department chose a strategy of rapid response to all comments.

- If the comment was obviously meaningful (unclear, missing information) we immediately updated the process description.
- Feedback that required discussion was duly registered and discussions were held in adequate forums in order to resolve the problem as soon as possible.

The advantage of this strategy has been that we have been able to keep up the positive attitude even up to the present time. Users understand and feel that the goal of the process is not to put load on the project team – but to provide true help and guidance. In addition we have a living process of process improvement within the organisation.

Process improvement was expected, a more surprising consequence of introducing the project web was the popularity of the process independent tools that was integrated. From the start we had implemented some tools we hoped would be of use:

- An action list tool with automatic e-mail alerts when due-date is passed.
- A Work Packet tool providing a template Work Breakdown Structure.
- A risk tool for following up risk and calculating project risk-level.
- A requirements tool for writing requirements according to our standard.
- A use-case documentation tool.
- A minutes of meeting documentation tool.

The response towards the tools was that they were regarded as one of the major benefits using the project web. We received loads of suggestions of improvements and requests for new tools.

In August 2003 we started logging the use of the project web. Analysis of the usage statistics shows that tool usage is by far the most frequently used part of the project web. Tools are accessed more than ten times as frequently as process descriptions and project status.

We believe tools are one major reason for the popularity of the project web. Since the tools are integrated closely with the processes they encourage the use of the process descriptions as well.

Since the introduction of the project web, many new tools have been added and more importantly the tools have been integrated against the company accounting systems:

- The Work Packet tool is now integrated with the accounting system so that hours actually spent is shown in the tool at all times.
- Order status, accounting status and invoicing status is integrated with the project web, allowing the project team to compare the plan against the actual status at all times.
- Project progress reporting is partially automatic, simplifying the monthly reporting.
- The company product databases are integrated with planning tools allowing the project manager to select from the range of available products when planning a delivery.

The below figure shows a tool to get an updated graphical presentation of project progress over time. The original plan for resource usage is compared with actual resource usage, estimated remaining work and project budget.

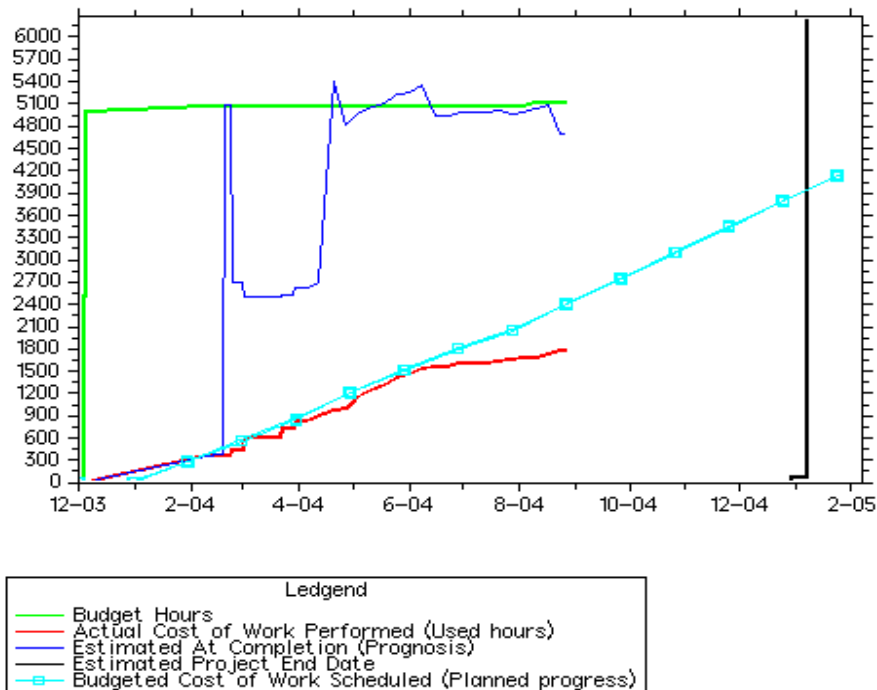


Figure 3: Graph showing project progress over time.

3.3 Keeping up the interest over time

As discussed previously, we believe that keeping up the interest and usage of an electronic process guide has been achieved by allowing the users direct influence on the continuous improvement of the tool. However there are many other aspects:

- Integrating the process description with useful tools seems like a great idea.

- Establishing forums for experience exchange between users have been working; we are now putting more focus on this.
- Management interest seems to be vital, when management level uses the tool for monitoring status and responding when problems are indicated seems to encourage use.
- Enthusiasm and ownership amongst the QA personnel helps generating a push towards improvement and use of the project web tool.

Today the status of use is that every project is using the project web. Furthermore statistics shows that all employees working with projects access the project web every month. An all time high usage frequency was reached May 2004 with an average of 180 hits per employee.

Watching the usage over time - it is clear that whenever new features are added to the project web and whenever we launch some kind of campaign, the usage immediately increases. In periods where no focus activities take place, the use drops slightly. These observations are interesting, as it seems like continuous improvements on the process endorse use of the processes.

4 Measuring performance

As early as spring 2003 we defined a GQM plan [2] together with our SPIKE researchers. The objective was to measure interest and performance of the project web. In order to measure the performance we launched a plan for collecting data:

- Hit statistics has been gathered from August 2003.
- We have launched a program running questionnaires amongst all project web users. At this time we have conducted three surveys; September 2003, January 2004 and August 2004.
- We have launched a program interviewing in depth a representative group of users. At this time we have conducted three rounds of interviews; September 2003, January 2004 and August 2004.

Preliminary results of the two first interviews and surveys have shown great expectations, a wide-spread use, an overall positive attitude against using the project web and indications that we are gaining performance benefits. In addition we have been able to identify weak spots in the process guide allowing us to focus on these in our SPI effort.

In addition we have measured project performance in terms of profit, delivery precision and planning precision. The goal is to be able to monitor whether or not we have a positive trend over time.

At the present we have indications of improvement, however the number of projects that has been run from start to finish using the project web is still too small to give base for reliable conclusions.

5 The way ahead

The interest for the process web has gained a momentum for process improvement within the company. We do not lack ideas for improvements nor interest. One of the key challenges is keeping the interest up over time. To do so we are on the edge of launching educational campaigns and we are well on the way towards establishing user discussion forums.

Based upon the positive experiences with the project web, both the support and the marketing departments have requested a similar tool for managing respectively maintenance contracts and proposals.

The project department is requesting that the project web is more tightly integrated with design tools and a Gantt tool for project planning.

At the present we see no reason for changing our strategy for process improvement using the project web as a focal point.

6 Conclusion

Kongsberg Spacetec has undergone a small revolution with respect to software process improvement by introducing an electronic process guide. We believe that our solution with an interactive guide closely integrated with simple but highly useful tools is a major success factor. Involving stakeholders from the start has given a sense of ownership and commitment amongst the employees that we believe could not have been achieved by buying a ready-made guide from an outside source.

In our case we developed the complete tool from scratch, today it may be possible to buy off the shelves products that provide much of the desired functionality. Development of the tool has not been a major cost factor, as we have not used extra resources outside the QA department for doing this. The most important factor to keep in mind is that the tool must be possible to adapt to the existing company infrastructure and culture.

We do not believe in buying ready-made processes enforcing them on to the employees. The major cost of introducing new processes is implementation in the organisation, not writing them. By developing the processes through workshops with the stakeholders, much of the implementation is already achieved.

Focusing improvement activities around a living tool like the project web has moved Kongsberg Spacetec from being a company relying on outside pressure to improve into being a company with a culture for continuous self improvement. Improvement activities take place with small steps every day instead of giant leaps every other year. In addition to being more according to modern philosophy this strategy is much more giving for both the QA personnel and the rest of the company.

A key challenge over time will be to document that the effort spent has a direct positive impact on the company performance, at the end of the day we will have to document a positive impact on the bottom line. At the current time we have indications of a positive impact of the SPI activities – and we believe time will show our effort has been fruitful.

7 Literature

- [1] Torgeir Dingsøy and Nils Brede Moe, "The Process Workshop - A Tool to Define Electronic Process Guides in Small Companies," in Proceedings of the Australian Software Engineering Conference (ASWEC), Melbourne, Australia: IEEE Press, 2004.
- [2] Tore Dybå, Torgeir Dingsøy, and Nils Brede Moe, Process Improvement in Practice - A Handbook for IT Companies. Boston: Kluwer, 2004, ISBN 1-4020-7869-2.
- [3] Nils Brede Moe, Torgeir Dingsøy, Tore Dybå, and Trond Johansen, "Process guides as software process improvement in a small company," EuroSPI, Nürnberg, Germany, 2002.

Author CVs

Ken Rune Nilsen

Graduated Master of Science 1992, University of Tromsø

2003 Vice President, department of Quality and Methodology.

2000: Director Infrastructure and Methods group, Developments department

1997: Head systems engineer generic ground station product, R&D department.

1996: Project manager and Systems Engineer, R&D department.

1995: Systems Engineer

1993: Employment at Kongsberg Spacetec as Software Engineer