EuroSPI'2005

European Software Process Improvement

9.-11.11.2005, Budapest, Hungary

http://www.eurospi.net

EuroSPI 2005 Experience Sessions 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

John von Neumann Computer Society

John von Neumann Computer Society

Ability, pride and creativity of our compatriots are for Hungary the fundament of progress
and the only spring-board into the future.
(Count István Széchenyi 1842)

Fields of activity:

As a significant professional body and learned society in the Hungarian IT community, the John von Neumann Computer Society (NJSZT) is dedicated to preserving values that can be included in today's knowledge-based society as well as to setting new directions that meet the requirements of the age and to actively forming the IS world of the future. The primal activities of our Society are IT support, ECDL (European Computer Driving Licence) Hungary, Hungarian Smart Card Forum, Organization of International and National Conferences.

Not-for-profit Professional Organisation for the Information Society

ECDL Hungary
Centre of the European Computer Driving Licence program in Hungary

URL: www.njszt.hu E-mail: titkarsag@njszt.hu
URL: www.ecdl.hu E-mail: ecdl@njszt.hu

ISBN 963 8431 94 6

TITEL EuroSPI 2005 Industry Proceedings	DATE 9.-11.11.2005
	NO. OF PAGES 273
AUTHOR/ Editor Miklós Biró (Corvinus University of Budapest) Richard Messnarz (ISCN)	

3 KEYWORDS

Software Process Improvement
Management
Measurement

EuroSPI 2005 Proceedings

Proceedings

The papers in this book comprise the industrial proceedings of the EuroSPI 2005 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 12th 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.

EuroSPI Chairs

General Chair Dr Richard Messnarz, ISCN

Industry Chair Bernd Hindel, ASQF

Publicity Chair not defined so far

EuroSPI 2005 Local Chair Dr Miklós Biró, Corvinus University Budapest

Scientific Programme Committee Chairs Ita Richardsson University of Limerick, Ireland, and Pekka Abrahamsson VTT Electronics , Finland

Industrial Programme Committee Chair Risto Nevalainen FiSMA and STTF

Industrial Programme Committee Co-Chairs Jorn Johansen & Mads Christiansen, DELTA, Torgeir Dingsoyr & Nils Brede Moe, SINTEF

Exhibition Chair Stephan Goericke, ASQF

Tutorial Chair Dr Richard Messnarz, ISCN

Industrial Programme Committee

Werner Achtert, TUV, Germany

Scott Ambler, Ronin International, Canada

Geir Amsjø, UiO, Norway
Bo Balstrup, Danfoss, Denmark
Nils Brede Moe, SINTEF, Norway
Lars-Ola Damm , Ericsson, Sweden
Carol Dekkers, Qualityplustech, US
Beatrix Barafort, CRPHT, Luxemburg
Andreas Birk, SD&M, Germany
Mads Christiansen, DELTA, Denmark
Alec Dorling, ISO-SPICE, UK
John Elliott, UK
Michael Ferchau, INVERSO, Germany
Lone Fjordstrom, Nokia, Finland
Johann Flachs, Giesecke & Devrient, Germany
Uwe Hehn, Methodpark, Germany
Tim Hind, Axa Sun Life, UK
Oliver Kraus, DaimlerChrysler AG, Germany
Nils Jakob Villmones, Kongsberg SpaceTec, Norway
Jorn Johansen, DELTA, Denmark
Richard Messnarz, ISCN, Ireland
Risto Nevalainen, STTF, Finland
Bernd Reh, Siemens VDO, Germany
Philippe Saliou, University of Brest, France
Hans Scherzer, APAC, Austria
Gerhard Thuencher, ZF Friedrichshafen AG, Germany
Tor Ulsund, BravidaGeomatikk, Norway
Hans Westerheim, SINTEF, Norway
Bruno Woeran, Danube, Austria

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 Miklós Biró and Dr Richard Messnarz, ISCN

Welcome Address by the EuroSPI General Chair

Dr Richard Messnarz

EuroSPI is 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 EQN (European Quality Network) which is funded by the EU Leonardo da Vinci Programme and establishes an European certification unit for T & Services professions. I therefore expect that EuroSPI partners will closely collaborate to form a group of national institutions in Europe representing a set of certified professions related with innovation and management.
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 for SPI and innovation".

Welcome to Budapest by Dr Miklós Biró

Dr Miklós Biró

Local Chair

It is a pleasure welcoming EuroSPI in Budapest at the heart of the historical Castle District, one of UNESCO's World Heritage sites eight years after the success of its predecessor also organised in this city. One of the key enablers of progress during this period was the spread of Information Society Technologies which had a definite impact on Software Process Improvement as well, making it possible to seamlessly produce software systems globally, with components developed in locations distant in space and time.
Being boosted by technology, globalisation is however a process whose history goes back to the times of Chandragupta Maurya founder of the first Indian empire (321 B.C.) and Alexander the Great whose troops were the first to open the route from Europe to Asia called Silk Road later. The business significance of globalisation was also clearly recognized by Adam Smith back in the 18th century.
The new opportunities offered by global Software Process Improvement are rooted in financial, and educational advantages, while the transfer of organizational maturity and the management of cultural differences means new challenges.
Among many of the countries represented at EuroSPI'2005, the above issues are highly relevant in Hungary as a new member state of the European Union. Budapest is by consequent a perfect ground for opening the scope of EuroSPI, reflected in these proceedings, to both business and research communities especially interested in the following issues:

- Process improvement, intellectual asset creation opportunities and concerns in outsourcing, offshore development center operation, acquisition.
- "Advanced capabilities in the engineering and management of software systems, services and applications" [EU FP6 IST priority, WP 2005-06]
- Innovation strategies and experiences.

The expanding scope together with the expanding network of experts joining the knowledge and experience exchange community contributes to the globally beneficial impact of EuroSPI.

Contents

Experience Session I: Extreme Programming

Experience Session 2: SPI Tools & Evaluations

Experience Session 3: Global and Virtual Development

Experience Session 4: Assessments

Experience Session 5: Knowledge & People

Experience Session 6: SPI Adoption Experiences

Experience Session 7: SPI and Processes

Experience Session 8: SPI and Innovation

Extreme Programming and Rational Unified Process – Contrasts or Synonyms?

Per Runeson and Peter Greberg
Lund University, Sweden
per.runeson@telecom.lth.se

Abstract

The agile movement has received much attention in software engineering recently. Established methodologies try to surf on the wave and present their methodologies as being agile, among those Rational Unified Process (RUP). In order to evaluate the statements we evaluate the RUP against eXtreme Programming (XP) to find out to what extent they are similar and where they are different. We use a qualitative approach, utilizing a framework for comparison. We conclude from the analysis that the business concepts of the two – commercial for RUP and freeware for XP – is a main source of the differences. RUP is a top-down solution and XP is a bottom-up approach. Which of the two is really best in different situations has to be investigated in new empirical studies.

Keywords

Agile, XP, RUP, comparison, framework analysis

1 Introduction

The agile movement has appeared the last years as an alternative direction for software engineering [1]. Among the agile methodologies, eXtreme Programming (XP) is the most well known [3][5]. In the current agile boom, many established software engineering methodologies try to present themselves as being agile. The Rational Unified Processes (RUP) [18] is among those, providing "plug-ins" to RUP for eXtreme Programming[1]. Thereby they offer a downsized version of RUP, which is stated to be lightweight, agile style.

Both methodologies share some common characteristics; they are iterative, deliver incremental releases, customer-oriented and role-based [2]. RUP is generally not considered agile; rather it is criticized for being too extensive and heavyweight. RUP comprises 80 artifacts, when fully instantiated, while XP stresses a few key artifacts; the code, unit tests, user stories and similarly. RUP has 40 roles while XP has five.

These issues lead us to the main research question in this paper: Do RUP and XP match together? Are they synonyms, or are they contrasts? There are existing comparisons, e.g. by IBM [12] and Ambler[2], [2], which compare the technical content and purpose of the two. Our research approach to investigate the question is a qualitative framework analysis. Using a modified version of a standard question framework, we investigate similarities and differences between RUP and XP.

[1] http://www-106.ibm.com/developerworks/rational/library/4156.html

[2] http://www.agiledata.org/essays/differentStrategies.html

small to medium sized software development teams. XP is intended to meet the demands of a context with unclear and volatile requirements. The methodology is not primarily commercial; instead there is a set of people – a community – who evolve and develop the methodology as such, as well as tool support (freeware), to advance and support XP development projects.

The origin of RUP and XP are similar. They are both based on experience from software engineering. Both are evolved during the same decade, although RUP has its roots earlier.

There are two different underlying philosophies behind RUP and XP. RUP takes to a large extent a technical management perspective while XP focuses on the development staff. RUP is originally designed to support large projects, while XP is originally designed for small to medium sized projects, for which type of projects several experience reports are published, see e.g. [10][14][21]. The distribution of the methodologies is different; RUP is primarily commercial and XP is primarily freeware, although RUP can be accessed through books, and XP is commercialized by consulting companies.

3.2 Why?

We analyze advantages and disadvantages for the two methods from three perspectives, technical, financial and social points of view.

Technical perspective. On the technical side, RUP is provided together with a large package of development tools and documents. It is delivered online via the web, and updated in new releases. It can be tailored and extended to suit the individual organization's needs. One major sales argument for RUP is the integrated tool-suite, although it is debated how well they integrate.

XP on the other hand strives towards simplicity. It comes with more loosely connected tools, which are developed in the XP community, to support specific practices, e.g. Junit for unit testing.

RUP is a large collection of processes, artifacts and roles. This must be scaled down for most projects except for the very largest ones. XP starts in the other direction, with a minimal core of values and practices, which has to be scaled up to fit larger contexts.

Financial perspective. The *financial* issues are different in the distribution and support of the methodologies, since RUP is a commercial product and XP is freeware as are most of the related tools. The financial power behind RUP is used for marketing giving more visibility to RUP. Rational Software is owned by IBM, which has good reputation in the software industry.

On the other hand, why should one pay for something that can be achieved for free? Effort must be spent on tailoring RUP, why should an organization then pay for it as well? [11] XP offers the freeware solution, which is financially advantageous, but may cause social reactions. Both approaches require tailoring and transfer effort before established in an organization.

Social perspective. The *social* aspects of RUP and XP are also related to the commercial versus freeware discussion. Larger software development companies are used to buying software licenses, and hence buying licenses for methodology is quite natural. The freeware principle behind XP is met with skepticism. Can something that is for free be good? The situation is very much like the open source situation. Free software is offered from the open source community and software is licensed from commercial companies, e.g. the Linux operating system versus Microsoft Windows.

The choice is of course primarily technical and financial, but there is a significant social aspect. Smaller organizations and technical staff show a tendency to be more in favor of the freeware/open source approach, while large organization and management are in favor of the license approach. The good reputation and financial strength behind RUP are management arguments, while on the technical level, people know that both approaches need tailoring and hard work tend to choose the method which is least complex, and puts the technical work in focus.

3.3 When and Where?

Regarding the *time* dimension, the development in RUP is organized in four sequential phases, *inception, elaboration, construction* and *transition*. Theses four phases constitute one development cycle, producing one release of the software. Within each phase, there are a number of iterations, and the four phases have their main focus on different activities, although all activities are run in parallel, see [13][18]. *Inception* stresses business and requirements, *elaboration* is architecture-focused, *construction,* is mainly implementation and test and *transition* has its main focus on deployment and change.

XP has its main focus on the produced code, independently of the time aspect. In the beginning of a project, the focus is on the product core, and later on features, but it is a code focus all the time. The design evolves as the software evolves. The *simplicity* value and the *simple design* practice emphasize that the design shall be as simple as possible for the current needs, not for future possible needs. Like in RUP, design and analysis activities are not concentrated to the beginning of the project, but intertwined with the development in the *planning* activity.

Both RUP and XP stress short iterations, although iterations in XP are even shorter than in RUP. In XP, iterations range from seconds in the pair programming activity, via days in the stand-up meetings to months in a release plan, see [3][5]. The iterations in RUP are less frequent, in the magnitude of weeks or months.

Both methods strive towards short lead-time and efficient use of resources. The XP principle of developing only what is absolutely necessary, indicates that XP will be the most efficient method. On the other hand, only empirical studies will provide sufficient answers to the question.

The *geographical* dimensions are not explicitly addressed in either methodology, but are present implicitly in both. RUP originates from a context of large distributed development projects, and its approach with artifact-based communication is intended to support this kind of geographical situation. The philosophy behind XP is based on direct, oral communication, both internally in the project and externally towards customers, hence requiring a limited geographical distribution. In practice, XP teams must be located in the very same room to gain the most benefits of the methodology. Even being located at different floors in a building has caused communication problems [15].

3.4 How?

This section deals with the technical content of the two methodologies. We analyze the *extent* of the methodologies, the *organization* of the methodologies and the *tools* support. Regarding the organization, we analyze common aspects, and try to find similarities and differences between the two. The analyzed aspects are *flexibility, project drivers, customer relation, releases and technical work.*

Extent. RUP consists of a large collection of documents, role descriptions, activities etc. RUP stresses the need for tailoring to a specific organization, which in most projects equals downsizing of the methodology. RUP is considered and criticized for being "heavy-weight".

XP is very lightweight, both in its presentation and in the practical application. Everything that is provided to start using XP in a project is covered in each of the sequence of books published on the theme, e.g. [3][4][5][6][7].

An indication of the difference in extent of the two methods is illustrated in Table 3, where all the roles of an XP project are presented, with their counterparts in RUP, constituting a small subset of the RUP roles. In total, RUP comprises more than 80 major artifacts, 150 activities and 40 roles [18].

In summary, RUP is a much more extensive methodology than XP, for good and for bad.

Table 3. XP roles and their counterparts in RUP

Team	XP roles	RUP roles
Customer team	Customer	Requirements specifier System analyst Project manager
	Tracker	
	Tester	Test analyst Tester Test system administrator
Development team	Programmer	Implementer Designer Integrator System administrator
	Coach	

Flexibility. Both methodologies stress the word flexibility. In RUP, it primarily means tailoring to different needs in different contexts and its focus on iterations. In XP flexibility means continuous change, based on the feedback loops. The short feedback loops require continuous actions. The 12 practices can be implemented differently in different projects. The values are the stable core of XP, while everything else may change.

Project drivers. RUP is defined as being use case driven, i.e. descriptions of use of the system are implemented, and continuously integrated and tested. XP applies test-driven design, i.e. test case are derived and implemented before the code is written. XP has user stories to guide what to implement. These user stories are less extensive descriptions, compared to the RUP use cases, where the complete scenario for the interaction between the user and the system is defined.

Regarding planning, both methodologies agree on that a complete project cannot be planned in detail. RUP proclaims continuous changes in the plans, while XP advocates planning only the very near future in detail.

Customer relation. Regarding the customer relation both methodologies stress the importance of a close relation to the customer, but still this issue is very different.

XP assumes the customer be involved in person in the team to "answer questions, resolve conflicts and set small-scale priorities" [3]. This is later turned into "an XP project is controlled by an assigned person, defining requirements, setting priorities and answering questions from the programmers". RUP is more flexible on the implementation of the customer relation. It is not always possible or even feasible that the customer is present in person.

Releases. RUP defines a release to be "a stable, executable version of a product and its necessary artifacts" [18], while XP defines it to be "a set of user stories creating a business value" [6]. The XP practice *small releases* and the RUP item d*evelop software iteratively* are very similar, assuming that a release can be both internal and external.

Technical work. XP involves two controversial practices, *collective ownership* and *refactoring*, which are tightly connected. They are also highly dependent on the *continuous integration* and *testing* practices, which constitute the quality assurance mechanisms. These practices are based on the principle of *sharing* responsibility. In RUP, which originates from larger systems, different project members are responsible for different subsystems, thus the underlying principle is *division* of responsibility

Tools. The RUP process as such is guided by a tool, and there are suitable tools for e.g. modeling that interface with the methodology. As the methodology is so extensive, this is absolutely necessary, to guide the user. This is also a part of the commercial success of RUP.

XP does not proclaim any specific tools. There are tools offered by the community, e.g. *Junit*, but any kind of CASE tools and project management tools can be used in XP. However, it is worth noticing, that in its original form, whiteboards, paper cards and pens are the most mentioned tools in XP.

Who? What characterizes the developers and organizations using RUP and XP respectively? XP focuses on the individual developer, empowering the technical level in the organization. It is based on direct communication between stakeholders, and requires courage, as openness and honesty are important. This requires the staff and organizations acknowledge and maintain these kinds of characteristics and values. It requires team workers solving problems in teams, and not feeling discomfort for peer reviews.

RUP does not focus on the individual developer, but emphasizes the roles, which are tailored to specific projects. It prescribes documentation, which puts demands on the staff to be motivated to spend effort on preparing and maintaining the artifacts.

The origin of the methods are different, RUP originates from large projects and organizations, and XP from the small. This fact permeates the methodologies as such, as well as its advocates and critics. RUP is a top-down methodology, typically advocated by management while XP is a bottom-up methodology, typically advocated by the technical staff.

4 Conclusions

In this paper, we have analyzed the similarities and differences between RUP and XP methodologies, based on a qualitative framework. Although many keywords and key values are the same, the two methodologies are quite different. Common values are user/customer involvement, iterations, continuous testing and flexibility. The implementation of these values are however very different. RUP offers an extensive process description, comprising artifacts, roles, activities, integrated tool-suites etc. XP on the contrary stresses values and principles, rather than prescriptive instructions, and focuses freedom and simplicity. The distribution channels are different, RUP being a commercial product by a large company, and XP is freeware, maintained by a community of volunteers.

We conclude from this analysis that the two in many aspects are in contrast. The situation is very similar to the Windows vs. Linux case. One is commercial, the other is freeware. One tends to be advocated by managers, the other by engineers. Still both are operating systems for personal computers. It is important to be aware of this social aspect in the selection of RUP or XP. Which of the two is best suited for certain types of projects needs to be further investigated in empirical studies.

5 Literature

[1] Agile Manifesto, http://www.agilemanifesto.org/ last visited 05-08-30

[2] Ambler, S., *Agile Modeling Effective Practices for Extreme Programming and the Unified Process*, John Wiley & Sons, 2002.

[3] Beck, K., *Extreme Programming Explained – Embrace Change*, Addison-Wesley, 2000.

[4] Beck, K. and Fowler, M. *Planning Extreme Programming*, Addison-Wesley, 2000.

[5] Beck, K. and Andres C., *Extreme Programming Explained – Embrace Change 2nd edition*, Addison-Wesley, 2005.

[6] Chromatic, *Extreme Programming Pocket Guide*, O'Reilly, 2003.

[7] Crispin, L. and House, T., *Testing Extreme Programming*, Addison-Wesley, 2003.

[8] Feller, J. and Fitzgerald, B., "A Framework Analysis of the Open Source Software Development Paradigm", *Proceedings of the 21st International Conference on Information Systems (ICIS)*, ACM, pp. 58-69, 2000.

[9] Fowler, M., Beck, K., Brant, J., Opdyke, W., and Roberts, D., *Refactoring: Improving the Design of Existing Code*, Addison-Wesley, 1999.

[10] Fraser, S., Beck, K., Cunningham, W., Crocker, R., Fowler, M., Rising, L., and Williams, L., "Hacker or hero? - extreme programming today (panel session)", *Proceedings of the conference on Object-oriented programming, systems, languages, and applications (Addendum)* pp. 5- 7, 2000.

[11] Henderson-Sellers, B., Due, R., Graham, I. and Collins, G., "Third Generation OO Processes: A critique of RUP and OPEN from a Project Management Perspective", *Proceedings. Seventh Asia-Pacific Software Engineering Conference*, pp. 428-435, 2000.

[12] IBM (Smith, J.), *A Comparison of the IBM Rational Unified Process and eXtreme Programming*, http://www3.software.ibm.com/ibmdl/pub/software/rational/web/ whitepapers/2003/TP167.pdf

[13] Jacobson, I., Booch G. and Rumbaugh, J, *The Unified Software Development Process*, Addison-Wesley, 1999.

[14] Karlström, D., "Introducing Extreme Programming - An Experience Report", *Proceedings Third International Conference on eXtreme Programming and Agile Processes in Software Engineering*, 2002.

[15] Karlström, D. and Runeson, P., "Integrating Agile Software Development into Stage-Gate Managed Product Development", submitted to *Empirical Software Engineering An International Journal*, 2005.

[16] Karlström, D. and Runeson, P., "Combining Agile Methods with Stage-Gate Project Management", *IEEE Software*, May/June, pp.43-49, 2005.

[17] Kitchenham, B., Linkman, S., and Linkman, S., "Evaluating Novel Software Engineering Tools", *Proceedings The 7th International Conference on Empirical Assessment in Software Engineering* (EASE 2003), Keele University, Staffordshire, UK, pp. 233-247, 2003.

[18] Kruchten, P., *The Rational Unified Process – An Introduction*, Addison-Wesley 2nd edition, 2000.

[19] Lindland O. I., Sindre, G. and Sølvberg, A., "Understanding in Conceptual Modeling", *IEEE Software*, March, pp. 42-48, 1994.

[20] Robson, C., *Real World Research*, Blackwell Publishers, Oxford, 2nd edition, 2002

[21] Schuh, P., "Recovery, Redemption, and Extreme Programming", *IEEE Software* December, pp. 34-41, 2001.

[22] Scott, K., *The Unified Process Explained*, Pearson Education, 2001.

[23] Seaman, C., "Qualitative Methods in Empirical Studies of Software Engineering", *IEEE Transactions on Software Engineering*, 25(4):557-572, 1999.

[24] Wohlin, C., Runeson, Höst, M., Ohlsson, M. C., Regnell, B. and Wesslén, A., *Experimentation in Software Engineering: An Introduction*, Kluwer Academic Publisher, USA, 2000.

Software Process Improvement in Europe: Potential of the new V-Model XT and Research Issues

Stefan Biffl[1], Dietmar Winkler[1], Reinhard Höhn[2], Herbert Wetzel[3]
[1]Vienna University of Technology, Institute of Software Technology and Interactive Systems,
Karlsplatz 13, A-1040 Vienna, Austria
{stefan.biffl, dietmar.winkler}@qse.ifs.tuwien.ac.at

[2]Knowledge Management-Associates GmbH
Lerchenfelder Gürtel 43, A-1160 Vienna, Austria
reinhard.hoehn@km-a.net

[3]IMG AG, Fürstenlandstrasse 101, CH-9014 St. Gallen, Switzerland
herbert.wetzel@img.com

Abstract

The goal of European industrial practice to support high-value software production in diversified domains has led to the development of a huge number of process model variants. However, these diverse models are hard to compare, which hinders efficient collaboration and software process improvement on a European level. Process managers see a growing need for approaches that support stakeholder collaboration, systematic process mapping, and transformation of processes to improve their leverage in software process improvement. In this paper we present the V-Model XT (VM-XT), a flexible software process model approach that has recently been announced as standard for public sector IT projects in Germany, as promising opportunity to help provide a unifying European software process model "umbrella". Based on the strengths of the VM-XT we suggest research directions for advanced support of software projects: (a) effective business value translation to engineering solutions that strengthens stakeholder collaboration, (b) process mapping that enables collaboration in projects that have to reconcile several process models, and (c) process "product lines" to capture the variability of software processes on domain and company levels and thus systematically help investigate best-practice approaches to software construction. We discuss these concepts, the contribution of the VM-XT, and conclude with next steps for research and validation.

Keywords

Software process improvement, V-Model XT, value-based software engineering, business value translation, software process mapping, process product lines.

1 Introduction

A common goal in software engineering is the construction of most valuable high-quality software products. However, concepts of high quality and value depend on the needs of involved project stakeholders to be able to define "business value" and translate it into software products [4, 5, 18, 20].

A subsequent ongoing challenge in modern software engineering is to establish a balance between product quality, functional range, development duration, and development cost. Software processes define development methodologies, independent of individual projects, regarding (a) process steps, performed in pre-defined sequences, (b) products, (c) activities and responsibilities, and (d) sets of methods and tools to support project work in order to improve project planning and execution.

A defined software process provides the basis to gather experience and data for systematic process improvement and can so lead to better products. However, the wide range of available software process approaches, which were often developed to fit specific application domains, project complexities, and project sizes, make it a challenge to compare best-practice experiences and select a ready-made process for any given project context. Therefore, many organizations customize their software processes according to their individual requirements based on well-known process models, e.g., the *System Engineering Method* (SEM) of Siemens PSE (Austria)[1] and *Promet* (IMG, Switzerland) [16].

However, if the underlying model needs to be changed, it is often very difficult to propagate these modifications to the models customized from the underlying model. Managers of individual projects and customers – especially in the public sector – may have a stake in the application of a specific software process model; in a multi-company project this often means to reconcile several process models to define an effective process model for the project at hand. These process diversifications and wide range of requirements fuel a need for a means to assess the comparability and compatibility of two or more software process models.

A first step can be to find out, to what extent two process models fit together in their current form; a next step could be to find a meta-model that presents a unified process model to which existing process models can be mapped. On European level this approach allows to compare and use process models, which implement national regulations and/or domain-specific needs, under a common umbrella. Such an approach for unification that allows systematic diversification is in line with the EU Lisbon declaration[2] goals to support European knowledge management in the area of software processes: software process models contain knowledge on how to make successful products; they can be used for knowledge acquisition and transfer, e.g., by providing structure to e-learning approaches.

Such an umbrella could be derived from the V-Model XT (VM-XT) [17, 23], a very flexible new software engineering process model that covers the whole life-cycle within a framework of IT solutions (systems engineering). After more than a decade of experience with previous releases [22], the VM-XT was released by the German Ministry of Interior in February 2005 as mandatory process model for public-sector IT projects in Germany. The process model is supported with a set of open source tools.

The VM-XT concentrates on the technical software engineering process with special attention to the "call for tender" as preliminary process step and an enhanced involvement of the customers within the project course. Additionally, the VM-XT allows tailoring individual approaches according to different project types (including the implementation and improvement of software processes) and combines systematic process guidance with more flexibility than most other software process models: the VM-XT provides a framework for software development, with the potential to grow into a European software process model.

Based on these strengths of the VM-XT we suggest three research directions for advanced support of software projects.

1. *Business value translation that strengthens stakeholder collaboration:* Economically successful projects are based on understanding stakeholder value propositions [4] and translating them into IT requirements and engineering solutions (see Figure 1). This remains a challenge, especially in IT projects with a public "call-for-tender". The VM-XT pays special attention to this preliminary phase and integrates success-critical customer contributions. Research can enhance the technical focus of the VM-XT with better integration of input from business process models, such as *Promet* [16]. Another challenge of value creation is to realign requirements and engineering solutions that evolve concurrently during the course of the project with proper synchronization mechanisms.

[1] http://www.pse.siemens.at

[2] http://europa.eu.int/eur-lex/lex/LexUriServ/LexUriServ.do?uri=CELEX:52003PC0509:EN:HTML

While software process models propose synchronization mechanisms that work well for in-house projects, collaboration in distributed projects, potentially across several companies and countries, needs more advanced process support.

2. *Process mapping that enables collaboration in projects that have to reconcile several process models.* VM-XT enables tailoring of particular process modules during the project course, e.g., adding individual process modules. This flexibility eases mapping other process models to VM-XT (e.g., for organizations, which want to use their in-house process model to bid for a public-sector IT project in Germany and have to show compliance with the VM-XT). Further, VM-XT can be used as common representation for several other process models and support more efficient mapping between any of these models (e.g., for process managers in multi-company projects, who have to derive a common process model that fits the project partners' needs and competences); in many cases today a difficult task.
3. *Process "product lines"* to capture the variability of software processes on domain and company levels and thus systematically investigate best-practice approaches to software construction. Many projects have to consider domain-specific business requirements, e.g., dependable systems, financial transaction services, security-related applications, embedded systems [12]. Consequently process managers see a strong need to describe domain-specific software processes (see also approaches in CMMI [7, 8, 13] and SPICE [9, 11, 15, 21]). We propose a process "product-line" approach [19] that uses VM-XT as basis for a software process meta model, and allows deriving process models that consider regulations from business domains, national standards, and company procedures. Systematically capturing process variability with such a product line approach opens up the chance for context-specific empirical investigation of best-practice approaches to software construction

The reminder of this paper is structured as follows. Section 2 describes basics of the VM-XT. Section 3 focuses on the contribution of VM-XT to business value translation. Section 4 outlines mapping and model translation approaches to support collaboration in projects that have to reconcile several process models. Section 5 describes enhancements to product lines to define process "product lines" that capture the variability of software processes on domain and company levels. Section 6 concludes and outlines next steps for further research.

2 The V-Model XT Process Framework

The V-Model XT (VM-XT) [23], successor of the V-Model 97 [22], is a software development framework for planning and execution of software processes. XT stands for *eXtreme Tailoring* due to the flexible customizing ability of the process framework to application needs, project types, and project complexity. In comparison to previous releases [22], new features in VM-XT cover regulations for hardware development, logistics, project management, and process improvement. Furthermore, VM-XT supports a preliminary call for tender as a pre-condition for the technical product development process, including bidding scenarios for contractors and customers. VM-XT assists three different project types: (a) system development from customer point of view, (b) system development from contractor point of view, and (c) implementation and maintenance of the organization-specific software process including continuous product and process improvement. The latter project type includes further development, extensions, and software process improvement of VM-XT as well as implementation and customizing approaches for specific organizations.

In addition to the three different project types, the VM-XT framework [23] describes a set of 99 products, 18 process modules, 18 decision points, and 7 project operation strategies:

(a) *Products* are results of activities, performed by defined roles, who apply a set of methods and techniques. Many of these products are supported by document templates for project application.

(b) *Process Modules (PMs)* include a set of products (product-centric approach), corresponding activities, and basic method suggestions. Project operation strategies enable customizing and tailoring of process *modules to achieve appropriate process course.* Some basic process modules are mandatory for all project types (VM-XT core components); others are optional depending on the application domain and the type of application.

(c) *Decision Points (DPs)* are related to a subset of products and represent the state of treatment

and the state of consistency.

(d) Finally, a *Project Operation Strategy (POS)* is defined as a sequence of decision points for project course (including repetition, cycles, and skipping). According to individual requirements of the organization, project operation strategies may be customized, e.g., regarding incremental, agile, component-based development as well as migration and maintenance purposes.

This set of concepts is the basic process model, which can be adjusted to the business domain and project types. Therefore, there are three possibilities for model application: (a) immediate support for project course without change of the basic concept; (b) application after tailoring according to several project criteria within the range of optional process modules, and (c) extension and customizing of the basic model according to individual needs using an XML-based editor, provided by the developers of VM-XT as an open source tool.

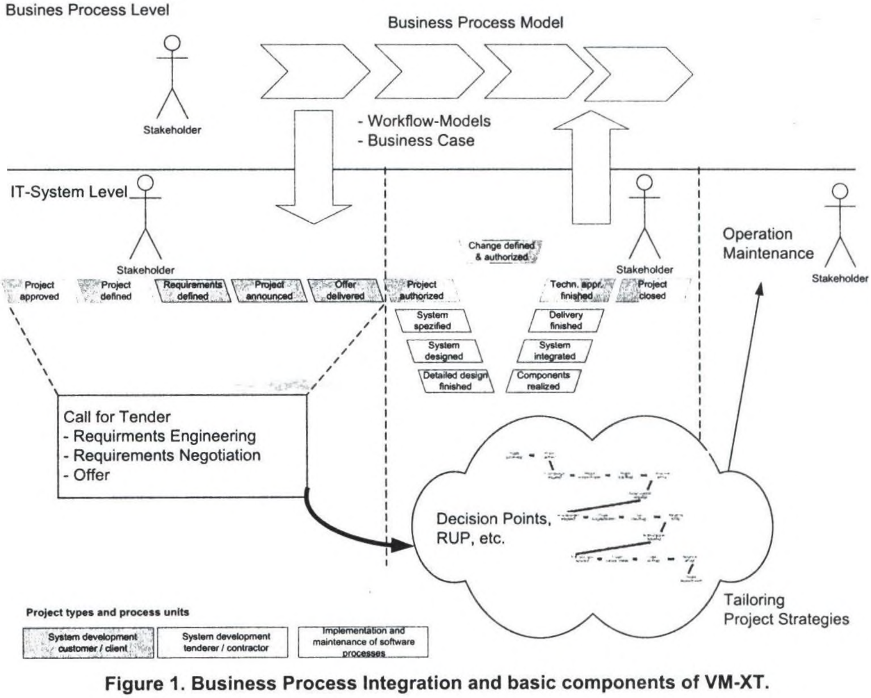


Figure 1. Business Process Integration and basic components of VM-XT.

Figure 1 displays the basic concepts of VM-XT and its integration with business processes, including the call for tender, regarding requirements engineering and negotiation, and offer scenarios. VM-XT is embedded within the IT-system layer, which consists of three steps: (a) call for tender as preliminary process for public domain customers, (b) the technical realization, and (c) the operation and maintenance phase. The three VM-XT project types and their involvement in the corresponding process modules can be identified as shaded marks in the process model. The software process schematically represents in form of the "V" the descending analytical development branch and the ascending synthetic development branch.

These horizontal interrelationships depict different views, e.g., implementation view (at the bottom level), architectural view, and user view (at the top level). Depending on the individual software processes of an organization, the schematic V-approach may be used as actual project process or be replaced with an alternative software process, e.g., the Rational Unified Process (RUP) [14] or some company-specific software process model. Nevertheless, to meet the requirements of public domain customers and to determine strengths, weaknesses, and possible extensions of the process applied one must be able to compare the VM-XT and the alternative software process model [17].

The following sections elaborate need from project practice for process model support: (a) the translation of stakeholder value propositions to IT requirements; (b) mapping approaches to compare software process models, and (c) process "product lines" for systematic customization and evolution of process frameworks such as the VM-XT.

3 Business Value Translation And Stakeholder Collaboration

A key element of *value-based software engineering (VBSE)* [4, 5] is to identify success-critical project stakeholders, to elicit and negotiate their main value propositions, and to translate them into IT requirements and engineering solutions. Software process models usually concentrate on providing a framework for project organization. Roles in the software process define responsibilities and competence to enact processes. However, there is often little consideration on how to align the needs and competence of the stakeholders, who give life to these roles, with the role(s) they should play [6]. Especially the project principal or customer is often represented by persons, who are not routinely involved in developing software systems. Thus it takes extra effort to help them play an active and constructive role; particularly at project inception in the context of a "call-for-tender" that determines the opportunities of value creation in this type of project.

The VM-XT is one of the very few process models that explicitly deals with the responsibilities of the customer and thus contributes to better include this crucial stakeholder in the project. However, we see the contribution of the VM-XT as a start to better address VBSE considerations; Figure 1 shows the VM-XT in the context of value creation: in a business environment the value that can come from a software development project is generated mostly from the application of the project results on the business process level. Thus, there is a research need to enhance the mainly technical focus of the VM-XT with better integration of input from the business process level, e.g., from results of *Promet,* a well-known open-source business process model [16]. The business value, expressed in business process modeling, needs to be translated to IT requirements that are the project basis, e.g., in a "call for tender". The next step is to develop an engineering solution that fulfills the IT requirements based on a software process tailored to the project needs. Finally, the operation of the IT system supports value creation due to new and improved business processes.

An important aspect of most effective value creation is that the stakeholders understand enough of the software development process options to describe requirements in a way that can be aligned with effective engineering solutions. This needs some way of communicating the scope of engineering solution options during requirements elicitation and negotiation [19]. Better integration of software process models and business process models can help to open up the currently typical one-way communication into an overall more effective two-way communication between stakeholders that understand the value of business requirements and engineering solutions.

Another challenge of value creation is to realign business requirements and engineering solutions that often evolve concurrently during the course of the project. For reasons such as technical development difficulties, finding inconsistencies of requirements, and changes in the real world, changes to requirements and development plans are common in software projects. Synchronization mechanisms during the project help to assess the need for realignment and provide concrete steps to re-negotiate for more consistent products and plans. Software process models propose synchronization mechanisms that work well for in-house projects, such as simple decision points and review procedures. However, collaboration in distributed projects, potentially across several companies and countries, needs more advanced process support, (e.g., effective tracing for requirements management and pro-active communication between different stakeholders to avoid risky "hidden consequences", such as interfering project course, with respect to possible changes of their interests) to deal with concurrent work and changes on different levels. The application of stakeholder-related principles from VBSE can help to address these value creation and collaboration aspects in software process models. Empirical analysis of collaboration risks in (distributed, multi-company) software projects can describe the gap between needs and currently available solutions in practice. The VM-XT provides an advanced software process framework to anchor this research.

4 Software Process Mapping

The major goal of European industrial practice is the production of high-value software in diversified application domains. To meet this goal several different process models were developed according to domain specific and company related requirements; these approaches led to a huge number of process model variants. Usually, an organizational unit selects an appropriate software process, which fits best to its business at a certain point in time, and then adapts this process according to the local requirements. As a consequence, a wide range of different software process models, architectures, and methodologies are in use [2, 3]. These diverse models are hard to compare (e.g., regarding structure and semantics), which hinders efficient collaboration and software process improvement on a European level. From an academic point of view, systematic mapping of process variants is a prerequisite for a conceptual unification and systematic construction of process models according to predefined needs.

Process managers, especially in the growing segment of multi-company software development, see a need for approaches that support stakeholder collaboration, systematic mapping, and transformation of processes to improve their leverage in software process improvement. Process mapping enables collaboration in projects that have to reconcile several process models. VM-XT enables tailoring of particular process modules during the project course, e.g., adding individual process modules. This flexibility eases mapping other process models to VM-XT (e.g., for organizations, which want to use an in-house process model and have to show compliance with the VM-XT).

Further, VM-XT can act as a common representation for several other process models and support more efficient mapping between any of these models (e.g., for process managers in multi-company projects, who have to derive a common process model that fits the project partners' needs and competences); in many cases today a time consuming task that comes with many new projects.

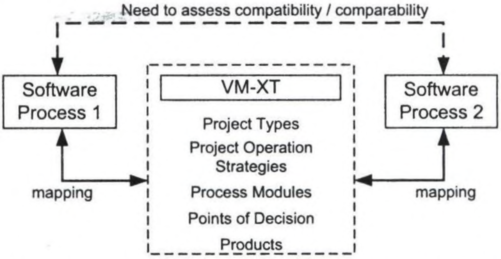


Figure 2: Basic Process Mapping Approach for comparability and compatibility.

Mapping to compare software processes requires the investigation of the specific software process model according to components, internal structure, products, activities, etc.

Figure 2 depicts the basic schema for software process mapping, based on VM-XT as bridging intermediate process model. Following this mapping schema, the analysis of a software process consists of two phases. The first phase is to map each software process to the VM-XT in 3 steps: (a) *structure analysis:* identify basic components of a process model, (b) *initial mapping:* compare this structural information to VM-XT elements according to syntactic and semantic elements in order to find similar elements, completely different elements, and mixed elements; and (c) *refine mapping:* continue to analyze elements with status "mixed" on a finer level of detail until the complete software process has been successfully mapped or a defined level of detail is reached. The result set contains a set of matching process elements and open issues for further investigation. This analysis allows determining the degree of compliance of the software process with the VM-XT structure.

The second phase analyzes the compatibility of the two software processes and builds on the results of the first phase. For each structural element the process manager has the following options: Choose one of the elements of the process models or to choose an element of the VM-XT. Using the VM-XT as intermediate model has two major benefits: (a) the VM-XT provides a consistent structure for the comparison and (b) if the manager needs to compare many process models over time the effort for structural analysis grows in a linear rather than exponential way. Following this mapping approach, small and medium enterprises (SMEs) can adapt their software processes according to VM-XT more easily in order to achieve competitiveness (e.g., to bid for public IT projects in Germany) and to produce software in collaboration with large companies.

Taking a broader view on software process improvement, quality management systems often provide a company-wide framework for continuous process improvement, e.g., CMMI [7, 8, 13], SPICE (ISO/IEC 15504) [11, 15, 21], and ISO 9000 [10]. VM-XT and its derivate process models may be embedded within a quality management system as a central business process for software construction. Because of these integration possibilities, VM-XT fits well to quality management systems, including bidding, parts of purchasing processes, and commissioning.

Initial research using this process mapping approach identified the challenges of how best to represent the mapping results and whether a simple model can cover all major process models. These difficulties point out the broader need for a consistent description of software engineering and software process semantics, e.g., building up an ontology that helps to converge the semantic variants and describe differences in meaning between process descriptions.

5 Extension of the VM-XT Framework to Process "Product Lines"

Many projects have to consider domain-specific business requirements, e.g., dependable systems, financial transaction services, security-related applications, and embedded systems [12]. Consequently process managers see a strong need to describe domain-specific software processes (see also approaches in CMMI [7, 8, 13]).

There are several dimensions of formal requirements, such as regulations from business domains, national standards, and company procedures, which need tailored software process models. Currently, such tailoring occurs typically in an ad hoc fashion. Typical follow-up problems from this approach are limitations of comparability, high effort for maintenance, and architectural deterioration, very similar to ad hoc reuse in software engineering. The flexible structure of the VM-XT makes it well suited to derive a wide range of software process variants. For future software processes this would be a clear common source to start from similar to a software library.

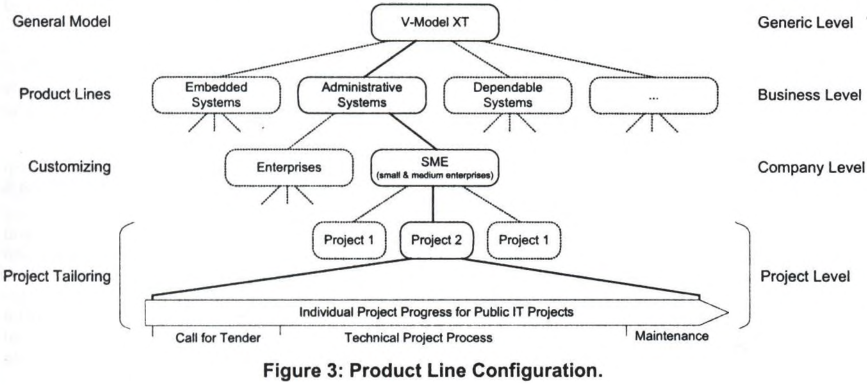


Figure 3: Product Line Configuration.

A major development step would be to take up the product line paradigm from software engineering to software process modeling. This approach allows capturing the variability of software processes on domain and company levels. Such a framework can provide decision making processes for project and process managers to derive an appropriate software process model to achieve robust and high-quality projects. Figure 3 illustrates decision levels of a product-line approach based on VM-XT. The model consists of 4 levels: (a) *generic level:* the basic VM-XT, (b) *business level*: describing VM-XT variants for various application domains, (c) *company level:* representing customizing of VM-XT approaches; and (d) *project level:* including tailoring for specific projects within an enterprise. Process product lines – a set of software processes derived from the basic process model – promise a repeatable and efficient configuration of process models with respect to specific business domains and application areas, such as dependable systems, embedded systems, and administrative systems [1]. Further benefits come from integration of state-of-the-art techniques and best practices due to a comprehensive tools and method support.

We see benefits from an extension of VM-XT to product lines in 3 ways: such an extension (a) allows to encompass different application domains, (b) more systematic insight in the courses of projects, and (c) a larger set of pre-defined methods and tools. Project and process managers may choose from a pool of application specific VM-XT variants (broader view) for special purposes, e.g., web applications, embedded systems (comparable to the Siemens PSE SEM process family[3]). Therefore preliminary customizing the basic process for individual application is necessary. This will lead to a set of process model approaches including a domain specific selection of process modules (PMs) of the basic VM-XT. The second way allows a deeper insight into software process enactment, i.e., the software process enhancement towards a more detailed view on specific process modules. This approach enables a better guidance through the software development process. The third way covers method and tool support of VM-XT. The current version of VM-XT provides a small set of methods and tools for application. To follow best-practice approaches, project managers need tool and method support including guidelines which individual method or tool to select. Product-line guidelines support the systematic creation of processes and tool support as they provide the ground to define a "market" for these products; further, they help project and process mangers to select appropriate methods and tools from the "market".

In the basic VM-XT framework, software process improvement approaches are derived from the underlying model (including continuous improvement with respect to further development and improvement of the VM-XT basic model). The VM-XT development team provides a VM-XT specific editor to customize the basic definitions of the software process model according to the individual requirements of an organization. Additionally, customized software process models may be extended with respect to special needs of the applicants following state-of-the-art approaches and new methods and techniques. Extensions may include additional modules, products, decision points, new project operation strategies, additional methods and tools. Therefore, this concept allows the construction of VM-XT variants with respect to different application and business domains, other types of software processes, and process improvement approaches. With such a product line approach it is possible to gather information on decisions in customizing a process model and to systematically evaluate and learn from them.

Due to their potential for learning from projects in similar domains and business areas, product lines promise repeatable and efficient configuration of process models. The product line approach supports (a) systematic investigation of best-practice approaches to software construction and at the same time (b) provides a structure to document process variability and capture lessons learned.

Best-practice approaches should be used to extend the basic model and all derived business domain specific versions of VM-XT product line approaches. Because of the basic model structure (XML data that describes process elements), experiences and empirical evaluation data can be used for training proposes, e.g., to set up an e-learning strategy. Product lines allow the consideration of national and domain-specific regulations and experiences. Therefore, VM-XT and the proposed product line concept provide a promising direction towards a broader usage of VM-XT on a European level.

Research is needed on how to best structure the product line refinement concept: what should be invariants, what are the variants. An empirical analysis can determine types of software projects that would be best addressed with the VM-XT product line approach. While the product line can provide

[3] http://www.pse.siemens.at

the structure of a derived software process model, a challenge can be that the derived processes still exhibit too many degrees of freedom to effectively support a project manager. The support for process enactment with templates and examples needs interfaces to feedback from project practice. A good example for successful process support on this level is the Siemens Austria PSE *stdSEM process family*, which consists of 6 major processes for project types, and examples for tailoring to support the project manager at work.

6 Summary and Outlook

The mandatory nature of the VM-XT for national public domain IT-Projects sets a significant impulse to the German software development scene: As consequence, all organizations have to meet the requirements of this process model. As Germany is an important industrial partner for many European countries, and as a market specifically targeted by companies in EU New Member States, the VM-XT is likely to become a leading software process model with the opportunity to grow to a European software process approach. The current version of VM-XT is available in German but will also be available in English in the second half of 2005 to help spread VM-XT across Europe.

During a workshop at the Vienna University of Technology in April 2005 researchers from several European countries found in the VM-XT potential for proposing a common European process model framework to support systematic software development and to build up empirical know-how on best practices in Europe. The structure of the new software process and the simple implementation within a company enable SMEs (small and medium enterprises) to better meet competitive and demanding bidding requirements in the software market across European national boundaries, e.g., for public projects in Germany.

In April 2005 the Austrian Computer Society[4] established a special interest group on "Software Process" with a focus group on "Process Improvement with the V-Model XT" in cooperation with the Vienna University of Technology and the University of Linz. A major goal of the special interest group is the application and advancement of process models compatible with the VM-XT framework according to research topics, practical application, including knowledge acquisition and transfer. Critical success factors for software process improvement are (a) balancing stakeholder interests by translating of business value into appropriate software engineering solutions, (b) integration of best-practice approaches, and (c) development of product lines to software processes to capture the variability of software processes on domain and company levels and thus systematically investigate best-practice approaches to software construction. A combination of the top-down product line approach with the bottom-up input from currently used processes (using the mapping and synchronization mechanisms) can get the most from both sides: systematic derivation of the software process framework combined with practical know-how from concrete project work. The concepts of the new VM-XT support these critical success factors as evolutionary parts of the VM-XT framework.

The strategic positioning of the VM-XT framework and its strengths to support process mapping and derivation hold the opportunity to set important impulses to software process improvement on a European scale, especially for the integration of software development efforts in the EU New Member States, and thus can make a significant contribution to realize major goals of the EU Lisbon declaration.

Acknowledgements

We want to thank the participants of the VM-XT research strategy workshop carried out at Vienna University of Technology in April 2005 and our research partners at Czech Technical University for their valuable discussion and comments to the research agenda presented in this paper.

[4] http://www.ocg.at

Literature

1. Abran A., Moore J.W. (eds.): *Guide to the Software Engineering Body of Knowledge*, IEEE, 2004.
2. Bernus P., Mertins K., Schmidt G.: *Handbook on Architectures of Information Systems*, Springer, 1998.
3. Bernus P., Nemes L., Schmidt G.: *Handbook on Enterprise Architectures*, Springer, 2003.
4. Biffl S., Aurum A., Boehm B., Erdogmus H., Grünbacher P. (eds.): *Value-based Software Engineering*, Springer Verlag, Heidelberg, 2005.
5. Boehm B.W.: *Value-Based Software Engineering*. Software Engineering Notes, 28(2), 2003.
6. Standish Group International: *Chaos Report*, 2004.
7. Chrissis M.B., Konrad M., Shrum S.: *Capability Maturity Model Integration*, Addison Wesley, 2003.
8. *Capability Maturity Model Integration*, Carnegie Mellon University, Software Engineering Institute, Internet: http://www.sei.cmu.edu/cmmi/, 2005.
9. El-Emam K., Drouin J.-N., Melo W.: *Software Process Improvement and Capability dEtermination*, IEEE, 1998.
10. ISO 900x:2000, International Organization for Standardization, 2000.
11. ISO/IEC 15504-x, SPICE Part 1-7, International Organization for Standardization.
12. Klemen M., Biffl S.: *Economic Aspects and Needs in IT-Security Risk Management for SMEs*, Proc. of the 6th Int. Workshop on Economics Driven Software Engineering Research (EDSER-6) at the Int. Conf. SE, Edinburgh, Scotland, 2004.
13. Kneuper R.: *Capability Maturity Model Integration*, dpunkt, 2003.
14. Kruchten P.: *The Rational Unified Process – An Introduction*, 3rd Edition, Pearson, 2004.
15. Loon H van.: *ISO/IEC15504-Process Assessment Standard*, Springer, 2004.
16. Ŏsterle H., Winter R.: *Business Engineering*, Springer, 2003.
17. Rausch A., Bergner K., Höhn R., Höppner S., Wetzel H.: *Das V-Modell XT, Grundlagen, Anwendung, Springer*, 2005.
18. Roel W.: Requirements Engineering, Wiley, 1995.
19. Schmid K., Biffl S.: *Systematic Management of Software Product Lines*, Software Process Improvement and Practice, 2004.
20. Sharp H., Finkelstein A., Galal G.: *Stakeholder Identification in the Requirements Engineering Process*, 10th Int. Workshop on Databases & Expert Systems application, IEEE Computer Society Press, p387-391, 1998.
21. *Software Process Improvement and Capability dEtermination*, Griffith University, Software Quality Institute, Australia, Internet: http://www.sqi.gu.edu.au/spice/, 2005.
22. V-Modell 1997, http://www.v-modell.iabg.de.
23. V-Modell XT, http://www.v-modell-xt.de, 2005.

Author CVs

Stefan Biffl

Stefan Biffl is an associate professor of software engineering at the Institute of Software Technology and Interactive Systems, Vienna University of Technology. He received an MS and a PhD in computer science from the Vienna University of Technology and an MS in social and economic sciences from the University of Vienna. He joined Vienna University of Technology in 1989. In 2001 he spent one year as researcher at the Fraunhofer Institute of Experimental Software Engineering, focusing on quality management and empirical software engineering. He has authored more than 50 publications in the areas of value-based software engineering, empirical software engineering, quality management, and software management. As consultant he was in charge of quality assurance and software process improvement for large IT projects in the public sector. He is a member of the ACM, IEEE, the Austrian Computer Society, the International Software Engineering Research Network, and the IFIP Technical Committee on Software Engineering.

Dietmar Winkler

Dietmar Winkler received an MS in Computer Science from Vienna University of Technology. He is researcher und lecturer at Vienna University of Technology, Institute of Software Technology and Interactive Systems and lecturer at University of Applied Sciences Vorarlberg. His research interests include software engineering and software processes, quality management and quality assurance, empirical software engineering and software inspection. Additionally, he is software engineering and quality management consultant in the automotive business area in industrial environment.

Reinhard Höhn

Reinhard Höhn studied mathematics at Wolfgang v. Goethe University Frankfurt, Germany. Afterwards he worked in the area of system and process analysis and software engineering for power plants at Lahmeyer International. He was the head of several IT projects, e.g., Toll Collect Austria, consultant for bidding scenarios and trainer for knowledge management methods and processes. Today he is working as product manager for knowledge management systems, methods and software processes with KMA Knowledge Management Associates GmbH, Vienna, Austria. He is a member of GI, the special interest group WI-VM of OCG.

Herbert Wetzel

Herbert Wetzel studied economics at Augsburg University, Augsburg, Germany. Afterwards he worked for Siemens in Munich and Augsburg as an internal consultant and project manager for about 14 years. He was the head of different projects most of them in context with book-keeping, cost accounting and SAP implementation. Since 11 years he is working for IMG - the Information Management Group, St. Gallen, Switzerland, a spin off of the University of St. Gallen. As a project leader and coach he took part in many projects in context with process improvement or SAP implementation for customers like Bayerische Landesbank (D), UBS (CH), Bank Austria (AT), DaimlerChrysler (D), VIW (AT) and Hirschmann (AT). Today he is Vice President of IMG, head of the Business Unit Methods & Tools of IMG and managing director of IMG Austria. He is a representative member of the special interest group for software processes of the German *Gesellschaft für Informatik*.

Tailoring Extreme Programming for Legacy Systems: Lessons Learned

Martin McAnallen
MIT Systems,
Armagh
Northern Ireland,
BT60 3JH
MMcAnallen@MiTSystems.co.uk

Gerry Coleman
Department of Computing
Dundalk Institute of Technology
Dundalk
Ireland
gerry.coleman@dkit.ie

Abstract

Updating and maintaining legacy systems creates significant challenges for software developers. Modifying legacy applications can be a time-consuming process which is fraught with architectural and code minefields. In many instances, the same developers, because of their specialist knowledge, and the same processes have been used to improve these systems over an extended period of time. Introducing new practices into such an environment presents problems, on both the human and the technological level. This paper reports on the experience of implementing a scaled-down version of eXtreme Programming (XP) into a small manufacturing company. How the difficulties, in creating the climate for such an implementation, were overcome, and the resulting benefits of the experiment are reported on. Finally, the conclusions and lessons learned offer support and advice to others who may also be considering such an approach.

Keywords: *extreme programming, software process, legacy systems, management*

1. Introduction

Legacy systems are systems that have outlived their original user requirements but have remained in operation long enough to be substantially modified until the system no longer resembles that which was first developed. The maintenance process continues because the system functions correctly but, in reality, a large percentage of the code is obsolete and the remainder frequently works in ways that are not fully understood by those maintaining it. According to Robertson [13], IS organisations are struggling to respond to demands for new system features on existing systems whilst simultaneously being expected to manage new technologies. The challenges to those charged with maintaining legacy systems include developing new functionality and enhancements often without a clear understanding of how the system works. In addition, most companies working with older legacy systems tend to follow traditional 'waterfall' models of software development. Together, these factors make the development of enhancements to legacy systems slow and cumbersome.

Legacy systems can be found in a range of industries but especially in established companies. The enhancements and modifications to functionality are usually carried out to accommodate a business change such as the arrival of new customers. For companies in possession of legacy systems, an inability to react to business changes such as these, because of a time-consuming process of feature upgrade, can often lead to lost revenue opportunities.

In this paper we describe an experiment carried out within the Irish office of 'Bayside', an American manufacturing company. The company maintain a legacy system running on IBM AS400s. The maintenance team are experienced programmers, who modify and enhance the system in response to users' requests, usually generated because of a new or changing business requirement. The company have the required technical ability to support the system but the pace of development can be slow and follows a very structured approach. Also some developments require specific programmers because of their knowledge of the system. The experiment documented here describes how a tailored version of the eXtreme Programming (XP) methodology was introduced into development in an attempt to meet users' needs earlier and reduce the cost of maintenance.

2. Extreme Programming (XP) and its use in the study

Though the agile movement has made significant inroads into software development departments, it is really eXtreme Programming [3] that is by far the most discussed agile method within the literature and the one with the most widespread industry base. Developed in the late 1990s, XP has 12 associated practices (**Table 1**).

Table 1 The set of 12 XP Practices

XP Practice	Description
Planning game	Used to determine the content and scope of system releases
Small releases	Release working versions of the system on short cycles
System metaphor	The collective vision of how the system works
Simple design	Produce the simplest design possible to satisfy requirements
Test first development	Tests are written, and must run successfully, prior to the continued development of the code
Refactoring	Restructuring of the system to simplify, reduce duplication or aid communication
Pair programming	All production code is written by two developers at the same machine
Collective ownership	Team owns system, so all are empowered to make changes
Continuous integration	Build and integrate the system many times daily
40-Hour week	Limit overtime to reduce tiredness and potential mistakes
On-site customer	Ensure that a customer representative is available at all times to answer questions
Coding standards	Agree conventions at the outset and ensure programmer adherence

With its now widespread popularity, Glass believes that XP "has evolved into a near religion" [6]. A number of authors including, [7], [8], [11], [12], and [14], have reported on how they have deployed XP in their own organisations. All of these articles report on the success of XP in the particular experiments. However a consistent trend in industrial XP usage is the difficulty in finding evidence of the implementation of all 12 XP practices on a single project or within a single organisation. Aveling [1], reports on an analysis of a number of case studies of XP implementations which examined the success rates in experiments using the methodology. His results suggest that, "partial adoption of XP is more common than full adoption". He reports that the practices that were most difficult to adopt were *system metaphor* and those requiring significant customer input, *on-site customer*, *planning game* and *small releases*. He feels however, that his results show that it is possible to deviate from complete XP and still enjoy the benefits afforded by the method.

McBreen [9], initially used the *test-first development* practice from XP, before experimenting with other XP practices, "not as a lead in to adopting XP, but as a useful process improvement step". On two separate projects. Murru et.al [11], used XP practices with a varying degree of success. The first project fully used 7 of the 12 XP practices whilst the subsequent project used 9 of the 12. The major differences in the second project were the use of the *planning game* and *coding standards*. The addition of these practices helped address the deficiencies inherent in the first project. The practices not deployed or partially deployed on project 2 were *system metaphor*, *continuous integration* and *coding standards*. Rasmusson [12], in his study, also fully utilised 9 of the 12 XP practices but in this instance it was *system metaphor*, *test-first development* and *on-site customer* that were not fully implemented.

What is shown from the above results is that companies are tailoring the XP method to suit their own particular environment. This is consistent with process models and process improvement generally in that certain contextual factors may influence what aspects of the process are suitable and what are not. In this study we attempted to see if eXtreme Programming could be used to speed the process of delivering system enhancements and improve the maintenance capability of the legacy team. In the environment in which this application is being used, changes to system functionality are usually required due to new business requirements and any improvement in the speed and quality of delivery has potential business benefit. It was felt at the outset that a total switch to pure agile programming practices would be both unsuitable to the application and too great a change to a company who were already successfully supporting this application. We believed that certain XP practices could provide real improvements in speed and quality of delivery without trying to revolutionise the company's existing development practices. A tailored version of XP was therefore introduced.

On initial examination, it was found that a couple of the practices that XP promotes were already part of the company culture. These are highlighted in **Table 2**.

Table 2 XP Practices in Current Use

XP Practice	Current Use
40-Hour week	The company culture already promoted a 40-hour week amongst its staff in all departments including software development.
Coding standards	Coding standards were found to be well documented and well used. This was due to a strong department manager who had many years experience and who enforced coding standards, documentation and revision control of software.

The practices detailed in **Table 3** were identified as **not** currently being used and therefore of being of potential benefit.

Table 3 Potential Benefits of using XP

XP Practice	Potential Benefit
Small releases	To make significant changes to the "internals" of legacy systems is futile because the internals already work and are invisible to the user. The greatest returns can be achieved from taking requests for change and prioritising this work into small releases where the user and company can see an immediate benefit. This analysis is supported by [10] who conclude there are economic benefits to splitting the project into small releases where the use of XP permits it.
Test-first development	In this particular environment testing had traditionally been carried out in line with waterfall approaches. It was believed that, because maintenance of a legacy system involved small incremental enhancements, just like XP small releases, the test-first philosophy of XP could suit. It was also felt this testing

	method would focus the developers more on what they were trying to achieve.
Pair programming	It was felt this technique would have two effects. Firstly it would potentially have the benefit of reducing defects and increase code quality but it was also felt if the more junior members were paired with more senior members then it would increase both their technical knowledge and their knowledge of the system.
Collective ownership	It was believed this would be important to the company as they had a dependence on key programmers' knowledge of the system. Furthermore, it was known that one of the COBOL programmers plans to take early retirement and that when he leaves knowledge of certain areas of the system will leave with him. Collective ownership could mitigate this risk.
On-site customer	Because this was an in-house application, and the end-user was already on-site, utilising them more during the development process could improve the quality of the modification and increase both communication and the overall relationship between the development team and the users.

3. Selecting the pilot project

Like most legacy systems this application, which was originally purchased by Bayside's American parent in the mid 1990's, is now maintained by programmers who were not involved in the original specification, design or development. Bayside took ownership of the source code in the late 1990's when the original development company went out of business. The original architects are therefore no longer involved with the system and, even if they returned, may not now even recognise it. The maintenance team supporting the system are split between the USA and Ireland and their challenge is to work on a system they do not fully understand.

The legacy application serves order entry, manufacturing, warehousing and shipping. The application is written in a mix of COBOL and RPG and it currently runs on an IBM iSeries Server although it was originally developed on an IBM System38 and over half the source code, and database physical files, were developed on a System38 and migrated to AS400 and then iSeries.

The project selected was a typical modification. It was chosen because it involved a change to a customer order entry screen due to a new business requirement, a typical legacy system enhancement and, if successful, would encourage more widespread acceptance and usage of the experimental techniques employed. Also, because it was an order entry screen, it would involve close work with the order entry team who were the development team's customer. This would also entail getting user support for the collaborative XP techniques used and buy-in for subsequent changes to working practices.

4. Preparing the Ground for XP

In preparation for XP's introduction presentations were made initially to upper management. Two particular managers were targeted as they came from a technical background and were seen as change agents who could encourage acceptance amongst the engineers. The presentations specifically highlighted the spirit of XP and, while all 12 XP elements were outlined, particular emphasis was given to the practices that were felt would have most benefit to the company. It was also emphasised that it was not proposed to revolutionise the development process but merely to evolve it. Once higher-level management had bought into the idea, and supported it, discussions then ensued with the head of development towards agreeing which practices could be experimented with and which project could be used as a pilot. Once this was finalised we then moved into a period of training the engineers. Engineers were presented with an overview of agile methodologies, and specifically XP. Attention was then paid to the practices to be used in the trial, what the aim of each practice was, and how we proposed to use it.

Opposition to our approach was found at both management and engineering level. Initially it was dismissed as "this week's craze" but, as it became a reality, those that opposed it most were the more established and experienced programmers. Some of this hostility stemmed from the belief that the change was being initiated in response to a perceived lack of technical ability to maintain the system. Because of these concerns, we held a final meeting in an attempt to address the fears of some of the group. During this session Cockburn's views on Agile Software Development [4] were presented to the group, pointing to how agile methods favour individuals and interactions above process. This had a positive effect and we followed this up by targeting one of the doubters in particular as, we believed, if we could convince him, we could convince the team. This approach was successful in getting the necessary commitment to commence the trial. Though residual misgivings remained amongst one or two of the team, we hoped that those would be assuaged through usage of the practices during the development period.

4.1. Facilitating XP and Knowledge Transfer

At the outset, we started with a review of how the development team in Ireland operated. There were only three team members and these were split between three small offices. Immediately we identified the need to pull the group together to facilitate tacit knowledge sharing and better communication. As Desouza noted, the major obstacle to knowledge sharing is not insufficient technology but ensuring people talk and share their individual know-how [5]. With this in mind, we tackled the physical space inhabited by the engineers. In one office we removed a partition wall and the door between this office and the next office was also removed. The desks were rearranged so that two members sat in one office and the third member sat in the second office but in clear view of the other team members so that oral, rather than telephone or e-mail, communication was fostered. The new arrangement was also purposely created so that if any user, acting as an on-site customer, came into the development area they would be visible to all members of the team and therefore any member could respond to their query and the others would be in audible range to hear any issues discussed or agreed.

An attempt was also made to make the project status more visible. The experiment was concerned with one particular project request so it was not appropriate within the time given to implement a full visible system of collecting and selecting requirements. We also had a limitation on wall space due to the rearrangement of the offices. What we implemented was the use of a simple white board. On this we placed information pertaining to the project name, customer, and date submitted, the owner, or developer responsible for the project, and a time for completion. It was a rudimentary system, which was all that was possible in the time allowed, but one that proved very popular with both prospective system users and developers.

5. Project Outcomes

On the whole, there was a very positive response to all of the changes made to work layout and practice. The tailored XP process generated major improvement in the development effort and most of the techniques, though not all, proved successful.

5.1. Analysis of the Tailored Model

5.1.1. Small Releases

Approach taken

In the experiment it was decided to break the project into distinct releases. The first release was the modification of the order entry screen and the second was the modification of the system functionality.

Outcome

This was found to be beneficial as the team could focus with the user and deliver the change to the screen quickly and secondly the team could finish the additional functionality that was invisible to the user. After the experiment we received some strong positive feedback from the users that they wanted to continue to see their requirements being broken into specific tasks with specific delivery times. The users requested that the company continue to develop in this way rather than hitherto, whereby they saw small changes being incorporated in a larger release which they had to wait some time for.

5.1.2. Simple Design

Approach taken

During the experiment it was found that this technique could not be implemented due to the intricate linkages within the existing system.

Outcome

A legacy system is one that is built then modified and re-modified with the result that the current architecture bears little resemblance to the original design. The programmers supporting it often do not have a clear understanding of how the entire system works. These factors lead to the need for careful consideration before modifying any part of the system. The simple design principle is more naturally suited to a "greenfield" project where there is no existing code to complicate the design process.

5.1.3. Test-first Development

Approach taken

Testing is an essential part of developing software regardless of which development model is used.

Outcome

The experiment did not allow for much testing however this approach was a new concept for the legacy developers and was found to have the claimed benefits of finding problems earlier. The writing of tests first focused the developers on what was required from the code. In the experience of the authors the new testing concept was a difficult one for the traditional legacy maintenance programmer to grasp and could be described as a paradigm shift in thinking but one that yielded significant results in terms of the developers understanding of the domain and the speed of delivery of code.

5.1.4. Pair Programming

Approach taken

Pair programming was tried both in Ireland and the US parent company but produced different results in each case.

Outcome

During the experiment, pair programming was conducted with the developers in Ireland and was found to be beneficial in the speed of delivering code and the reduction of errors. These developers who participated in the pair programming were aged around 25 to 30. Pair programming was also tried in the parent company but it ran into difficulties. There, the average age of the programmers, involved with the experiment, is slightly over 45 and these individuals have been writing this code for 20 to 30 years. These individuals found pair programming a very difficult and unnecessary practice. Their experience of the language and the system was such that pair programming did not improve their performance. The age of developer and experience of coding would be unique to legacy systems and raises interesting issues about the use of the pair programming technique.

5.1.5. Collective Ownership

Approach taken

The experiment highlighted the importance of introducing this practice into the organisation in the future.

Outcome

Bayside follow the traditional method of having one engineer responsible for each subsystem. The engineer is responsible for design, implementation and maintenance, and is, in other words, the "owner". Within the company this was highlighted as a problem when one engineer reached retirement age and another took early retirement. This created a gap in knowledge in the organisation and a skills deficit. Since the experiment the company has adopted collective responsibility and a corresponding process of "swapping roles".

5.1.6. On-site Customer

Approach taken

This was found to be one of the areas that the company had not been taking full advantage of despite having the development team and customer in the same location.

Outcome

As a result of the experiment the company has adopted a change in policy where the developers work more closely with the user. They have found this has had the effect of increasing the developers' knowledge of the problem domain and has led to a release of software that is a closer match to the users original requirement. The users also gave very positive feedback on this practice. Having commented that it was not always possible to document exactly what they wanted, which often meant a release some months later that did not meet their original intentions, they found major benefits in being able to discuss their needs directly with the developers on an ongoing basis.

6. Lessons learned/Conclusions

With regard to the practical implementation of agile methods in a legacy system environment it can be concluded that some of the agile methods work well for legacy systems and some do not but that moving to a hybrid model has some notable advantages over the original waterfall-style model.

In the case of Bayside, it was found that small releases and customer collaboration yielded benefits, both in speed of delivery, and improvements in user requirements being met. It was also found that the agile, test-first approach brought reduced delivery time and generated higher quality applications with fewer errors. The company also profited from the changes with regard to code ownership.

In practice, pair programming did not work particularly well with the older developers. This may be due to their age and experience profile. However, the younger developers embraced the concept and this proved very worthwhile in the Irish context.

On the downside, there were clear difficulties in attempting to use simple designs as the implications for making any modifications to the existing code were far-reaching. The inability to use simple design, naturally meant that the XP practice of refactoring could also not be used. Coding standards had previously been well defined prior to the experiment and other configuration management procedures were already in place so these elements of XP were not implemented. The issue of coding standards is a difficult one for legacy developers in that they may be attempting to introduce a new format onto a substantial existing code base. Therefore, implementing coding standards means that they would apply only to existing and future work. Whether they could be re-engineered into legacy code is ultimately context-dependent.

7. Further work

It is planned to adapt the hybrid model created during this pilot and to apply it to further development projects. While the project was successful it will still take some time for the new concepts to become part of daily life.

Further development will now be examined regarding the visibility of user requirements. A more suitable board and/or card system could be introduced, however some consideration would have to be given to how this could be made visible across both development sites. An electronic system, incorporating a virtual whiteboard may be more appropriate. XP's use of user stories may support the issue of visible requirements and this should in turn support further trials where some of the method's other practices, such as the planning game, can be tested.

8. References

1. Aveling, B., 2004, "XP LiteConsidered Harmful?", Proceedings of the 5th International Conference of Extreme Programming and Agile Processes in Software Engineering, Springer, LNCS 3092, 94-103.
2. Avison, D, Lau, F., Myers, M. and Nielsen, P., 1999, "Action Research", Communications of the ACM, January, Vo. 42, No. 1, 94-97.
3. Beck, K, 2000, *Extreme Programming Explained: Embrace Change*, Addison Wesley.
4. Cockburn, A., 2001, *Agile Software Development: Software Through People*, Addison Wesley.
5. Desouza, K., 2003, "Facilitating Tacit Knowledge Exchange", Communications of the ACM, Vol. 46, No. 6, June, 85-88.
6. Glass, R. 2001, "Extreme Programming: The Good, the Bad and the Bottom Line", IEEE Software, November/December, 111-112.
7. Grenning, J., "Launching Extreme Programming at a Process-Intensive Company", IEEE Software, Nov./Dec. 2001, pp. 27-33.
8. Grossman, F., Bergin, J., Leip, D, Merritt, S. and Gotel, O., 2004, "One XP Experience: Introducing Agile (XP) Software Development into a Culture that is Willing but not Ready", Proceedings of the 2004 conference of the Centre for Advanced Studies on Collaborative research, IBM Press, 242-254.
9. McBreen, P., 2000, "Applying the Lessons of eXtreme Programming", Proceedings of TOOLS 34, IEEE Computer Society, 421-430.
10. Muller, M.M., and Padberg, F., 2003, "On the Economic Evaluation of XP Projects", Proceedings of the 9th European Software Engineering Conference, ACM Press, 168 – 177.
11. Murru, O., Deias, R., and Mugheddu, G., 2003, "Assessing XP at a European Internet Company", IEEE Software, May/June, 37-43.
12. Rasmusson, J., 2003, "Introducing XP into Greenfield Projects: Lessons Learned", IEEE Software, May/June, 21-28.

13. Robertson, P., 1997, "Integrating Legacy Systems with Modern Corporate Applications", Vol. 40, No. 5, May, 39-46.
14. Sliger, M., 2004, "Fooling Around with XP: Why I lost interest in PMI and took up with something more Extreme", Better Software, May/June, 16-18.

Experiences with the selection and use of an open source web load testing tool - A Case Study

Hans Westerheim

SINTEF ICT
hans.westerheim@sintef.no

Abstract. The use of Internet is increasing, and more and more the Internet site of a company is crucial to the image of the company. To be able to keep a good image the Internet site needs to incorporate several qualities, speed of use being one of them. This paper describes a company which is running three different web sites, and as an attempt to be more efficient and safe, all three sites were to be moved onto a common infrastructure. When this was done, this physical infrastructure became one of the most accessed web sites in Norway. In advance of the transfer, the new infrastructure was to be tested for the expected number of simultaneously users and their use of the infrastructure. Almost 40 possible tools were evaluated according to predefined criteria. Five tools were finally judged as almost equal with respect to the criteria. From these tools, OpenSTA, an open source tool, was selected. The tool was used to conduct the stress test of the web site with variable results. The infrastructure is now hosting the three web sites. Using action research as research method, this paper discusses the selection and use of an open source tool when doing web load testing. The company in this case study did a conscientious evaluation of tools before OpenSTA was selected, and the evaluation proposed a well-working tool.

1. Introduction

Internet has become an ordinary part of the everyday life for most people in the western part of the world. Internet is widely used a channel for the diffusing of different kind of information like news, public information, advertising, traffic information etc. Lately, also a lot of services are offered by the means of Internet; ticket purchasing, bank services, it is possible to file one's tax return, reporting etc.

Traditionally this information and these services have been accessed by the means of a personal computer and a browser, but lately also the use of mobile and handheld devices for accessing Internet has become more widespread.

The use of different types of devices and browsers used to access internet, along with the use from a very different kind of users, have forced the developers of internet information and internet services to put more emphasis on the capability of the services and the technical infrastructure to handle *the load*. Load testing is a helpful tool in this respect [1].

This paper covers the experiences of selecting and implementing OpenSTA, an open source tool, as the web load test tool in a Norwegian software development organization. The evaluation of different open source and commercial web load test tools showed that the functionality of OpenSTA was as good as the functionality of the commercial tools, with respect to the criteria defined by this company. The tool was chosen mainly because the cost, even though cost was not one of the initial criteria for the selection. The cost structure of the commercial tools, usually depending on the number of "virtual users" made a very negative effect for this company.

1.1 Web applications and Internet

In the early days of web and Internet the pushing of news and information was the primary use. This is not the actual situation anymore. More and more intelligent services are offered by the use of Internet, the different internet services are also becoming more and more linked and integrated. The internet applications and services more and more need to conform to a certain level of quality, *Quality of Service* [1]. One of the quality factors is the ability to serve the user in an efficient way.

The infrastructure serving the user with web services and applications must then be able to handle a certain *load*. The load can be measured as the maximum number of concurrent users accessing an Internet site. This quality can be verified by the means of web load testing [1]. Web load testing is a supplement to functional testing of web applications [2].

There have been a lot empirical studies covering performance testing of software and applications [3-5]. As a result of the transition to the use of web user interfaces and Internet, the applications and services need to be tested from the web user's point of view. In addition to the more traditional performance testing, this testing must also include testing of the perceived performance from a user's perspective. Such testing cannot be performed only in a test environment, but must include the use of the technical infrastructure actually running the services.

1.2 The Company

In this section we give a brief description of the company in this case study. The technical infrastructure as well as the software development in the company is described.

1.2.1 The Software Development Department

The main company, responsible for the web sites, has organized all of its interactive services into a separate company. The other departments and companies within the main company buy development and implementation services from this company. The main company is a media house located in Norway. The employees are mainly sitting at two different geographic locations.

At the time of the load test tool selection, there were eight system developers working in the software development department. Two web designers were employed as well as a total of twelve persons in the support services. Developers were located at two locations. The specification and development of the web applications were attempted to be accomplished in a structured and professional manner. This complies with the general maturity of web development, now very often looked upon as industrial software development [2].

The department acts as a service centre for the other departments. There was a very close relationship between the development department and the organization responsible for operating the applications.

1.2.2 The technical Infrastructure

The company is running three different web sites. One of the web sites is a newspaper web site; the main aim is to serve updated news. The second web site is a traditional news and entertainment web site. This web site is running several applications and services which include a lot of interaction with the user. The third web site is a dating service; also the services on this web site require an interactive dialogue with the user.

The development is mainly done in Java. The services are all using a database running on an Oracle server. To support the publishing of news and information, Escinic is used as a standard publishing tool. The infrastructure also includes the Oracle Application Server.

The incoming requests from the internet users are cached by a load balancer which then disperses the requests to two web caches. Requested information which can be served directly from the web caches will be delivered to the request from the actual web cache. Seven application servers are providing information and services to the web caches. These application servers use two database servers to store and update information. New or updated information is pushed from the application servers onto the web caches. This mechanism ensures updated static information at the web caches. If the users request information not presently available on the web caches, the application servers are handling the request. The application servers are running Oracle Application Server.

A relevant piece of information worth noticing is that the company is administering a relative limited set of hardware resources. This had implications when the tool for web load testing was selected, and it also affected the importance of having a web load test tool.

1.2.3 The need for web load testing

Internet is a very important medium for the company. This counts not only for users accessing services and applications using a traditional web browser, but also for users browsing on mobile devices as well. The cost of launching a service or application not able to serve the users is not calculated, but since it is within the core of services the company wants to offer, it is reasonable to assume that the cost is high. The company cannot simply afford to launch a service which does not have the desired quality and speed. For some of the services there exist alternative web sites, e.g. news. It is then likely to anticipate that the users will change service if the company cannot offer effective services. The cost for switching to another information or service provider is almost non-existing for the users. The consequences for the company would be a decreasing numbers of users. This would influence on the advertising income from the web site. The testing of web applications must therefore include the testing of the performance as experienced by the end user.

One of the differences which were introduced by web applications was that many users were given the possibility to access an application or a service simultaneous from different locations and by the use of different browser and different internet speeds. An internet service must therefore in some cases be able to adapt to these differences, and the load testing should be able to cope with this matter [1, 6].

1.3 Software testing and web applications testing

Traditionally the main focus for the performance testing tools has been to test the internal efficiency and effectiveness of the software. Testing has been oriented to the functions implemented in the software and the different modules have been tested separately. In earlier days users, or the groups of users, accessed the software through a defined user interface, using defined terminals and also using quite controllable quality on the communication with the main computers and servers. In summary, this made it easy to simulate the operating environment of the total software in a closed and controlled environment.

When new functions were to be deployed or existing functions to be updated, the individual functions could be tested with respect to efficiency in advance, and then deployed into the software if the tests were passed [3, 4, 7].

In addition, the load testing is concerned with the service efficiency as perceived by the end user. When it comes to internet, this count for service efficiency delivered to the user's browser. The individual users may be using different hardware accessing Internet, using different operating systems, connecting at different speeds and using different browsers allowing different functionality.

For companies earning their money by serving news, like the case is for the company described here, it is of great importance that the web services are able to deliver information and services during peak periods. When there are important things happening in the world, the peak can be tremendous, e.g. like during September the 11th 2001.

1.4 The use of Open Source Software

Open Source software, or free software, is distributed under a different paradigm than commercial tools [8]. The main difference is that when someone acquires open source software, it is not only the right to use the executable software, but also the source code is provided, as well as the right to change the source code. The open source community often promotes the users of the software to change the code, and to publish the changed code so that other users can also access the new code [9].

The use of open source software has increased significantly in the latest years, and also the use in commercial public sector organizations has increased [10, 11]. The discussion about several attributes related to software quality, and the ability of commercial software and open source software to fulfill these qualities is ongoing. In this case study though, commercial software and open source software are looked upon as equivalent when it comes to fulfilling software qualities.

2. Method

In this case, case study has been deployed as method [12]. In this case the researcher wanted to catch the complete process of selection going on in the company, and the opportunity the follow the process closely was present.

2.1 Data Collection

The close cooperation between the company and the researcher was ensured by the use of email, phone conferences, meetings, workshops and a common web based project hotel tool [13]. From the company's side there was mainly one contact person, enabling a close and informal cooperation.

Observation was most used as data collection method. During meetings and workshops, MindManager was used as a tool for documentation [14]. This is a tool for drawing mind maps, an effective way to take notes. The notes were taken by the researcher, and summaries from the meetings and workshops were made subsequently based on the notes.

2.1.1 Data collection, test tools

Internet was used as the primary information source when data about the various tools was collected. Google[1] was used as search engine, combined with written information from some of the vendors of test tools.

All tests were logged, both initial setup of the test, the test script generation as well as the accomplishment of the tests. MindManager and MS Word were used as tools for documentation.

2.2 Data Analysis

Constant comparison [15] has been used as method to analyze the information given during the observation, the interviews and informal conversations. The interviews were partly recorded and transcribed; in some cases the researcher took notes during the interviews. The workshops were mainly documented using mind maps. The written material were compared to see if the company did follow the process they had agreed upon, and to see if there were any derogations. In latter case, the reasons for such were identified. The analysis was not visible to the company, and all the material was kept and analyzed by a single researcher.

3. Results

In this section of the paper the results from the work are presented. The evaluation process is described, as well as the resulting tool and initial experiences from the use of the tool.

3.1 The Evaluation of possible Tools

In this section we describe the evaluation process, as well as the criteria we used for selecting the web load test tool for the company.

3.1.1 The evaluation process

To be able to select a test tool in a qualified way, some sort of process is needed. In this case the evaluation process was conducted in a relative traditional way; the characteristics of the tools were evaluated against a list of desired characteristics. The scores for each tool were used a discriminator. The process is similar to processes reported also from other cases [16].

Initially 42 possible tools for load testing were identified. Information regarding these tools was collected. The initial evaluation of the tools was conducted by the company itself. This initial evaluation resulted in a shortlist of 20 tools to be further considered. The company and the researcher jointly conducted a literature study. Scientific papers regarding web load testing, and the experiences with different tools were collected [1, 6, 17-20]. The papers were studied with the aim to be able to exclude individual tools, or tools with specific qualities. The result of this study was that nine tools were removed from the shortlist, and one new tool was added.

The list with the twelve remaining tools was in focus of a common workshop with the company and the researcher. During this workshop the evaluation of the twelve tools against the criteria list was discussed. The most important criteria for the company were in focus of the discussion (see section 3.1.2). The results from the test with respect to these criteria were again evaluated resulting in a shortlist with seven tools still on it.

The company's representative and the researcher did a final test and evaluation of these seven tools. The final shortlist contained five tools. These five tools were considered quite equal to the functionality demanded by the evaluation criteria, and they were all considered to be compatible with the technical infrastructure, as well as with the software development tools and processes. From this list of tools, OpenSTA was chosen because of the costs compared to the commercial tools.

3.1.2 The criteria for the tool evaluation

The list of criteria was divided into five main areas which were considered to be important for the company:

[1] www.google.com

Operating System: The tool should be able to run on clients with different operating systems, the requirement was Windows XP, Windows NT, Windows 2000, Sun Solaris, AIX, VMS and Linux.

User Friendly: The tool should be easy to configure. The tool should be provided with good user documentation, and should be easy to use. The tool should be able to be language configurable. Finally, the tool should have good configuration documentation and a good troubleshooting guide.

Support: There should be a helpdesk available in the office hours. There should be upgrades available on a reasonable basis, and the upgrade path should be straightforward.

Structural Issues: The product should be secure. The tool should not cause any co-existence problems with systems already running on the company's infrastructure. The tool should not cause undue load on the systems or the infrastructure. The load caused by the tool itself should be measurable.

Functionality: This area did cover most issues, like:

- The tool should be able to simulate load volumes between 1 and 1.000.000 concurrent hits
- The tool should be able to handle streams from logs
- The tool should be able to generate load in a random or controlled way depending on requirements
- The tool must be able to generate output to log files, spreadsheets, diagnostic screens, prepared formats, and it must be possible to save and open previous results, and it must be possible to print the output
- The tool must be able to handle pop-ups on a web-site
- It should be possible to version-control and label the separate tests
- It should be possible to reuse tests
- The tool should be able to handle both internal load as well as load from the web
- The tool should be able to analyze components that contribute to load on the network, the applications, the database etc.
- The tool must be able to identify what resources in the technical infrastructure that are being hit during a load test

3.1.3 The final shortlist

There were five tools left on the final shortlist. These were:

- Radview – Webload
- Empirex – E.Load
- Opendemand – openload
- Softlight – Sitetools Loader
- Web Performance Trainer
- OpenSTA
- Novosoft – WAPT

Demo versions or evaluation versions of these tools were downloaded from the internet and installed on computers both within the company's network, and also on computers outside the company's network. The same types of test scripts were developed for all the tools, and the test scripts were run with different numbers of virtual users (where applicable) to simulate a real load on the company's web sites.

The test scripts were designed to simulate different types of use of the web sites. The first case was users just opening the main web pages of all the three web sites just to look for information. Secondly users clicking on main news articles were simulated. These articles were most likely stored in the web caches since it is rather static information. Third users asking for updated prices were simulated to also activate the application servers. Finally users logging onto a secure area with user name and password were simulated. This was done to make sure that the ability of the infrastructure to handle several logins at the same time was tested.

The results from the test, together with downloaded documentation of the tools, showed that all these tools were more or less able to fulfill the criteria set by the company. This final shortlist was presented to the IT management in the company, and OpenSTA was chosen. The main reason for choosing OpenSTA was the price structure of the commercial tools. Most of the commercial tools have a basis license cost. On top of this basis license cost the user has to pay per virtual user which is to be simulated. For the company in this case study, a high number of virtual users, the ideal number was 40 000, were needed to test a realistic load, giving very high cost if they were to choose one of the commercial tools.

3.2 The implementation and initial use of the selected tool

When OpenSTA was chosen, the tool was downloaded from Internet. It was installed on one computer in the beginning. One of the developers taught herself how to use the tool. Scripts for running tests against the company's web site were developed, and initial testing was done. During the tests, several monitoring tools were running on the application servers and database servers. After the initial tests, server logs were studied. The aim was to confirm that the scripts really did run, and that the intended actions implemented in the scripts had taken place within the systems and infrastructure. The server logs were also used to measure the actual load on the application servers and the database servers.

As a second stage, OpenSTA was also installed on a computer running outside the company's network. The same scripts were deployed, and run from this computer. The scripts were also run in parallel from computers inside and outside the company's network. This was done as a pre-test for increasing the load, and also to be better able to control the deployment and use of the scripts.

The tool seemed to simulate a number of users in a proper way. It was not possible from all the server logs to identify how the technical infrastructure did react to all the requests.

4. Analysis and Discussion

In this section of the paper the results from the tool selection are analyzed and discussed.

4.1 The tool selection process

The tool selection process managed to support the company in selecting a web load test tool. The tool did fulfill the requirements set by the company, and it was possible to deploy the tool. The definition and management of the test script did work well, as did the definition and control of the virtual users needed to perform the testing.

From the data and analysis it seems clear that adopting a tool selection process as described in this paper is a proper way to ensure the selection of an appropriate web load test tool.

4.2 Selecting an open source test tool

The download, installation and deployment of OpenSTA in this company worked out well. The downloaded version did support the software and hardware infrastructure in the company according to the facts about the tool available on Internet.

The people working in the IT department could not point out any weaknesses when compared to acquirement of a commercial tool. Frequently asked questions, and answers posted on the internet site worked as a good replacement of written manuals and instructions. The long-term use of the web load testing tool within the company is not analyzed. The conclusion after the initial use, and short term use, covered by this paper is that an open source tool did work properly for this company, and that the use of open source tools should be considered by companies looking for a web load testing tool, applying the same sort of evaluation process as in this case study.

5. Possible further research

It would be of interest to collect empirical data from several case studies where different tools have been used for web load testing. These data should include evaluation of both strengths and weaknesses of the selected tools, described in context of the organizations using them as well as the types of web applications, or services, being tested. The technical infrastructure on which the tools have been used also needs to be included in the context. By doing such an evaluation it should be possible to map different properties by a software developing organization to different tools, and also hopefully, different properties by the tools.

The development process for web applications should include the configuration and use of some tool for web load testing. How to incorporate the load testing into the development process should be further explored.

Acknowledgements

The author would like to thank colleagues at SINTEF ICT, especially Geir Kjetil Hanssen and Torgeir Dingsøyr for discussions and internal reviews of the paper. The ability to be present together with the company has been a matter of necessity to be able to publish this paper, so thanks to the company and the contact person in the company.

This work was conducted through the Software Process Improvement based on Knowledge and Experience (SPIKE) project, supported by the Norwegian Research Council.

References

1. Menascé, D.A., *Load Testing of Web Sites.* IEEE Internet Computing, 2002. **6**(4): p. 70-74.
2. Ricca, F. and P. Tonella. *Analysis and Testing of Web Applications.* in *The International Workshop on Web Site Evaluation.* 1999. Atlanta, USA: IEEE.
3. Smidts, C.S. and D.W. Sova. *A Comparasion of Software-Testing Methodologies.* in *Annual Reliability and Maintainability Symposium.* 1995: IEEE.
4. Weyuker, E.J. and F.I. Vokolos, *Experience with Performance Testing of Software Systems: Issues, an Approach, and Case Study.* IEEE Transactions on Software Engineering, 2000. **28**(12): p. 1147-1156.
5. Heiser, J.E., *An Overview of Software Testing.* 2000.
6. Nguyen, H.Q., *Testing Applications on the Web.* Wiley Computer Publishing, ed. I. John Wiley & Sons. 2001: John Wiley & Sons, Inc. 402.
7. Vokolos, F.I. and E.J. Weyuker. *Performance Testing of Software Systems.* in *1st International Worskshop on Software and Performance.* 1998. Santa Fe, New Mexico, USA: ACM Press.
8. Johnson-Eilola, J. *OPen Spurce Basics: Definitions, Models and Questions.* in *The 20th annual inmternational conference on Computer documentation.* 2002. Toronto, Canada: ACM Press.
9. de las Heras Quiró, P. and J.M. González-Barahona, *Free Software Today.* Upgrade, The European Online Magazine for the IT Professional, 2001. **II**(6): p. 4-11.
10. Muñoz Esteban, J.J., *European Initiatives Concerning the use of Free Software in the Public Sector.* Upgrade, The European Online Magazine for the IT Professional, 2001. **II**(6): p. 36-40.
11. Robert, G. and F. Schütz, *Should Business Adopt Free Software?* Upgrade, The European Online Magazine for the IT Professional, 2001. **II**(6): p. 12-19.
12. Yin, R.K., *Case study research: Design and methods.* 2nd ed. Applied Sicoial Research Methods Series. Vol. 5. 1994, London: SAGE Publications. 171.
13. eRoom, *eRom v. 7.* 2005.
14. MindJet, *MindManager.* 2004.
15. Avison, D., et al., *Action Research.* Communications of the ACM, 1999. **42**(1): p. 94-97.
16. Hanssen, G.K. and T. Dingsøyr. *A Comparasion of Automated and Manual Functional Testing of a Web-Application.* in *EuroSPI 2003.* 2003. Graz, Austria.
17. Sisalem, D. and H. Schulzrinne. *The Adaptive Load Service (ALS): An ABR-Like Service for the Internet.* in *The Fifth IEEE Symphosium of Computers and Communication, ISCC 2000.* 2000: IEEE.
18. Avritzer, A., J. Kondek, and D. Liu. *Software Performance Testing Based on Workload Characterization.* in *WOSP 2002, Third International Workshop on Software and Performance.* 2002. Rome, Italy: ACM.
19. Dustin, E., J. Rashka, and D. McDiarmid, *Quality Web Systems, Performance, Security and Usability,* ed. Addison-Wesley. 2002: Addison-Wesley. 318.
20. Shea, B. and K. Gallagher, *The Load Test Revolution.* 2003, Newport Group: Massachusetts. p. 15.

Defect Management System

A success story with Process Automation

Sundaresan.J, Sudha.Y, Raghavan.T.S

Abstract

Defect management Systems (DMS) helps to cut down the large amounts of time required for manual Document / Code reviews and additionally support many aspects of software quality assurance for the project. DMS is a automated Management tool which aids in automating the defect collection, organization and reporting of defects in review process as captured in the Reviews (Documents & Code) and testing stages of the Software Process Life Cycle. Also it is capable of giving very useful Review / Test metrics from which root cause analysis can be done. In addition, visualization of Root causes in the form of Pareto, Fish-bone analysis is also possible. The detailed reports on key aspects of the Product related defects could be easily generated.

This paper presents the experience of the team, DMS (Defect management Systems, a SPI team of the Software Technology Appreciation Group in the organization, PSC (Philips Software Centre (P) Ltd., Bangalore), in making the tool possible. The team has developed an in-house tool and this paper provides insights regarding how to integrate with existing tools in the organization & successful deployment of the in-house tool.

Keywords

DMS, Defect management Systems, Review Process, Review Effectiveness, Test Metrics, RCA, Fishbone Diagram.

1 INTRODUCTION

Philips Software Centre Private Ltd, operating from Philips Innovation Campus (PIC), Bangalore, India was established in August 1996 as a wholly-owned subsidiary of Royal Philips Electronics N.V. Major work at Philips Software Centre are done using state-of-the-art software engineering paradigms and platforms including real-time systems, component-based software engineering. PIC is an ISO 9001/TickIT, SEI CMM Level 5 company and has emerged as a critical partner in the development of strategic and futuristic technologies for Philips worldwide.

Product market is a very competitive affair these days. For almost every other product or service, there is more than one organization trying to sell the same and in each of these, the core product is essentially the same. What then determines the organization's success rate is the amalgamation of the following 3 factors: -

- Cost effectiveness
- High Quality
- Fast Time To Market

Key objectives in a product company is to reduce cost is in terms of "CYCLE TIME REDUCTION ". This can be achieved via Automation of various process to enable fast development of the product

A lot of effort was being directed towards collation and analysis of Review findings at Philips Software Centre, thus adversely affecting productivity. DMS tool was conceptualized to reduce this non value adding effort. This tool automates the collection, organization, reporting & management of defects captured in the Reviews (Documents & Code) and testing stages of the Software Process Life Cycle

DMS intends to significantly reduce the effort spent on collecting defect data, manual calculation of metrics, & Perform RCA and come up with a Defect Prevention actions. DMS tool is presently deployed across PIC and has become a way-of-life in the organization. Infact PIC-Bangalore is currently deploying this tool at various other Philips sites across the world.

The paper also describes how the DMS was used as an effective solution in aiding Software Process Improvements and describes how it was deployed seamlessly within the organization. In particular, the topics covered include:

- Decomposition of the problem – definition statement of attributes for the DMS System
- Development of the DMS system
- Key challenges faced
- Results, Lessons learned and way forward

2 BACKGROUND

The DMS STAG team started the work with the following main objectives:

1. To evaluate the usage of existing tools in the organization
2. To evaluate the functional scope of these tools with respect to the requirements of a standard Defect Management System
3. To create an appropriate DMS tool
4. To bring in the changes needed in the related processes and technologies in the organization to improve the effectiveness of Defect Management

3 *TOWARDS CREATION*

3.1 Organization Goals for DMS

<u>Goal 1:</u> Eliminating manual collection of defect data, which will lead to errors from review and testing process.

The results to be achieved from this goal are:

- Automate Defect collection and reporting
- Cycle time reduction in calculating the defect metrics of the project.
- Environmental friendly – Paper wastage is reduced, as on-line review and review preparation are possible.
- Multi-site reviews made simpler and easier (The review softcopies are consolidated with the use of macros).

<u>Goal 2:</u> Web Enabled Database Repository

The results to be achieved from this goal are:

- Centralized repository of defect data for the present and future projects.
- Data representation in graphical format – automated.
- Query based data repository.

<u>Goal 3:</u> Web Enabled Defect Management

The results to be achieved from this goal are:

- Reference defect trend to be selected for the project, as basis for the current defect prevention plan.
- Pareto analysis to identify top Signature & Causes.
- Creation of an automated Fish bone diagram for aiding RCA.
- Planning and Tracking of DP using PDCA cycle.

3.2 Approach

On interaction with the cross-section of practitioners in Organization, it was felt the present form of review templates for defect logging was too cumbersome. Manual collection of data from the review reports and Problem tracker (Continuus PT/Clearquest) is prone to inadvertent errors. Lot of effort is spent towards consolidation of Review and Testing defects, thus adversely affecting productivity. A need was felt to have a tool, which will significantly reduce the effort spent on manually collecting the data and calculating the metrics associated with review and testing processes.

A Survey was conducted to collect the requirements of the organization for automating the data collection part of the review and testing process. The key requirements of the practitioners are summarized as follows:

- Metrics/Defect Classification should be web enabled & automated
    - Capturing of the defects should be done using a GUI based tool.
    - Tool should be integrated to PR/CR Tracker tool such as Clearcase DDTS and Continuus PT
- Common Defects Repository
- Tool would aid in Pareto & Root Cause analysis
- Tool should also generate various defect related metrics & graphs
- Tool should be simple, short, less time consuming.

The need was clear to develop a tool, which will automate the defect management mechanism of review and testing process.

4 General Description of the Tool

The DMS tool consists of 3 parts:

1. The review part
2. The testing part &
3. Defect Management Part

The tool eliminates manual typing of **Review Comments** after the review is over. The participants review the work product on the desktop or PC screen. A dialog box appears on the screen wherein the reviewers enter the defect severity, Cause for the defect, defect type and Qualifier. All the reviewers send the review comments to the author/moderator. The tool consolidates the comments automatically for the author / moderator and the consolidated review report is taken to the review meeting for discussion. Provision is made for capturing the defects, from the review meeting also and added to the consolidated review report. Using **Macros** the review defects are picked from the Consolidated review sheet and sent to a database, to calculate various metrics like Review effectiveness, Defect densities for life cycle stages, Review & Testing defects, Effort related metrics such as Review and Review rework effort, automatically. The query database is dynamic in nature and act as a data repository for projects.

Testing defect metric is also automated. Defects logged onto Synergy (CM Tool) database is converted to an excel format (comma separated value <csv> file format) and exported to the database. Using this information, we are able to calculate Post release defect metrics and the ratio between review and testing defects.

Defect Management assists in implementing the DP and CAR Process at the Project/Line of Business Level, by automating Pareto analysis of the defects and help in Root cause analysis and Resolution. This will also aid in planning and tracking the Defect prevention actions. DMS institutionalises a PDCA cycle to help, implement the DP and CAR Process.

5 CHALLENGES

- **Resistance to change**
    - Members used to the manual reporting mechanism , had reluctance to use the new tool as it involved downloading from the web and changing the templates to use the macros. This was overcome by creating an installation shield, which will automatically install the tool in the users desktop.

  - SEPG mandated the tool to be used in all projects and the same was notified through the organization process manual.

- **Mis-conception about the tool**
  - It was highlighted that the tool is not a substitute to the review process and just helps in automatic collation and reporting of defects.
  - Demo sessions conducted by the SPI members in their respective business units and also to the SEPG group.

6 *The Results*

- Cycle time reduction as review process is automated.
  - Eg: After a review meeting with 50 defects it took an average of 120 minutes to manually enter all the review comments in the word document. Now with the tool, it hardly takes 5 minutes.
- The review part of the tool is used in projects in all the business units of PIC.
- All Review & Test Related metrics are reported to Corporate Quality using the DMS tool

For ROI Details – Please refer to ROI Table 1.

For sample user feedback – Please refer to
APPENDIX

7 *Best Practices*

- 'Proof of concept' carried out by proto-typing, before building the actual system
- Back-ups for key people, so that work does not suffer, even during project pressure, travel etc.
- Technical documentation was very good and followed the peer review process. It was emphasized that the aim should be **"Building it Right" than "testing it right".** Also,
  - Requirements, Architecture and detailed Design artifacts and Test Specs well documented
  - Help files & User manual for operating the system in place
- Daily focus meetings for tracking
  - Daily Audio bridge followed by weekly face to face meetings

8 *Learnings*

- SPI activities in organization are not less important than projects.**Need an "I Can Attitude and I will Attitude"**
  - **"I can is more important than IQ"**

– Quote from the book "The monk who sold his Ferrari"

- For SPI activities , Aggressive commitments are difficult to match (Project deadlines are the one and only 'holy cow')
- Involvement from the Organization stakeholders for Knowledge Transfer and takeover upfront.
- Issues with deployment can be addressed early.

9 ACKNOWLEDGMENTS

We wish to sincerely thank and would like to express our appreciation for the following for all their support and assistance offered during the course of this work:

Mr. K.V. Ramachandran, Head of the Technology Change Management Committee, Philips Software Centre (P) Limited, Bangalore

Mrs. Veena Y., Technology Manager of the Technology Change Management Committee, Philips Software Centre (P) Limited, Bangalore

All those project leaders, software developers who extended their co operation for the evaluation of the tools and answering the tool-survey questionnaire

10 Literature

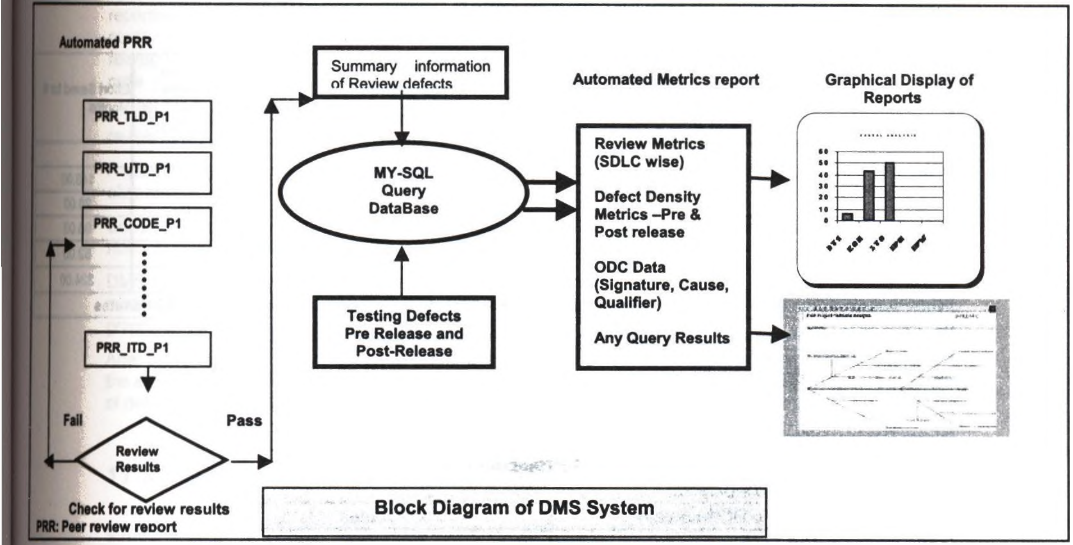
[1] An Article: Defect Management in Development and Test available at www.stickyminds.com/se/S2152.asp

[2] An Article talks about the Experiences with Defect Prevention available at http://domino.research.ibm.com/tchjr/journalindex.nsf/0/594603ab3cf18da185256bfa00685c2b?OpenDocument

[3] The purpose of Defect Prevention is to identify the cause of defects and prevent them from recurring available at http://www.teraquest.com/SW-CMM/static/DefectPrevention.html

[4] Concise description of **root cause** analysis from a systems perspective, with reference to classic system archetypes. By Gene Bellinger available at http://www.systems-thinking.org/rca/rootca.htm

APPENDIX 1


Automated PRR
PRR_TLD_P1
PRR_UTD_P1
PRR_CODE_P1
PRR_ITD_P1
Fail
Pass
Review Results
Check for review results
PRR: Peer review report
Summary information of Review defects
MY-SQL Query DataBase
Testing Defects Pre Release and Post-Release
Automated Metrics report
Review Metrics (SDLC wise)
Defect Density Metrics –Pre & Post release
ODC Data (Signature, Cause, Qualifier)
Any Query Results
Graphical Display of Reports
60
50
40
30
20
10
0

Block Diagram of DMS System

Appendix 2

Return on Investment (ROI)

Data

LOB	No. Of Reviews	Time taken by Author for consolidating the comments Manually (Person Hours)	Time taken by Tool (person Hours)	ROI Using the Tool (Time taken before-Time taken after) * 100 /(Time taken after)	Effort Saved for 6 Months
DS	74	222	74.00	200.00%	148.00
PBC	14	42	14.00	200.00%	28.00
SPG	48	144	48.00	200.00%	96.00
TV	26	78	26.00	200.00%	52.00
					324.00

Note: Average time taken for entering/consolidating 50 – 60 review comments for a review is 180 minutes

Review Effectiveness (If tool used only for consolidation of review comments)	
Time taken by SQE To DO Manually	45 minutes
Time taken by The Tool	5 minutes
ROI	800.00%
Note: Review Effectiveness is Calculated After Every Review	
Estimated ROI for Testing Defects after 1 Projects piloting	
Testing Defects for (119 defects)	
Time taken by SQE To DO Manually	30 minutes
Time taken by The Tool	3 minutes
ROI	900%

ROI Table 1

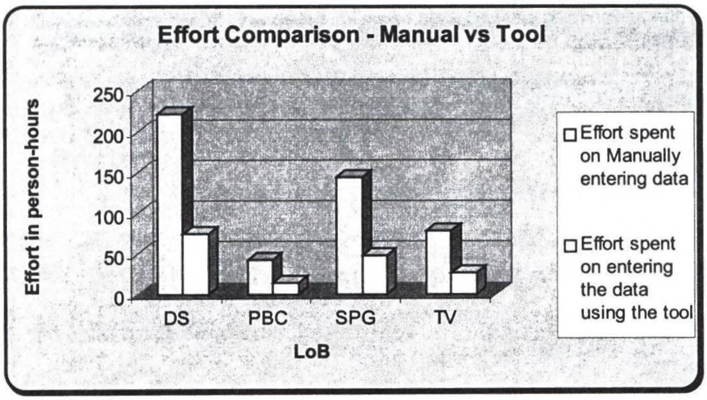


Figure 1: Effort Comparison Manual Vs Tool

APPENDIX 3

Feedback from Users

I find DMS (with review tool) really convenient to assess product quality and have already used it in reporting the monthly metrics (for peer review effectiveness) and also to view the status of product quality. The ability to generate data across lifecycle stages and also projects is useful. If the charts related to monthly product quality metrics are all provided in the future, it would be the icing on the cake - Components

A very well developed UI and an excellent tool which aids in preparing, conducting and completing the review cycle - MCE/TV

A good tool to optimize review process! User friendly tool during review preparation, during review & after review! -MCE/TV

Considerable improvement in Productivity; Ease of creation and consolidation of review report; Reduction of cycle time of review process-. MCE/PBC

DMS tool is a catalyst in continoooooooooous Improvement. All one needs to do is few clicks of mouse button to consolidates review defects, defect density data, testing defects and viewing graphical representation of defect data - MCE/DS

After using the DMS the effort involved for entries of the defects for documents has reduced by 1/2; the effort involved for entries of the defects for code has reduced by almost 3/4; Effort for consolidation of defects has reduced by 3/4; Also it is a very user friendly tool. MCE/Special projects.

11 Author CVs

Sundaresan.J

Sundaresan J is working at Philips Software Centre, Bangalore, in the capacity of a Quality Leader from the past 4.5-year. He has a total of 12.5 years experience in the Telecom and IT Industry in which 5.5 years is in the areas of Quality Control and Quality Assurance. Current responsibilities include definition and deployment of processes and institutionalising way of working in various TV projects. He is also performing the role of SPI co-ordinator for the TV projects executed from Bangalore. He has also led various improvement forums within the organization successfully

T.S. Raghavan

Raghavan.S.Tirumale, presently, is working in the Philips Consumer Electronics division of Philips Software Center (P) Limited, Bangalore as a Senior Technical Leader since April 1999. He is Respected member in the competence center for Tools and Test Engineering Group, playing a part on deploying various Project life cycle related tools. Mainly concentrated on development of various tools related to Software Testing and Design in Philips consumer electronics worldwide, Philips Electronics.N.V. He was also intensively involved in the organization wide software process improvement activities His interests include: Software testing, software engineering and Embedded software design and development.

Sudha.Y

Sudha Y is working at Philips Software Centre, Bangalore, in the capacity of a Senior Technical Leader – Quality since 2001. She has a total of 7 years experience in the IT Industry in which 4.5 years is in the areas of Quality Control and Quality Assurance. Current responsibilities include definition and deployment of processes and institutionalising way of working in Test Engineering & Services Group. She was also intensively involved in the organization wide software process improvement activities & has led a couple of them successfully. Her major areas of interests are in Project Management, Statistical Process Control, Software Metrics and Process Trainings.

Experiences of Process Introduction and Tool Evaluation in the Domain of Technical Product Development

Dr. rer. nat. Hans-Joachim Rabe, Dipl.-Ing. Rolf Rainer Moritz

dSPACE GmbH

Technologiepark 25

D-33100 PADERBORN

Abstract

For all projects, the need to set up well defined processes obviously grows in proportion to team sizes and the complexity of project goals. Rapidly growing organizations in particular often experience the problem that the development of processes does not keep pace with increasing project complexity, resulting in a gradual loss of efficiency. To improve this situation, new processes and tools often have to be introduced in existing organizations and in project teams with a high level of experience. Such introduction projects are even more demanding if the development environment is heterogeneous and if a high acceptance threshold towards tools is expected. In our paper we describe the situation in our company, which undertakes ambitious development projects in the field of technical applications with the main focus on simulation software and hardware. In the last 3 years we gained experience in 2 consecutive process management projects in large organizations, the first targeted at introducing a configuration management tool and the corresponding processes, and the goal of the second being the introduction of a new uniform project management process and tool. We report on the organization and responsibilities of evaluation teams, on methods of selecting tools reliably, and on how the acceptance thresholds were lowered by setting up new accepted processes coupled with appropriate training. The lessons learnt from the first project are presented, and we describe how these were applied to the second, the project management project. In conclusion, we present a bundle of best practices that we aim to pursue in future projects.

Introduction

People in young, successful, and rapidly growing high-tech companies often remember the foundation phase of their company: In the initial phase, only small development teams with a few members or even "one-man teams" focus their work on features, usability, and robustness. In many cases, there is little or no necessity to introduce stringent and uniform development processes and tool support of these processes, unless customers explicitly require them. With increasing market success and a growing customer base, the products become more and more complex. Further development, customer-specific modifications, and maintenance and support tasks lead to staff growth. To meet customer timelines, more and more tasks have to be done in parallel and several people have to work on the same version or different versions of an item such as a software component at the same time. The situation becomes even more complex when numerous products must be handled at the same time. By this stage at the latest, setting up, documenting and maintaining uniform processes, and introducing tools in support of those processes, become vital elements of development activities. Neglecting such activities would lead tc a substantial loss in efficiency and quality.

Our company is 17 years young, with a history similar to the one sketched above. We develop real-time simulation systems (hardware and software), which is mainly used in the automotive industry for developing and testing control systems, such as engine control, ABS, ESP and many more. The simulation systems comprise hardware components, embedded software components such as real-time operating systems and I/O drivers, code generation software, and visualization and test software. All components are designed as off-the-shelf products; there is only little engineering work to be done by the product development teams.

Several reasons motivated us to work on processes and introduce them along with appropriate support tools:
The software components are developed in different environments and languages. Embedded software is written in C for various target platforms. The code generation software parts are mainly based on scripting languages provided with the widely used control modeling tool MATLAB®/SIMULINK® and C++. The visualization and test components run on WINDOWS® platforms and are predominantly written in C++ and the freeware scripting language Python. Thus, the development environments, and the methods and expertise of the development teams, are quite heterogeneous. As members of the different development teams usually have to work together on common projects, it is crucial to have unified processes between the development teams to avoid significant efficiency losses caused by different understanding of methods, tool usage, and even terminology.
Complex tool functionality and demanding quality requirements, particularly in safety-critical applications, necessitate the traceability of all documents produced throughout the development process, e.g., requirement specifications, test results, release decisions, and many more. This, in turn, requires defined processes throughout the entire product development and the traceability of all relevant changes to the documents. Configuration management is one of the essential processes in this context, particularly when talking about parallel development work on tools composed of several megalines of source code and hundreds of individual components. In addition, some customers requested us to assess our product development process according to standard process models which require –among others- defined configuration management processes.
As customers often have tough schedules for their projects, they must rely on the suppliers' release date commitments. Consequently, the credibility of the release date is an important element in maintaining customer confidence. We strive at improving our release date predictability, especially by introducing and improving the project management process.
Last but not least, the documentation of well defined processes helps new employees joining the development teams to become productive quickly.

This paper aims to present our experience of introducing both configuration management and project management.

The Configuration Management Project

The Initial Situation and Goals

Our products are maintained over years. Development projects usually lead to new versions of the same product containing new features and/or improvements. All new versions of our software product portfolio are bundled and issued with a new release, and many of the products are interdependent. The releases of various types (major, minor, maintenance) add up to about 12-14 releases a year.

For all maintenance tasks such as support, bug fixes and customization, defined access to all versions released is indispensable. We therefore introduced simple version control systems many years ago. With growing team sizes, the increasing demand for parallel development, and growing product complexity with respect to both the number of features and the number of product interdependencies, the weaknesses of the initial approach became increasingly apparent. The major drawbacks were that our tool:

Was not appropriate for larger teams,

Did not support temporary development paths (branches),

Did not support bug tracking and change requests,

Had no process or workflow support; moreover, no uniform processes were defined for versioning,

Did not support configuration management. Relations between bug lists, change requests, and the corresponding software versions were maintained manually or even did not exist.

Our goals were to get rid of the drawbacks to be ready for further team growth and increasing product complexity, and to establish appropriate processes along with tool introduction. In addition, both processes and tool usage had to be standardized throughout the entire product development. Hence, the introduction project had 2 facets, one being the introduction of a new versioning tool and migration of existing data, the other being the introduction of the new configuration management process and the corresponding tool features.

Setting up the Team

When we decided to initiate a project to introduce configuration management, we anticipated the following problems:

The developers always work on very high-level products using cutting-edge technology. They are often confronted with challenging demands from our customers with regard to the features, usability, and robustness of the products. In turn, the developers make challenging demands on the tools they use, particularly with ambitious project schedules. Hence, the acceptance threshold for new tools and processes was expected to be extraordinarily high. We had to ensure that developers were closely involved in the decision process.

We anticipated there would be a wide range of different requirements for the tool due to the heterogeneous development environment (see above).

Basic knowledge of configuration management tools and processes was poor. A significant effort had to be planned to acquire the know-how.

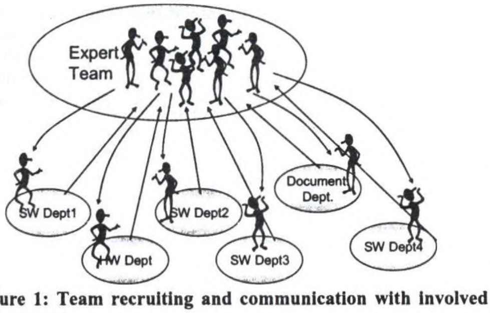


Figure 1: Team recruiting and communication with involved departments

We chose to set up 2 teams, one responsible for the processes, the other for tool selection. The first team (CM Process) was filled by management people, e.g., group or department leaders, the second (CM Experts) by highly experienced developers with high reputations in their teams. We involved delegates from all the development departments, in total 5, in both teams in order to establish close and direct communication channels between the selection teams and the departments in charge of implementing the processes and tool usage. Essential team results were reviewed by colleagues in the departments, who were involved in final discussions where necessary in order to achieve broad common agreement. This ensured information flow and reduced acceptance thresholds when new processes, methods and tools were implemented. *Figure 1* shows the team structure and the communication channels. The documentation department was involved in the project described below.

Both teams were directed by the head of one of the development departments, a person involved in other cross-departmental tasks. During the project we acquired an employee who was fully in charge of quality management. He took over project management step by step.

In an initial 2-day workshop held by external consultants, we got basic know-how on configuration management and gathered our initial requirements. The results were a rough overview of the methods of configuration management and of the most important vendors of tools in this domain, and a structured collection of our own requirements. After the workshop, both teams worked in parallel and communicated closely.

Tool Selection Process

From marketing material, trade fairs and independent reports, the CM Experts preselected 6 tools to be assessed in detail in workshops together with the vendors. To achieve comparability between the vendors, we took 2 measures:

Working out a questionnaire asking questions resulting from daily development work and questions with respect to support of the configuration management process. The latter questions were less detailed and less concrete as we did not have experience in this domain. The job of the process team was not yet completed. All preselected vendors were requested to answer the questionnaire prior to the assessment workshops or during the workshops. Identifying assessment criteria suitable for a quantitative assessment of the tools. We identified 19 criteria and determined weighting factors. Every tool was assessed numerically shortly after the workshops and every member of the evaluation team gave a mark for each criterion.

Numeric assessment was supplemented by qualitative assessment of the answers to the questionnaire, the general impression of tools and vendors, and pricing information. This led us to a decision which we thought to be reliable within a few months. However, in view of the high financial investment, we opted for an additional pilot project with the rank 1 tool. We identified 3 real development projects to be supported by the tool and had the project team members trained. Surprisingly, the tool failed. We encountered unexpected quality problems and a lack of functionality. This was despite the fact that the lack of functionality had been recognized in the earlier phases of the selection process during the workshops, and workarounds had been suggested. Unfortunately, they turned out to be unusable in practical work. Apparently, the pilot project was essential to detect the tool's weaknesses, as this was the only way in which realistic and daily work tasks were solved with it. However, for time and cost reasons we were not able to repeat pilot projects with two more tools (rank 2 and 3). We had to think about a more efficient method of obtaining a more thorough knowledge of the alternative tools prior to any further pilot projects and a more reliable basis for our decision. We identified characteristic and daily work use scenarios from the various development teams and documented them in the form of detailed step-by-step instructions. The use scenarios comprised setting up projects, defining configurations, going through integrations, and many more. We also included detailed questions about the handling of change requests and task and workflow support. These questions were driven by the results of the process team rather than by practical work, because there was only little experience with configuration management at this time. The document was written by experienced developers and reviewed by the tool team. We called it the "screenplay", and it comprised 35 scenarios. We organized 3-day workshops for each tool and requested the vendors to set up their software and provide a coach for that period. Under the guidance of the coach, our expert team and qualified developers tried to solve the tasks described in the screenplay during the workshops. Each member of the team recorded all findings in their individual copies of the

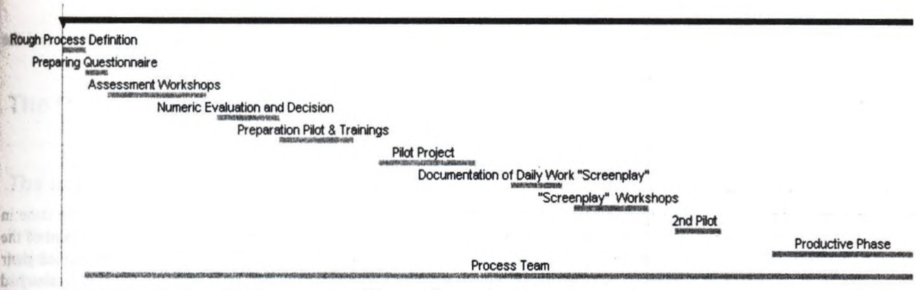


Figure 2: Activities in the selection process of the configuration management tool

screenplay. At the end of every workshop day, all members briefly discussed the pros and cons of the tool being evaluated, and voted. After completion of the workshop, the final assessment was compiled by the team. As all the workshops were held within a very short time of a few weeks, it was easy to compare the findings from each workshop. We felt that we got a much more profound insight into the evaluated tools and much more information on their usefulness in our own daily work. After budget clearance we purchased the tool. An outline of the team activities during the selection process is shown in *Figure 2*.

Setting up the Configuration Management Processes

As pointed out earlier, we set up a CM Process team comprising 5 employees. Each team member headed an organizational unit and therefore each person was in charge of quality issues. Their main task was to define a uniform configuration management process applicable to all development teams. The problem report/change request process is essential for configuration management. Consequently, the team focused on this kernel process and worked on:

Selection of a suitable method to uniformly document processes
Definition of a standard problem report/change request sheet and all data assigned to it
Determination of the workflow through the change process by defining a state model for problem reports
Definition of roles working with the problem reports, their responsibilities and rights.

As a result we identified 6 roles and the corresponding activities of the problem report workflow. Both roles and activities were documented using a combination of activity-chart-like diagrams and brief textual descriptions. In order to achieve the traceability of each problem and to get workflow support from the tool, we defined 15 states of a problem report and transitions between them. Only authorized roles were allowed to initiate the transition, i.e., we defined precise transition-role mapping. The work turned out to be quite time-consuming for several reasons. First, we did not have a documentation standard to describe processes. We had process documents in the various development departments, but they only covered certain aspects of process documentation and were mainly textual and hence not very suitable as a development-wide standard. We looked for several options and decided to introduce the scheme described above. Second, the heterogeneity of the developments teams led to very different requirements regarding the problem report sheet. It took a substantial amount of time to homogenize the manifold demands. Third, the various tools examined by the CM Experts team at the same time showed very different approaches and terminology with respect to workflow support.

These tool characteristics affected the definition of the change process. Consequently, we were unable to complete process documentation before the tool had been selected. Thus, the essential process for configuration management was well defined when we started to introduce the tool. However, new standardized versioning processes were not addressed in time, due to the erroneous assumption that the old ones would still be suitable for the new tool environment and due to the lack of time. The folder structure of the tool repository in particular was not standardized in time, which resulted in extra work during tool roll-out.

Tool Roll-Out

The selection process, the problem report process, and the tool decision were presented to a larger community of developers in an info forum to provide them with information on further steps and to give them the opportunity of asking questions. It proved to be a great advantage that the teams had been recruited from experienced individuals with high reputations in their teams. We found that acceptance of the assessment method and the decision itself was high.

The tool roll-out consisted of two phases:

The migration of the data from the old versioning tool to the versioning part of the new tool
The introduction of the new configuration management part

Because we did not want to jeopardize ongoing projects, all migration and introduction activities were done in the inter-project gaps, e.g., immediately after a release of a product and before starting the development of the next version. Developers from the various teams were trained on the tool and they, in turn, trained their colleagues, which was easy as we had tool and process experts from each department. Because tool selection had been thorough, we encountered no more essential technical or functional problems. The drawbacks of the new tool and potential workarounds were known and well communicated prior to widespread tool usage. Every department and every project team evolved its own plan for migration into the new world. During this phase we found that some definition and processes were not yet defined (see previous chapter). This was solved quickly during the introduction phase but quite late, resulting in some extra migration work.
Now, after 2½ years, almost all departments have finished and the tool has been successfully und uniformly introduced in the development departments. Furthermore, it is increasingly being used in departments other than product development. We are very satisfied with this project.

Currently, the tool environment is administered by two employees, one responsible for technology and versioning, the other for workflow support and process administration.

Lessons Learned

The following practices proved to be useful:
Setting up 2 different but closely communicating teams, one working on the processes, and one working on tool evaluation.
Recruiting highly skilled people from the departments involved to lower the acceptance threshold and to ensure easy know-how transfer to the departments.
The "screenplay" method is very effective for comparing tools by applying them to realistic use scenarios.
Step-by-step tool introduction minimizes the risks for ongoing projects and allows for gradual growth in experience with both tool and processes.
Pilot projects are crucial for revealing tool weaknesses.

The weaknesses we experienced are:
The initial tool preselection approach turned out to be insufficient as it did not provide in-depth knowledge.
Process awareness must start earlier in the project, particularly when new processes are to be introduced or processes are to be standardized within larger organizations.
The process activities are quite time consuming, particularly if there is no experience of documenting processes and if the processes to be introduced are completely new, such as configuration management in our example.
The process and tool teams should work consecutively so that the tool requirements can be derived from the process definitions.

The Project Management Project

The Initial Situation and Strategic Goals

In the light of our experience with the configuration management project, we decided to completely investigate project management processes prior to the evaluation of tools. In view of the fact that the development departments previously worked on their own project management processes, we started to analyze the project management environment of each development team. Using the process analysis method of Philip B. Crosby (TQM), we received comparable results. We discovered various methods, project management tools and procedures. Departments where short-term projects were the daily work developed more sophisticated methods than in areas where long-term projects dominated. The tool environments created also differed. In some teams, MS Project was established, while other teams used their home-made project management environment based on MS Excel. Combinations of various tools were also used to manage the projects. Moreover, the project management terminology used was heterogeneous. This analysis resulted in a documented process description for all departments containing all subprocesses, process ownership, activities, process inputs and outputs, tools and infrastructure, roles combined with responsibilities, know-how and abilities, as well as internal departmental standards.

Setting up the Team

Encouraged by the experience of the previous project, we established a task force, again involving members from all development departments, and this time including the documentation department, which is involved in any development project (refer to *Figure 1*, chapter "Setting up the Team"). The selected team members had experience in their individual project management methods and knowledge of the specific tool environment, and worked in cross-department projects. It was their responsibility to define process and tool requirements which would lead to common processes to be implemented across all development areas.

The goals for the project management task force were:
Unify the project management process
Standardize project management terminology
Identify efficient workflows
Achieve
more and more cross-departmental teams
more reliable release date forecasts
transparency of resource workload
unified project management tools
multiproject management to identify cross-project influences
Extract our requirements for a shared project management tool

To ensure continuous work, we set up redundancies to keep the work going despite business trips, vacations, or other unpredictable absences.

Setting up the Project Management Processes

Comparing the results of the process analysis, we defined subprocesses for specified project management subprocesses.
Project definition
Rough project planning
Detailed project planning, replanning
Project control
Project completion
Project aborting

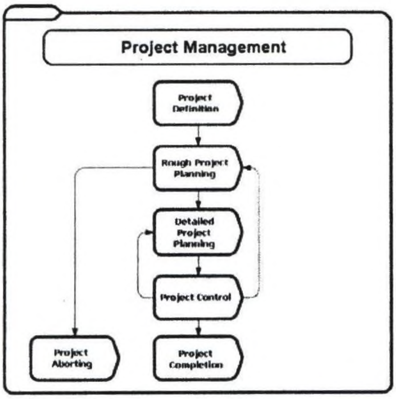
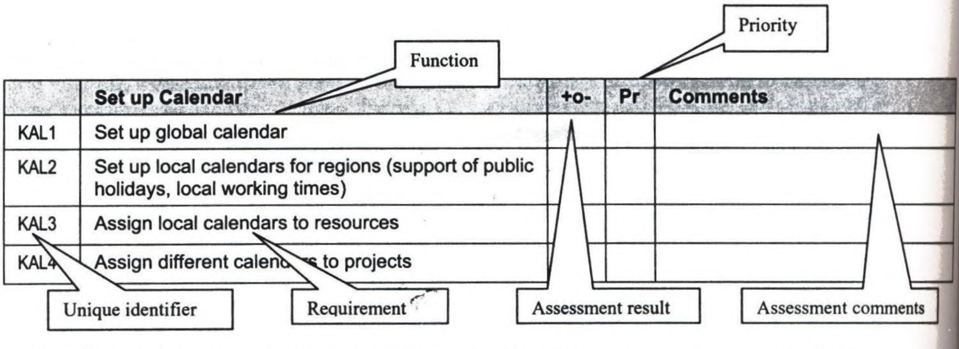


Figure 3: The top level structure of the project management process

Starting with the six individual project management subprocesses, we identified common activities, mapped differences to a common subprocess-based model, and modified the processes for future needs. Several meetings, chaired by a neutral moderator, and lots of discussion led to commonly accepted results. During the sessions the vocabulary was gathered, and a common understanding was acquired and documented in a project management glossary. We identified the roles necessary for project management, process inputs and outputs, environment and procedures. The project management process model was created and documented in a business process model (refer to *Figure 3*).

Following our encouraging experience with evaluating the configuration management tool, we again set up the "screenplay" method. Along the defined process, we extracted requirements for project management tools. We assigned process goals and our requirements for tool functionality for each project management subprocess. In addition to the project management process, we realized the necessity to set up requirements for project management tool administration, such as resource management, multiple calendar maintenance, project navigation in a multi-project management environment, access from different locations, user privileges, report management, multiviews of project data, etc. A large number of new requirements were then derived from these processes. To get comparable results we organized the requirements in chapters and documented the requirements in tables as shown in *Table 1*.


Priority
Function
Unique identifier
Requirement
Assessment result
Assessment comments

	Set up Calendar	+o-	Pr	Comments
KAL1	Set up global calendar			
KAL2	Set up local calendars for regions (support of public holidays, local working times)			
KAL3	Assign local calendars to resources			
KAL4	Assign different calendars to projects			

Table 1: Extract from the "screenplay" showing activities related to the use scenario"set up calendar"

Each requirement was identified by a unique identifier for communication purposes. Sub-requirements detailing functionality were listed for each major requirement. Each sub-requirement was prioritized for evaluating the results. The whole screenplay was reconciled with all the departments concerned and reviewed by all task force members. The preselection of tool candidates was launched in parallel sessions. In contradiction to the classic approach that resources of the IT department select candidates, preselection was done by a small team staffed with members of the departments concerned, which would use the project management tool in their daily work. The basic information was taken from a study by the University of Osnabrück, containing assessment results for 28 project management tools. The study was published close to the beginning of tool evaluation (Feb 2003) and described, investigated, categorized and assessed project management tools for different project management necessities. We looked for a plan-oriented tool in the multi-project management system area.

The team decided to perform evaluation in 3 steps:

Preselection against a specified common criteria list
Tool demonstration by manufacturer, distributor or representative
Workshop with end users "acting the screenplay"

Step 1

Extracting highly relevant requirements from the screenplay, we listed the criteria for preselection. About 20 candidates met the criteria in the list. We investigated on the Internet, visited fairs, made telephone inquiries, and selected the top 8 candidates for the next step of evaluation.

Step2

The second hurdle for the candidates was a tool demonstration and a small questionnaire containing questions about the most relevant requirements identified in the screenplay. In most cases an Internet presentation combined with a telephone conference took place, the time consumption being about 3 to 4 hours including preparing the telephone conference and Internet session.

Step3

Finally 3 candidates passed the preselection phase. We sent the screenplay to the vendors of these candidates, asked for their statements, and requested them to self-assess their tool with respect to our requirements ("score"). The vendors could score their product as meeting the requirements:

+ = fully, no limitation for a particular requirement
o = partially, it is possible to fulfill the requirement but with limitations or workarounds
- = not fulfilled, the requirement cannot be fulfilled by using the tool

The vendors were requested to comment on their scores, even where they could not fulfill a requirement completely.
In parallel, we asked for a quote for a 2-day workshop to learn more about the tool and tool concepts, act our screenplay, and assess usability for tool adoption in our process environment. To see how interested the representatives were in gaining our custom, we tried to reduce the quote. At the end we got an equivalent price from all competitors. In preparation of the workshop we formulated an agenda used for all workshops to get similar conditions for all candidates. To catch the given time frame of 2 days, we reduced the screenplay to the most important and most used functionalities.
The agenda contained:

½ day tool presentation by representative
½ day project planning
½ day project control
½ day multi-project management facilities, templates and reporting

Not only members of the task force joined the workshop team. It was important for us to include in the workshops all departments that would use the tool in the future, to investigate their needs for project management. Again the screenplay was the guideline through the workshops, and the participants voted in the same manner as the tool representatives before them. A team of two members worked together in the workshop. One member acted the selected parts of the screenplay, the other documented the results in the screenplay.

At the end of each day a summary was produced and impressions were collected. Two candidates passed the workshop. One workshop was stopped and aborted when the team detected that major functions in the area of multi-project management were missing.

Each department collected the results of the workshops in an assessment report separately. The criteria in the report referenced the main chapters of the screenplay processed in the workshops. A priority (high, middle, low) and score (best, medium, inapplicable) had to be documented in the following table per criterion:

General:
Roles and views
Resource management
Project work:
Define project

Rough project planning
Project visualization
Detailed planning and replanning
Project control
Project completion/aborting
Project template support
Multi-project management:
Shared resources
Dependencies between projects
Information flow support
Project team view:
General requirements
Time sheet and feedback
Analyzing project data
Reports
Interfaces
Workflow management
Work product management

The scores were derived from the workshop result documentation. There was no score for the tool evaluated in the aborted workshop. When we analyzed the results for the remaining candidates, we got a dead heat. There was no significant lead for any of the tools. Both tools had advantages and weaknesses in different areas.

As the next step we decided to contact companies who had implemented the tools in their project management world. We searched for a power user for each tool who had had the tool in place for several years as a reference. Two members of the task force worked out a questionnaire as a basis for interviewing members of the responsible project management teams of the companies. A one-day visit was organized to learn their experience of using the tool, and to understand what the advantages are, what kind of daily problems occur, and how they are solved. We discovered additional areas we had not covered when setting up the screenplay: the update mechanisms of the implemented tool. One tool had a huge gap in this area. We had not anticipated that this could be a problem for an accepted project management tool. We learned that there is in enormous maintenance workload when new tool versions are distributed. Furthermore, we were told that it is very valuable to have three servers in place to perform tasks for production environment, for training and for updating the project management system. Also we learnt about the project management processes tailored for project management tool assistance and about the experience of tool implementation and process introduction.

The Final Vote

After presentation of the results of external experience, we started the final voting. All the departments concerned were involved in the final voting, which took into account all the information gathered during evaluation, from own and external experience. Eventually the candidate for first pilot projects was nominated.

The evaluation project ended at this particular point. We agreed to end the task force and to set up a team of experts to implement the process, methods, and tool. These were experts who had to be able to administer the tool, to serve as first contacts in their departments, and to define conventions and modifications for using the process and tool.

Preparing the Next Steps

We arranged external training sessions in tool administration and the methods used with the tool especially for the expert team, held by consultants from the distributor. The group of trained experts worked out conventions for tool adoption and the tool settings necessary for customizing the project management tool to our needs, followed by sessions where tool consultants reviewed our tool settings and conventions for the project management tool environment used in our company scenario.

To improve and finalize the project management process we defined at the beginning, we hired an external consultant to develop training courses for project managers. In preparation we discussed our project management process with the consultant and modified the project management process by adopting state-of-the-art project management methods, documents and procedures. dSPACE documentation on the process, responsibilities,

methods, glossary, and work products was drawn up as a handout for the training sessions and as the basis for project management at dSPACE. There was no focus on tool dedication in these training sessions, but on standard methods of managing a project throughout the company. To bring project management process and tool together, we integrated the project management tool in the process environment by slightly modifying process definitions.
Before introducing the new project management process and tool throughout development activities, we decided to run a few pilot projects to verify the process and tool decisions and definitions. The pilot projects were defined by the expert team. The tool distributor agreed to a half-year pilot phase, where we paid for tool installation and consulting services, using the necessary licenses for free during this period.

To train the pilot project teams, we prepared two training courses with different objectives:
First, a training course for project leaders and group or department heads
Second, a training course for team members who deliver feedback to their project leaders.

The focus of the first course is on:
The new project management process
Definitions, roles, responsibilities, methods, work products
How to define a project
Getting started, project planning
Using project management methods
Project management tool introduction
Project planning in the tool
Project control supported by the tool
Collecting feedback
Interpreting tool data
Resource balancing
Reporting

The second training addressed the following subjects:
What does project management mean?
What are my tasks for project management?
How to deliver tool supported feedback?

The project leader training is organized as a workshop and lasts 3 days; the second training course is a presentation of 4 hours. This training program has to be passed before being admitted to the project management tool, to ensure a common understanding and a common view of the process, methods and tool. The pilot projects were accompanied by the expert team to support the teams, to detect weaknesses and recognize best practices, and to discuss und solve problems occurring in handling the tool or living the process. The pilot phase was driven by a fixed end date. Product development management, the expert team, and the members of the task force came together for the final decision. By reviewing the results, the experience gained, and users' impressions, we came to the decision to buy the tool.

Getting Productive

Setting up a live database server for the entire dSPACE project management world, independently of pilot projects, training, and the test environment, was the first step. Some projects of the pilot phase were imported to the productive server, and new projects were started on the new server platform.

The training courses were modified for greater efficiency. Project teams still have to be trained before getting access to the new tool and process. All new projects have to be created in the project management tool according to the project management processes. A common understanding and a project management culture are growing. The process is ongoing. *Figure 4* illustrates the activities throughout the project management project so far.

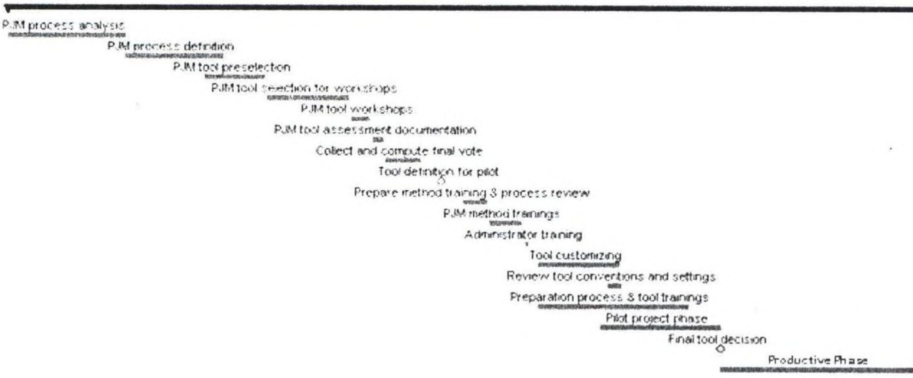


Figure 4: Activities in the project management project

Lessons Learned So Far

Process awareness in early stages drives a goal-oriented procedure
Without a well defined process, no sophisticated requirement acquisition
No theoretical process definition, processes must be accepted by process owners
New process design or process modification always carried out with process users concerned
Screenplay method again very successful
Developing our project management process world
Taking future needs into account
Structured documentation
Overview of all requirements with priorities and ratings
Essential for comparison of candidates
Directly comparable
Project management board PJM Experts to conduct all necessary activities
Preparing the ground for new tool and processes
Direct contact in the departments, quick response
Driving introduction
Finding problem solutions
Collecting best practices
Acceptance
Involving all necessary colleagues
Collect requirements from colleagues concerned
Training reduces thresholds
Recruiting multipliers for process infiltration
Time consumption and effort
Method of long- and medium-term evaluation
Investment for future earnings
Long preparation period to define project management process (6-8 months)
Preselection phase (20 → 8 candidates) 1 month
Selection phase (8 → 3 candidates) 2 months
Workshops 0.5 months
Final evaluation phase (3 → 1 candidate) 2 months

Summary and Best Practices

In two projects we established and refined methods for effective process definition, tool selection, and tool roll-outs. The first project was successfully finished; the second one is in the roll-out phase and we are very positive that it will be successful as well. From our experience we find as best practices:

Start with the process definitions and derive tool requirements from the process needs
Involve the process owners
Set up a coordination board
Generate a single screenplay to precisely reflect requirements in daily work situation
Get an early tool presentation from vendors (WEB very efficient) for preselection
Execute the workshop strictly according to the screenplay
Make necessary process modifications for best tool implementation
Train users in process and tool
Set up pilot projects
Collect feedback from the pilot to improve process and tool adoption

CASE STUDY: A practical approach for SPI in Spanish large companies

Román López-Cortijo García, Javier García Guzmán, Antonio de Amescua Seco,
Gonzalo Cuevas Agustín
{rlopez,jgguzman,amescua,gcuevas}@progresion.net

PROGRESION SMP
Parque Científico Leganés Tecnológico
Avda. Mediterraneo, 22, Office 1.66, 28914, Leganés, Madrid, Spain

Abstract

The purpose of this paper consists of describing a real case of a Software Process Improvement (SPI) in a Spanish large company. This paper also provides a set of experiences and recommendations for performing a software process assessment in large companies with several vertical organizational units. Moreover, it is presented two tools for coordinating this kind of assessments. Finally, the strategies for defining and deploying improvements in large companies are discussed.

Keywords

Software Process Improvement projects, software process assessment, improvement action plan, practical experiences

1 Introduction

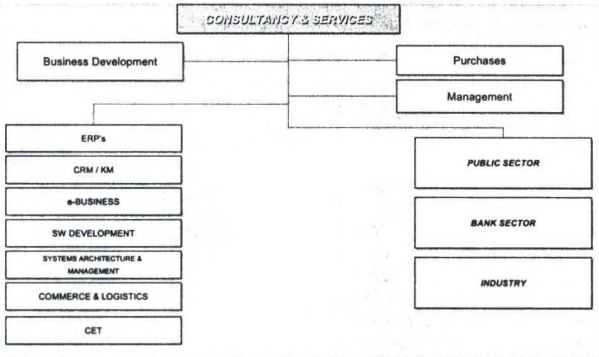


Figure 1: John Doe's Company Organizational Structure

John Doe's Company[1] is a Spanish enterprise that is leader in the sector of consultancy and IT services providing. The invoicing of the company is near the 700 Million € and has more than 2500 people in staff. Its scope of action includes practically all the market sectors and its organizational structure is shown in figure 1.

In summary, this organization is structured in the following operational

[1] Due to contractual obligations, we are not authorized to mention the real name of the company related to this case study. Currently, we are negotiating the permissions to mention the real name of the company in the final version of this paper. All the data provided are real.

areas: High level management, logistics and administration staff, business development staff, operations staff that is differentiated by specialties, that are organized in centres of experts (CEX) and, finally, management of clients whose activity is oriented to satisfy immediate user needs. This structure is in accordance to a business strategy highly centred in the client, to whom it is provided very with specialized solutions that are made in the centres of experts.

John Doe's Company wants to begin a SPI project oriented to improve the performance of the Information Technologies (IT) Project Management and Software Engineering areas.

The organizational units of John Doe's Company that are affected by this SPI project are:

- ERP's (29 project managers, totalizing 120 technical people)
- CRM / KM (55 project managers, totalizing 200 technical people)
- e-Business (28 project managers, totalizing 200 technical people)
- Software Development (28 project managers, totalizing 100 technical people)
- Commercial & Logistics (25 project managers, totalizing 60 technical people)
- Engineers Pull (20 project managers, totalizing more than 1000 technical people)

In accordance with collected information at the beginning of the SPI project, the main highlights of the current situation of John Doe's Company are:

- At organizational level, unified criteria and practices for project management are not well established and documented.
- The techniques and procedures for the projects management have not been standardized as the documental management.
- Software Engineering techniques and procedures have not been standardized.

In order to begin this SPI project and to obtain the established objectives, John Doe's Company contracted the services of PROGRESSION SMP [1]. This company is a Spanish leader in consultancy related to ISO 15504 [2] and CMMI [3].PROGRESSION SMP provides a set of services relative to planning SPI programs, change management, implementation of risks management strategies and metrics programs, as well as, design and development of knowledge management strategies relative to software businesses.

John Doe's Company contracted PROGRESION SMP due to the large experience of their members (totalizing the participation in more than 70 SPI projects), its independence from John Doe's competitors, the large knowledge related to the software engineering market and the condidence and condidentiality provided by PROGRESION SMP.

2 *SPI Action Plan*

The solution used is based on practices widely extended and guaranteed by the excellent results that they have obtained during many years in the industry. On the one hand, the software process improvement practices proposed by the IDEAL model [4] and, on the other hand, the ISO 15504 and CMMI reference models.

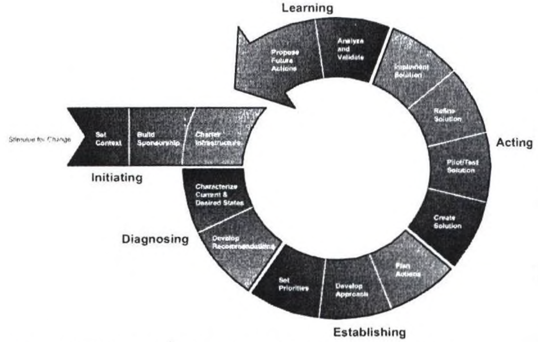
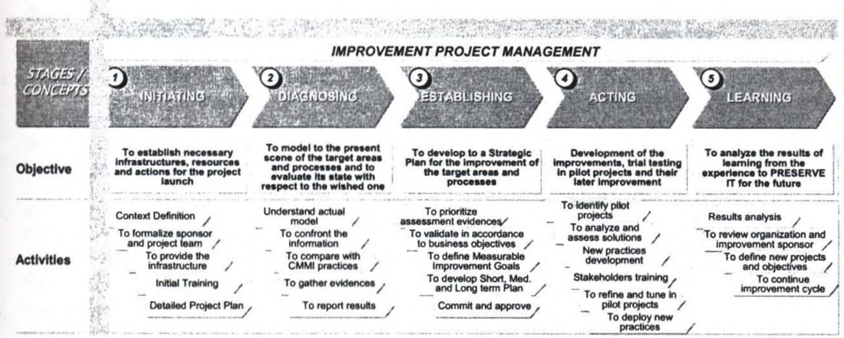


Figure 2: Software Process Improvement life cycle

From our point of view, the initiating phase of the SPI project includes several activities that are essential to achieve successfully the project objectives. For these initiating activities is necessary to consider the following aspects:

- The project motivation and its starting point only can be founded in key necessities for the business that conform the project requirements. The business needs identified by John Doe's Company managers were:
  - To improve company competitiveness
  - To reduce terms of accomplishment
  - To open new business lines
  - To increase the satisfaction of the client
- From this alignment with the business needs, it is essential to establish the infrastructure necessary to make the project. For this purpose, it is necessary:
  - To create a summarized SPI action plan (the detailed plan is shown in figure 3)



IMPROVEMENT PROJECT MANAGEMENT

STAGES / CONCEPTS	1 INITIATING	2 DIAGNOSING	3 ESTABLISHING	4 ACTING	5 LEARNING
Objective	To establish necessary infrastructures, resources and actions for the project launch	To model to the present scene of the target areas and processes and to evaluate its state with respect to the wished one	To develop to a Strategic Plan for the improvement of the target areas and processes	Development of the improvements, trial testing in pilot projects and their later improvement	To analyze the results of learning from the experience to PRESERVE IT for the future
Activities	Context Definition To formalize sponsor and project team To provide the infrastructure Initial Training Detailed Project Plan	Understand actual model To confront the information To compare with CMMI practices To gather evidences To report results	To prioritize assessment evidences To validate in accordance to business objectives To define Measurable Improvement Goals To develop Short, Med. and Long term Plan Commit and approve	To identify pilot projects To analyze and assess solutions New practices development Stakeholders training To refine and tune in pilot projects To deploy new practices	Results analysis To review organization and improvement sponsor To define new projects and objectives To continue improvement cycle

Figure 3: Software Process Improvement Action Plan

  - To determine the organizational structure that support the SPI project (see figure 4)
  - To adequate the participants' profiles, updating its knowledge related to the techniques and process capability model to use to guide the SPI Project.
  - To identify project's short, medium and large term objectives. These objectives should be measurable.

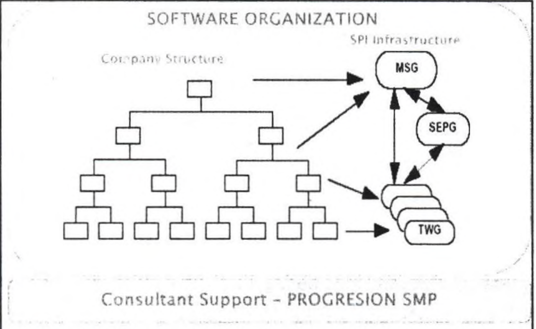


Figure 4: Software Process Improvement Organization

  - To prepare the communication strategy related to SPI project
  - To launch the SPI project
- The organizational infrastructure of a project like this should be settled down during the initiating phase. This organization should be structured by the following groups that are graphically described in the figure 4.
  - <u>John Doe's Company</u>: MSG, Management Steering Group; SEPG, Software

Engineering Process Group; TWG, Technical Working Group; Sponsor y Coordinator

- PROGRESION SMP: Project Management, Strategic Group, Tactical Consultancy, Experts Group, Reference Models and SPI trainers and assistants.

2.1 Assessment

In accordance with objectives of the definition of the assessment phase of the SPI project, the evaluated process areas have been: Requirements Management (REQM), Requirements Development (RD), Project Planning (PP), Project Monitoring and Control (PMC), Project Quality Assurance (PPQA), Configuration Management (CM), Analysis and Measurement (MA), Technical Solution Definition (TS), Verification (VER), Validation (VAL), Product Integration (PI), Risks Management (RSKM) and Organizational Training (OT).

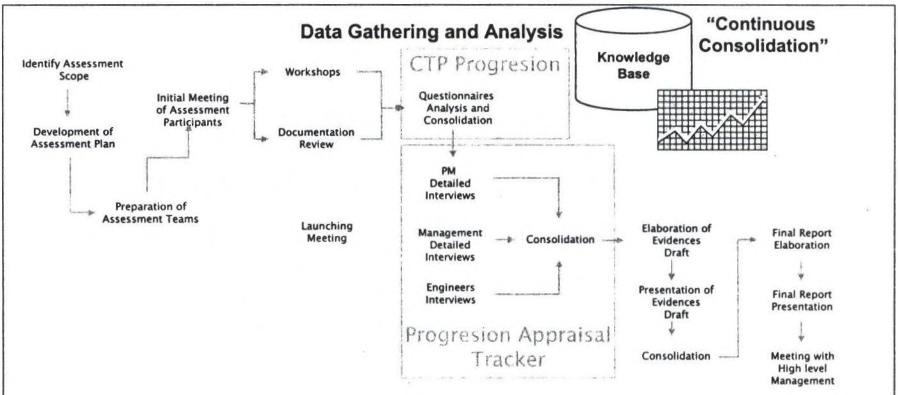

Data Gathering and Analysis
Knowledge Base
"Continuous Consolidation"
Identify Assessment Scope
Development of Assessment Plan
Preparation of Assessment Teams
Initial Meeting of Assessment Participants
Workshops
Documentation Review
Launching Meeting
CTP Progresion
Questionnaires Analysis and Consolidation
PM Detailed Interviews
Management Detailed Interviews
Engineers Interviews
Consolidation
Progresion Appraisal Tracker
Elaboration of Evidences Draft
Presentation of Evidences Draft
Consolidation
Final Report Elaboration
Final Report Presentation
Meeting with High level Management

Figure 5: Assessment Methodology

Due to the complexity of the assessment, that consists of analyzing five organizational units separately, in order to establish the photo of the actual state for each division and a global one, gathering the main problems common to all organizational units, PROGRESION has been used the process of evaluation shown in figure 5.

One of the greater risks of the evaluation process consists of a possible lack of motivation of the John Doe's Company personnel because usually the assessment is considered as an unnecessary activity. Our experience says that many software engineering personnel consider more useful beginning directly with the improvement action plan. The recommended contingency action to reduce this risk consists of make a preliminary assessment very short in time, but sufficient to have a general view of the organization to be used as a guide to prepare the exploratory interviews.

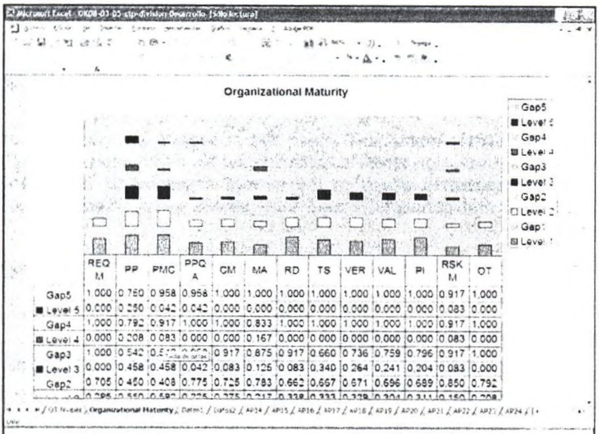
The assessment is organized in two main stages.

- The first stage consists of a taking of contact between the evaluators and the software organization, in order to obtain an approximated photo of the organization's software process capacity. You have to be conscious that the obtained data only gives an approximated view that should be confirmed during the second stage of the assessment.

  This first photo of the organization's software process capability is obtained by means of evaluation questionnaires (easy to complete) that ¡are filled up by representative project managers, after the celebration of workshop dedicate to explain the basic concepts of the process area that is being assessed.

The project managers that fills up the questionnaires, in many cases, do not correctly understand the reference models (ISO 15504 and CMMI) terminology, so, in many cases, the information provided by the project managers does not represent faithfully their current practices.

In many large companies, like John Doe's Company, want to have a detailed analysis by organizational unit and, moreover, to have a global view of the software organization. There are several tools for helping the evaluator to analyze the achievement of reference models objectives for software process, but these tools does not permit easily the "drill up and down" analysis required for a large company like John Doe's. For this reason, PROGRESION has developed a software tool based on Microsoft Excel for supporting, in an integrated way, the organization global analysis and the detailed analysis for each organizational unit (see figure 6). This tool is prepared to implement SPICE [5] and SCAMPI [6] assessment models and allows to gather and analysis data coming from different units, helping to identify global conclusions.



	REQM	PP	PMC	PPQA	CM	MA	RD	TS	VER	VAL	PI	RSKM	OT
Gap5	1.000	0.750	0.958	0.958	1.000	1.000	1.000	1.000	1.000	1.000	1.000	0.917	1.000
■ Level 5	0.000	0.250	0.042	0.042	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.083	0.000
Gap4	1.000	0.792	0.917	1.000	1.000	0.833	1.000	1.000	1.000	1.000	1.000	0.917	1.000
▨ Level 4	0.000	0.208	0.083	0.000	0.000	0.167	0.000	0.000	0.000	0.000	0.000	0.083	0.000
Gap3	1.000	0.542	[illegible]	[illegible]	0.917	0.875	0.917	0.660	0.736	0.759	0.796	0.917	1.000
■ Level 3	0.000	0.458	0.458	0.042	0.083	0.125	0.083	0.340	0.264	0.241	0.204	0.083	0.000
Gap2	0.705	0.450	0.408	0.775	0.725	0.783	0.662	0.667	0.671	0.696	0.689	0.850	0.792
[illegible]	0.295	0.550	0.592	0.225	0.275	0.217	0.338	0.333	0.329	0.304	0.311	0.150	0.208

Figure 6: Questionnaires Analyzer – CTP PROGRESION

- The second phase of the assessment consists of obtaining a precise vision of the organization current practice in relation with the previously selected process areas (in this case, those related to project management and product engineering) and guided by the preliminary evaluation results.

This phase is performed by means of detailed interviews to Division Managers, Managers of Centres of Experts, Project Managers and Engineers. The interviews should be prepared based on the results of the preliminary evaluation. Also the documentation gathering should be used . in this phase because provides objective evidence of the practices accomplishment.

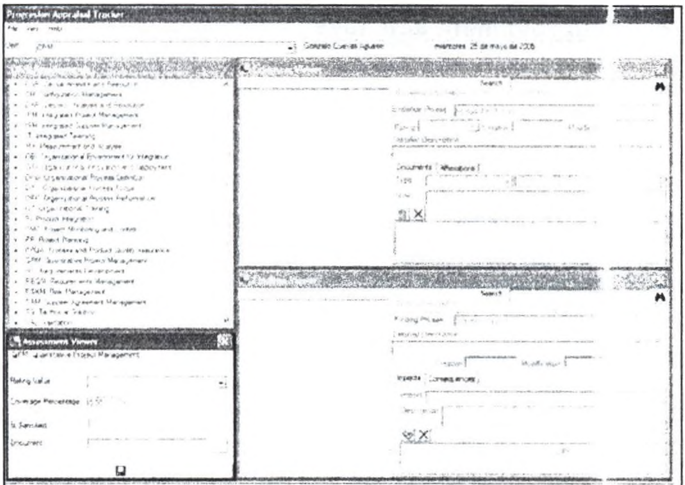


Figure 7: PROGRESION Appraisal Tracker

During the assessment of John

Doe's Company, PROGRESION detected more than 200 evidences, but many of them were very similar and conducted to the same consequences.

The consolidation process needed to integrate evidences and deduct consequences is very complex and should be completed as the interviews are being celebrated.

PROGRESION has developed a software tool (called PROGRESION Appraisal Tracker, see figure 7) for helping the evaluator to identify evidences, consolidate them and deduct the consequences derived from the evidences. This tool allows the user to import the preliminary results (in Microsoft Excel format) from CTP PROGRESION and link them with the objectives and requirements established by the reference model.

PROGRESION Appraisal Tracker also manages an evidences and consequences taxonomy that allows the evaluator, when is consolidating his information, to search previously stored evidences and consequences and attach to the current assessment. This search utilities manages with several synonyms and concepts related that permits a more flexible and efficient search in the taxonomy database.

The human resources employed in the assessment phase have been:

- On the John Doe's Company side:
  - 6 Division Managers
  - 16 Managers of Centers of Experts (Commercial and Logistics:3; Software Development: 2; e-Business: 4; ERPs: 4 and CRM: 3)
  - 108 Project Managers and Engineers (Commercial and Logistics: 20; Software Development: 22; e-Business: 22; ERPs: 17; CRM: 23; Technological Center: 1 and 3 Client Commercials
- On the PROGRESION SMP side, the organizational structure was configured by a Project Manager, an Assessment Coordinator, three workshops conductors, three interviews responsible and three interview assistants.

The assessment final reports audience was John Doe's Company assessment participants, Responsible of organizational units evaluated, and high managers of John Doe's Company. All of these groups were informed separately, but the information transmitted was the same, it only differed in the level of detail.

2.2 Improvement activities

The strategy, which we recommend to follow in the improvements definition and deployment phases, is based on the evolutionary development of processes and process asset libraries, like techniques, tools, etc. This strategy implies the early development of an operative product, so that later successive versions are developed. Each nucleus fulfills minimum and essential requirements and must be modular and flexible, anticipating the future inclusion of more capabilities in a simple way.

This model is being applied in three phases of progressive improvement, on the same process areas that have been analyzed during the assessment phase. Each process area (PA) could be improved in different and sequential phases (one PA in each phase) but we consider that is better to work in several PAs at the same time.

There are two ways to implement this incremental strategy:

1. Deployment of SMALL CHANGES in numerous units
2. Deployment of GREAT CHANGES in not numerous units

The advantages and disadvantages of these two alternatives are shown in figure 8, but for John Doe's Company, we selected the first one.

During the assessment process, high level managers of John Doe's Company mentioned in several occasions that the average time to finalize successfully an improvement program (18 months [4]) is very high and that was necessary to reach earlier achievements visible to all organizations members, in a term of 6 months.

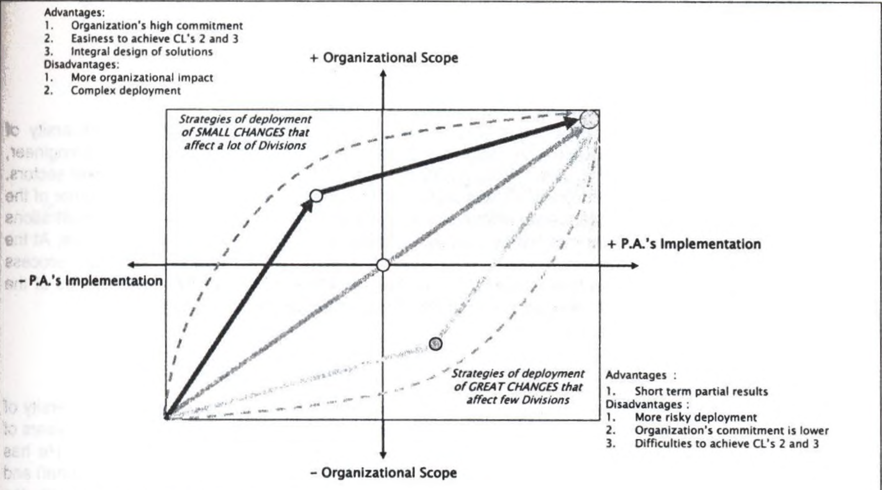

Advantages:
1. Organization's high commitment
2. Easiness to achieve CL's 2 and 3
3. Integral design of solutions
Disadvantages:
1. More organizational impact
2. Complex deployment
+ Organizational Scope
Strategies of deployment of SMALL CHANGES that affect a lot of Divisions
+ P.A.'s Implementation
- P.A.'s Implementation
Strategies of deployment of GREAT CHANGES that affect few Divisions
Advantages :
1. Short term partial results
Disadvantages :
1. More risky deployment
2. Organization's commitment is lower
3. Difficulties to achieve CL's 2 and 3
- Organizational Scope

Figure 8: Transformation Strategy

Due to this circumstance, the improvement deployment strategy is materialized by means of small scale improvement actions performed in the target processes that consist of the introduction of improvements in the processes on small scale, which once tested they are quickly deployed to the whole organization. These small improvements are incremental, and their effect is a gradually improvement of the process capacity visible since the first months. This strategy contributes to increase of the moral of the personnel involved in the SPI project by means of visualizing tangible progresses since very early stages.

It also results in the tendency to simplify the management, support and software engineering processes definitions that must be independent of the technologies used and operated by the different units. These technologies should be considered when we are implementing the procedures to deploy the new processes.

3 Literature

[1] PROGRESION SMP Web Page http://www.progresion.net

[2] International Organisation for Standardization and International Electrotechnical Commission. "ISO/IEC 15504 Software Process Improvement and Capability dEtermination Model (SPICE)", 1997.

[3] Software Engineering Institute. "CMMI for Systems Engineering, Software Engineering, Integrated Product and Process Development, and Supplier Sourcing (CMMI-SE/SW/IPPD/SS, V1.1)", Carnegie Mellon University, March 2002.

[4] McFeeley, B. IDEAL: A User's Guide for Software Process Improvement Handbook CMU/SEI-96-HB-001 Software Engineering Institute. Carnegie Mellon University. February 1996

[5] Emam, Khaled El; Drouin, Jean-Normand; Melo, Walcélio: SPICE: The Theory and Practice of Software Process Improvement and Capability Determination; IEEE Computer Society, Los Alamitos, California, 1998.

[6] Software Engineering Institute. "Process Maturity Profile CMMI V1.1 SCAMPI V1.1 Appraisal Results 2004 Year End Update", Carnegie Mellon University, March 2005.

4 Author CVs

Román López-Cortijo García – Progresion Founding Partner

He received an Engineering degree in Computer Science at Polytechnical University of Madrid. He initiated his career in 1982, since then he has made works as software engineer, consultant and director in Software Engineering companies related to different market sectors. In addition, he has been director and partner of important enterprise groups. Co-author of the methodology Spanish standard software development methodology for Public Administrations called METRICA V3, He has made numerous software process improvement projects. At the moment he is manager director of PROGRESSION SMP, making software process assessment and improvement projects (CMMI and ISO 15504) in several companies of the market sectors such as, telecommunications, bank and industry.

Javier García Guzmán – Progresion Consultant

He received an Engineering degree and PhD in Computer Science at Carlos III University of Madrid. He is software process improvement consultant in PROGRESION. He has 7 years of experience as software engineer and consultant in public and private companies. He has participated in numerous research projects, financed with public (European and national) and private funds, in relation to software process improvement and its integration with the organizational business processes. He has published books and international scientific papers related to software engineering and collaborative working environments. His current research interest is formal measurement of processes improvement, ISO 15504 assessments, software capacity rapid audits, Evaluations for pre-diagnosis according to ISO 15504 and CMMI and management of knowledge related to software engineering.

Antonio de Amescua Seco – Progresion Founding Partner

He received an Engineering degree and PhD in Computer Science at Polytechnical University of Madrid. He is University lecturer of Software Engineer at Carlos III University of Madrid and founding partner of PROGRESION. Their main research areas are software development methodologies and software process improvement. He has been the researcher responsible of the Spanish standard software development methodology for Public Administrations called METRICA V3. He has published several books and more than 100 technical publications on software engineering and management. He is member of the Spanish Association of Software Metrics (AEMES) and of SPIN-Spain (Software Process Improvement Network).

Gonzalo Cuevas Agustín – Progresion Founding Partner

Gonzalo Cuevas attended the High School of Telecommunications, Polytechnical University of Madrid, Spain, where he received an engineering degree in Telecommunications in 1965 and a PhD in Telecommunications in 1974. He also received an MS in Computer Science from the Polytechnical University of Madrid in 1972. He is presently vice dean of the Computer Science faculty at the Polytechnical University of Madrid, where he has been a full professor since 1970. His main research field is software engineering, including both technology (methods, techniques, and formalisms) and management. His current research interest is models and methods for process assessment and process improvement, and also transition packages for software process improvement.

Experiences with Managing Collaborative Projects in an eEurope Environment

Richard MESSNARZ[1],Miklos BIRO[2]
[1]*ISCN LTD, Bray,Co. Wicklow, Ireland*
Tel: +43 316 811198, Fax: + 43 316 811312, Email: rmess@iscn.com
[2]*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

Abstract: This paper discusses the goals , implementation and results of a study where defined quality processes have been implemented in an e-working infrastructure in distributed multinational EU projects. The paper highlights which functions and quality concepts have been highly used and in which areas there are still skills gaps or missing acceptance to apply quality criteria.

The paper concludes which skills are missing and describes a set of initiatives which the EU is supporting until 2007 to resolve the skills gaps.

1. Goals of the Study of Processes and Integrated Systems

1.1 – Applying ISO 15504 Principles on Distributed EU Projects

ISO 15504 defines capability levels for processes. At level 1 the process is existing , at level 2 the process is managed (tracked against objectives, work products under control), and at level 3 the process is defined (defined, tailored and implemented in an integrated infrastructure).

In the EU IST – 2000 – 28162 project we applied these principles on a set of different EU projects in the fields of the

- IST programme
- EU Leonardo da Vinci Programme
- INTERREG programme
- Minerva programme

and

- Defined scenarios of work for different types of EU projects
- Implemented the scenarios on an integrated online available system with tracking features, version control, templates for results, and tailoring opportunities
- Analysed the teamwork and resulting success criteria in distributed projects in 59 European organisations

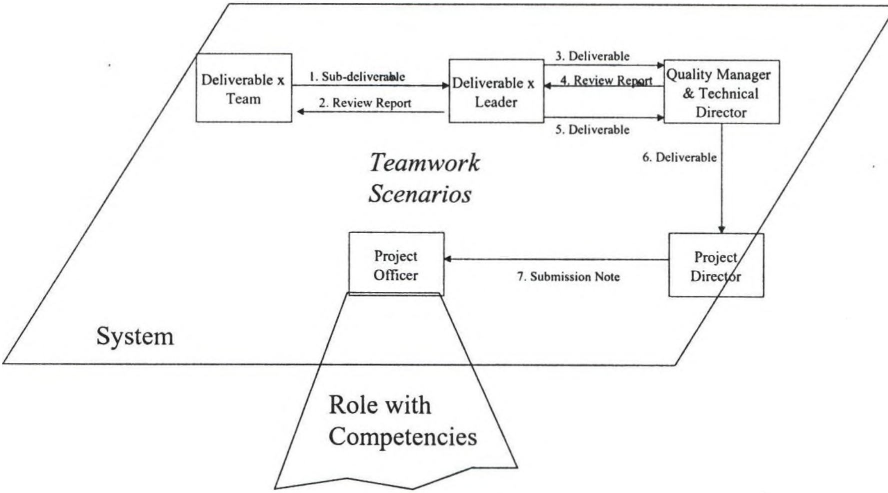


Figure 1: System Architecture

1.2 – Additional Factors Emphasized by the EU

The EU wanted to focus on additional topics such as

- Social competencies required in multinational European teams which act in a distributed manner
- A wizard and fully generic system where the system can be adapted easily to different team-working scenarios (what works in one project might need an adaptation in other projects; sometimes this is even driven by regional demands where the coordinator is placed)
- Thinking about new approaches (even virtual) to analyse if people have sufficient competencies to fulfil certain roles in the teamwork scenarios
- A Re-Use Pool of scenarios concerning best practices for managing IST, Leonardo, Minerva, etc. projects.

1.3 – Innovation as a Key Factor

The driver for the above set of activities between 1996 – 2006 (now continued in the ManagEUr project) were the different innovation studies :

- 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 .

- TEAMWORK (2001-2003) tested an e-working platform applying ISO 15504 principles 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 44 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 result was a pool of best practice working scenarios for different EU project types and a social guide for required team-working competencies.
- A study from WIT at 126 multinational companies and success factors for learning organisations highlighting the stages of learning in a corporate organisation (2002).

2. Implementation of the Study of Processes and Integrated Systems

2.1 –Using a System in Distributed EU Projects

In the projects we used a highly configurable e-working platform TEAMWORK which allows to configure work paces and e-working teamwork scenarios with a wizard (Figure 2). This way the configuration of a new scenario (if necessary) took only 15-20 minutes time. To analyse the required set of scenarios all partners followed a mission and team analysis methodology called BESTREGIT. This methodology provides guielines about how to analyse

- Mission and goals
- Roles and responsibilities
- Teamwork communication charts and work flows

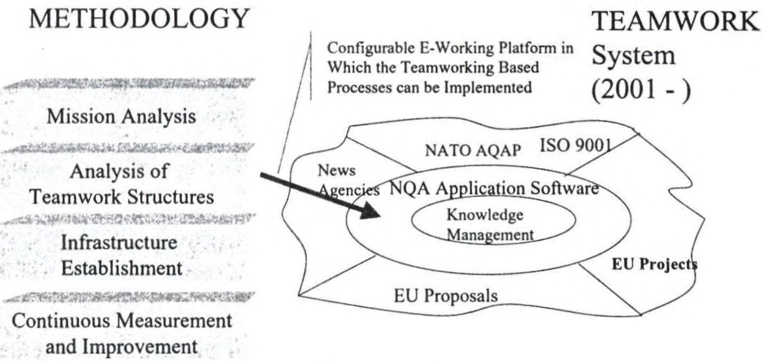
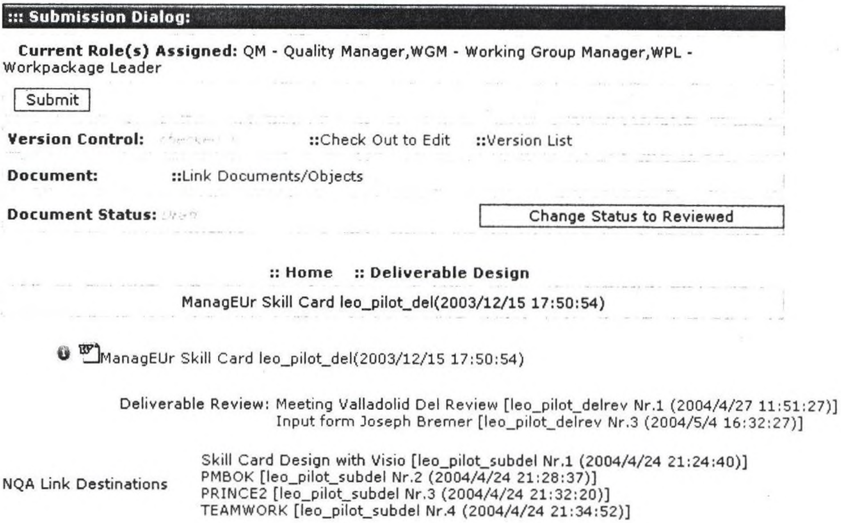

METHODOLOGY
Mission Analysis
Analysis of Teamwork Structures
Infrastructure Establishment
Continuous Measurement and Improvement
Configurable E-Working Platform in Which the Teamworking Based Processes can be Implemented
TEAMWORK System (2001 -)
NATO AQAP
ISO 9001
News Agencies
NQA Application Software
Knowledge Management
EU Projects
EU Proposals

Figure 2: TAMWORK System



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: open

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 Environment – Deliverable Linked With Sub-Deliverables Online

Once the system had been set up with the refined scenarios of work the system offered all projects the below functionality:

Project Types / Work Spaces: The system 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.

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.

2.2 – Implementing Additional Factors Emphasized by the EU

- *Social competencies required in multinational European teams which act in a distributed manner*

Questionnaires and interviews have been used by a social experts 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/).

PATTERNS is a method where a significant number of people is interviewed, all feedback s recorded and then a systematic search of typical answer patterns is done. This pattern recognition method (also as a tool supported analysis) allows to extract typical social patterns from different teams interviewed.

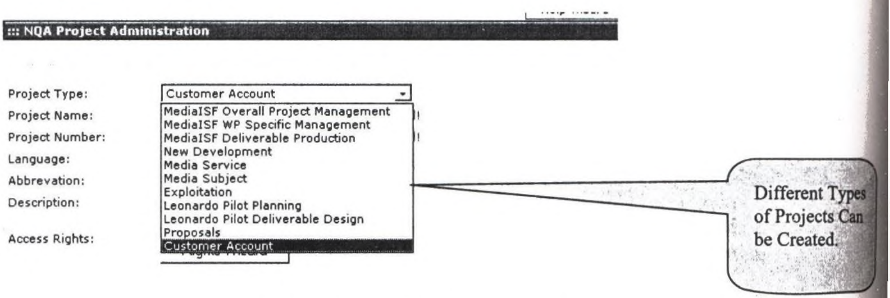
- *A wizard and fully generic system where the system can be adapted easily to different team-working scenarios (what works in one project might need an adaptation in other projects; sometimes this is even driven by regional demands where the coordinator is placed)*

Originally the wizard was not part of the underlying TEAMWRK system and had to be developed during the project.

- Thinking about new approaches (even virtual) to analyse if people have sufficient competencies to fulfil certain roles in the teamwork scenarios

As outlined in Figure 1 the system supports roles and communication flows. Users who receive an account are being assigned to roles inside the working scenarios. The major question in distributed EU projects is, do the users (partners) have sufficient skills for the role they play (e.g. assigning someone to be the quality manager of the project). Thus the idea was created to integrate the TEAMWORK system with a skills analysis system in he future. If someone is assigned to a role we can configure and test if the user would have skills gaps to fully occupy that role.

- A Re-Use Pool of scenarios concerning best practices for managing IST, Leonardo, Minerva, etc. projects.



::: NQA Project Administration

Project Type: Customer Account

Project Name:

Project Number:

Language:

Abbrevation:

Description:

Access Rights:

MediaISF Overall Project Management
MediaISF WP Specific Management
MediaISF Deliverable Production
New Development
Media Service
Media Subject
Exploitation
Leonardo Pilot Planning
Leonardo Pilot Deliverable Design
Proposals
Customer Account

Different Types of Projects Can be Created.

Figure 4: Re-Use Pool of Working Spaces Structure

Originally this re-use pool was not part of the underlying TEAMWORK system and had to be developed during the project. If we defined once a best practice set of working scenarios about how an EU Leonardo project is to be planned, the server saved this structure as a project type. Whenever a user in future has to plan an EU Leonardo project he just selects and re-uses that structure and automatically creates a new work space with this defined best practice structure.

3. Results of the Study of Processes and Integrated Systems

3.1 –Feedback concerning the System Usage in Distributed EU Projects

The system meanwhile is used in a set of different EU initiatives (INTERREG, Leonardo, IST , Minerva) . The feedback is :

- In general the users feel that the system helps to control the project and supports working in virtual teams.
- However, there seems to be a missing acceptance for the required set of quality criteria as follows –
  - The system forces users to apply version and change control. If a user uploads a new file the user has to select the previous version, document changes, and upload the new file as a new version . This way the system administers a set of versions for one file, displaying version histories. It seems that EU managers still think I terms of files (10 versions = 10 files on a server) instead of version control (one file with a history of 10 versions). About 50% of the EU managers in EU projects need an extra training to understand basic configuration management technologies.
  - The system forces users to use certain templates and to publish files under pre-defined document categories. It seems that EU managers still think in terms of file directories (many files shared in a directory on a server) instead of

structured document control (to categorise ach file by type – plan, deliverable, review, etc. – and publish under document categories). About 30% of the EU managers in EU projects need an extra simple publish script so that they do not need to understand what document category they use.
    - The system forces users to set a certain status (draft, reviewed, approved) for uploaded materials. None of the EU managers used this and it always had to be done by the quality manager in a review afterwards.
- The following functions have been extensively (by more than 80% of the EU managers)
    - The submission function which automatically submits the materials to the corresponding team members.
    - The discussion portal to discuss and agree deliverables.
    - The search function to find materials on the server.
    - The download of materials, once they were structured and available on the server.

This result illustrates that EU projects (if not managed by a good quality manager in the middle) would (even if the system is place) have troubles achieving a defined and deployed level 3 in ISO 15504. On the other side the analysis of 44 projects in the trial sample showed that those who could manage quality in the results were hose who delivered in time at proper quality (with a chance to sustain after the funding ends).

This again resulted in two further decisions / initiatives supported by the EU Leonardo da Vinci programme

(a) There must be a qualification programme which trains EU managers in the above skills (so that quality functions are used as well) – *the ManagEUr project.*

(b) There must be a certificate (issued by an EU supported qualification body) which future EU managers should have before leading EU projects – *the EQN certification initiative.*

3.2 – Feedback Concerning Additional Factors Emphasized by the EU

Some selected findings of the pattern analysis were [3], [4], [8]:

COMMUNITY OF TRUST

„If you are building any human organization Then: you must have a foundation of trust and respect for effective communication at levels deep enough to sustain growth.“

Remark: The interviews and our experience show that a community of trust was established during the trials. The formation of a “community of trust” was the result of meetings creating positive relation between the participants, of a common interest in the success of the project, and an efficient and open management style. For applications of TEAMwork in the future it should be explicitly checked which factors in an organisation could endanger mutual trust. It is decisive to create and maintain human relationships between the participants around the formalised communication via the NQA server.

WORK FLOWS INWARD

„If you want information to flow to the producing roles in an organization Then: put the developer at the center and see that information flows toward the center, not from the center."

Remark: This pattern was realised by the configuration of the roles of the participants. The result was, that none of the interviewed persons complained about missing information. In the future this pattern should be used explicitly when flows of information in a TEAMwork project are defined.

TEAM PER TASK

„If A big diversion hits your team, Then: Let a subteam handle the diversion, the main team keeps going."

Remark: This pattern was a guiding principle of the organisation of the trials with different teams working in different domains and subteams of these teams responsible for individual tasks. During the trials there was no threat of a major diversion. In future applications of TEAMwork an „emergency team" should be defined and nominated in order to be able to handle such cases.

HOLISTIC DIVERSITY

„If Development of a subsystem needs many skills, but people specialize, Then: Create a single team from multiple specialties."

Remark: This pattern was implicitly realised by the whole TEAMwork team and the communication of people with very different skills and professional background. Here again TEAMwork seems to be especially appropriate for heterogenous teams.

The role based approach of TEAMwork makes this pattern a useful supplement to the TEAMwork methodology. The roles themselves say nothing about individual skills of the TEAM members. The combination of people with different skills in a team is especially necessary if the team has to fulfil tasks that cannot be completely formalised and represented by the configuration of the NQA-Server.

3. Conclusion and the EU Manager Certificate

The study showed that systems would be ripe enough to enable a defined quality level for distributed EU projects but there is a significant skills gap to accept the basic quality concepts offered by systems.

Therefore the EU Leonardo da Vinci programme supported an EU project to develop these skills for managers of distributed (EU) projects.

The project ManagEUr (Leonardo da Vinci HU/B/03/F/PP-170028) has developed (2003 – 2006) -

- a skills set ad self assessment portal so that people currently managing EU projects can register online, do a self assessment, see their skills profile (matched against the skills requested by the studies) and can browse recommended learning references. See http://www.iscn.com/projects/manageur_skill_portal/ .

- a set of training courses for 23 learning objectives identified in the studies and they are offered by partners since early 2005.
- a standard ECDL like test which is supported by a European portal and dynamically generates multiple choice based certification – test questions for managers performing the test.
- a certificate which managers who pass the test can achieve.
- a demo and product CD based on the IST-2000-28162 results to offer EU project management demo portal systems.

In the EQN (European Quality Network) project the EU Leonardo da Vinci programme supports the establishment of a certification unit for qualifications like the certified U project manager (2005 – 2007).

References

[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/biro.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/biro.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, Wash-ington, 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.

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, Media-ISF –

e-working 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 also a lead SPICE assessor and has worked as consultant over 15 years for many automotive firms.

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 designes 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 impovement (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 Jonh 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.

Globalisation - Sourcing by Virtual Collaboration?

Kerstin V. Siakas[1], Bo Balstrup[2]

*[1] Alexandreio Technological Educational Institute of Thessaloniki,
Department of Informatics, P.O. Box 141, GR-57400 Thessaloniki,, Greece*
E-mail: siaka@it.teithe.gr

*[2]Center for Software Innovation,
Stenager 2, Sonderborg, Denmark*
E-mail:bb@cfsi.dk

Abstract

In today's rapidly changing and highly competitive global environment offshore outsourcing (contracting of a portion or all of the software development function to software engineers outside the home country), teams working across national borders and virtual organisations have become a fact. The globalisation of the software market has also changed the contextual boundaries of IS research and practices to include the wider societal context. Management experiences difficulties when applying traditional management approaches, because of the increased complexity of global organisations and their dependency on people with different underlying norms, values and beliefs. Researchers call for caution in managing global workers. They argue that behavioural traits of diverse work groups can contribute to dysfunctional organisations. Cultural sensitivity has become a core issue.

This paper discusses the implications of the globalisation in software development and seeks to make more explicit the human dynamics that bear on the success of outsourcing and virtual collaboration. Two models, namely the e-Sourcing Capability Model for Service Providers (eSCM-SP) and the Software Quality Management – Cultural and Organisational Diversity Evaluation (SQM-CODE) Model, are proposed to bring added value for service purchaser in their search for, selection of and collaboration with service providers.

The eSCM-SP, a capability maturity assessment model related to IT-enabled sourcing, is briefly discussed and the SQM-CODE model, a tool for assessing the fit between organisational and national culture, is presented and its importance for identifying cultural factors and taking appropriate action in order to achieve a cultural fit between the service provider and the contractor in outsourcing business partnerships is revealed.

Keywords

Sourcing, Virtual Collaboration, Cross-cultural Management, Culture

1 *Introduction, Motivation and Perspectives*

The management of cultural diversity is becoming a significant issue for companies. Due to the emergence of global organisations, increasing number of joint ventures and cross-national partnerships, businesses need to embrace people from a variety of ethnic backgrounds and cultures. This has created a new awareness of the importance of understanding other cultures and has contributed to the need to develop a cultural sensitivity. Problems between mother organisation and subsidiaries, resistance, low motivation and low productivity can arise in international operations because of cultural ignorance or insensitivity (Siakas et al., 2003).

Being a global organisation implies having a universal culture (Joynt and Warner, 1996). The objectives of the global organisations are to create a universal culture in the whole organisation and to integrate multi-domestic operations with individuals who hold opposed work-related values.

The contextual boundaries of Information Systems (IS) research and practices have changed to include the wider societal context due to the globalisation of the software market. Cross-cultures comparative studies have made attempts to document differences in value systems of managers. There are two views on managing IS in a global context.

One view proposes that managing IS in a global context is largely the same as managing IS in a domestic context and the managers will display similar managerial values despite their cultural differences (Sparrow et al., 1994; Ein-Dor and Segev, 1993). In global companies with strong organisational culture managers usually demonstrate similar management style (Land, 1992). This implies that the impact of culture on organisational systems will decline.

The other view proposes that cultures are deeply rooted in individuals and thus the managerial value differences exist due to cultural aspects, different business and legal environments, different languages and varying technology availability (Ives and Järvenpää, 1991; Järvinen, 1997; Tractinsky and Järvenpää, 1995). This implies that organisations are affected by national cultures and seem to resist the convergence effect of international business (Clark and Mueller, 1996; Hofstede, 2001).

The cultural orientation of a society reflects the complex interaction of values, attitudes and behaviours displayed by its members. They are all part of the cultural learning and give rise to misunderstandings and misinterpretation of intent. In global teams there is an acute need for workspace awareness due to limited access to informal communication. Awareness of cultural differences backed up by cross-cultural training becomes an important factor for success in international business.

2 Globalisation

Globalisation is a business fact, expanded worldwide beyond domestic boundaries, which is creating an interconnected world economy in which companies do their business and compete with each other anywhere in the world, regardless of national boundaries (Cullen, 1999). Globalisation today is an inevitable reality that cannot be ignored. Saee (2002) states that globalisation has been beneficial to nearly all countries around the world, including advanced industrial countries, the emerging economies and many of the world's low-income countries with exception of a few countries which for ideological reasons have chosen not to pursue free trade. New Information and Communication Technology (ICT) capabilities have increasingly facilitated globalisation. Results will be dependent of development policies and strategies.

The globalisation does not imply homogeneity of cultures (Walsham, 2001). In general, research in IS has not considered culture when investigating the process of software development. Particularly quality related issues seem to be missing in the literature. However, many researchers have compared national cultures with organisational cultures (Clark and Mueller, 1996; Morden 1999; Smith, 1996).

Hofstede (2001) provided strong evidence that national cultural differences shape organisational behaviour at a local level, and that differences in national and regional cultures affect work values. He argued that culture is a collection of characteristics possessed by people who have been conditioned by similar socialisation practices, educational procedures and life experiences. Krishna et al. (2004) affirm that major differences in norms and values cannot be harmonised since they derive from deep-rooted differences in cultural background, education and working life.

In global organisations the organisation culture in the form of assumptions, beliefs, attitudes and values are shared by existing members and taught to new members of the organisation and by promoting a strong organisational culture without disproving and demolishing local converging values and attitudes success is more likely. Similarly shared stakeholder values are considered to be important for success (Hofstede, 2001; Land,1992; Schein, 1985). Organisational culture affects directly individual behaviour by imposing guidelines and expectations for the members of the organisation. One of the key issues for managers in global organisations is integration across geographic distance and cultural diversity (Siakas, 2002).

Organisational culture is mainly created and maintained in existing frameworks by the founders and the leaders of an organisation through their value system (Bryman,1992; Schein, 1985). Three of the most important sources of organisational cultures are according to Brown (1998):

- societal or national cultures within which an organisation is physically situated;
- the vision, management style and personality of the founder and other dominant leaders of the organisation;
- the type of business an organisation conducts and the nature of its business environment.

Within a single culture certain values, attitudes and behaviours are either favoured or suppressed. In a global organisation conflicts and dissatisfied employees will most likely be the result if the values of the employees of the mother organisation are divergent from the values of the employees in a subsidiary (Siakas, 2002).

The challenges globalisation offer, originates from social, economical, legal, political and technological differences between nations, together with cultural differences regarding work values, attitudes and preferences both of employees and consumers. Given the complex nature of globalisation, organisations need to develop high cross-cultural understanding, intercultural communication skills and intercultural management competencies. Management of global organisations that can take account of the cultural context of their endeavours experiences better success.

In a truly global company the top management and board must be prepared to travel to the subsidiary in order to enable the company to make fast decisions. Large companies have a tendency to be inflexible and do not realise the difficulties they face when trying to remotely control overseas subsidiaries. The large company must be as diligent as its local, small competitors, especially on the emerging markets like China, where the local management needs more involvement from the mother organisation in order for the internal decisions not to become a bottleneck for business growth. (Thomsen, 21.4.2005).

Nothing slows down an organisation more than the inability to make even smallest decisions quickly (Jennings and Haughton, 2000). Instead of bureaucracy global organisations need to be flexible. Flexibility can be improved by the creation of simple and understandable guiding principles regarding decision making, new actions, initiatives, new products and new directions. The guiding principles need to be communicated to subsidiaries and shared by everyone in the organisation in order to strengthen the organisational culture.

3 Sourcing

Since the Industrial Revolution, companies have struggled with how to exploit their competitive advantage in order to increase their market share and profits. Sourcing is an activity where one organisation provides services to another organisation. Outsourcing entails purchasing a product or process from an outside supplier rather than producing this product or process in-house. The concept of allocating business activities to another organisation was initially used for manufacturing of industrial components. The emphasis for large integrated companies was how to own, manage and directly control their assets (Northfield, 1992). In this sense outsourcing has been used since the mid twenties. In the fifties and sixties, the emphasis was on diversification and to broaden corporate bases. In the seventies and eighties many large companies developed new strategies focusing on their core business, identification of critical processes and choice of processes that could be outsourced (Northfield, 1992). Thus data processing entries was commonly outsourced and in the nineties outsourcing was used for entire IT operations and IT intensive business processes.

The evolution of the internet has increasingly facilitated organisations to delegate part of their business activities to external service providers and to established business partnerships beyond geographical boundaries. The business activities being outsourced range from resource intensive operational tasks to critical strategic business processes. More recently IT intensive projects and tasks are being increasingly outsourced. Companies may have multiple sourcing relationships in different countries.

Although the primary motivation for outsourcing is cost-effectiveness, there are many transaction costs that should be taken into consideration including the costs of continuous liaison with outsourcing providers, including organising and monitoring projects, as well as evaluating outcomes, handling legal matters such as signing contracts, ownership and copyright issues. Also potential risks need to be quantified and taken into account. Such potential risks may e.g. include potential costs of contract failure.

The activities in sourcing can be grouped into (Siakas et. al. 2005):

- Process Outsourcing: The entire business process, often including personnel and resources, such as computers and software, is outsourced and the process is transferred to the service provider
- Project Outsourcing: A subset of activities with a specific dead-line, such as e.g. a software development project, is outsourced
- Task Outsourcing: Outsourcing of usually high demand tasks, such as unplanned demand which can not be met by the internal resources in the organisation
- Insourcing: People are brought into the organisation in order to meet certain requirements and activities within a specific dead-line or to add knowledge and skills in order to reduce risk

Outsourcing allows organisations to focus on broader business issues and to redirect resources from non-core activities toward research, development and activities that provide a greater return. Simultaneously organisations, by having operations accomplished by outside service providers, who usually are experts in the field, gain access to world class capabilities, such as new technologies, tools, methodologies and procedures that the organisation may not currently possess. Also advantages, such as improved business focus, competitive advantage through expanded skills, continuous improvement and adoption of best practices are likely to take place. As a result customer companies will be enabled to achieve faster, more efficient, effective and more economical business processes.

When companies outsource they become more flexible, more dynamic and more able to meet the changing opportunities. However, organisations should be aware of the risks and dangers that outsourcing can cause like dysfunctional organisations due to loss of control and in-house expertise, cultural differences and dilution of the company knowledge base when outsourcing. Outsourcing is also a vehicle for sharing risks across many companies. Service providers make investments not only for their own company but on behalf of their many contractors. By sharing these investments, the risks are significantly reduced. Also for organisations that lack required resources for applications needed to be developed or modified outsourcing can be a good solution.

Regarding software outsourcing there seem to be more awareness of cultural issues than in the literature about management of software development and information systems in general. The literature about outsourcing seems to propose:

- Recognition of the fact that cross-cultural training is needed both in advance and continuously (Foster, 2000)
- Use of 'cultural bridging staff' (people rooted in the country of the sourcing service provider as well as in the country of the client) for informal sharing of experiences (Krishna et.al., 2004)
- Use of common systems, common processes and common compatible technologies (Heeks et. al., 2001)

- Recognition of the importance of the communication language (Foster, 2000).

In the Knowledge Society skill based work will be outsourced and creative, innovative knowledge intensive work will remain, but high quality design requires a profound experience and insight in the methodologies and tools used to implement and produce the product.

4 Virtual Collaboration

"Unlike conventional teams, a virtual team works across space, time, and organisational boundaries with links strengthened by webs of communication technologies" (Lipnack and Stamps, 1997).

The leadership traits and skills needed with virtual teams are not different from those used with collocated teams. The difference is in the way they are exerted to create the desired results.

Collaboration has three facets (Balstrup, 2004), namely:

- Collaboration within each collocated group
- Collaboration between dispersed group of the virtual team
- Collaboration between the groups and the leader

A potential conflict arises when the team consists of members from different organisational units, because the team does not know where to place its loyalty. In virtual environment this is amplified, because informal communication is reduced (members seldom meet face-to-face). Lewis (1999) stated that "Language is a poor communication tool unless each word or phrase is seen in its original cultural context". Therefore a successful leader of a virtual team must excel in applying the right choice of communication means along with a profound knowledge of the effect of applying it.

Teamwork is in essence a result of human interaction, but, in an environment where organisations formulate strategies for becoming global, working in a common place becomes less common. Two important factors for supporting collaboration are loyalty and commitment. The individuals of the virtual team and the leader must build a cohesive team committed to the common goal and through interdependent interaction generate group identity and create the feeling of belonging to the "we" group (Balstrup, 2004). Creation of cohesion is fragile and requires effective interpersonal leadership.

The cultural dimension divides the teams into culturally homogeneous and heterogeneous teams. Culture is the most difficult to assess as it embraces facets like language, tradition, values, core beliefs, humour and many more. The virtual leader must posses a profound understanding of the cultural differences within the team. Additionally the leader needs an employee at the distant location who is a valid substitute for him. An employee who through self-management supported by a trusting delegation performs leadership and who loyally exerts the chosen strategy and direction. The benefit of the local leader is to have a person who is able to transform the leadership into the local cultural context and a person to whom the local team can attach in the absence of the virtual leader (Balstrup, 2004).

In a cross-national study of managerial value (Bigoness and Blakely, 1996), used on 567 managers from twelve nations it was found that regardless of nationality the value dimension that included the instrumental value "broadminded, capable and courageous" was ranked as the most important value dimension by all managers.

Particular traits are positively related to successful leadership. This requires the successful leaders to acknowledge and reflect on their strengths and weaknesses. They must be motivated to continuously develop themselves and be aware that a present strength can turn into future weakness. Additionally they must be able to compensate for their own weaknesses by selecting employees or external service providers with complementary strength and empower them to take on the tasks they are more qualified to perform (Balstrup, 2004).

5 Assessment of capabilities of IT-enabled service providers

The effective management of cultural diversity in a global context is a challenge and a competitive advantage. Potential for increased ambiguity, complexity and confusion occurs in situations where a single agreement has to be taken or when overall procedures have to be developed (Adler, 1997). Managers who are involved in cross-cultural communications and negotiations need to develop characteristics such as cultural sensitivity, flexibility and adaptability. However, will these characteristics guarantee success in a multicultural environment? Managers seem to be worried about their own capabilities to be successful in an increasingly complex global context. In order to help service purchasers to search, select and collaborate with service providers we propose the use of two models, namely the eSCM-SP (a capability maturity assessment model) and the Software Quality Management Model - Cultural and Organisational Diversity Evaluation (SQM-CODE) model (a model for assessment of the fit between national and organisational culture).

In the eSCM-SP it is argued that it is important to identify cultural attributes that impact on service and implementation actions in order to support the close coordination necessary to meet client requirements. The model states that "*multi-national and organisational differences between the client and the service provider, and cultural differences within the provider may impact the quality of interactions and the overall quality of the sourcing relationships*" (Hyder et al., 2002). It is mentioned that important cultural factors need to be identified and appropriate action to be taken in order to achieve a cultural fit between the service provider and the client/contractor. However it is not mentioned which cultural factors are important, how to identify them or what kind of appropriate action should be taken. Nevertheless, the fact, that a cultural fit between the client and the service provider may impact the quality of interactions, is explicitly mentioned, seems to be a first step in recognition of the importance of a cultural fit between factors from the external environment (national culture) and the internal organisational environment (organisation).

Research has tried to identify cultural factors including national factors that influence the effectiveness of Software Process Improvement (SPI) (Biró et. al., 2001; 2002, Siakas, 2002, Siakas and Balstrup, 2000). Siakas (2002) found that a fit between national and organisational culture plays an extremely important role in all kinds of organisations that promote a climate of satisfied employees and decreased misunderstandings and conflicts due to cross-cultural issues. She also found and statistically proved that a cultural fit between national and organisational culture in global organisations is significant for obtaining commitment and avoiding resistance when introducing change in software quality management issues. A model, called SQM-CODE, Software Quality Management – Cultural and Organisational Diversity Evaluation was developed as a result of her research to assess the organisational and the national culture in order to find the cultural fit (Siakas et al., 2003). The model can also be used as a tool for assessing the cultural fit between a service provider and a contractor in an outsourcing business partnership.

5.1 The e-Sourcing Capability Maturity Model

The eSCM-SP (eSourcing Capability Maturity Model for Service Providers) (Biró et. al., 2003; Hyder et al. 2004a; 2004b, Hyder et al., 2002) provides IT-enabled sourcing service providers with a reference model that addresses critical issues related to IT-enabled sourcing (eSourcing) and aims to help them in establishing, managing and continuously improving relationships with contractors and in improving their capabilities in developing products and services. It was initially developed to be used by the service providers for capability determination and improvement in order to continuously deliver high quality services, as well as to prove their capabilities by certification at a capability level. The eSCM-SP has been designed to complement existing quality models so that service providers can capitalise on their previous improvement efforts.

Version two of the eSCM-SP, which was released in April 2004, has 3 purposes (Hyder et al. 2004a; 2004b):

1. to give service providers guidance to help them improve their capabilities across the sourcing life-cycle

2. to provide clients with objective means of evaluating the capabilities of service providers
3. to offer service providers a standard that they can use when they want to differentiate themselves form their competitors.

The eSCM-SP consists of 84 practices. Each practice in turn is organised into three dimensions, namely sourcing life-cycle, capability area and capability level. The sourcing life-cycle dimension describes the specific phase to which the sourcing service belongs in the life-cycle. The sourcing life-cycle dimensions are: ongoing (covers entire life-cycle), initiation, delivery and completion. The capability area consists of ten logical groupings of the practices and aim to help users better remember the content of the model for managing more effectively, build and/or demonstrate capabilities in each critical sourcing function. The capability level contains five capability levels starting from a desire to provide eSourcing services, meeting clients requirements, controlling through measurement and enhancing through innovation until the highest level of sustaining excellence. The focus in all capability assessment models is the disciplined examination of the processes used by an organisation against a set of criteria to determine the capability of those processes to perform within quality, cost and schedule and the ultimate goal is to improve the capability of the processes (Siakas, 2002).

In the eSCM-SP there are many similarities with other capability assessment models like CMMI (Capability Maturity Model Integrated), Bootstrap or SPICE (ISO-15504), as to the belief, deriving from the Total Quality Management philosophy, that the quality of products is mainly determined by the quality of the processes, which produce them, that process performance depends to a great deal on individuals performance and the more mature a process is the better the performance. To improve the capability of a process, the first step is to understand the status of the process and to have a frame or path to follow for improvement.

The eSCM-SP, like other similar models, places proven practices into a structure that helps organisations to assess their maturity, establish priorities for improvement and guide the implementation of these improvements. However, the emphasis in the eSCM-SP model is clearly to a higher degree on human issues, like developing and sustaining stakeholder relationships and building and keeping competent workforce than earlier capability assessment models. The situation is also different, because in a sourcing agreement there is always two partners, the serviced provider and the contractor, who both have strong interest in a successful partnership and reduction of risks.

The use of the eSCM-SP is valuable for IT-enabled sourcing service providers, who want to appraise and improve their ability to provide high quality sourcing services, reduce risks, add value to their operations and to differentiate themselves from competitors. The service providers contribute to the overall success of the contractor organisation. Thus it is in the contractor's interest to choose a technically competent service provider, with a cultural and ethical attitude compatible to the contractor.

In this paper we argue that it would be valuable for contractors to use the eSCM-SP model as a means to assess the capability of potential service providers during the searching and selection process and to determine and continuously improve the capability of existing service providers during the collaboration process. In eSourcing there are also critical issues associated with initiation and completion of the contract between the service provider and the contractor.

5.2 SQM-CODE

The main objective of the research study that lead to the development of the SQM-CODE (Software Quality Management: Cultural and Organisational Diversity Evaluation) model was to add to the knowledge of what factors influence successful software quality management systems (Siakas, 2002). The research in particular examined those factors, which form a cultural and organisational perspective. The research question that the study addressed was to what extent cultural factors influence the successful adoption and implementation of a Software Quality Management System. The analysis considered factors from both national and organisational areas. The existence of quality oriented management procedures (similar to the procedures identified in Capability Models), was investigated empirically, together with the awareness of quality issues amongst the workforce.

The research method used was a contemporary comparative multimethod also called triangulation using both quantitative (307 questionnaires) and qualitative investigation (87 interviews) in organisations developing software in Denmark, Finland, Greece and the UK. Consequently, and by its very nature the investigation utilised the strengths of cross-national comparative studies.

The SQM -CODE model assesses the cultural fit between national culture and organisational culture and comprises two sub-models, namely the C.HI.D.DI. typology and the Authoritarian-Participative model (Siakas, 2002).

The C.HI.D.DI typology, which is based on Hofstede' s Power Distance and Uncertainty Avoidance dimensions, classifies organisations into four dimensions namely ***C**lan, **HI**erarchical, **D**emocratic* and ***DI**sciplined.* This classification defines the national culture. Simultaneously a suitable software quality management system is proposed.

The Authoritarian-Participative sub-model defines the organisational culture considering organisational characteristics such as organisational structure, degree of formalisation, management style, leader's role, handling of rules and degree of control.

The final two-axed values obtained from both assessments show the cultural fit. The closer the values are the better the fit. The values can be plotted into the four quadrants of the C.HI.D.DI typology for graphical representation.

The self-assessment of the SQM-CODE will give a fast response regarding the basic underlying cultural fit or dichotomy between organisational and national culture. A full SQM-CODE assessment includes an in-depth analysis aiming to identify critical cultural factors and to propose appropriate action in order to achieve a cultural fit. Our findings from a field-study (Siakas, 2002) showed statistically significant evidence that if there is a fit between the organisational and the national culture, then there is a higher employee satisfaction and problems are solved more smoothly. A dichotomy is highly likely to generate dissatisfaction, conflict and ultimate failure.

Global organisations would benefit of using the SQM-CODE in their subsidiaries. The organisational culture in the mother organisation might not be suitable in other countries. The mother organisation has to be aware of the differences in cultures and be flexible enough to take into consideration differences between the organisational and the national culture.

For organisations which aim to delegate part of their business activities to external service providers and/or to established business partnerships beyond geographical boundaries an assessment of the cultural fit between the contractor (the global organisation) and the business partner (service provider) would be beneficial in order to save a lot of effort and money on business which do not have the cultural foundation to be successful without a lot of dynamism and hard work.

5 Conclusion

Globalisation today is a reality having created numerous of challenges for managers worldwide. Increased and improved capabilities of ICT facilitate continuous expansion of globalisation. Outsourcing and virtual collaborations prompt for cultural sensitivity, flexibility and adaptability, together with high awareness of risks and dangers due to cultural differences. Globalisation is a competitive advantage if handled in a right manner. In order to help service purchasers to search, select and collaborate with service providers we proposed the use of two models, namely the eSCM-SP (a capability maturity assessment model) and the Software Quality Management Model - Cultural and Organisational Diversity Evaluation (SQM-CODE) model (a model for assessment of the fit between national and organisational culture).

Top quality and just in time approaches together with low-cost products and services have become core values. A critical core factor is culture. Emphasis is put on understanding, managing and taking advantage of cultural differences not only amongst employees in the global organisation or between service providers and contractors in outsourcing business partnerships but also of cultural differences between clients in the global market. Apart from the use of the eSCM-SP and the SQM-CODE mentioned above it is crucial for the purchasing company to master the "art" of virtual collaboration and be highly aware of the risks and dangers that outsourcing can cause like dysfunctional organisations due to cultural differences and dilution of the company knowledge base when outsourcing. In the Knowledge Society skill based work will be outsourced and creative, innovative knowledge intensive work will remain, but high quality design requires a profound experience and insight in the methodologies and tools used to implement and produce the product.

Literature

Balstrup Bo (2004): Leading by Detached Involvement – Success factors enabling leadership of virtual teams, MBA Dissertation, Henley Management College, UK

Bigoness William J., Blakely Gerald L. (1996): A Cross-national Study of Managerial Values, Journal of International Business Studies, Fourth Quarter

Biró M., Deák G., Ivanyos J., Messnarz R., Zámori Á. (2003): Using the eSourcing Capability Model to improve IT enabled business process outsourcing services. *EuroSPI 2003* (European Software Process Improvement Conference), Graz, Austria, 10-12.12.2003, pp.III.1-III.16

Biró M., Messnarz R., Davison A.G.(2002): 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

Biró M., Messnarz R., Davison A.G.(2001): Experiences with the Impact of Cultural Factors on SPI. *EuroSPI 2001*(European Software Process Improvement Conference)

Brown, A. D. (1998): Organisational Culture, Financial Times Management, Pitman Publishing

Bryman, A. (1992): Charisma and leadership in organisations, London, Sage Publications

Clark P., Mueller F. (1996): Organisations and Nations: From Universalism to Institutionalism?, British Journal of Management, Vol. 7, pp. 125-139

Cullen J. B. (1999): *Multinational Management: A strategic Approach*, Cincinatti, Ohio, South Western College Publishing

Ein-Dor P., Segev E. (1993): The Effect of National Culture on IS: Implications for International Information Systems. Journal of Global Information Management, 1, pp. 33-44

Foster N. (2000): Expatriates and the impact of Cross-Cultural Training. *Human Resource Management Journal*, Vol. 10, No 3, pp. 63-78

Heeks R. B., Krishna S., Nicholson B., Sahay S. (2001): Synching or Sinking, Global Software Outsourcing Relationships, IEEE Software, 18, 2, March/April, pp. 54-61

Hofstede Geert (2001): Culture's consequences: comparing values, behaviours, institutions, and organisations - 2nd Ed. - Thousand Oaks California, Sage Publications

Hyder Elainei B., Heston Keith M., Paulk Mark C., (2004a): The eSourcing Capability Model for Service Providers (eSCM-SP) v2, Part 1: Model Overview, CMU-ISRI-04-113, Pittsburg, PAL Carnegie Mellon University

Hyder Elainei B., Heston Keith M., Paulk Mark C., (2004b): The eSourcing Capability Model for Service Providers (eSCM-SP) v2, Part 1: Practice Details, CMU-ISRI-04-114, Pittsburg, PAL Carnegie Mellon University

Hyder Elaine B., Kumar Bennet, Mahendra Vivek, Siegel Jane, Heston Keith M., Gupta Rajesh, Mahaboob Habeeb, Subramanian Palanivelrajan (2002): The e-Sourcing Capability Maturity Model (eSCM-SP) for IT enabled Service Providers, v1.1, CMU-CS-02-155, School of Computer Science, Carnellie University, Pittsburg

Ives B., Järvenpää S. (1991): Applications of Global Information Technology: Key Issues for Management, MIS Quarterly, Vol 15 March, pp. 33-49

Jennings Jason, Haughton Laurence (2002): It's Not the Big That Eat the Small... It's the Fast That Eat the Slow : How to Use Speed as a Competitive Tool in Business, HarperBusiness Publishers,

Joynt Pat, Warner Malcolm (1996): Introduction: Cross-cultural perspectives Managing across Cultures: Issues and Perspectives, by Joynt Pat, Warner Malcolm (Ed.): International Thomson Business Press

Järvinen, P. (1997): On cultures and information technology applications in organisations in An Ethical Information Society, J. Berleur, D. Whitehouse (eds.): IFIP, Chapman and Hall

Krishna S., Sahay Sundeep, Walsham Geoff (2004): "Managing Cross-cultural Issues in Global Software Outsourcing", Communications of the ACM, April Vol. 47, No 4

Land F.F. (1992): The Management of Change: Guidelines for the Successful implementation of Information of Information Systems, in Brown, A. (eds): Creating a Business Based IT Strategy, Chapman & Hall

Lewis Richard D. (1999): When cultures collide: managing successfully across cultures, London, Nicholas Brealey

Lipnack Jessica, Stamps Jeffrey (1997): Virtual Teams: Reaching across space, time and organisations with technology, Wiley, New York

Morden, T. (1999): Models of National Culture – A Management Review, Cross Cultural Management, Vol. 6 No 1

Northfield Dianne (1992): Outsourcing of IT Services: Case Study of the Contract Between the Tricontinental Royal Commission and ICL Australia Pty Ltd, *CIRCIT*, April 1992, http://www.circit.rmit.edu.au/rr.html

Saee John (2002): *Strategic Global management: Cross-Cultural Dimension*, , France, Normedia Publishing House

Schein E. (1985): *Organisational Culture and Leadership*, London, Jossey-Bass Ltd.

Siakas Kerstin V., Balstrup Bo, Georgidaou Elli, Berki Eleni (2005): Global Software Development; the Dimension of Culture, *IADIS (International Association for Development of the Information Society) International Virtual Multi Conference on Computer Science and Information Systems (MCCSIS 2005) - SEA (Software Engineering and Applications)*, 26 April 2005

Siakas Kerstin V., Berki Eleni, Georgiadou Elli (2003): CODE for SQM: A Model for Cultural and Organisational Diversity Evaluation, *EuroSPI 2003* (European Software Process Improvement Conference), Graz, Austria, 10-12.12.2003, pp. IX 1-11

Siakas Kerstin V., (2002): *SQM-CODE: Software Quality Management – Cultural and Organisational Diversity Evaluation*, PhD Thesis, London Metropolitan University, UK

Siakas Kerstin V., Balstrup Bo (2000): A Field-study of Cultural Influences on Software Process Improvement in a Global Organisation, European Software Process Improvement Conference, EuroSPI '00, Copenhagen 7-9 November

Smith Peter B. (1996): National Cultures and the values of organisational employees: time for another look in *Managing across Cultures: Issues and Perspectives* Edited by Joynt P., Warner M., International Thomson Publishing

Sparrow P., Schuler R., Jackson S. (1994): Convergence or divergence: human resource practice and policies for competitive advantage worldwide, *International Journal of Behaviour*, Vol. 5, No 1, pp. 167-199

Thomsen, J., 21.4.2005: China Demands a Mobile Board of Directors, *Borsen Executive*, p. 7, cols 4-6

Tractinsky Noam, Järvenpää (1995): Information Systems Design Decision in a Global versus Domestic Context, *MIS Quarterly* / December, pp. 507-529

Walsham, G., (2001): *Making a World of Difference: It in a Global Context*, Wiley, Chichester

3 Author CVs

Kerstin V. Siakas

Kerstin Siakas is an Assistant Professor in the department of Informatics at the Alexandreio Technological Educational Institute of Thessaloniki, Greece. Her teaching includes Management Information Systems, Software Quality Management and Project Management. She is engaged in research in Information Systems Engineering, Multidisciplinary Approaches for Software Engineering, Knowledge Management and Software Quality Management. She has a particular interest in human and cultural approaches. She has an extensive industrial experience (since 1975) in software development on different levels from many European countries and mainly from multinational organisations. She has a number of academic and industrial project partners in many countries and she has published around 30 papers about her research.

Bo Balstrup

Bo Balstrup is Managing Director of the Center for Software Innovation in Sonderborg, Denmark. The centre provides research, technology transfer and matchmaking between industries and universities in the area of complex embedded mechatronic systems. He holds a B.Sc.EE from the Danish Technical University and an Executive MBA from Henley Management College, UK. He has formerly been Manager of Software Technology at Danfoss Drives, where he built the global software organisation. He has an extensive 26 years of industrial experience in software development and management from a number of Danish and multinational companies combined with experience in university-industry collaboration projects. He has implemented software quality systems in global organisations and headed an acquisition project as part of company globalisation. His research areas are Software Process and Methodology and Leadership of Virtual Organisations.

How European Software Industries can prepare for growth within the Global Marketplace - Northern Irish Strategies

Fergal Mc Caffery[1], Darja Šmite[2], F.G Wilkie[1] & D. McFall[1]

[1] Centre for Software Process Technologies, University of Ulster at Jordanstown,
Shore Road, Newtownabbey, Nothern Ireland BT 37 0QB
F.Mccaffery@ulster.ac.uk, fg.wilkie@ulster.ac.uk, d.mcfall@ulster.ac.uk

[2] Riga Information Technology Institute
Kuldigas iela 45b, Riga, LV-1083, Latvia
Darja.Smite@riti.lv

Abstract

Global market expansion forces the competition in the field of software development to grow extremely fast. Accordingly, countries that base their brand in the field of information technologies are looking forward to quality and productivity improvement along with price reduction. One of the major trends in the global market is outsourcing, which is considered to be an opportunity to achieve best knowledge, extra resources and assurance of quality development, speed time-to-market, and cut costs. The role of emerging nations in this context increases every day, pushing out the old players. This paper will therefore suggest how software development should change within the Northern Ireland (NI) software industry and adopt a global software development model in order to continue as a major player within the global software development industry.

Keywords

Northern Ireland software industry, global software development.

1 Introduction

Along with the new era of globalization, where the previous competition in the local market expands world wide overriding the borders, each country tries to find its own place under the sun. Globalization provides such new opportunities as achieving best knowledge and extra resources, speeding time-to-market by using "follow-the-sun" or 24 hour development, and cutting costs. In addition, global software development service providers often make significant investments in software process improvement (SPI) as this contributes to well implemented processes and the quality of the product. Naureen Khan in his evaluation of offshore IT outsourcing in India [1] reports that more than 200 Indian firms are quality accredited and from these around 36 firms have [1]CMM® level 5 and 19 firms have CMM® level 4. According to McKinsey Quarterly [2] the situation in terms of the global software development market considering labor costs, quality and productivity is as follows (see Figure 1).

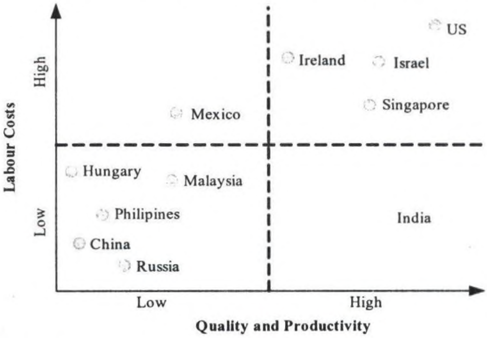


Figure 1. Outsourcing Market

In the era of globalization each country strives to popularize its brand. Being considered one of the major partners in software development outsourcing for a long while, the Republic of Ireland succeeded due to several success factors [3]:

- A strong software development and telecommunications infrastructure,
- A highly educated workforce,
- English is the first language,
- Legislative support,
- Quality,
- Government funding for research and development.

Unfortunately, global market is frequently changing. Therefore, the Irish (both north and south) software industry can no longer rely solely on the factors that originally contributed to the success of its software industry. Emerging software development nations now can compete with Ireland in terms of the above factors and for a lower labour cost. This paper will therefore suggest how Irish (focusing particularly on NI) software development should change and adopt a global software development model in order to continue as a major player within the global software development industry.

[1] ®CMMI/CMM is registered in the U.S. Patent and Trademark Office by Carnegie Mellon University

2 The Current Software Development Picture within NI

The NI software industry has approximately 110 software companies that develop software either as a prime business activity or as an important part of their business function. The Centre for Software Process Technologies (CSPT) performed a survey within the NI software industry in 2003 [4]. The survey was extensive in its scope encompassing more than half of all of NI's software development companies, which together employed 80% of NI's software engineering employees (the companies surveyed employ a total of 2428 software employees out of an entire industry with approximately 3000 software employees). For the purpose of this paper we determine software engineering employees as those directly concerned with lifecycle work.

It emerged that even though 71% of the software companies in NI are home grown, they only employ 40% of the entire 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.

Like the Republic of Ireland (ROI) and the rest of the United Kingdom, the NI software industry is exposed to any downturn in the global software industry, and global economy in general, and this has been witnessed in recent years in the lay off of personnel from our larger software companies [5]. From research into software growth generally (including in other regions [3,6,7,8,9,10]) it appears that common factors influencing the growth of a successful software industry include:-

- Financial support (either through government or venture capital) to establish and expand software companies, as well as to enable research and development;
- Prioritisation of software as an important sector with special tax incentives offered by the government;
- An excellent telecommunications infrastructure;
- Protection against software piracy.

NI provides world class financial support and has an excellent telecommunications infrastructure. Perhaps more interestingly and surprisingly, three other factors have also been identified as a concern including; limited influence from overseas, lack of experience of our entrepreneurs and a lack of encouragement for our entrepreneurs. 61% of founders of indigenous companies had no more than 10 years experience upon start-up and significantly 27% of founders of indigenous companies had no more than 5 years experience upon start-up. Many of these entrepreneurs are united on the view that culturally NI is adverse to entrepreneurial activity, many have expressed exasperation at the resistance to new ideas and the 'can't do' or 'you do' attitude which permeates the culture. The nature of NI's new World Wide Web order is a likely explanation for the youthful character of it's entrepreneurs but nonetheless for some companies at least, this may be a growth limiting factor. Many have identified a need for a history of hardened entrepreneurial veterans, who like in countries such as the U.S have been allowed to gain wisdom from a history of failure, failure being regarded as part of the learning process. In NI, failure is equated to limited ability.

Notwithstanding the positive influence from the presence of some multinational software organisations, many have suggested a limited influence from overseas and an inward looking culture. Even in the percentage of software engineering staff employed from overseas we can see issues. Home grown companies have on average only 2% of their staff from overseas. While apparently NI can satisfy it's workforce demands locally, evidence suggests that others around the world cannot [6,7]. Some have seen this as a sign that NI universities do a great job of satisfying the local demand for computer science skills, others have seen this as an indication of under performance in the NI software industry.

Many small companies engage in the development of relatively small-scale web-applications, often competing aggressively with each other for a share of a limited local market. Overseas markets have been targeted, particularly in the U.S, but the percentage of revenue generated in those markets is relatively small, often not exceeding 20% (this overseas percentage value also includes companies whose products are produced for an American parent [4]).

2.1 Current Software Development Practices within NI

Only 18% of NI software development organizations use a formal methodology such as SSADM[4]. Many software development companies instead adopt in-house methodologies that are far from formal and indeed many are no more than liberal embellishments to waterfall lifecycles. The surveyed companies typically perform full lifecycle activities that include the following stages which are performed either iteratively or sequentially depending upon the development methodology adopted by the company for a particular project:

- System/Information Engineering and Modelling;
- Software Requirements Analysis;
- Systems Analysis and Design;
- Code Generation;
- Testing;
- Delivery and Maintenance.

2.2 Areas of concern for the NI software industry

Figure 2, illustrates the main areas of concern within the NI software development industry. 65% of companies have experienced difficulty with managing risks and estimating task size. This is due in many instances to a lack of project management expertise and a lack of experience in developing products for a wide client base, which consequently means there is little opportunity to build experience in task size estimation. Small companies are immature in their handling of risks. This is a consequence of having insufficient historical project data which acts as a yardstick on which to assess risk, with many companies failing to implement any form of risk management strategy, content simply to gamble on outcomes rather than assess and mitigate potential problems. Smaller companies in particular, exhibit low adherence to standards and low tool usage which lessens the chance of maintaining a repository of historical project information, with consequences for maturity.

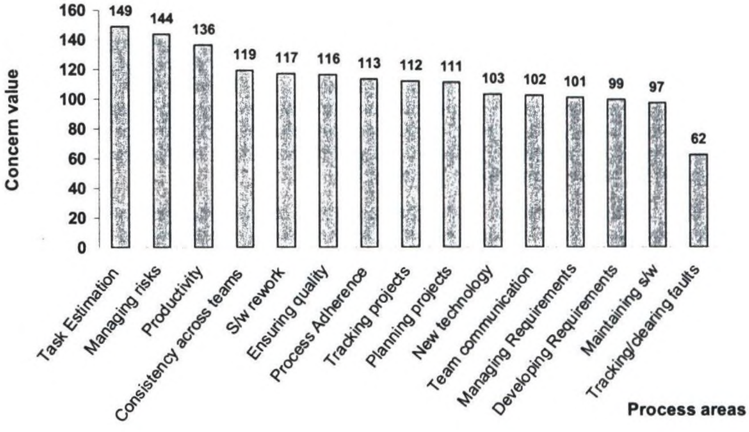


Figure 2. Areas of Concern for all organizations

For larger organizations, who demonstrate proficiency in task size estimation the issue is one of certainty. Increasingly, in our competitive market, companies are being asked to produce confidence measures, illustrating how confident the company is in it's estimations for future projects. There is a need here for expertise in more analytical methods of task size estimation, where confidence measurement is a key feature.

Productivity is closely related to task estimation and is a major issue for both small and large companies alike, with many companies unsure as to how much actual time is spent working on tasks. Keeping up-to-date with new technology is a real issue for smaller companies as their business need typically requires them to move with the latest technology in order to remain competitive, but they often suffer from a resource issue in that they cannot dedicate resources to such research, whereas this is less of an issue for larger companies, also larger traditional companies often use more stable technology.

As you would expect smaller companies are generally not concerned with either consistency across teams or team communication as typically there are few teams and the teams have on average three team members. Whereas with larger companies more teams exist and teams may also be larger and may even be distributed across different sites, therefore presenting consistency and communication issues.

After comparing the NI survey findings with reports from the emerging Eastern software development nations [6,7,8,9,10] it appears that currently NI is not well placed to attract overseas contracts as it cannot currently compete with emerging software development nations such as India, Turkey and China in terms of cost and quality certification.

Two types of software development contract exist with NI - home-based or overseas. To date most domestic contracts are performed nationally, with all stages of the software development lifecycle being performed within the UK and Ireland. However during the past year some of the larger British and Irish companies are beginning to outsource some of their software development work. Such companies have been attracted by the lower cost of software development within the emerging nations compared to domestic software development rates. There is a fear within the NI software industry that this may be a trend that will increase over a period of time particularly if the initial outsourcing projects are successful and the proposed benefits of improved quality (due to higher certification), quicker time to market, reduced cost, and the need for less full-time (more expensive) local software development staff required are realised. The fear is that if home-based companies are shipping their software development work overseas how can the local industry possibly compete in terms of attracting overseas contracts (particularly from the U.S). The answer perhaps is to concentrate on the more appealing aspects of the NI software development industry. For example, focus on the lifecycle areas that rely on good communication skills. As English is the first language of NI, this presents a natural advantage for western interaction (in particular to U.S interactions). Additionally, the time difference between NI and the western industry presents more of a window of opportunity in terms of overlapping work times. As NI tends to produce a higher percentage of entrepreneurs than the emerging Eastern nations, perhaps we are then more suited to working with the more initial stages of the software development lifecycle [4].

In light of these findings it seems that a possible solution to the problem of sustaining the NI software development industry is to both focus upon the strengths of this industry but to also discover a way to utilise the potential benefits of the emerging Eastern software development nations. This strategy may help to ensure not only the survival of the NI software development industry but perhaps even lead to long-term growth by changing the role that software development companies with NI fulfil within the global software development picture. What we propose is a way for the NI software industry to gain a market share within the global software industry, and we are aware that given the scale of the Eastern software industries that these industries will also perform separate full lifecycle development projects in addition to working with the NI software industry to fulfil western contracts.

3 Possible Growth Scenarios

The reality is that as Eastern software development nations go from strength to strength over the next few years in terms of their ability to attract outsourced contracts [10] this will greatly impact the scale of the software industry within NI, as domestic companies will have great difficulty competing in terms of CMMI® certification or cost. This means that a number of options exist for the NI software industry:

- Option1: continue to approach software development as they have for the past number of years;
- Option 2: try to improve the software industry by following a SPI path and try to compete with the emerging Eastern nations in terms of quality;
- Option 3: revise their role within the global software development picture.

3.1 Option #1 – Continue as Previously

In terms of planning to make the NI software industry successful, option 1 may prove to be an alternative that could lead to the phasing out of software development within NI.

3.2 Option #2 – Improve the Quality

Option 2, if selected would be of benefit to the NI software development industry as it would eventually enable NI software development companies to compete with the Eastern software development companies in terms of quality to attract contracts. However, the path to SPI is a long one and given the fact that most companies have previously not been engaged in SPICE (ISO/IEC:15504) [11] or CMMI® [12] certification programmes it will be a number of years before a significant percentage of NI software companies would be in a position to seriously compete with the Eastern software development companies in terms of quality. In relation to the current software development infrastructure in NI software, many medium to large software companies are using sound processes governing a strong methodological base, but often many smaller indigenous companies are driven by their entrepreneurial managers and directors who know the processes well and act as mentors to other staff members. In such companies methodology is sacrificed in favour of a set of loosely defined techniques, with training being event driven for the most part even though it is part of almost all strategic plans. Many companies use what they regard as computer aided software engineering (CASE) tools during software development, however few of these instances are examples of full CASE tools. This is a consequence of limited methodology adoption and indeed the small size of local organizations with limited funds for investment. 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 home grown 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. There is a reliance on individuals in a fire fighting environment, low awareness of standards and problems experienced both 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.

3.3 Option #3 – Act Globally

The adoption of option 3 is crucial to the long-term future of the software development industry within NI. Most software development projects in the NI software industry are produced by following a complete software development lifecycle such as the waterfall model for larger projects, and more agile iterative models for smaller, shorter lifecycle projects. However in order for the NI software industry to remain viable as a software development nation, local software development companies may need to change from focusing upon full-lifecycle development to focusing upon higher value software development areas. This means playing to the strengths of NI software industry and concentrating on the areas that involve critical interactions with western clients (in particular cultural aspects and time zones favour U.S customers interacting with Irish and British based companies as opposed to companies from the Eastern software development nations).

4 The Proposed Way of Developing Software within Northern Ireland

The preferred option to increase the chances of developing a successful software industry within NI is therefore to adopt both options 2 and 3 in parallel.

It is predicted that a quarter of all Europe's IT jobs will have moved to the Eastern software development nations by 2010 [13]. It is therefore important that NI software development companies start to focus on changing the current software infrastructure from focusing upon full lifecycle coverage to concentrating on the core areas of system engineering, and move towards outsourcing some of the processes to offshore service providers.

Considering the various options of distributing software development processes between the engaging party and offshore service providers, the first idea about splitting responsibilities is sending coding offshore. However, this scenario overestimates the value of coding against entire project expenses. The average industrial estimate of the software implementation process is around 20% of the whole project life cycle. Today's IT outsourcing frequently involves a much greater range and depth of services than in the past, with an increasing number of IT functions being transferred to IT service providers[14].

To illustrate possible context that could be outsourced, the authors offer three different options from the market choices and experience:

- Option # 1 - MachroTech's Global Software Delivery Model [15];
- Option # 2 - Skytechsolutions Global Sotware Delivery model [16];
- Option # 3 - Latvian experience in global software development improvement [17].

Table 1. Sourcing options

	On-site processes	**Joint processes**	**Offshore processes**
Option #1	• Analysis • High Level Design • Development Coordination • Deployment	• Integrated Testing • Client Acceptance	• Planning • Detail Architecture • Development • Testing
	On-site processes	**Joint processes**	**Offshore processes**
Option #2	• Business Requirement Planning • Conceptualization • High Level Design • Coordination • Implementation	• Architecture Design	• Detail Level Design • Construction • Documentation • Testing • External Testing • Maintenance
	On-site processes	**Joint processes**	**Offshore processes**
Option #3	• Testing • Implementation	• Planning and Coordination • Requirement Analysis	• Design • Coding • Unit Testing • Maintenance

Anand Ramakrishnan [18] reports that such areas as system design and architecture, research and development outsourcing provide higher returns (up to 60%). Besides, such evolution scenario as strategic partnership and moving from "body shopping" to wide-ranging services is seen as the future of outsourcing [14, 19]. Accordingly, there should be a deliberated distribution of software lifecycle processes between the outsourcing provider and the engaging party in order to make the multi-site development reasonable.

In addition, using the findings of the NI survey it is apparent that in order for NI companies to work effectively with the emerging nations as described above certain parts of their software development infrastructure will need to improve. To assist this SPI programmes should be given priority within the NI software development industry as it's software development companies will be responsible for managing the delivery of the overall project and will therefore require a higher level of software development maturity than was discovered in the findings from the NI software development survey. The CSPT is currently assisting the NI software industry with SPI programmes [20] and assessments based upon CMMI® [21]. Also a large number of companies within the Eastern software development nations (particularly in India) are CMMI® certified and will expect their business partners to work to a similar level. Additionally, western contracts may at some stage in the future only be awarded to companies that are CMMI® certified.

The software professionals required to work within this new proposed lifecycle will differ from the majority of existing software professionals and recent graduates that received a university education that devoted the majority of the curriculum to generating code. Software professionals will therefore require retraining to focus on business and management areas of software development, as well as consulting. Universities should also provide computer science postgraduate courses in SPI to help grow an improvement culture. To assist this development the University of Ulster (NI) plans to provide a postgraduate course in SPI in 2005/2006 (this course will be officially recognised by the Software Engineering Institute, Carnegie Mellon University). Additionally, universities should design new computer science undergraduate courses to focus upon the business and management disciplines associated with software engineering such as requirements management, project planning, managing outsourcing projects, configuration management and risk management, in addition to development disciplines such as requirements development, software analysis, design and testing.

5 Conclusions

If the NI software industry is to grow within the global software development industry it may have to change the way its companies perform software development. The authors would like to emphasize the following major changes necessary for the industrial growth:

- Explore the options of outsourcing some of the software development life cycle processes to an offshore service provider, by this using the opportunities offered by global market, including product cost reduction;
- Proven by the findings from the NI software industry survey, companies within the NI software industry should adopt SPI programmes to increase their software development maturity so that they would be able to work more effectively with both high maturity Western and Eastern software development companies;
- To accompany the industry in there aspirations, the universities should also refocus the education providing computer science postgraduate with knowledge about business and management areas of software development, consulting, and SPI to help grow an improvement culture.

This paper focused upon how software development practices should change within the NI software industry in order to enable growth within the global software industry; however the above conclusions may also be applied to other Western European software industries wishing to grow within the global marketplace.

6 Acknowledgement

The Centre for Software Process Technologies is supported by the EU Programme for Peace and Reconciliation in NI and the Border Regions of Ireland (Peace II).

7 Literature

[1]. N. Khan, W.L. Currie, V. Weerakkody, B. Desai, "Evaluating Offshore IT Outsourcing in India: Supplier and Customer Scenarious", Proceedings of the 36th Hawaii International Conference on System Sciences, IEEE, 2002

[2]. McKinsey Quarterly, "Programmers Abroad" 2001 No.2

[3]. Cochrane, R., IEEE Software, March/April 2001, pp 87-89.

[4]. F. Mc Caffery, F.G. Wilkie, D. McFall & N. Lester. (Apr 2004) "NI Software Industry Survey", Proceedings of Fourth International SPICE Conference on Process Assessment and Improvement, Lisbon, Portugal, 28-29 April 2004, SPICE User Group (Lisbon, Portugal), ISBN 972-9071-73-X, Pages 159-161

[5]. Momentum Survey 2004. Advancing NI through I.C.T.

[6]. Moitra, Deependra, India's Software Industry, IEEE Software January/February 2001. pp 77-80

[7]. Ju Dehua., China's Budding Software Industry, IEEE Software May/June 2001. pp 92-95

[8]. Matsubara, Tomoo, "Japan: A huge IT Consumption Market", IEEE Software, September/October 2001, pp 77-80.

[9]. Aytac Turgay., Ikiz Seckan., Aykol Meric., A SPICE-Oriented, SWEBOK-based Software Process Assessment on a National Scale: Turkish Software Sector Survey – 2001, Proceedings of the Joint ESA-3rd International SPICE Conference on Process Assessment and Improvement, 2003, pp 135.

[10]. EIU ViewsWire, "China industry: Software exports to expand rapidly over next several years", Oct 16, 2003

[11]. ISO/IEC TR 15504:1998(E), Information Technology – SPICE, Software Process Assessment, Parts 1-9, Type 2 Technical report.

[12]. Capability Maturity Model® Integration (CMMISM) for Software Engineering (CMMI-SW, V1.1, Version 1.1, August 2002)

[13]. "One IT job in four to go abroad" BBC News, http://news.bbc.co.uk/2/hi/business/3516742.stm

[14]. J.N. Lee, M.Q. Huynh, R.C.W. Kwok, S.M. Pi, "IT Outsourcing Evolution – Past, Present, and Future", Communications of the ACM, May 2003/Vol. 46, No.5, pp.84-89

[15]. http://www.machrotech.com/company/offshore-outsourcing-model.asp

[16]. http://www.skytechsolutions.com/corporate/global_delivery.htm

[17]. D.Smite, "A Case Study – Coordination Practices in Global Software Development" accepted for publishing in the Proceedings of the Profes conference, Oulu, Finland, 2005

[18]. A. Ramakrishnan "Offshore Insight: Maximizing the Global Delivery Model", Kanbay Incorporated, 2004, http://www.kanbay.com/executiveconnections/artifacts/ec5_print.pdf

[19]. "Global Software Teams Take Flights", Outsourcing Journal, March 1999, http://www.outsourcing-journal.com/mar1999-academic.html

[20]. F.G. Wilkie, D. McFall & F. Mc Caffery, (Mar 2004) "The Centre for Software Process Technologies (CSPT): A Model for Process Improvement in Geographical Regions with Small Software Industries", Proceedings of Software Engineering Process Group Conference, Orlando, Florida, USA, 8-11 March 2004, The Software Engineering Institute (Pittsburgh, USA), Pages 5-(pages on CD-ROM)

[21]. Appraisal Requirements for CMMI, Version 1.1 (ARC,V1.1)

8 Author CVs

Dr Fergal Mc Caffery

Fergal is a full-time researcher in the CSPT. His research is focused in the areas of Requirements Management, Risk Management, Software Process Improvement within small to medium sized companies, the development of a software process improvement framework for the medical device domain and automating software process assessments. He has 6 years experience as a project manager/team leader/senior software engineer for Nortel Networks and 1 year as a software engineer for Aldiscon Ltd. This period in industry was preceeded by 2 years as an academic member of staff at the University of Ulster. Prior to this Fergal completed a D.Phil within the area of intelligent adaptive multimedia systems.

Darja Šmite

Darja is an IT consultant and auditor at the Riga Information Technology Institute, and an academic member of staff at the University of Latvia. She is also currently working towards her Ph.D. degree at the University of Latvia. Her major research interests are related to software development and software process improvement in distributed environment. Darja has 5 years of industrial experience as a Software Engineer, Systems Analyst and Project Manager. Therefore, her research has been taken in close relationship with practice.

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. ergal 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 preceeded 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 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.

FROM COMPLIANCE TO BUSINESS SUCCESS: Improving outsourcing service controls by adopting external regulatory requirements

Miklós Biró[1], Gáborné Deák[2], János Ivanyos[2], Richard Messnarz[3]

[1]CORVINUS University, Veres Pálné u. 36. H-1053 Budapest, Hungary
[2]MEMOLUX, Thököly ut 137, H-1146 Budapest, Hungary
[3]ISCN, Florence House, 1 Florence Villas, Bray Co. Wicklow, Ireland

Abstract

New generation of general models referring either IT or Internal Control – like COBIT or COSO - are extended with executive management perspective. The practice shows, this opening solely does not enough to reach a breakthrough, because models became more complicated than it could be applied without some difficulties. The best catalysts of improvement programs are the more and more mandatory rules being issued, mainly from financial reporting area. The Sarbanes-Oxley Act for US SEC registrants and its affiliates and 8th Directive on company Law in the EU require strict internal control of reporting processes. In this paper we concentrate on the application of these rules successfully in such a relation where IT enabled services has major effect on the compliance of the user organisation. We investigate the effects of high maturity level on compliance for both the service and the user organisations. The paper refers to the applicability of the well-known capability models like CMM, eSCM and some other sources like COSO, BSC and SAS 70. For presenting implementation practices of the general risk based control model via key control processes, effectiveness measurement and innovative technologies, the knowledge management platform resulted by former Software Process Improvement experiences were used.

Keywords

Process Improvement, Business Objectives, Control Frameworks, Independent Audit, IT-enabled service, Outsourcing, Risk analysis, Knowledge Management

1 Introduction

Software process improvement models and practices have become more and more accepted and incorporated by not only the international standards like ISO 15504[1] and ISO 9001 series[2], but even by the more business management-oriented control frameworks, such as COBIT[3], the open standard of Control Objectives for Information and related Technology. COBIT *Management Guidelines* provides tools to help IT managers improve IT performance and link IT objectives to business objectives, which consist of Maturity Models, Critical Success Factors (CSFs), Key Goal Indicators (KGIs) and Key Performance Indicators (KPIs). This concept delivers a significantly improved framework responding to management's need for control and measurability of IT by providing management with tools to assess and measure their organisation's IT environment against the 34 IT processes COBIT identifies. In addition, to help focus on performance management, the principles of the Balanced Business Scorecard[4] were used.

New generation of general models referring to either IT or Internal Control – like COBIT or COSO[5] - are extended with business perspective willing to gain top management's ear. But the practice shows, this opening solely does not enough to reach a breakthrough, because models became more complicated than it could be applied without some difficulties. Very frequently exposed that the best catalyst of improvement programs are the more and more mandatory rules coming into force, nowadays mainly from financial reporting area. Sarbanes-Oxley Act for US SEC registrants and its affiliates and 8th Directive on company Law in the EU require strict internal control and effectiveness conclusion performed by the executive management.

Compliance and maturity issues have come into the view of the management as the huge cost of compliance readiness projects calls the attention of the sustainability and the added business value of such efforts.

In this paper the authors summarise their experiments at a Hungarian SME, which has run both IT-enabled outsourcing services and software and business process improvement projects for more than 15 years.

2 Experiences with SPI at 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) and professional organizations (IIA, EOQ). The payroll and accounting service lines are represented among the biggest ones in the Hungarian market provided by an independent Hungarian SME with about 150 clients.

Memolux has run software process improvement practice for more than 15 years. The ICT department achieved maturity level 3 of Bootstrap methodology[6] and ISO 9001 certification by 1998 and has been successfully participating in EU research projects. Memolux was the prime user and contractor of the PASS project, which was the first Central and Eastern European ESSI Process Improvement Experiment (FP4 PIE) project directly supported by the European Commission[7]. Memolux was a co-developer of the Media-Information sans Frontiéres (NQA based teamwork) system and was the technical coordinator of the Media-ISF Best Practice (FP5 IST Take-up) project[8].

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 adapt 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.

3 Lessons learnt from using Capability Models

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 [9], as well as contributed to the global understanding of business motivations for software process improvement [10], [11]. 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 [12], [13].

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 [14]. In another paper of "Stages of Software Process Improvement Based on 10 Year Case Studies" [15], the authors described that the above global processes drove Hungarian companies to orient their further process improvement initiatives towards their business needs. Memolux, whose core business is payroll and accounting service provision, introduced the published eSourcing Capability Maturity Model for IT-enabled Service Providers (e^{scm}) [16].

There were four main lessons learnt[18] 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 in professional outsourcing services, especially in managing control activities supporting the internal and external compliance requirements.

4 External regulatory requirements and customizing internal control models for compliance

External legal regulations regarding control of business processes such as SOX and the new 8th Directive in public financial reporting draws the attention towards knowledge management support and technologies of internal controls. The related external assurance requirements in the US, EU and its Member States are summarized by the discussion paper[17] issued in March 2005 by the Fédération des Experts Comptables Européens (FEE) - the representative body of the European accounting profession.

The US Sarbanes-Oxley Act of 2002 provides for new corporate governance rules, regulations and standards for specified public companies including SEC registrants. The US Securities and Exchange Commission (SEC) has mandated the use of a recognized internal control framework. The SEC in its final rules regarding the Sarbanes-Oxley Act made specific reference to the recommendations of the Committee of the Sponsoring Organizations of the Treadway Commission (COSO)[5].

By the definition of the COSO framework, internal control is a process, effected by an entity's board of directors, management, and other personnel, designed to provide reasonable assurance regarding the achievement of objectives. Memolux via adapting the models and experiments of software process improvement has developed applicable knowledge management platform for supporting the design, implementation and measurement of internal controls. For the customizing process we used the public resources of the IIA[18] and ISACA[19] websites.

IT Governance Institute recently published the paper of "IT Control Objectives for Sarbanes-Oxley - The importance of IT in the design, implementation and sustainability of internal control over disclosure and financial reporting"[20] in order to give advices how the compliance and sustainability requirements can be achieved through implementing maturity practices defined by the COBIT framework.

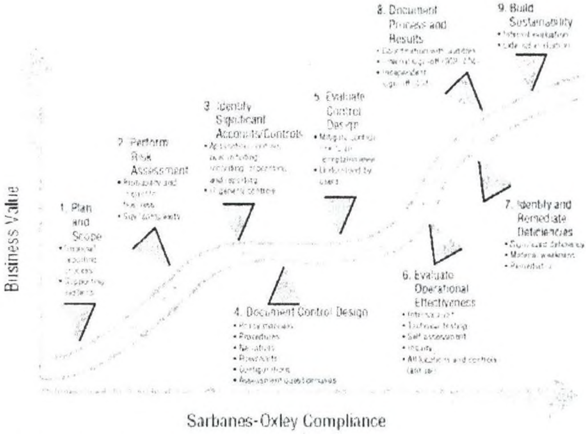

1. Plan and Scope
2. Perform Risk Assessment
3. Identify Significant Accounts/Controls
• IT general controls
4. Document Control Design
• Procedures
• Narratives
• Flowcharts
• Configurations
• Assessment questionnaires
5. Evaluate Control Design
• Understood by users
6. Evaluate Operational Effectiveness
• Self assessment
7. Identify and Remediate Deficiencies
• Material weakness
8. Document Process and Results
9. Build Sustainability
Business Value
Sarbanes-Oxley Compliance

Figure 1: Compliance Road Map

This compliance road map, illustrated in **figure 1**, provides direction for IT professionals on meeting the challenges of the Sarbanes-Oxley Act. Compliance is not a stand-alone process, as it must be integrated within the overall business-led compliance process. However, even these should be based on business (financial reporting) requirements, signed off by the business, and not left to the IT provider. This is especially true when IT is outsourced. For IT application controls, the business, not IT, should define the control requirements, especially for financial systems that are often complex in nature from a business process perspective.

When an organization uses external service organizations to perform outsourced services, these services are still part of the organization's overall operations and responsibility and need to be considered in the overall internal control program. Organizations should review the activities of the service organization in arriving at a conclusion on the reliability of its internal control. Documentation of service organization control activities will be required for the attestation activities of the independent auditor, so an assessment is required of the service organization to determine the sufficiency and appropriateness of evidence supporting these controls. Traditionally, audit opinions commonly known as SAS 70 reports[21] have been performed for service organizations.

The IT Governance Institute provides maturity profiles for internal control design and effectiveness model presented in **figure 2** demonstrating the stages of control reliability that may exist within organizations.

For the purposes of establishing internal control, some organizations may be willing to accept controls that fall somewhere short of stage 3. However, given the Sarbanes-Oxley Act's requirements for independent attestation of controls by external audit, controls will more than likely require the attributes and characteristics of stage 3 or higher for key control activities.

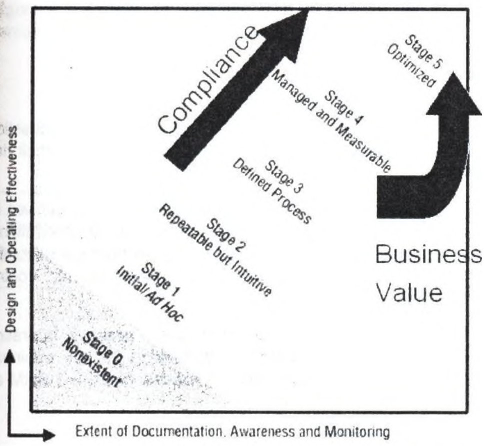


Figure 2: Internal Control Design and Effectiveness Model

This model doesn't indicate that the whole organization and the key process areas should be at the high maturity levels to achieve successful business results. This model refers only to the return of investment regarding the compliance readiness efforts of the specific external regulation.

From the viewpoints of sustainability and added business value the following stage 5 level practices implicate that the internal resources are used effectively and efficiently:

- An enterprise wide control and risk management program exists such that controls and procedures are well documented and continuously re-evaluated to reflect major process or organizational changes.
- A self-assessment process is used to evaluate the design and effectiveness of controls.
- Technology is leveraged to its fullest extent to document processes, control objectives and activities, identify gaps, and evaluate the effectiveness of controls.

By this conclusion the following parts of the paper concentrate on risk management, effectiveness measurement and related technology issues.

5 Implementing Enterprise Risk Management by using Key Control processes

The Committee of Sponsoring Organizations of the Treadway Commission (COSO) issued *Enterprise Risk Management — Integrated Framework*[22], as well as a detailed practical application guide in 2004.

Designed to offer organizations a commonly accepted model for evaluating risk management efforts, the framework expands on internal control concepts by providing a more robust focus based on the broader subject of enterprise risk management (ERM). Detailing the essential components of an effective ERM process, the framework provides guidance to help organizations build effective programs for identifying, measuring, prioritising, and responding to risk.

Embedded within an organization's strategies and objectives, ERM's value is maximized when a balance is reached between growth, returns, risks, uncertainties, and opportunities. How much risk the entity is prepared to accept is inherent in ERM's capabilities, which encompass the following key components:

- Aligning risk appetite and strategy.
- Enhancing risk response decisions.

- Reducing operational surprises and losses.
- Identifying and managing multiple and cross-enterprise risks.
- Seizing opportunities.
- Improving deployment of capital.

In addition, the new framework presents a standard definition of risk and ERM and provides direction to enhance risk management, including criteria for companies to use in determining whether their risk management is effective, and if not, what is needed.

Considering activities at all levels of the organization, the ERM framework views entity objectives at the entity, division, business-unit, and subsidiary levels, in four key categories: strategic, operations, reporting, and compliance. At the same time, the framework focuses on eight interrelated components: internal environment, objective setting, event identification, risk assessment, risk response, control activities, information and communication, and monitoring.

Key Controls[23] are those significant controls within our business processes, which if operating correctly will both ensure and give assurance that the organization is achieving its key business objectives. We use Key Controls concept to simplify the implementation process of COSO ERM as presented in **figure 3**:

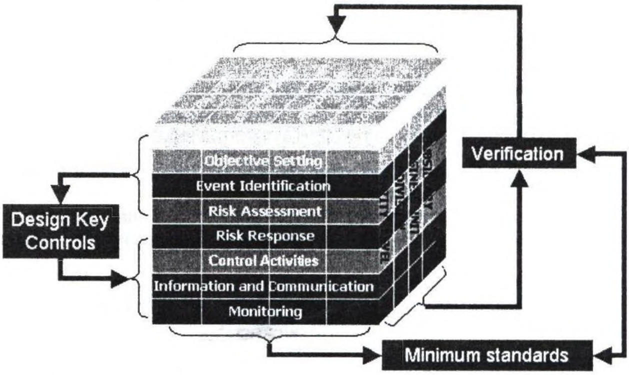


Figure 3: Enterprise Risk Management and Key Control Process

Minimum standards of control should ensure that the key control is completed in a manner that achieves our control objectives in a complete, timely, and accurate manner. Therefore, minimum standards of control are actually related to control objectives that the key control is attempting to achieve. In fact, when properly formulated, minimum standards are derived based on the control failure risks that the key control is attempting to prevent.

The implementation of Key Control process consists of the following steps:

- Customize the generic (e.g. financial) control objectives to be specific to the organization type, size, etc.
- Review and document "performance" reporting processes
- Search for missing controls
- Test identified controls
- Developing "minimum standards of control"
- Reviewing key controls and control exceptions
- Verification against the established minimum standards of control

Figure 4 presents an example Key Control process of Financial Reporting in an EU-funded multi-partner (co-sourcing) project experienced by Memolux:

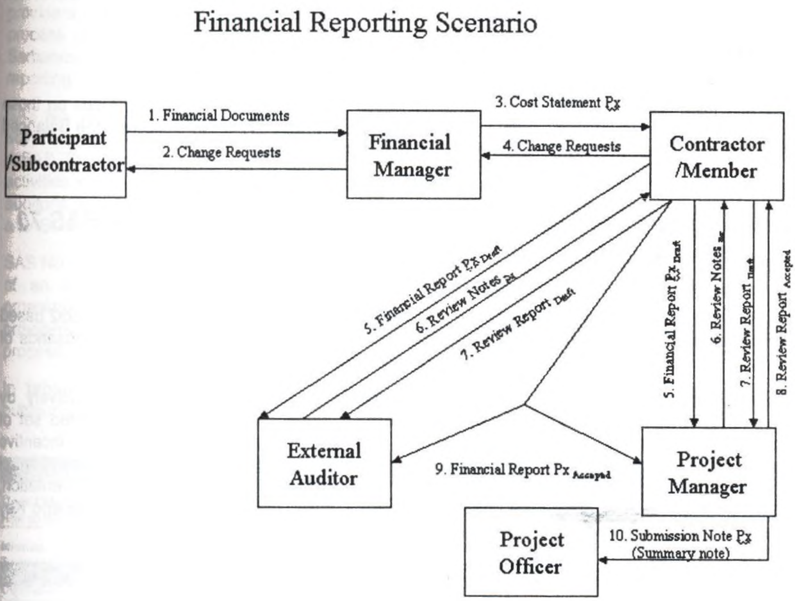


Figure 4: Key Control Process of Financial Reporting in EU-funded co-sourcing project[24]

Customised Control Objectives for Financial Reporting:

1. Authorization – Financial reports are appropriately authorized by project partner/financial manager
2. Accuracy – Reported costs are actual, economic and necessary for the implementation of the project
3. Valuation – Reported costs are determined in accordance with the usual accounting principles of the contractor
4. Completeness – All requested (periodic/final) reporting forms and evidences provided
5. Classification – Cost model instructions for direct and indirect costs are appropriately used
6. Existence – Proper justification of the resources deployed by each contractor, linking them to activities implemented
7. Timeliness/cutoff – All reported costs incurred during the duration of the project/period
8. Safeguard assets – All the original documentation likely to be examined by the auditors are availableSegregation of duties – External audit certificate provided

Example of minimum standards of control for Financial Reporting:

1. Financial reports are submitted by all the project partners in time (min. 2 weeks scheduled before official deadlines)
2. Actual template, formats, calculations are used (e.g. provided by teamwork tool)
3. Consistence with the actual Periodic/Final Activity Report and with the reported deviations from project plan ensured
4. Access to original documentation provided
5. Submission of reports, change requests, external reviews and approval by the Project Manager are documented (e.g. via teamwork submission tool)

This Key Control process example has relevancy for practices of many types of virtual organizations and co-sourcing cooperation/service controls as well. Integrated audit principles regarding co-sourcing models has the following considerations:

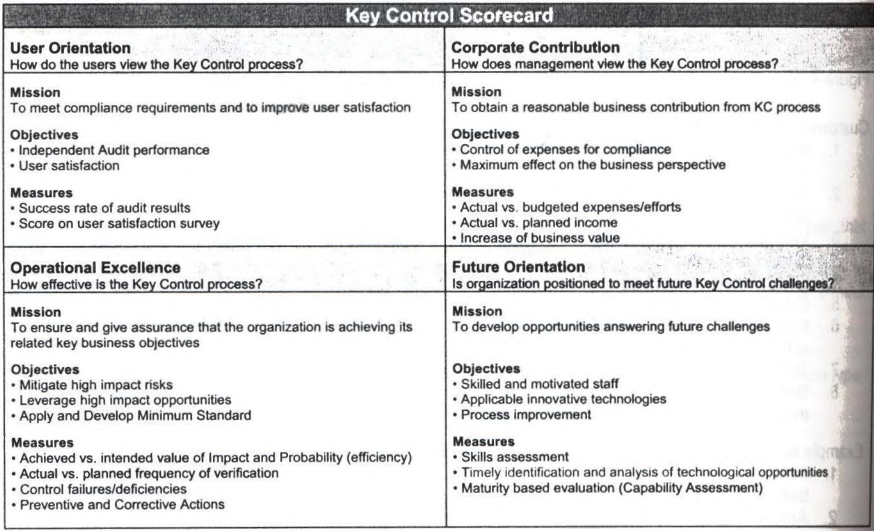
- Understand the business risks/opportunities over which the controls are to be assessed and tested (i.e., a common scope and objectives built around identified business risks).
- Identify the key controls, including both their manual and automated elements.
- Document, assess, and test *only* those key controls.

The conclusion from risk and opportunity management is that Key Controls have to focus on those events, where the related business objectives are really measurable. Therefore we use the Balanced Scorecard[4] model for presenting possible effectiveness measurement.

6 Measuring control effectiveness by using Key Control & SAS 70 Balanced Scorecards

The balanced scorecard concept was created by Robert S. Kaplan and David P. Norton in 1992 based on the simple premise that "measurement motivates." Today, it has been utilised by thousands of corporations, organisations and government agencies worldwide.

The balanced scorecard allows organisations to implement strategy rapidly and effectively by integrating measurement with the management system. It allows you to assess a detailed set of objectives and activities on an ongoing basis, as well as to measure links between incentive compensation and individual performance. An organization should build its specific Key Control scorecard based on the four elements of the framework: Financial/Corporate Orientation, User/Customer Orientation, Operational Excellence, and Future/Growth Orientation. A generic Key Control Scorecard applied by Memolux is shown in **table 1**:



Key Control Scorecard	
User Orientation How do the users view the Key Control process?	**Corporate Contribution** How does management view the Key Control process?
Mission To meet compliance requirements and to improve user satisfaction **Objectives** • Independent Audit performance • User satisfaction **Measures** • Success rate of audit results • Score on user satisfaction survey	**Mission** To obtain a reasonable business contribution from KC process **Objectives** • Control of expenses for compliance • Maximum effect on the business perspective **Measures** • Actual vs. budgeted expenses/efforts • Actual vs. planned income • Increase of business value
Operational Excellence How effective is the Key Control process?	**Future Orientation** Is organization positioned to meet future Key Control challenges?
Mission To ensure and give assurance that the organization is achieving its related key business objectives **Objectives** • Mitigate high impact risks • Leverage high impact opportunities • Apply and Develop Minimum Standard **Measures** • Achieved vs. intended value of Impact and Probability (efficiency) • Actual vs. planned frequency of verification • Control failures/deficiencies • Preventive and Corrective Actions	**Mission** To develop opportunities answering future challenges **Objectives** • Skilled and motivated staff • Applicable innovative technologies • Process improvement **Measures** • Skills assessment • Timely identification and analysis of technological opportunities • Maturity based evaluation (Capability Assessment)

Table 1: Key Control Scorecard

By focusing on measuring internal control effectiveness of co-sourcing partnership, we investigated the applicability of SAS 70 audit procedures as Key Control processes. Statement on Auditing Standards (SAS) No. 70[21] is an internationally recognized auditing standard developed by the

American Institute of Certified Public Accountants (AICPA). A SAS 70 audit or service auditor's examination is widely recognized, because it represents that a service organization has been through an in-depth audit of their control activities, which generally include controls over information technology and related processes. In today's global economy, service organizations or service providers must demonstrate that they have adequate controls and safeguards when they host or process data belonging to their customers. In addition, the requirements of Section 404 of the Sarbanes-Oxley Act of 2002 make SAS 70 audit reports even more important to the process of reporting on effective internal controls at service organizations.

SAS No. 70 is the authoritative guidance that allows service organizations to disclose their control activities and processes to their customers and their customers' auditors in a uniform reporting format. A SAS 70 examination signifies that a service organization has had its control objectives and control activities examined by an independent accounting and auditing firm. A formal report including the auditor's opinion ("Service Auditor's Report") is issued to the service organization at the conclusion of a SAS 70 examination.

SAS No. 70 is generally applicable when an auditor ("user auditor") is auditing the financial statements of an entity ("user organization") that obtains services from another organization ("service organization"). Service organizations that provide such services could be application service providers, bank trust departments, claims processing centers, Internet data centers, or other data processing service bureaus.

In **table 2** we provide a sample SAS 70 audit balanced scorecard applicable for IT-enabled outsourcing service organizations.

SAS 70 Scorecard	
User Orientation How do the users view the audit process?	**Corporate Contribution** How does management view the audit process?
Mission To meet audit requirements of users and to improve user satisfaction **Objectives** • Audit performance • User satisfaction **Measures** • Acceptance rate of SAS 70 audit results • Score on user satisfaction survey	**Mission** To obtain a reasonable business contribution from audit process **Objectives** • Control of expenses for audit • Maximum effect on the business perspective **Measures** • Added value for the user organizations • Positive effect on the service management • Actual vs. budgeted expenses
Operational Excellence How effective is the audit process?	**Future Orientation** Is organization positioned to meet future audit challenges?
Mission Effective audit process **Objectives** • Improvement of audit process • Efficient account audit • Efficient audit result presentation • Efficient management of audit findings **Measures** • Audit maturity level • Number of successful account audits • Rate of accepted audit reports • Number of failures to manage nonconformities in time	**Mission** Develop opportunities to answer future challenges **Objectives** • SAS 70 audit training and education of service personnel and user contact persons • Monitoring audit requirement revisions • Internal and external benchmarking research **Measures** • Educational budget as percentage of total audit budget • Percentage of service staff and user contact persons involved in training and education activities • Percentage of budget spent on audit requirement revision monitoring • Number of successful renewal projects initiated by research team

Table 2: SAS 70 Scorecard

The provision of an SAS 70 audit report doesn't mean a Key Control for the user organization, as it can be implemented in the Service Level Management or Third party Management processes. However it can reduce the total audit costs and provide higher confidence towards the service organization.

From the service organization viewpoint, the SAS 70 audit, such as any other type of quality certification process can be handled as a Key Control process having relevancies to measure the achievement of specific business objectives.

7 Applicable SPI technologies supporting control evaluation

In January 2005, The IIA Research Foundation published a survey of "Sarbanes-Oxley Section 404 Work - Looking at the Benefits"[25]. The survey identifies control improvements that have taken place as a direct result of SOX evaluations and the lessons learned that could improve the efficiency and effectiveness of control evaluations in the future.

Regarding the enhanced documentation and control evidence, there are two components of improved documentation that were mentioned by the survey respondents:

- Documentation of the processes, workflow, and controls, and
- Documentation of the evidence that the controls are working.

Improving the documentation of controls and processes is not surprising because it has been mandated by regulation and auditing standards. In completing the readiness effort, organizations have better captured not only the process flow and associated controls, but also updated the associated policies, procedures, handbooks, job descriptions, and other pertinent documents.

Respondents believed that the development of adequate documentation would pay future dividends in areas such as training new employees, enabling backfill and succession planning for key positions, and identifying process improvement opportunities. Many respondents mentioned that the improved documentation is an important control from a global control perspective.

A major finding is that there was little documentation or evidence that existing controls were working. For example, how would an organization determine that there was a proper review of an exception report, or a proper reconciliation, if there was no documentation that the review of the reconciliation was performed? Respondents noted the improvement in documenting the evidence of supervisory reviews and approvals, management committee actions and decisions, and the investigation and resolution of un-reconciled or outstanding items. The need to properly and clearly develop evidence of the operation of each key control has become a more common practice.

These issues referred by the survey are strongly connected to the compliance and maturity conclusions presented in **figure 2**. The Key Control Scorecard Future Orientation measures shown in **table 1** reflect to these experiments. Hereby we identify the following applicable technologies resulted by former software process improvement developments also practiced by Memolux:

The **NQA TEAMWORK**[26] Environment - A combination of methodology, technology and social skills to run project administration, quality management, internal control processes as teamwork over the Internet.

NQA is highly configurable and adaptable. Companies can define their own project administration structure and by configuration of scripts the user interface is adapted. Companies can use their (so far developed) documentation by inserting their documentation guidelines in a template pool. Teamwork is highly emphasized by the underlying methodology (role based work flow models) and the assignment of team members to roles and the structuring of workflows are done by the quality administrator through a menu system. NQA is pre-configured for scenarios supporting international guidelines and standards.

The **Process Assessment Portal**. Within a process improvement context, process assessment provides the means of characterizing the current practice within an organizational unit in terms of the capability of the selected processes. Analysis of the results in the light of the organization's business needs identifies strengths, weakness and risks inherent in the processes. This, in turn, leads to the ability to determine whether the processes are effective in achieving their goals, and to identify significant causes of poor quality, or overruns in time or cost. These provide the drivers for prioritizing improvements to processes.

Process capability determination is concerned with analysing the proposed capability of selected processes against a target process capability profile in order to identify the risks involved in undertaking a project using the selected processes The proposed capability may be based on the results of relevant previous process assessments, or may be based on an assessment carried out for the purpose of establishing the proposed capability.

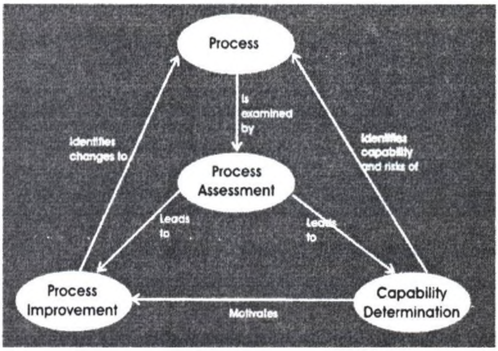


Figure 5: Process Assessment

ISO/IEC 15504-2[27] defines a reference model of processes and process capability that forms the basis for any model to be used for the purposes of process assessment. The reference model comprises a two-dimensional approach to the evaluation of process capability - one dimension defines the processes to be assessed, the other describes the scale for measurement of capability. Any model(s) compatible with the reference model may be used for assessment, and the results of any conformant assessments will be able to be translated into a common base.

Each process in the reference model is described by a statement of the purpose of the process, which includes an outline of the intended outcomes of process implementation

The **Skills Assessment Portal**[28] - A skills portal configured with the skills card and supports the steps of browsing required skills, self-assessment, formal assessment, evidence collection, generation of skills profiles, and learning recommendations.

By integrating the Capability Adviser Process Solutions and Capability Adviser Skill Card Solutions, with the web based training portal Moodle and with the NQA Teamwork Portal System, the result will be a system available for assessment, learning, and joint development of knowledge and training.

8 Conclusion

Outsourcing and co-sourcing partnerships have major effects on implementation of Enterprise Risk Management (ERM) based control systems. By using the maturity practices of COBIT Management Guidelines we presented the Key Control concept as a sustainable interpretation of risk and opportunity management in relation with measuring effectiveness of the related business objectives. The innovation of achieving software process improvement skills and experiments performed by Memolux from the beginning of the 1990's resulted appropriate capability to provide knowledge-based skills and technology, supporting the design, implementation and measurement of internal controls for outsourcing and co-sourcing projects. As having been member both of the software process improvement and business process outsourcing communities for many years, Memolux develops new business service applicable for internal control of outsourcing and co-sourcing activities as well.

This new service will be set-up by implementing the same knowledge management platform already used for software quality processes. The uniqueness of the proposed service is that it comparatively supports adequate internal control processes of both the service providing organisation and the user sides, being assessable by independent auditors.

For further extension of the experiments presented in this paper, authors are considering to set up ISO/IEC 15504 conform process reference model and measurement framework applicable for the internal audit community assessing control effectiveness of outsourcing and co-sourcing business processes.

9 References

1. ISO/IEC 15504 Information technology -- Process assessment

2. ISO 9001:1994 Quality Systems. Model for quaility assurance in design, development, production, installation and servicing

   ISO 9001:2000 Quality management systems. Requirements.

3. Information Systems Audit and Control Foundation, IT Governance Institute: COBIT - Control Objectives for Information and related Technology

4. Kaplan,R.S:; Norton,D.P.: The Balanced Scorecard: Measures That Drive Performance,1992

5. The Committee of Sponsoring Organizations of the Treadway Commission (COSO): Internal Control — Integrated Framework

6. The Bootstrap Assessment scheme, Bootstrap methodology www.bootstrap-institute.com

7. 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)

8. Baksa,C.; Ivanyos,J.; Ziaja,Z.;Messnarz,R.: Measuring e-business effectiveness: Adaption of process driven approach in web-publishing. EuroSPI 2002, 18th-20th September 2002 in Nurenberg

9. 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.

10. 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).

11. 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/biro.html)

12. 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)

13. 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)

14. 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/biro.html)

15. Balla,K.; Biró,M.; Ivanyos,J.; Messnarz,R.: Stages of Software Process Improvement Based on 10 Year Case Studies EuroSPI 2004, Trondheim

16. 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.

17. Fédération des Experts Comptables Européens (FEE): Risk Management and Internal Control in the EU, March 2005

18. The Institute of Internal Auditors (The IIA): www.theiia.org

19. Information Systems Audit and Control Association (ISACA): www.isaca.org

20. Information Systems Audit and Control Foundation, IT Governance Institute: IT Control Objectives for Sarbanes-Oxley - The importance of IT in the design, implementation and sustainability of internal control over disclosure and financial reporting, 2004

21. American Institute of Certified Public Accountants (AICPA): Statement on Auditing Standards (SAS) No. 70, Service Organizations

22. Committee of Sponsoring Organizations of the Treadway Commission (COSO): Enterprise Risk Management — Integrated Framework, 2004

23. Vorhies,J.B.: Key Controls: The Solution for Sarbanes-Oxley Internal Control Compliance, 2004
24. Certified European Project Manager HU/B/03/F/PP-170028 project, carried out with the financial support of the Commission of the European Communities under the Leonardo da Vinci Programme
25. The IIA Research Foundation: Sarbanes-Oxley Section 404 Work - Looking at the Benefits, January 2005
26. Messnarz,R.; Nadasi,G.; O'Leary,E.; Foley,B.: Experience with Teamwork in Distributed Work Environments, EuroSPI 2001, Limerick
27. ISO/IEC 15504-2:2003 Information technology -- Process assessment -- Part 2: Performing an assessment
28. Biró,M.; Messnarz,R.: ManagEUr – Success Criteria for EU Project management, EuroSPI 2004, Trondheim

Topic: Experience Report ISO/IEC 15504 and ISO/IEC 61508 Fears about models and standards in an SME overcome

Vera Grimm
IAE GmbH, Gesellschaft für angewandte Elektronik,
55252 Mainz-Kastel, Anna-Birle-Str. 1a, Germany
(mailto:Vera.grimm@iae-gmbh.de)

Introduction:

IAE Gesellschaft für angewandte Elektronik GmbH is a typical SME, which delivers requirement specifications, software modules, hardware prototypes and services – related to project management and quality management – within the automotive and avionic domain. Because of the required software quality and because of many safety critical software functions IAE develops, the introduction of ISO/IEC 15504 (SPICE) and ISO/IEC 61508 (SIL) has become a necessity. Most of the developers and the management feared the introduction of the standards. They expected a lot of arbitrary additional work and an overload of needless established processes; imposing a lot of paper work to programmers and service providers.

The contrary was the case at IAE. There is now a good balance between adherence to standards, high quality in project work and documented working procedures. Standards and methods are widely accepted by the staff members and by executive staff and managers.

This experience report shows that deeply rooted objections turned into valuable help, given by the standards ISO/IEC 15504 and ISO/IEC 61508.

KEY WORDS: Assessment, Safety, ISO 61508 (SIL), ISO/IEC TR 15504 (SPICE), Process Improvement, Review, Inspection

Annotation:
The listed examples are derived from real practice and can be proved. IAE GmbH is certified via ISO 9001:2000. In 2005, five projects were successfully assessed according to ISO/IEC 15504 and most of the assessed H.I.S. processes reached level three.

Thesis 1: The introduction of capability maturity models and safety related standards is too expensive

The developers argued against the introduction because they feared a lot of extra work especially due to additional documentation resulting in missed milestones or delayed delivery dates. The management argued against the introduction, because feasibility analysis showed that ISO 61508 needs additional resources particularly with regard to the testing phase. Developers and managers doubted, that the standards would allow a balanced relation between effort and result.

Example IAE:

- Because of the standardized proceeding in project work, newbie's are easily introduced in the team and get early involved in their daily project work. Changes or additions of staff members into a project need only a very little extra time for adjustment into the project organization, project methods and tools, e.g. IAE uses a project monitoring sheet („Projektbegleitblatt") that enables a quick overlook of the whole project and its structure. Key documents help to assess the current status of the project. Saving of costs is the result of standardized proceeding in this point.
- Together, the staff members of IAE have created an inner structure with the help of standards, which is accepted and applicable. Based on this inner structure, IAE was successful certificated according to ISO 9001:2000 in 2004 and every re-audit succeeded without any major deviations. This certification means an increase of the company's value to our customers.
- IAE project leaders monitor and control their projects with defined status reports and assessment checklists, which have been derived from ISO/IEC 15504. Based on these existing standards, controlling of projects is comfortable and quickly fulfilled.
- Furthermore, IAE defined metrics considering the requirements of ISO/IEC 15504. These metrics are part of the used project-controlling tool. Based on these metrics, project reports and management reports are prepared automatically. This saves a lot of time during project work for the project manager and for the customers.
- IAE's quality manager engaged all developers and managers during assessment preparation and the assessment itself. As a side effect, some of them became experts with special knowledge about process and quality -oriented software development. They extended obviously their knowledge in project management or quality management. Since this time, our customers ask for these experts many projects, so IAE gets more requests for quality experts and attracts new business.
- Due to the utilization of ISO/IEC 15504, the quality department and the developers could realize the guidelines of ISO 61508 very quickly and without big changes in existing processes. As a result, IAE realizes many safety critical software projects and the developers continually increase their knowledge. Because of this practical appliance extra costs for additional coaching get less and less.
- One of the lessons learned during assessment was the awareness that an enhancement of effort during analysis phase, helps to reduce change requests and bug fixing. Tripling the effort in person days during analysis- and requirements phase effected adherence to defined milestones to 96 percent. As a result of following the guidelines of ISO/IEC 15504 and ISO 61508 deliveries where in the majority of cases punctual and of high quality, so customer satisfaction increased measurable.

Thesis 2: Models and standards are not sufficiently flexible

The developers feared that they would be handicapped in their work because of strict regulations. Especially in fire-fighting projects there would be no time to follow the standards strictly.

- The standard itself helps to adapt to specific IAE methods. Tailoring means the flexible handling of models and proceedings and IAE uses the standard itself to adapt to specific IAE methods and processes. For example, the work break down structure (WBS) is standardized at IAE but has flexible characteristics that are stored in the project controlling tool as a collection of project phases. IAE project managers may tailor the given WBS to the demands of their new project. Our standard WBS template helps to create defined work packets. As a result of using the tailored processes, creation and documentation of projects and file structure is simplified
- To monitor and to assess projects, IAE project teams use a special project checklist. This checklist covers every important checkpoint during life cycle of the whole project and includes base practices of the ISO/IEC 15504. So every team member knows the main focus to care about without any additional training. The project manager doesn't need to create a list of her or his own. As a result of using the guidelines provided by the ISO/IEC 15504, management practices to do high quality project work are well known and established. Required metrics and data can be generated automatically.
- Reporting and status evaluation are very important to control project progress and effort deviations. IAE offers defined templates for status reporting. The project decides if it is more effective to create the status manual by using this template or to use the automatic status report of the controlling tool. Both variants are allowed and can be combined with special customer requirements very easily, because typical required data are always included.
- With the help of specifications of ISO/IEC 15504 and a fussy requirement documentation based on ISO 61508, guidelines for coding were defined. Following these guidelines and the guidelines for the use of the c language in vehicle based software (MISRA C) helps the developers to write well laid out structured and expressive code. The use of MISRA encourages training and enhances developer's competence in general c programming. For example, because of this established knowledge, developers are able to read and check third party code and to deliver high quality code reviews.
- Helpful are the created requirement documentations and definite function descriptions out of the software requirements analysis process (ENG 1.2). Based on the requirement documentation test cases and software functions are derived. Inspectors of code are able to see very quickly which functions are required and compare these with the code under inspection. Because of this IAE performs code checks with diminished effort and develops testable software. During development of test cases it is possible to check very early, if any requirements exist, which can't be tested. Based on this standard process, at an early stage of development errors in requirements are detected and developers can flexible react and start a change request.
- Some more of the desired documents have to be created very early during development life cycle and very often information's are missing. Tailoring rules help to handle this problem and no show stopper appear, because every project team member knows what to do in such a situation.

Thesis 3: Realization is only possible using sophisticated tools

Developers complained that they would waste their time during project work with feeding lots of data into special tools, customer use to support their development process. They noticed that these expensive tools are rarely used in the correct way and so the input data are needless. So developers doubted for a lot of project phases, that these tools really would bring the expected benefit and help. On the other hand, the developers whished to reduce the amount of existing work sheets to document project results and project data and to replace them with a kind of supporting tool or database. They hated the intricate way to merge data manually together or to make them traceable. They complained about missing interfaces between different tools.

IAE management worried about cost, effort and that it would take a lot of time to establish such tools.

- The use of a project controlling tool is absolutely indispensable for IAE. A qualified project needs much more than a plain time recording tool with milestone recording features. Project managers at IAE monitor and control their projects with defined data and workflows. All data, which have to be collected, are precisely analyzed and derived from the company's objectives. These data generate the defined metrics and the metrics are collected within the project controlling tool. No one has to do redundant work or to merge several sheets manual together. Project managers and top level managers derive statistical data and reports from the project controlling tool, there's no manual step inside this process.
  Return on invest had been possible in a very short time. After one year intensive work with this tool, every ordered report is generated in a few seconds and the tool involves every input, output, activity, progress and effort of every work package of a project.
- Quality management and project management use this tool to improve their processes and workflows. There's no additional tool, because project phases are maintained within this tool and so improvement of the development process is possible too. As an example, the expert group also used the tool practically for their field-tests during the last year and still will do so in the future.. Every half year, upper management holds a meeting with IAE people to talk about the pros and cons in using the tool. As a consequence IAE continually improves all process areas, the work with the tool itself and the reports provided by the tool.
- Developers and managers agreed, that manual version control means an extreme extra work. Together they analyzed different tools and decided in favor of a freeware tool. There are some minor costs for the GUI but no more extra license costs for the tool itself. The online documentation is nearly perfect and a lot of support is guaranteed without any user help desk fees. The acceptance of this tool is a fast selling item, because the developers prefer open source tools. The tool itself helps to handle change management, requirements monitoring and supports version management. Information quickly passes on, less paper is needed (eco friendly) and rebuilding of sources and data is comfortable.
- The necessary set of tools is reduced to a minimum and applicable depending on the size of projects in the most flexible way. E.g., IAE does not use a special requirements database. Developers follow the defined activities based on the standard process. They use and produce the obligatory inputs and outputs during requirements phase and collect those analyzed data into the requirement specification, they don't need to feed them additionally to a database. No redundant work is needed and fewer defects occur.

Thesis 4: Capability maturity models and standards are introduced for their own sake

Developers and engineers at IAE wondered about the required processes, because during their whole studies, nobody talked about any quality processes or models in this intensity. Quality assurance like testing was known and accepted, but a straight red line during the whole development and system cycle was unknown. They feared a severed discipline due those standards that would permanently thwart project progress.
Mangers welcomed the idea of process orientated development, because they hoped for a higher productivity in development with increased quality.

- The defined business objectives at IAE are supported and achieved by the compliance to the predetermined methods of ISO 9001:2000 and ISO/IEC 15504. ISO/IEC 15504 helps to reach highest quality and to avoid bugs during development. This assists in customer satisfaction and guarantees follow-up-orders.
- IAE follows the instructions of ISO 61508 and learned to involve stakeholders very early and in time, to straighten out needs and requirements. As a result of a very good performance during planning and requirement phase, the expenditure for change requests decreased at more than 50 %. The number of changed requirements has been reduced significantly.
- During preparation for the official certification and assessment, all engaged engineers could increase their competences regarding project- and quality management. Now they are able to rely on a lot of lessons learned and experiences, which help them to run a project successfully.
- Apart from the official certification a lot of positive side-effects appeared. IAE developed new templates which support daily project work and decrease effort in documentation. For example standard status and to do lists arose out of this team work and as result from the assessment report meeting. Also the revision of all process descriptions made them more clear and applicable.
- Especially ISO/IEC 61508 limits the risk of hardware and software bugs and raises the safety of the whole system under development. For example, the execution of the FMEDA is required and this helps to detect failure cause and their consequences for the whole system.
- The use of ISO 61508 forces to examine malfunction of hardware, - software- or electronic components. IAE engineers now are able to analyze and define safety functions and to develop fail/safe concepts.
- ISO/IEC 61508 and ISO/IEC 15504 presume a clear structure of responsibilities. IAE Engineers know escalation steps exactly and whom to ask to get a quick decision. For example, every project repository includes a team folder, which shows participants, managers and supporters of the project. The content of this folder helps to add people during project documentation and to assign to them working with the software development plan or the test plan. This avoids misunderstandings referring to questions of liability, because stakeholders and suppliers are included in the responsible matrix.

Thesis 5: Processes given by means of models and standards are not accepted

IAE developers and engineers argued they would never need such defined standards to code in a correct way, because compiler and code checker tools help enough to avoid mistakes. Moreover, keeping with the rules would only be possible with intensive coaching because process models often impose constrains during development which doesn't fit with real business processes and often they are not up to date. Worst of all, internal processes are not accepted by customers and redundant work has to be done.

- All department managers at IAE created together an improvement and training concept for all employees and especially for prospective project managers. They all need to understand the required activities, inputs and outputs of ISO/IEC 15504 and ISO/IEC 61508, to use them for daily work and to create internal rules. Based on these activities a training group has been established. The members of this group are living and imparting the ideas of quality and business objectives quite naturally and teach them to newbie's as mentor. Training on the job is done during project work by help of these experts and under regard of special needs to the current project. The whole project team is involved in tailoring processes and the way to improve processes for special needs. Process are well understood and highly accepted.
- At IAE, project managers and their representatives are intensively involved in developing and optimizing the structure of projects. All experiences collected due to the implementation of the defined standards and their tuning to a project brings up even more expertise without additional training. For example, IAE QM and some of our engineers created a standard test plan by analyzing several different projects and under regard of customer requirements and demands based on ISO/IEC 15504, TPI-model and ANSI/IEEE 829. This test plan includes tailoring rules which ensure, that every project can work with this template in a most comfortable way.
- One very important component of IAE's quality system is the quality manager in charge. Together with the staff, quality management developed and realized a running quality system and several interfaces to project management, configuration management and top level management. Quality division itself is part of the top level management. At present all defined guidelines, workflows and processes are derived from current practice and they are accepted and established without resistance.
- At the beginning of a project in 2004 concerning an electric power steering for a demo car, the usage of ISO/IEC 61508 was hardly discussed. Because of the demo car, no SIL Level was required on customer's side. The developers and quality management decided to adopt ISO/IEC 61508 within this project, because in fact of a follow-up-order they were able to reuse processes, documentation and methods. Another benefit of this decision was to get a lot of experience with ISO/IEC 61508. The customer was very pleased to get project work of highest quality and detailed documentation. Tailoring of existing processes with manageable extra work helps to fulfill requirements of ISO 61508 and by using the recommended methods of this standard - as result, the project team develops safety critical software functions the right way. Right now, in April 2005 the follow-up-order had come and most of the developed data and documentations are in reuse, for this new safety critical project which requires SIL Level 3. Additionally another project (YawRate) with safety critical development tasks is ordered.
- All employees took part at the assessment meeting (debriefing) after the first qualified assessment (ISO/IEC 15504). One weak point was the risk tracking during project time, because project leads didn't like to monitor risks via the existing risk sheet. The new idea was to track risks via projects "To-Do Lists". The idea was tested in some projects and the tracking of risks worked fine with this standard method.

Thesis 6: Process models generate only puffed up documentation – „shelf ware“

IAE developers were angry about redundant documentation in different project phases. Without some rules and standard templates double work often occurred and those documents were hard to maintain. IAE people agree that some helpful documentation is important to understand what is going on in one project, but developers grumped, because they didn't want to document obvious and natural things. "Everybody knows what to do and knows team members, why do we need to write this down?"
Management and engineers had the opinion, that a plan is a fine thing, but in reality nobody cares about this documented plan, when unexpected things during the project happen.
Too much individual stuff would be documented and nobody would have an ounce of time to read all these documents about work packages, project boundary condition, analysis results and so on. No work would be done; team would only be reading large piles of documents.

- Supporting documents, as templates and checklists are really helpful for developers daily project work. These tools include contents and requirements of norms (ISO 9001:2000, 15504, 61508, etc.) For example, the code check list includes MISRA rules and parts of the style guide. The check list can be supplemented with special needs of different customers in an easy way. The QM toolbox was developed with the help of ISO/IEC 15504 and ISO 61508 and the usage of this toolbox avoids redundant and puffed up documentations.
- By means of tailoring rules and through strict model proceeding, the whole documentation of IAE's quality management system is reduced to fistful power point presentations and a few manageable key documents. For example, the quality manual, which includes the development manual, simply consists of 44 pages. For coaching of our employees and for newbie introduction, a small power point presentation is used, which is derived from this manual. As a result, knowledge transfer about the whole quality- and development process is done very fast and intensive.
- By using the appendices of ISO/IEC 15504 and 61508 it is possible to define the inputs and outputs for project and project phases very easily. The standards explain very well, what und why something is required. Derived from these additional explanations, the process inputs, activities and outputs were defined for IAE projects. Tailoring guidelines are required by these standards/models and we use tailoring guidelines to avoid redundant and needless documentations, therefore the definition of projects work products is done well and quickly, redundant products are avoided.

References:

1. DIN EN ISO 9000:2000 Dezember 2000
2. DIN EN ISO 9001:2000 Dezember 2000
3. DIN EN 61508-1 November 2002
4. DIN EN 61508-2 Dezember 2002
5. DIN EN 61508-4 November 2002
6. DIN EN 61508-5 November 2002
7. DIN EN 61508-6 Juni 2003
8. DIN EN 61508-7 Juni 2003
9. DIN ISO/IEC TR 15504 1-9 von 1998
10. MISRA Guideline April1998

It is Time to Increase the Focus on Software Acquisition

Angelvik, Bratthall, Mestl @ DNV Research (contact: Lars dot Bratthall at dnv dot com) +47 6757 9900
Stålhane @ NTNU (contact: Tor dot Stalhane at idi dot ntnu dot no) +47 7359 5000

Abstract

Software procurement is an area where suppliers now point at their customers and ask them to improve – or there are challenges in supplying software intensive systems of the right quality. This paper investigates published research about software procurement. Based on this, it seems that there are well-founded advice on software procurement in "a general software procurement situation", and "extremely large projects", such as the procurement of a new weapons system. However, it appears that the advice on how to procure software-intensive systems, for organizations with portfolios or systems of long-lived systems is limited compared to advice for simpler contexts. Initial results from further research in this direction are outlined.

Keywords

software, procurement, acquisition, archival research

1 Introduction

Software has become an integrated part of modern life from cashless shopping via online terminals, ticketing systems to huge municipal or governmental database systems managing not only more but also increasingly diverse amounts of data. As software spreads and establishes itself in new areas the number of software *acquisitions* is also growing, as well the number of people responsible for this. One study indicates that US enterprises spend more than $250 billion annually on procuring software products and services [Getto, G. et al]. At the same time, analyses show that about two-third of all IT projects become significantly more expensive or have to be redefined / terminated [The Challenges of Complex IT Projects], e.g., for public IT projects the average overruns in work-hours was more than 60% [Moløkken et al. 2005]. Studies like these may give an indication that the risks of unwise software investments are growing similarly.

This article investigates some of the research available and suggests provides initial results from a joint industry project aiming at decreasing the risk for high total cost of ownership for long-lived procured systems.

2 State of how to procure

2.1 How is software procurement researched?

Software procurement may be seen as a sub-class of general purchasing which has evolved from a mere buying function to an important strategic business process [Carr et al. 1999] that often has direct effect on the bottom line of a company's income [Buvik et al, 2000]. Purchasing in general may be divided into three main classes depending on the view point, i.e. supply chain management, relationship / network perspectives and organisational buyer behaviour [Ulkuniemi, 2003].

The supply chain perspective appears natural when considering purchasing as an integrated part of a larger logistic production system. For instance, standardised program packages, often termed as commercial off the shelf (COTS) components, like Microsoft Office or Outlook are relatively easy upgradeable, often automatically, and may be delivered from different vendors. This aspect gives birth to the concept of competing 'supply chains' [Farbey, 2001] where the one(s) is chosen that fit best the procuring organization's requirements and strategic perspectives. Although COTS software are usually cheaper than custom made systems, they have some side effects that can not be neglected, often prohibiting their implementation, such as customisation aspects, short lifetimes, or unwanted functionality [Craig Meyers 2001, Gerlish, 1998].

So far relative little attention has been paid to the supplier and customer relationship perspective as it does not provide operational models or normative guidelines in software acquisition [Brereton 2004]. This perspective is on a higher business strategic level [Håkansson et al. 1995] focussing on co-operative effects in business networks. Here, even co-operations with competitors are considered e.g. to develop common industry standards, to achieve higher *efficiency* of purchasing software intensive systems.

The organisational buyer behaviour perspective has in contrast an entirely internal point of view focussing on organisational rather then individual goals where internal procedures, processes and phases are central [Tanner, 1999]. It represents the most widespread approach to procurement and considers it as a series of decision making situations that may successfully be applied for the development of practical models and guidelines. In this category there is much of the work done within the military [The Defense Acquisition University, Gallagher et al], the industry [Getto et al] and in more academic (but often with a degree of military funding) environments [Choi et al 2001, ISR, SEI/ASP].

2.2 Risk mitigation

Software acquisition is especially risky in case of bespoke software systems, systems specifically designed for genuine tasks with a relatively long expected life time (e.g. 20 years) like computer controlled baggage handling systems or governmental software. Decisions for their replacement are usually postponed until maintenance cost or functionality limitations become too overwhelming. Due to the longevity and the often huge costs of new software systems (not only procurement costs but also costs connected with training and changes in business processes) make the replacement of bespoke software intensive systems a business critical strategic decision.

To reduce the risks and costs several systematic approaches and even standards have been developed over time, ranging from software lifecycle perspectives, requirement analysis, risk assessments, detailed process planning and development of standards and best practices [Hansen et al 1999, Software Acquisition Gold Practice]. Especially the military with their huge software intensive weapon systems, e.g. missile guidance, fighter plain software etc, are in sore need for acquisition guidance [Zambrana et al. 2004, Guidelines(...) 2004]. The same is true for non-military governmental software acquisition projects [Graff] in e.g. highway control [The road... 1998], social and health sector, [CITPO 2005, Krouse 1999] employment services etc. which in addition are very exposed to various political pressures, peculiar legal issues [Aigner et al, 2004, SIMAP 2005] and usually have to struggle with one-year budget planning.

Many of the software acquisition approaches have a life cycle perspective in that they already include the design and production phase of the software attempt to ensure a high software quality (e.g.. rapid prototyping, security aspects etc). There are software life cycle process reference model standards [ISO 12207] with their main focus on software development and software process assessments [ISO 15504, Wang et al 2000, Wang 2000, Bootstrap 3.2].

2.3 Capability determination models

Another approach focuses directly on the involved acquisition processes and assesses the company's capabilities in software procurement (ISO 15504, SA-CMM, CMMI-AM). These models provide detailed generic process oriented descriptions, benchmarking and improvement suggestions [SAGP-P]. Many of these initiatives originate from the US military and were originally developed for very large acquisitions. Interestingly for an impatient practitioner, the IEEE Std 1062, SA-CMM and the CMMI-AM approach all focus on what should be done, rather than how it should be done. This has two results: a) They are excellent for assessment; b) they can, and need to, be adapted/tailored to an existing process context. The quality of this relies on the assessing party, and, the quality of accompanying material. Other attempts model the maturity, or the abilities, of a provider of the software [eSCM].

3 Motivation for further research

There is a puzzling discrepancy between the available amount of research on software procurement on one side and the in industry perceived relatively low success-rate of actual software procurement on the other. Is this just due to a wrong perception or is there actually a potential for improvement of software acquisition within the business environment? A joint research project, called LCSP (**L**ife **C**ycle **S**oftware **P**rocurement), has been initiated by DNV in cooperation with one very large oil & gas company, a large international financial institution, an international telecom operator, and four government agencies, plus students. The project budget is several MNOK, plus time spent by partner organizations.

The objective of the project is to identify key practices that should be in place during procurement of software-intensive systems. The key practices are intended to together decrease the risk for high total cost of ownership for systems having a life-span of five to thirty years. The project also delivers a hazard identification that can be used as a checklist during procurement projects, along with lists of common risk-mitigating actions for each hazard. Compared to other research projects in the area, the key characteristics are: a) Explicit focus on achieving low total cost of ownership; b) explicit focus on long-lived systems that will experience change; c) explicit focus on the portfolio of both systems and projects ongoing. Partner organizations mainly participate because they see a potential for significant changes in TCO for their ICT systems; a rough estimate of the yearly total ICT budget among partner organizations is estimated to be between 6000 and 10000 MNOK; that is, the savings potential is significant. The organizations include different items in their ICT budgets, hence the measurement span.

4 Research method

The project partner organizations are studied in detail with respect to their past and present experiences with software acquisition. Through extensive focused interviews and short non-directive interviews [Frankfort Nachimias & Nachimias 1992] with multiple roles in the participating organizations, their specific challenges together with their own solution paths are uncovered. The interviewers have worked jointly during early interviews, followed by inter-researcher discussions, to calibrate understanding of answers. Apart from interviews, we also perform document studies (process/organizational descriptions), as well as multi-datasource case studies of particular projects from the partners. Each interview, or document studied is assigned an evidence ID, which is recorded in an evidence log. Thus, we retain complete traceability from evidences to suggestions, allowing the researchers to assess the strength of recommendations given.

When the research project passed 100 evidences, each evidence was jointly analysed by the researcher group for practises that affected TCO, and, for explicit or implicit risks wrt TCO that could be detected. This was assembled in a large electronic mindmap, because that way of organizing information a) allows the simultaneous visibility of large amount of information; and b) fast and easy clustering and changing of clusters. Clustering of practises and risks have taken place iteratively, with test workshops with project participants to ensure both partner understanding, but also validation of clusters identified.

5 Preliminary findings

Preliminary findings from practice indicate that

a) existing published software procurement models are not used. They are conceived by the practitioner as being too complicated (typically, the CMM set of models) requiring too much tailoring;

b) software procurement frequently has a few-systems focus, i.e. is considered as an event where it is not necessary to take all existing related systems into account;

c) software procurement does not take well enough into account the inherent flexibility of software, i.e., in many cases it is already known in the beginning of the procurement phase that the software will be subject to large scale changes in its functionality during its life time. Example: Top management gives an order about the procurement of a software-intensive system, for example a tailored CRM system. It is also given order about minimizing the TCO for the system. A project manager then takes over – and he is measured on if he delivers within time and budget (i.e., no TCO consideration any longer), and finally, some supplier organization is internally measured on how little time they can use to deliver adequate quality. Thus, one ability to affect TCO significantly, i.e., before thought has been transferred into in form, was lost because one did not take anticipated changes into account during engineering.

From initial analysis of the interview results three main categories of key process areas (KPA) have emerged that seem to play an important role in procurement processes. These are shown in Figure 1.

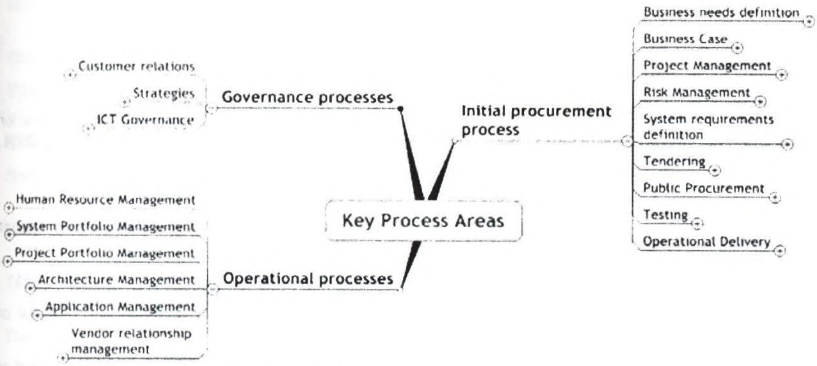


Figure 1 Key Process Areas identified

- *Governance Processes*: Typically, upper management is accountable for these KPAs.
- *Operational processes:* Typically, some line or corporate function are accountable for these KPAs.
- *Initial Procurement Processes*: Typically, these processes are handled by a procurement project. Initial analysis suggests that much of IEEE recommended standard 1062 can be reused as a base.

6 Next steps

In the research project, work still remains in grouping and extracting information about the software procurement processes and derive guidelines to reduce the total cost of ownership of long lived software systems. Guidelines are structured in a way that draws on practical experiences from both capability models (CMM variations, SPICE), and from the patterns community. For example, the structure contains both a "what/goal" part, and examples from the project partners of "how" they achieve the "what/goals". An early comparison to substantial work by [Bird et al 2002, Hansen et al. 1999] indicates that our work can reconfirm and thus increase the validity of many earlier described best practices in the initial procurement process category, while we find evidences for multiple other practices that we place in the two other categories, that seem to have received less attention in the software procurement literature. We estimate that all guidelines will be made available to the public towards the latter part of 2005; present non-anonymized material is restriced to project participants only. It is expected that the material will have value for a) organizations that wish to improve themselves wrt to reducing TCO for software-intensive systems; and b) organizations that wish to provide consultancy services in the area. The practises will be complemented with indications of which practises might be wiser to start improvement work on, to simplify incremental improvement. At the same time, there is nothing that outrules a broad assessment of an organization's degree of goal fulfillment in all recommended key process areas, and improvement initiatives based on that.

7 Literature

Aigner, W;Regner, P; Wiesinger, T; Küng, J. Supporting Public Software Acquisition Workflows – Implications for Data Models. Database and Expert Systems Applications, 15th International Workshop on (DEXA'04), 2004. http://www.faw.uni-linz.ac.at/PublicationFullText/2004dexa/bpim04_prtwjk.pdf

Bootstrap 3.2 (ISO 15504:1998 compatible)

Bird, Hansen, Peña, Gomez, Kirchhoff, Sundmaeker, Pereira, Marques, Norberg, Dorling, Pitette. IT Procurment – Best Practise Guide. Impronova, 2002.

Brereton, P. The software customer/supplier relationship. *Communications of the ACM, v* 47, n 2, 2004, p 77-81

Buvik, A; John G. When does vertical coordination improve industrial purchasing relationships? *Journal of Marketing* 64 (4) 52-64, 2000

Carr, A.S; Pearson J.N. Strategically managed buyer-seller relationships and performance outcomes. *Journal of Operations Management* 17 (5) 497- 519, 1999

Choi, S.J. / Scacchi, W., Modeling and simulating software acquisition process architectures. *Journal of Systems and Software*, Dec 2001

CITPO (Clinical Information Technology Program Office), http://www.tricare.osd.mil/peo/citpo/default.htm, 2005

CMMI Acquisition Module Version 1.0, Technical Report CMU/SEI-2004-TR-001, ESC-TR-2004-001

Craig Meyers, B; Oberndorf, P. Managing Software Acquisition: Open Systems and COTS Products. *ISBN: 0-201-70454-4, 2001*

eSCM: eSourcing Capability Model for Service Providers: http://itsqc.cs.cmu.edu/default.aspx? currentDDState=escmForSP&m=theModel

Farbey, B; Finkelstein, A. "Software Acquisition: a business strategy analysis," presented at *Proc. of Requirements Engineering* (RE 2001), 2001

Frankfort-Nachimias, C., Nachmias, D. Research Methods in the Social Sciences, 4th Ed. Edward Arnold, London, GB. 1992.

Gallagher, B. P; Shrum, S. Improving Defense Software-Intensive Systems Acquisition with CMMI-AM. http://www.sei.cmu.edu/news-at-sei/features/2005/1/feature-1-2005-1.htm

Gerlich, R. Lessons learned by use of (C)OTS Proceedings DASIA 98. - Data Systems In Aerospace, 1998, p 517-23

Getto, G. et al. Software Acquisition: Experiences with Models and Methods, *DaimlerChrysler, Research and Technology*, Ulm, Germany, http://www.iscn.at/select_newspaper/procurement/chrysler.html

Graff, M. Public Sector Challenges with Software Acquisition, Ownership & Software Asset Management, http://www.ncsl.org/programs/lis/CIP/PPT/graff02/

Guidelines for Successful Acquisition and Management of Software-Intensive Systems, Dept. of Air Force, Software Technology Support Centerv3.0, 2000, http://www.stsc.hill.af.mil/resources/tech_docs/gsam3.html

Hansen, S; Dorling, A; Schweigert, T; Lovett, J. IT Purchasing Guidebook for small enterprises. 1999 *QAI Europe*

Håkansson, H; Snehota, I. Developing Relationships in Business Networks. *Routledge*, London, 1995

IEEE Recommended Practice for Software Acquisition", IEEE Std. 1062, 1998 Edition (R2002), ISBN 0738103349 (Print), ISBN 0738115142 (PDF), 21 January 2003

ISO/IEC 12207:1995 and ISO/IEC 12207:1995/Amd.1:2002

ISO 15504:2003 (also known as SPiCE)

ISR, UCLA, Irvine, CA, USA.

Krouse, M Eight steps to successful hardware and software procurement [in healthcare]. *Healthcare Financial Management, v* 53, n 6, June 1999, p 60-2, 64

Moløkken-Østvold, K.J; Jørgensen, M; Sørgaard, P; Grimstad, S. Management of Public Software Projects: Avoiding Overruns. 2005. http://www.simula.no/publication_one.php?publication_id=818

SEi/ASP http://www.sei.cmu.edu/acquisition/acquisition.html

SIMAP (European Public Procurement) http://simap.eu.int/A/2330efb7-0aab-b34c-04c0377ccce3f48d_en.html 2005

Software Acquisition Capability Maturity Model, SEI, http://www.sei.cmu.edu/arm/SA-CMM.html

Software Acquisition Gold Practice: Acquisition Process Improvement. DACS (Department of Defense, http://www.goldpractices.com/practices/api/

SAGP-P: Software Acquisition Gold Practice: Focus Area: Process http://www.goldpractices.com/dwnload/practice/pdf/Acquisition_Process_Improvement.pdf

Tanner, JF Jr. Organizational Buying Theories: A Bridge to Relationships Theory. *Industrial Marketing Management.* 28 (3) 245-255, 1999

The Challenges of Complex IT Projects. *The report of a working group from the Royal Academy of Engineering and the British Computer Society*, April 2004, www.raeng.org.uk , ISBN 1-903496-15-2

The Defense Acquisition University http://www.ndu.edu/irmc/programs/dau.html

The Road to Successful ITS Software Acquisition. *US Dept. of Transportation, Federal Highway Adm.*; 1998 http://www.fhwa.dot.gov/tfhrc/safety/pubs/its/architecture/rdsuccessvol1.pdf

Ulkuniemi, P. Purchasing Software Components at the Dawn of Market, *PhD thesis, University of Oulu*, 2003

Wang, Y, Dorling, A. and Pitette, G. (2000), PULSE: An ISO 15504 Extended Model and Methodology. *Proc. 1st Intl. Conference on SPICE (SPICE 2000)*, Limerick, Ireland, June, pp.195-204

Wang, Y. (2000), PROBE: Development of a European Benchmark on IT Acquisition Processes, *Proc. of European Software Process Improvement (EuroSPI'00)*, Copenhagen, pp.8.24-8.34

Zambrana, M; Singer, D; Space and Missile Systems Center (SMC). *Software Acquisition Handbook, SMC/AXE, v. 1.0*, 9. Feb. 2004

8 Author CVs

Lars Bratthall

Dr. Bratthall received his PhD in software engineering from Oslo University in 2001 and his MSc from Lund University in 1996. He has professional background from software process improvement and from construction of industrial software-intensive systems. His current work is mission-based, at DNV Research: Help manage the risks to life, property and the environment caused by software-intensive systems.

Endre Angelvik

Angelvik received his MSc from Trondheim Technical University in 1998. He has spent ten years of his life working with mission-critical large transactional software intensive systems, mainly in the finance and telecom industries. He has held leading roles both at the supplier side, and with the procuring party. His current work is mission-based, at DNV Research: Help manage the risks to life, property and the environment caused by software-intensive systems.

Thomas Mestl

Dr. Mestl holds a PhD in mathematics and a masters degree in precision engineering. He has worked several years with information quality and interface design of mobile ITC equipment. His current work is mission-based, at DNV Research: Help manage the risks to life, property and the environment caused by software-intensive systems

Tor Stålhane

Tor Stålhane was born in 1944 and became a M.Sc. at the Norwegian Institute of Technology, University of Trondheim (NTNU) in 1969. From 1969 to 1985 he worked at SINTEF - RUNIT, department for languages and compilers. From 1985 he worked on his Ph.D. studies and finished his thesis on software reliability in 1988. From 1988 he mainly worked with quality assurance and software safety and reliability at SINTEF. In 2000 he became professor at the Norwegian University of Technology and Science in Trondheim. During the latest decade he has mainly worked on safety analyses of software intensive systems and measurement based process improvement

Assessment and improvement of IT departments: the case study of a large retail company

Paolo Salvaneschi[1], Daniele Grasso[2], Maurizio Besurga[2]

[1] University of Bergamo, Faculty of Engineering, Dalmine (BG), Italy
and Salvaneschi & Partners, Bergamo,Italy
pasalvan@tin.it

[2] Mediamarket S.p.A, Curno (BG), Italy
dgrasso@mediaw.it

Abstract

This paper describes a process assessment and improvement case study of the IT department of a large retail company. The characteristics of the company and the IT department are shortly introduced. The business motivations and the main approaches and results of the assessment process are presented and discussed. The assessment led to an improvement project that is currently running. We shortly highlight the improvement process and finally we discuss some aspects that may be useful for assessing and improving similar types of organizations. Among them the role of global service provider of IT departments, the relevance of communication and knowledge management processes and the need of assessment tailoring and people involvement. We also discuss the role of the ISO 9001-2000 certification approach in the improvement process and the need of a blend of technical and organizational aspects as well as communication and training / mentoring actions.

Keywords

Software process assessment, Software process improvement, IT Departments

1 Introduction

Assessment and improvement of software processes are important and critical issues for IT industry and have been extensively considered both in research and industrial practice [1]. Industrial process assessment experiences have been published for the case of large IT companies (see for example [2] for Siemens and [3] for Nokia) and also for SMEs (see for example [4] and [5]). These issues are certainly important for the IT vendors, but they are also of great relevance for IT departments of non-ICT organizations. Large organizations relay on business critical processes managed by IT departments. The quality of such processes (acquisition processes as well as in house development and services delivered to internal users) is important for the ability of the whole organization to compete and maintain / improve the market position. This is the reason why methods and technologies of software process assessment and improvement may be usefully applied to IT departments of the client side in the ICT market. This paper presents a process assessment and improvement case study of the IT department of a large consumer electronics and domestic appliances retailer. A short presentation introduces the company involved in the case study and the organization of the IT department. The discussion of the case study starts from the presentation of the business motivations of the project. The following chapters present the methodological approach of the assessment, based on the ISO 15504 [6] tailoring, the results of the assessment and the actions planned to improve the processes. The improvement process is currently running. A final chapter discusses the lessons so far learned and highlights some topics that may be of specific interest while assessing an IT department and may be reused in other similar cases.

2 The company and the IT department

The Mediamarket company is part of the Media-Saturn Holding GmbH, the European leading company in large-scale retail of Consumer Electronics and Domestic Appliances with a total of 503 stores in 10 countries, more than 12 billion Euro turnover and more than 30.000 employees (year end 2004). The Italy based company manages the brands MediaWorld and Saturn and operates 58 stores in the whole country, with 1,4 billion Euro turnover and more than 4.000 employees (year end 2004). The IT department is organized in several areas according to a functional structure (see Figure 1).

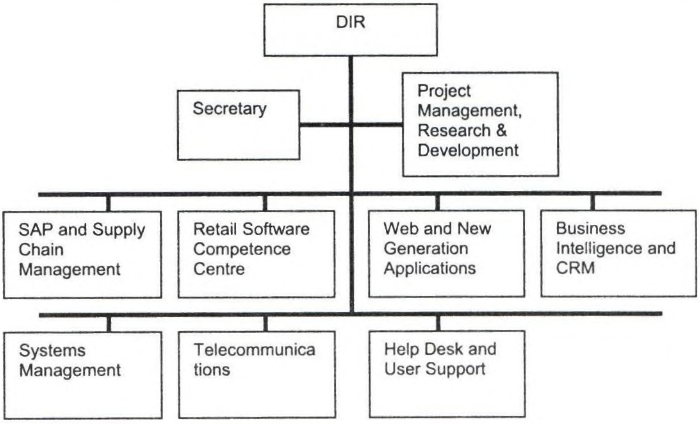

DIR
Secretary
Project Management, Research & Development
SAP and Supply Chain Management
Retail Software Competence Centre
Web and New Generation Applications
Business Intelligence and CRM
Systems Management
Telecommunica tions
Help Desk and User Support

Figure 1: IT Department structure.

Sap, Retail software, Web and Business Intelligence are involved in the development and maintenance of the application software. They develop new applications both trough internal resources and external contractors. Systems management, Telecommunications and Help Desk are service-oriented areas. They manage the data centre and the wide area network and support a large number of users geographically distributed (about 200 servers and 3000 clients). The IT department employs about 50 people.

3 Business motivations

In the past years, the IT department experienced a fast growing process, due to the expansion of the whole company. This led the IT management to run an improvement process whose aim was both to reach a new stability level of the department and support a higher service level for the internal clients. The management involved a third party as consultant. The aim was to cooperate with the R&D area to assess the existing processes and suggest possible improvements, according to the business priorities. Two points were essential in this decision. The first one was the Direction involvement and strong support from the beginning of the project. The second was a continuous action, during the project, for stimulating and maintaining the cooperation of the area managers as well as the persons involved in the operations of each area.

4 The assessment process

The assessment process is based on the ISO 15504 approach and is composed of the following steps:

1) Selection of the interesting processes
2) Evidences collection and measurement of the quality profile of processes
3) Definition of business goals and required quality profile
4) Diagnosis and suggested actions

We shortly present the characteristics and results of each step.

4.1 Selection of the interesting processes

The first step is the identification of the interesting processes. This is done on the basis of the ISO 15504 and ISO 12207 [7] lists of process. Figure 2 defines the processes of interest, according to the results of a set of meetings with the managers of the IT department and of each area. They basically include the whole set of ISO 12207 processes. Some rewording makes them more adherent to the local vocabulary (for example the infrastructure process is included in the system operation support). Some processes are reduced or included in other chapters (for example the supply process is included into the development process) due to the characteristics of an IT department. The processes are grouped in four chapters. According to the ISO 15504 classification, the first two groups include the customer-supplier and the engineering processes. The processes take into account both system and software aspects. Specific relevance is given to the quality control activities. The third group is related to the project, support and organization processes. The last group is partially outside the classification of the above-mentioned norms but is deemed to have a significant interest. It includes the activities related to the establishment of a good communication channel between the partners of the IT business in the company: the internal users and their departments as well as the areas involved in the IT department. It also includes the knowledge management processes required to maintain and reuse the rich set of structured information owned by the IT department. The list includes the whole set of processes of the IT department. Only part of them will be of interest for each specific area.

Process	Code	Definition
DEVELOPMENT, OPERATION AND EVOLUTION		
Acquisition	ACQ1	Request for proposal and contract management
Development	SVIL1	System requirements analysis
	SVIL2	Software requirements analysis
	SVIL3	System architectural design

	SVIL4	Software architectural design
	SVIL5	Software detailed design
	SVIL6	Coding
	SVIL7	Integration and delivery
	SVIL8	User and support documentation development
Operation	ESE1	System operation support
	ESE2	Software operation support
	ESE3	User support
Maintenance	MAN1	Problems acquisition and management
	MAN2	Problems analysis and solution
	CONTROL	
Quality Control	CON1	Software test
	CON2	Acceptance test
	CON3	Reviews and Audits
	CON6	Suppliers monitoring and acceptance test
Improvement	MIGL1	Measures acquisition and manag. for improvement
	SUPPORT	
Document Management	GDOC1	Document management
Conf. / Version Management	GCNF2	Configuration and version management
	GCNF3	Release management
Planning	GPIA1	Global and project level planning
	GPIA2	Project monitoring
	GPIA3	Project termination
Human Res. Management	GPER1	Skill profile def. and evolution
	COMMUNICATION	
Communication Management	COM1	Communication with users and with other departments
	COM2	Communication between Dept. Areas
Knowledge Management	KNM1	Dept. archive management
	KNM2	Documents re-use

Figure 2: IT Department interesting processes.

4.2 Evidences collection and evaluation

The information providing evidences for process scores is collected for each process of interest of each area. The score are based on the ISO 15504 levels. An additional requirement identifies the processes of figure 2 that are not interesting for the specific area under assessment:

NA– The process is not interesting for the specific Area
0 – Not performed or only partially performed
1 – Performed informally
2 – Planned and tracked with documented evidences
3 – Based on standard methods and, if required, tailored methods
4 – Quantitatively controlled
5 – Continuously improving

The type of information collected for each area includes:

- General description of role, structure and characteristics of the area;
- Interaction with other areas;
- Definition of the interesting processes;
- Classification of projects / activities (e.g. for a software development area, we classify low complexity and high complexity projects) and definition of significant examples for each class;
- Collection of the available documents for each example;
- In case of not available evidences based on documents, the area manager is requested to describe the current practices.

The area manager is also required to express, from his point of view, the critical issues and the suggested business goals. The assessment actors are the IT Manager, the R&D manager (acting as "internal sponsor" of the project), the area managers and the external assessor. The information is collected and processed through the following steps:

- Introductory discussions with the R&D manager and distribution to each area manager of a short presentation of the project, its goals and methods.
- Kick off meeting. The IT manager presents the project to the area managers and enforces its relevance. The aims, roles and methods are discussed and a plan of actions is defined.
- The assessor meets every area manager collecting the available information.
- The assessor analyses the information (using the ISO 15504 practices as a guideline) to extract the evidences for the score of each process and writes a report for the area manager. It includes the ranking of processes and the explanation of each score, given the collected evidences.
- The report is submitted to the area manager
- The assessor discusses it with the area manager. When the consensus is reached, the report becomes part of the final report.
- The final results are collected and summarized in a final report The report is discussed with the R&D area manager and the IT manager and is presented to the area managers in a final meeting.

For example figure 3 shows the measured profile for one of the software development areas (see figure 2 for process coding). The main part of the development processes is planned with documented evidences (level 2). The interesting processes for operation and maintenance are at level 1. The majority of quality control processes are at level 1, while no improvement process is running. The support and communication processes are at various levels. Some process like document and configuration / version management is very well performed al level 3, while some other one (e.g. planning) requires a significant improvement. The large differences in maturity of processes are mainly due to the business priorities (e.g. the need to manage the configuration and delivery of complex systems), the time pressure and the locally available skills. If we look at the complete assessment, we may classify the quality profiles into two classes: development oriented areas (the four boxes of the intermediate level of figure 1) and service-oriented areas (the three boxes of the bottom level of figure 1). Inside each class, the profiles are quite similar.

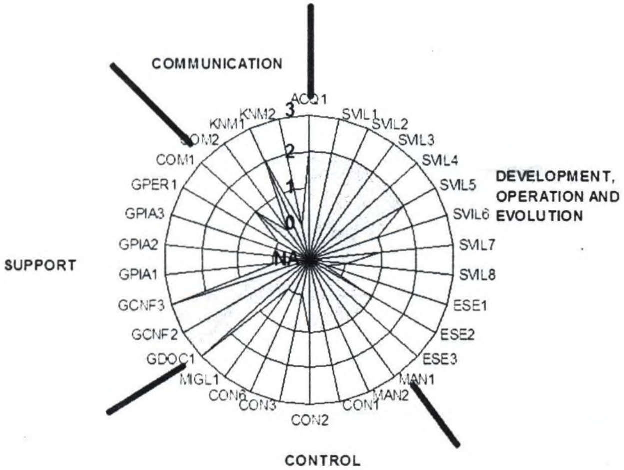


Figure 3: The measured process profile for a software development area.

4.3 Business goals and required process profile

The IT manager evaluates the assessment report to identify the critical issues, according to the business goals and the priorities of the department and the whole company. The result is a required quality profile for the processes of each area. For example figure 4 shows the measured profile referred in figure 3 (grey area) and the required improvements (black area). The black part of the diagram shows the high priority improvements. The maintenance and quality control processes as well as some development processes are required to move to the same level 2 of the development processes majority. The planning processes have to be formalized and documented and a significant effort is required in the communication and knowledge management area. The reason is that, due to the growing complexity of the organization, the need of communication and coordination between the areas became more and more important (for example, many projects involves actors coming from various areas). Moreover, the stability and the evolution of the information system are based on the availability of a consolidated body of information and knowledge to be owned by the organization and not by the individuals (for example data models, systems configurations, operation procedures).

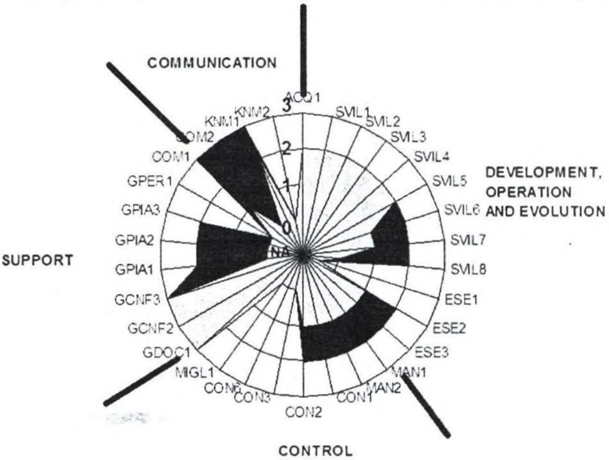


Figure 4: The required process profile for the area referred in figure 3.

The above mentioned required quality profile is an example from a development area. If we examine the whole set of the quality profiles, the required improvements are similar for each profile class (development oriented and service-oriented). In addition, each profile of the two classes may have specific required improvements.

4.4 Diagnosis and planning for process improvement

The assessment report includes a detailed set of evidences and examples that are the basis for the assignment of process scores. This supports the explanation of the measured quality and the identification of the suggested actions for process improvement. The result is the definition of an improvement plan including, for each process to be improved, a list of actions and associated tasks as well as the estimated effort and timing. The actions belong to the following categories:

- Identification of adequate methods and adaptation to the specific cases.
- Development of procedures and technical standards. Tools evaluation and delivery.
- Training through tailored courses. Projects mentoring and monitoring.

The improvement process is organised using the ISO 9001:2000 [8] certification approach, even if the organization is not interested to the certification by itself. The first step of the plan will last one year and is currently running. A new assessment is also planned at the end of this phase.

5 Lessons learned

In the following final considerations we want summarize the lessons learned and the aspects that may be useful for assessing and improving similar types of organizations.

5.1 Service vs. software: the system approach

A first question may relate to what is specific of an IT department of a large non-IT organization, for example versus a software development company. An important point is that an IT department does not produce software, but delivers a global service. This means that the system view is mandatory. The service level depends on many aspects: the application software development, evolution and maintenance as well as the management of the technological infrastructure and the user support during the every day operations. This also means that communication and coordination is very important. This was also recognised by the area managers. In the first phase of evidences collection, the majority of them included in the suggested business goals the need for a better communication and coordination. The final service level is an emerging property of the system and depends on the ability of teams to cooperate in a proactive way. It requires a mixture of planning, commitment and available knowledge. The planning is the general framework, while the commitment and knowledge are required to effectively react to every day not forecasted events. This is the reason why the communication and knowledge management processes are so important in the process assessment.

5.2 Assessment tailoring and people involvement

As stated in other cases studies (see, for example [9] for a comparison of two different approaches in two Norwegian IT companies), the success of the assessment and improvement effort strongly depends on the following two factors:

- The management commitment and the cooperation of every involved actor;
- The ability to tailor the assessment and the improvement to the specific goals and needs.

In our case the R&D manager acted as internal sponsor, cooperating with the external consultant to define the project. The IT manager was directly involved not only in the decision phases, but also in the presentation of the project to the area managers. The seven area managers were involved as direct actors of the projects (they discussed with the assessor each aspect to reach the consensus). The project was presented to them not only as a company requirement, but also as a way to improve their professional knowledge. The contribution to the project was also included in the rewarding system parameters. The seven areas in the IT Department were interviewed during the assessment. The process involved the area managers and their main reference people. The improvement process is progressively involving all the persons working in the areas. The tailoring for the organization was also carefully taken into account. The assessment was based on the ISO 15504 method, but the content was tailored (see for example the list of the interesting processes or the specific required quality profiles) according to a cooperative analysis made by the assessor and the management.

5.3 Using the certification approach as a tool

In the improvement process, an interesting aspect is the role of ISO 9001-2000 and the certification approach. The certification is not considered interesting by the organization. Nevertheless, the improvement actions were organized as a certification of the IT department. The process improvement is delivering a set of procedures and document schemas with the same approach of the quality manual for the certification. It seems that the quality manual approach provides a practical tool for organizing the decisions, the rules and the best practices. Writing a procedure is a way to define the rules of the organization, discuss them and reach the consensus. Only specific aspects more strictly related to the details of the certification requirements, are not considered. The general approach is to find a solution as simple as possible, discuss it, test in service and modify if required. The R&D manager has the role of Quality Manager and the responsibility of the improvement process, while the area managers are process owners. The interesting point is that the ISO 9001 is useful, even if the formal and commercial aspects (e.g. the approval of a certification body) are not required.

5.4 Training and mentoring

A final consideration is that, in the improvement actions, the technical and organizational aspects are not sufficient. Simple and effective procedures or good and well-tuned tools are important, but they are not sufficient. Training, mentoring and monitoring are mandatory for the success of the project. Our improvement plan includes a significant effort to support the delivery of the new approaches. For example, for the development process, a set of pilot project is defined for each area and the external assessor acts as consultant discussing periodically the results and supporting the improvement. Specific training actions are also delivered: for instance a seminar on how to use the documentation schemas, a short course on requirements analysis, and a short course on software design. Our experience shows that moving an organization from an existing quality profile of processes to a better one is basically a cultural problem. This is the reason why this type of actions is difficult and prone to failures. Moreover the actions require time to become effective. It is not sufficient to define a method, write a procedure and deliver a training seminar. A mentor has to support the every day operations until the new method becomes part of the usual cultural framework.

6 References

[1] Fuggetta A., 2000, Software Process: A Roadmap, in "The Future of Software Engineering", Anthony Finkelstein (Ed.), ACM Press

[2] Lamprecht W., Messer T., 1998, Software Process at the Gate to the Top, EuroSPI98

[3] Lindholm T., 1998, Implementing SPI: Combining business process improvement and system development at Nokia, EuroSPI98

[4] Bucci G., Campanai M., Cignoni G.A., 2000, Rapid Assessment to solicit Process Improvement in SMEs, EuroSPI2000

[5] The SPIRE Project Team, 1998, The SPIRE Handbook,, Marty Sanders (ed)

[6] ISO/IEC TR 15504-1-9 Information technology -- Software process assessment -- Part 1-9

[7] ISO/IEC 12207 Information technology -- Software life cycle processes

[8] ISO 9001:2000 Quality Management Systems Requirements

[9] Tore Dybå, Nils Brede Moe, 1999, Rethinking the Concept of Software Process Assessment, EuroSPI99

7 Author CVs

Paolo Salvaneschi
Associate professor of "Software Engineering" at University of Bergamo, Faculty of Engineering and director of "Salvaneschi & Partners" (Software Engineering consulting company). He joined the University after spending twenty-five years as software architect and head of software development groups in industry and research labs. His professional experience is in software engineering and A.I. application to monitoring and data interpretation as well as ICT applications in various areas (industry, civil engineering, environmental control). He published some hundred papers.

Daniele Grasso
Project Manager and R&D manager at Mediamarket S.p.A IT Department

Maurizio Besurga
Chief Information Officier of Mediamarket S.p.A.

Performing ISO/IEC 15504 Conformant Software Process Assessments in Small Software Companies

Christiane Gresse von Wangenheim[1], Timo Varkoi[2], Clênio F. Salviano[3]
[1]Universidade do Vale do Itajaí (UNIVALI) - Computer Science
São José/SC - Brazil
gresse@univali.br
[2]Tampere University of Technology - Information Technology
Pori - Finland
timo.varkoi@tut.fi
[3]Centro de Pesquisas Renato Archer (CenPRA)
Campinas/SP - Brazil
clenio.salviano@cenpra.gov.br

Abstract

Small software companies face similar issues regarding software quality improvement and process assessments as larger companies. The main difference is that smaller companies seldom have specialized or competent resources to solve the problems. Therefore, the development of assessment methods also from the viewpoint of small companies can support the software industry. Based on experiences applying ISO/IEC 15504 in small software companies in Finland and Brazil, we present a flexible approach to efficient process assessments. Flexibility requires a continuous assessment model, so that the scope of process improvement and assessment can be defined based on the prioritized needs of an organization. Our experiences show that 15504 can also be applied with success in small software organizations. The paper presents how the assessments were run and lessons learned on applying 15504 in this kind of organization.

Keywords

ISO/IEC 15504, Software Process Improvement, Software Process Assessment, Small Organizations

This work has been realized with support of the CNPq, an entity of the Brazilian Government directed to scientific and technological development.

1 Introduction

Today, in many countries small companies (SC)[1] are important for the economy and employ a great share of people (for example, in Brazil they represent about 70% of the total number of organizations in the software sector [10]). Typically, this type of company faces similar problems as any type of company, e.g., regarding the quality of their products, although, in general, SCs normally face these problems to an extreme due to the informality of their processes and lack of resources. These characteristics can harm SCs in relation to their quality, productivity and competitiveness, or even their survival on the market. Therefore, the identification of problem areas and the systematic establishment of improvement actions are vital for their long-term success.

However, it seems that small software companies still find it difficult to run process assessments [2,8,9]. One reason is that many small companies are simply unaware of the existence of such methodologies [2,10]. Often there is also the perception that software process assessments conformant to these models/standards can be expensive and time consuming [11]. There is also often the perception that assessment models and standards are more targeted towards large organizations [2,8,11], including, e.g., practices for documentation and process formalization [9], which have been criticized to be inappropriate in the context of small companies [8,11].

In this context various initiatives have been started focusing specifically on software process improvement and assessment in the context of small software companies. Two examples are the project 15504MPE [1] in Brazil and the SataSPIN project [17] in Finland. The projects focus on the development of an assessment method customized to small software companies in order to enable effective software process assessments in conformance with the standard IS/IEC 15504 directed to their needs at low cost. Both methods, which were developed independently, are going in the same direction being similarly structured and based on ISO/IEC 15504 allowing a flexible adaptation of the process assessment in accordance with an organization's specific characteristics and goals.

In the context, also first experiences of both projects have been gathered. The 15504MPE project started in February 2003 and so far 7 trials have been performed in Brazilian small software companies. The SataSPIN project was organized in two phases; the first phase started in August 1998 and the second phase in June 2001. The project was ended in May 2003. Altogether 20 small and medium sized software companies took part in the projects, which resulted in an established software process improvement network.

In this paper, we present a unified view on the assessment methods developed and present a summarized overview on our experiences on the applicability of the international standard ISO/IEC 15504 with small organizations. We discuss the effectiveness and efficiency of the process assessment method as well as its effects on software process improvement.

2 Software Process Assessment Models and Methods

Software process models aim at continuous improvement. Small organizations require special guidance to be able to benefit from the models and to be successful in the process improvement. Process assessments provide lots of information as the basis for improvement activity planning. Today, the two most prominent software process improvement and assessment approaches are the CMMI framework [3] and the international standard ISO/IEC 15504 [6]. In general, both frameworks are directed to any type of company and being based on a continuous representation both offer flexible approach adaptable also to small software companies [14,15]. An advantage of the standard ISO/IEC 15504 specifically in the context of small software companies is its broad scope of, not only, technical processes. However, as the standard itself defines only a minimal set of generic requirements for an assessment, it does not provide in itself sufficient support for its ready application in a small software company.

[1] A common definition used classifies small companies with less than 50 employees.

In order to effectively and efficiently support process improvement in the context of small software companies, considering their specific characteristics and limitations, various ISO/IEC 15504 conformant assessment methods have been developed, such as, for example, RAPID [13], FAME [4], SPINI [9], MARES [1], SPIRE [14], and TOPS [2] among others.

In the context of our research, we developed the tailored assessment methods:

- SPINI: The assessment method has been developed during the SataSPIN project/Finland as part of a SPI Initiation Framework (SPINI). The framework is based on the ideas of IDEAL and SPICE, and the experience acquired when implementing software process improvement in small organizations. The framework expresses the essential activities that were found useful in starting up SPI using external support.
- MARES: The method has been developed as result of the research project 15504MPE/Brazil based on our experiences applying 15504 in small software companies focusing on continuous improvement based on the CenPRA approach for SPI [1] based on IDEAL.

In this section we provide a general overview and comparison of both methods. The methods are basically composed of a process assessment model and an assessment process.

2.1 Process Assessment Model

The process assessment models used by both methods are based on the exemplar assessment model of Part 5 of ISO/IEC 15504 [5,7]. The capability dimension is adopted as-is from capability level 0 up to level 3. Capability level 4 and 5 are not considered due to the, generally, low level of capability encountered in small software companies.

The process dimension has also been developed based on ISO/IEC 15504-5. However, due to the characteristics of small software companies, some processes of the exemplar model have been disregarded as being irrelevant in most cases in the process assessment model used by the MARES method. For example, as most small companies do not have subcontractors, processes related to the Acquisition Process Group have not been considered. However, if any of these turns out to be an important process in a specific context, it can easily be re-integrated based on ISO/IEC 15504-5. In addition, some processes (e.g., Project Management and Risk Management) have been re-grouped into one process.

Assessments during the SataSPIN project were performed according to the prioritized needs of the participating companies. As most of them were in the very beginning of process improvement, the selection of processes for the assessments consists mainly of engineering processes together with project management. Detailed distribution of the assessed processes is presented in Table 3.

2.2 Process Assessment Process

Both methods are based on the requirements for performing an assessment as defined in ISO/IEC 15504-2. In addition, a phase in the beginning of the assessment is added in order to characterize the company, to obtain a general understanding on the complete software process and to systematically support the selection of the processes to be assessed. In both methods, the assessments are performed similarly, but the assessment planning related activities are organized in a different way (Table 1). The MARES method pays more attention to the control and feedback of the assessments.

Table 1. Comparing the basic phases and activities of both methods

SPINI		**MARES**	
Phase	*Activity*	*Phase*	*Activity*
Needs analysis	Alignment General assessment Scope definition	Planning	
		Contextualization	Data collection Data analysis

Process assessment	Assessment planning Data gathering Assessment reporting	Process assessment	Preparation Data collection Consolidation of observations Validation Process attribute ratings and general results Reporting
			Monitoring & Control
			Post-mortem

Planning: During the planning phase, the assessment is organized and planned, including the definition of the purpose(s) for performing the assessment, its scope, available resources, constraints, schedule and the required output. In addition, participants and their responsibilities are identified. In general, the assessment is performed by an assessment team, composed of a competent lead assessor and a support assessor. Participants from the organization are the sponsor of the assessment and representatives involved in the software process.

Needs Analysis/Contextualization: The activities in the first step are alignment, general assessment and scope definition. The alignment provides a common understanding within the organization of process thinking and related concepts. A general assessment is used to study the overall situation of the organization's business and improvement goals and its software processes. As results of this phase, target profiles are defined, indicating high-priority processes and their capability level to be achieved in order to meet the organization's goals. Organization's needs and present situation are considered, and the assessment strategy and scope is defined including the processes that will be investigated in detail. All results of the contextualization phase are documented and revised. Based on the results of the needs analysis/contextualization phase, the assessment plan is updated and completed determining schedule, participants, source material and restrictions to the assessment.

Process Assessment: The second step concentrates on a planned, detailed assessment of the selected processes. Therefore, the data collection is prepared, including the refinement of the strategy and techniques to be used for collection and analysis, as well as the definition of process attribute indicators. Data gathering consists of studying the source material and interviewing process performers in assessment sessions. The collected evidence is then analyzed by the assessors mapping the data to the indicators of the process assessment model. This is explicitly documented by stating the consolidated observations and briefly indicating the level of achievement of the indicators in order to facilitate later on the rating for the process attributes. The observations are then validated by representatives of the organization during a feedback session in order to ensure that they accurately represent the assessed process(es). In addition, the assessors validate them in order to ensure their objectivity, consistency and sufficiency with respect to the scope of the assessment. Then, the capability of each process is rated unanimously by the assessors resulting in the definition of a set of assessed process profiles based on the validated evidence. By comparing the assessed process profiles to the target profiles and considering the results of the contextualization phase, strengths, weaknesses are identified. In addition, potential risks and improvement suggestions are identified by analyzing the gaps between the target and the assessed process profiles. The assessment report records the findings and process ratings and profiles in a detailed manner and is presented to the SPI stakeholders. It gives precise input for improvement actions.

Monitoring and control: All activities during the assessment are monitored and controlled with respect to the assessment plan. If necessary, corrective actions are initiated and the plan is updated accordingly.

Post-mortem: Once the assessment is finished, the assessors hold a brief post-mortem session in order to discuss the performance of the assessment as a basis for the continuous improvement of the assessment method. Further feedback on the execution of the assessment, its results and benefits is also provided through a questionnaire to be answered by the assessment sponsor.

3 Performing Assessments in Small Software Companies

Focusing on software process improvement in small software companies, we have run eight software process assessments in small companies in Brazil in the context of the project 15504MPE during

2002-2004, and thirteen in small software companies in SataSPIN project in Finland during 1999-2001. The companies varied from recently founded ones (partly associated to incubators) to companies founded 15 years ago, representing different growth stages. Their size varied from 2 to 56 people. One trial was run in a rapidly growing company with about 56 employees at the moment of the assessment, which strictly seen, would be considered a medium organization. However, as the company only recently before the assessment employed several people, the results are still considered relevant. Some of the recently founded companies were very small with only 2 to 5 employees.

Most of the companies developed their own software products. The types of products and services ranged from the development of individual applications, development and/or customization of standard products, or services requiring the development of software systems. Application domains include e.g. commercial and industrial applications, enterprise information systems, distance learning support, logistics and telecommunications. Only two of the companies were primarily subcontractors.

The assessments have been performed in accordance with ISO/IEC 15504 using either the SPINI or MARES method. Summary of the assessments analyzed in this paper is presented in Table 2.

Table 2. Summary of assessments

	SataSPIN	15504MPE	TOTAL
Companies	8	8	16
Assessments	13	8	21
Projects	22	12	34
Participants	61	37	98
Process instances	77	27	104

Due to the time frame in which the assessments took place, different versions of 15504-5 have been used (ISO/IEC TR 15504-5:1999 and ISO/IEC CD 15504-5:2003), as this part of the standard is still under development. In each assessment, different processes have been assessed up to different levels of capability depending on the company's specific characteristics and goals. Table 3 presents an overview on the number of process instances assessed with respect to the processes of 15504-5.

Table 3. Overview on the number of instances assessed per process

Process	Version	15504MPE	SataSPIN	Total
Supply	TR	4	1	5
Supplier Tendering	CD	1	0	1
Contract agreement	CD	1	0	1
Software release	CD	1	0	1
Software acceptance support	CD	1	0	1
Requirement elicitation	TR / CD	2	16	18
Customer Support	TR / CD	2	2	4
System requirements analysis and design	TR	0	6	6
Software requirements analysis	TR	0	14	14
Software design	TR	0	10	10
Software construction	TR / CD	3	3	6
Software integration	TR / CD	1	2	3
Software testing	TR	1	2	3
System integration and testing	TR	0	1	1
Software installation	CD	2	0	2
System and SW maintenance	TR	0	1	1
Documentation	TR	0	3	3
Configuration management	TR	0	1	1
Change request management	CD	1	0	1
Project management	TR / CD	6	13	19
Risk management	TR	0	2	2

Measurement	TR	1	0	1

Here we can observe a strong focus on project management, requirements elicitation and analysis, as well as design. Comparing the processes assessed in both projects, we can observe a stronger focus on the engineering process in SataSPIN, with exception of the software installation process which did not exist in the TR version and was exclusively assessed in the 15504MPE project. Projects also show some emphasis on the supply process, as well as in the case of 15504MPE on the customer support process covering specifically early and final phases of the software process. As typical for companies just starting SPI, most assessments resulted in either capability level 0 or 1 in the first assessment.

We also can observe that only some of the processes identified as high-priority for SPI in the small companies confer to process areas focused, e.g., by the CMM-SW/staged framework on maturity level 2 [12]. Others, such as Supplier Agreement Management, Measurement and Analysis, Process and Product Quality Assurance, and Configuration Management, were not or considered only in one assessment. This may indicate the importance of the flexibility of the assessment method enabling the selection of the processes to be assessed in accordance with the specific characteristics and business goals of an organization.

Effects of the assessments on software process improvement

In general, all participating companies considered the assessment as very beneficial and have already begun to implement improvement actions. Based on surveys after the assessments, the companies agreed on that the assessment contributed to a better understanding of their software process. As result of the assessment, strengths and weaknesses were identified that only partially had been recognized before the assessment, and which were confirmed by the companies. Especially, the explicit indication of improvement suggestions was considered helpful.

The key business benefits of SPI were reported to be: 1) Improved control of outcomes, 2) Development of knowledge and skills, and 3) Improved manageability of the operations [16].

However, an interesting tendency, we observed, is that the very small companies with less than 5 employees (including in some cases only 1 or 2 full-time employees), and which had only been recently founded, started improvement actions (e.g., developed tools), but did not effectively established them, as they considered them only to be useful once the size of the company increases.

Methods were created as a part of national projects to support SPI and are therefore intended to be used in a larger framework. Both methods are similar in their assessment approach and produce the desired outcomes. Methods can be developed further by integrating the experiences gathered during their use. The ISO/IEC 15504 can efficiently and effectively be applied also in the context of small companies.

4 Lessons Learned

Our experiences indicate the applicability of the standard ISO/IEC 15504 for software process assessment also in the context of small companies. The process assessment models based on the exemplar process assessment model defined in 15504-5, as well, as the assessment processes were considered adequate providing findings that were confirmed by all participating companies.

Some lessons learned:

- **Flexibility of assessment model** based on a continuous representation has shown to be important in order to support the adaptation to this kind of organization and each company. However, most assessments methods based on continuous models do not yet provide detailed support for the identification of target process profiles and the selection of processes to be investigated in detail.
- **Focus on the principal high-priority processes** in alignment with the company's business goals, characteristics and resources available for SPI. This is important to keep the assessment focused on the most relevant processes to be targeted for improvement, as well as to keep the assessment cost as low as possible with maximum coverage of relevant processes.

- **Coverage of the process reference model** has shown to be important, especially as in the context of small companies, processes in direct contact with the customer (such as supply, software installation, customer support or change request management) have turned out of high priority.
- **Data collection** based on group interviews has turned out to be another activity critical to the success and costs of an assessment. The interviews were performed in an open style, not using any kind of questionnaire or checklist, based only on an interview plan, which lists all issues to be elicited during the interview. This was considered very adequate enabling a valid data collection, as the companies' representatives could freely describe how the processes are executed, leaving the mapping to the processes from the process assessment model to the assessors. This was considered especially important, in those cases where we observed a low level of Software Engineering knowledge among the companies' representatives and, therefore, their incapability to do this kind of mapping on their own.
- **Identification of risks and improvement opportunities**. As the objective of the assessments was on process improvement, we observed that besides the minimum requirements regarding the assessment output, principally including the process profiles, it is necessary to point out also the principal strengths and weaknesses related to the assessed processes, as well, as risks and improvement suggestions in order to provide initial support for the planning of improvement actions.
- **Availability of documents templates and tool support.** Based on our experiences, the assessment effort can be considerable reduced when templates for the documents to be produced during an assessment are available (in the respective native language, if needed). Further cost reduction could be achieved by the usage of tool support for the management of documents during all assessment activities and identified possibilities enabling the partial semi-automatization of the handling of information as a basis for the creation of initial versions of some documents to be produced.

Overall, systematic process assessments support management of small software companies and provide valuable information for their process improvement. As small software organizations are in the beginning depending on external support for SPI, the main issue is to convince them on the expected business benefits.

Acknowledgments

The authors would like to thank the funding organs of the research project 15504MPE including CNPq, UNIVALI and CenPRA, as well, as FUNCITEC and SEBRAE. We would also like to thank the representatives of the companies that participated in the assessments.

5 Literature

[1] A. Anacleto, C. Gresse von Wangenheim, C. F. Salviano, R. Savi. A Method for Process Assessment in Small Software Companies. In Proc. of the 4th Int. SPICE Conference, Portugal, 2004.

[2] G. Bucci, M. Campanai and G. A. Cignoni. Rapid software process assessment to promote innovation in SMEs. In Proc. of the European Software Day at the 25th Euromicro Conference, Italy, 1999.

[3] Capability Maturity Model® Integration for Software Engineering, Version 1.1 (CMMI-SW, V1.1), Continuous Representation. Technical Report CMU/SEI-2002-TR-028, Software Engineering Institute, 2002.

[4] Fraunhofer Institute for Experimental Software Engineering. FAME: A Business-Focused Method for Process Assessment. http://www.iese.fraunhofer.de/fame

[5] ISO/IEC CD 15504-5:2003 Information Technology - Process Assessment - Part 5: An exemplar Process Assessment Model (not publicly available).

[6] ISO/IEC IS 15504: Information Technology – Process Assessment, Part 1 to Part 5. International Organization for Standardization, 2003-2005.

[7] ISO/IEC TR 15504-5:1999 Information technology - Software process assessment - Part 5: An assessment model and indicator guidance.

[8] D.L. Johnson and J.G. Brodman. Tailoring the CMM for Small Businesses, Small Organizations, and Small Projects. Elements of Software Process Assessment and Improvement, K. El Emam and N.H. Madhavji, eds., IEEE Computer Society, 1999.

[9] T. Mäkinen, T. Varkoi and M. Lepasaar. A Detailed Process Assessment Method For Software SMEs. In Proc. of the European SPI Conference, Denmark, 2000.

[10] Ministry of Science and Technology (MCT), Indicators of the Information Technology Sector, 2004 (in Portuguese). http://www.mct.gov.br/Temas/info/Dsi/indsetor/Indsetor.htm

[11] M. C. Paulk. Using the Software CMM in Small Organizations. In Proc. of the Joint 16th Pacific Northwest Software Quality Conference and 8th Int. Conference on Software Quality, Portland, 1998.

[12] Paulk M., Curtis B., Chrissis M. B. & Weber C.: Capability Maturity Model for Software, Version 1.1. Technical Report CMU/SEI-93-TR-24, SEI 1993.

[13] T. P. Rout, A. Tuffley, B. Cahill and B. Hodgen. The RAPID Assessment of Software Process Capability. In Proc. of the First Int. SPICE Conference, Ireland, 2000.

[14] M. Sanders (ed.). The SPIRE Handbook - Better, Faster, Cheaper: Software Development in Small Organisations, Dublin: Centre for Software Engineering, Ireland, 1998.

[15] SPICE Project. Final Report of Phase 2 of the SPICE trials. http://www.cis.strath.ac.uk/research/papers/SPICE/p2v2rp100.pdf

[16] Varkoi, T.: Management of Continuous Software Process Improvement, Proceedings of IEMC 2002, Cambridge, United Kingdom, 2002.

[17] Varkoi, T., Mäkinen, T.: Software Process Improvement Network in the Satakunta Region - SataSPIN, Proceedings of the EuroSPI´99, Pori, Finland, 1999.

6 Author CVs

Christiane Gresse von Wangenheim

Christiane Gresse von Wangenheim is professor at the UNIVALI/Brazil. Her interests are in the area of software process improvement focusing on small companies. Previously, she worked at the Fraunhofer Institute for Experimental Software Engineering in the area of software measurement. She received her PhD in Production Engineering at the Federal University of Santa Catarina (Brazil) and a PhD in Computer Science at the University of Kaiserslautern (Germany).

Timo Varkoi

Timo Varkoi is currently working on his dissertation as a research project manager at Centre of Software Expertise (CoSE) in Tampere University of Technology, Pori. His working experience includes both software development and management responsibilities in industrial software organizations. He is a competent SPICE assessor and the president of Finnish Software Measurement Association (FiSMA). His current interests include software process assessment and improvement related research, expertise and training combined with development of software intensive organizations.

Clênio F. Salviano

Clênio F. Salviano is a researcher at "Renato Archer" Research Center (CenPRA) in Brazil. His interests include software process capability models, process improvement, process assessment and software patterns. He holds a master degree in Computer Science from the Federal University of Minas Gerais and is a PhD student at Campinas State University (UNICAMP).

Software developer motivation in a high maturity company: a case study

Nathan Baddoo, Tracy Hall, Dorota Jagielska
Systems and Software Research Group
School of Computer Science
University of Hertfordshire
Hatfield, UK

Abstract

In this paper we discuss the impact of software developer motivation on projects. Motivation has been reported to be an important determinant of productivity and quality of work in many industries. In this paper we explore specifically how motivation impacts on development work in software engineering. We describe work previously done to suggest that software developers may have a different profile of motivators to other professionals. We present data collected from 9 developers working in a software organisation that has been assessed at CMM level 5. We find that the developers working in this high maturity development environment are highly motivated. We also report relationships between motivation and progress in development work and describe the impact of motivating factors on specific aspects of software development. Our main conclusion is that good software developers are proactive, flexible and adaptable, prepared to share knowledge with team and follow good practice by, for example, documenting work. Also, and in particular reference to this high maturity company, good software developers are able to resolve complex problems, innovative and eager to try new technology. The biggest motivators to such performance in a high maturity organisation are pay and benefits, recognition and opportunities for achievement. These are closely followed by technically challenging work, job security and senior management support. Cost, time, product quality – in terms of reliability – and user satisfaction are all moderately good indicators of project success. User satisfaction, however, is the best indicator of all four as its importance assumes a temporal significance as a project progresses. Finally, we found that technical competency, interpersonal skills and adherence to good practices impact very favourably on software project success.

Keywords: Motivation, Performance, Project Outcomes, Software Developer

1 Introduction

In this paper, we present results from a study of a high maturity software development operation in the UK. In the study, we attempt to synthesise the relationship between the impact of software developer motivation on good software performance and the resultant impact of good developer performance on software project success.

We argue that despite the overwhelming evidence of the impact of human factors on software project success [Hall and Wilson 1997, Wilson and Hall 1998, Willson et al 2000] , the software industry tends to favour a techno-centric approach to improvement [McDermit and Bennett 1999]. This is manifest by the variety of new development methods, tools and techniques that continue to fill the literature. We suggest that many of these new technical developments fail to make it into industrial practice and this has some relation to why the software industry has been slow to improve its overall performance. In this paper we suggest that improved understanding of the dynamic and multifaceted human factors within the software development process will not only improve managers' capability to design and implement effective development processes, but will also allow new methods and tools to be deployed more successfully. The result will be a software industry with a more mature understanding of the human variables within the development process. One such variable is the motivators to developer performance. We argue that managing these human variables can enable companies to move towards achieving significantly improved development process performance.

Motivators of performance are some of the most important human factors reported in the literature. Studies have shown that motivators, in addition to skills and time, have been important factors in deploying defined software processes [Kaltio and Kinnula 2000]. Indeed, our own studies show that developer motivation is related to successful process improvement [Baddoo and Hall 2002, Baddoo and Hall 2003]. In these studies, we were able to identify specific motivators that led to success in process improvement. We showed that the key motivators of software practitioners are visible support and commitment from senior management and empowerment of practitioners, whereas the key de-motivators were related to constraints on resources and a failure to secure practitioners' buy-in for process initiatives [Baddoo 2001]. We did not, however, specify how these factors directly influenced developer performance and how such performance, in turn, impacted on project success. So in this study, we return to the field to trace the direct relationship between factors that affect individual developer performance and how such performance impacts on project success. To do this, we must also ascertain developers' perception of good performance and good software project. Overall in this study we are attempting to answer the following research questions:

RQ1: What are the characteristics of a good software developer?
RQ2: What motivates good performance in software developers?
RQ3: What are the characteristics of a good software project?
RQ3: How do good software developers impact on the outcome of software projects?

We report on a study that is part of a larger longitudinal study of a CMM level 5 project, which involves analyzing a combination of quantitative and qualitative data. The overall design of our data collection strategy allows us to collect data tracking both technical and human project factors. However, specifically in this study, we collect and analyse qualitative data on human factors and their impact on project success. We use a high maturity company so that we can proper isolate the human factors that influence software development. We assert that high maturity companies would have better technical competence. Therefore, conducting a study like this in a high maturity company can provide a better indicator of what human factors best impact project success. We assert that this is less likely where the company is of a lower maturity and therefore less able to provide evidence of project successes.

The rest of the paper is structured as follows: In Section Two, we provide background to this research. We discuss concepts that underpin the factors that we have explored in the study. In Section Three we present the research methodology. We discuss how the design of the study enables us meet the research objectives and also discuss the data collection and analysis methods used. In Section Four we present analyses of our research results. In Section Five we synthesise the research results in light of the research questions. We conclude this paper in Section Six.

2 Background

2.1 Motivating performance.

In this study we are interested in how good performance is motivated. Therefore we aim to identify factors that underpin good performance. Studies on motivators have tested different motivation theories on groups of software practitioners in a variety of environments. A review of these studies indicates that factors that motivate software developers' performance vary only marginally with respect to social or political environments. However, significant differences exist between what motivates software developers and what motivates other practitioner groups. Many studies done on software developers' motivators show that software developers have a high need for growth and therefore are only truly motivated by factors that are intrinsic to the job that they do [Fitz-Enz, 1978; Mata Toledo and Unger, 1985; Couger, 1988]. This is evidenced by the fact that software developers are highly motivated by the nature of the job itself. Other intrinsic factors that motivate software developers highly are opportunity for advancement and growth [Couger and O'Callaghan, 1994], recognition [Warden and Nicholson, 1995], increased responsibility [Couger and Adelsberger, 1988] senior management support [Mellis, 1998, Willis et al., 1998, Diaz and Sligo 1997, Ahuja 1999 and Pitterman 2000].

2.2 Developer Performance

We suggest that a skilled team is a base for projects' success. In fact, Boehm [1988] found out that there are great differences in productivity and error rates between the most and least productive software developers.

In this work we suggest that if we are able to identify which skills/professional capabilities have the biggest influence on performance, then these can be transferred into software development practice to increase the rate of successes on projects. We suggest that a good way of identifying these skills and professional capabilities is through studying the performance of the most skilled developers. This approach employed by Turley and Bieman [1995] showed that exceptionally well performing software engineers considered themselves to be mòre proactive with management, more willing to exhibit and articulate strong convictions, more able to maintain a "big picture", better at mastering skills and techniques and better at helping others than non-exceptionally performing engineers.

2.3 Project Outcome

The most widely cited general definition of project success is a project completed on time, within budget, and meeting customer requirements [Jones, 1995, Baccarini, 1999 and Linberg, 1999]. In this study, we are planning to verify if practitioners' perception of project success corresponds with this wide definition. According to Procaccino et al [2005], factors that contribute to practitioners' perception of project success are at the same time aspects of their job that they most value and take the most pleasure in. In this work, we are trying to identify the specifics of these factors so that they can be included in management's motivating strategy.

3 Methodology

In this section we provide an overview of the project in the study, explain the research strategy- based around the concepts of motivation developer performance and project outcomes - as explored in Section Two and expand on the data collection and analysis methods used

3.1 The project

The findings presented in this paper are from an intensive one-week case study conducted in July 2004. The focus of the case study is a large complex embedded software development project (LEDS), based in a large UK engineering company.

The company considers LEDS to be a prestigious project with a high profile. LEDS has been set up by managers to showcase the high quality work of the company. The project is composed of several disciplines including a hardware team, a requirements modelling team and a software development team. The project is managed by a dedicated project management team.

All members of LEDS' software development team participated in this study as did representatives from the other disciplines in the project. The software development team were members of the company's Software Department. The Department was assessed at CMM level 5 in 2004. The processes used by the Department were, therefore, of high maturity.

The LEDS software team were considered the *crème de la crème* developers. They had been hand picked by the Department's managers to ensure a high quality outcome to this high profile project.

3.2 Research strategy

In this study we pursue the following research strategy:

MOTIVATION → PERFORMANCE → PROJECT OUTCOMES

Figure 1. Inputs and outputs of developer performance in a high performance team

Figure 1 shows the three elements of this research and encapsulates what we are trying to cover in this study, which is:

- what motivates software developer performance?
- what is good software developer performance?

- how does good software developer performance impact on software project outcomes?
- what are the indicators of software project success?

Clearly there are other elements of performance that could be explored (for example team dynamics) however these are outside the scope of this study.

During the study we spent one week on site and administered a questionnaire, see http://homepages.feis.herts.ac.uk/~comqnb/Index.html , to 9 team members. We did this in addition to conducting other studies as part of the overall larger scale longitudinal study. Questions in the questionnaire were informed by literature accounts of motivation, performance and project outcomes as discussed in Section 2. These questions were also a representation of some of the responses offered by software developers in earlier one-to-one interviews and focus group discussions that were held as part of the larger scale longitudinal study. By constructing the questionnaire study in this manner we are not only able to validate findings from other data collection exercises, but are also able to confer importance to some issues, dismiss other issues of minor importance and overall identify a fuller range of issues that warrant consideration.

3.3 Questionnaire data collection

Data collection methods significantly influence data analysis processes that can be used in research. As a result, selection of the data collection process needs to be carefully considered since its impact on the rest of the research process is significant. In this particular study we used questionnaires because they are best suited for the nature and type of data that we wished to analyse.

Questionnaires are a more convenient method of collecting primary data than most survey methods. Well-designed questionnaires collect less biased data because the respondent is not influenced by the attitude or opinion of an interviewer or *vice versa* [Berdie and Anderson, 1974]. Also, administering questionnaires overcomes the inherent problems in replication because all respondents receive the same set of questions.

We use questionnaires to collect developers' perceptions of good software development, motivators to software performance, project success and the impact of good software performance on project success. Questionnaire responses were measured on a Likert scale [Likert, 1932] of 1 to 7, where 1 indicates strong disagreement with a statement and 7 strong agreement.

3.4 Questionnaire analysis

One of the first ways of organising raw data is to group scores or values into frequencies [Black, 1999]. Frequency analyses are useful for reporting descriptive information from research. Frequency tables are used to report numbers of occurrence of each data variable. These frequencies can then be presented either in tallies or in percentages. Frequencies are useful for comparing and contrasting within groups of variables or across groups of variables.

Frequency analyses can be used for both nominal/ordinal data and also numeric data. Frequency analyses can also be used to conduct elementary statistics on both subjective and objective data. We use frequencies to analyse the perception data collected from the questionnaires. The result of this analysis is presented in Section Four.

4 Results

In this section, we present analysis of the data collected from the questionnaire survey.

4.1 Traits of good developer performance

Table 1: Traits of good developer performance

Table 1: Traits of good developer performance

Good developers are:	In general	On my current project
Self-dependent	6	5
Innovative	6	6
Proactive	7	6
Driven by the work itself	5	5
Has high technical competence	6	5
Resolves complex problems	6	6
Has strong knowledge of problem domain	4.5	5.5
Flexible and adaptable	7	6
Eager to try new technology	5	6
Prepared to work long hours	4	5
Communicate well with stakeholders	6	5
Shares knowledge with the team	7	6
Adheres to process	5	6
Fully documents their work	7	5

Median on Likert scale 1- 7, N=9

Table 1 shows the results of what developers perceive as the traits of a good software developer. Developers generally agreed that all the traits in Table 1 are characteristics of a good developer. However, there are varying degrees of agreement with respect to particular characteristics. We discuss some of the interesting patterns of responses and their implication below:

Traits of good developers, generally

Developers in our study very strongly agreed that being *proactive*, flexible and *adaptable*, *fully documenting work* and *sharing knowledge with the team* are the traits of a good developer. Developers also agreed, though less strongly, that *self-dependency*, *innovativeness*, *high technical competence*, *resolving complex problems* and *communicating well with stakeholders* are also traits of a good developer. The weakest agreement on traits of a good developer relate to developers who are *driven by the work itself*, *eager to try new technology* and *adhere to the process*. Respondents remained neutral with respect to whether having a *strong knowledge of problem domain* or *being prepared to work long hours* are traits of a good developer.

Traits of good developers on current project

Judging their own performance on the current project developers in our study considered themselves to satisfy all the traits presented the questionnaire as traits of a good developer. However, the strength of agreement differs with respect to particular traits, as discussed below.

Developers strongly agree that people working on their current project are *innovative*, *proactive*, able to *resolve complex problems*, *eager to try new technology*, that they *share knowledge with team* and *adhere to process*. However, they agree less with the rest of the traits in relation to their current project team. These are being *self-dependent*, *driven by the work itself* and *prepared to work long hours*, having *high technical competence*, *strong knowledge of problem domain*, *communicating well with stakeholders* and *fully documenting their work.*

4.2 Motivators of good developer performance

Table 2: Motivators of developers' performance

Performance is motivated by...	Generally	My performance
Technically challenging work	5	6
Opportunities for achievement	6	7
Opportunities for promotion	5	5
Pay and benefits	7	7
Recognition	7	7
Increased responsibility	5	5
Technical support / supervision	5	5
Job security	5	6
Work conditions	6	5
Senior management support	5	6
Company policy	5	5
Autonomy	5	5
Sense of ownership	5	5

Median on Likert scale 1-7, N=9

Selected factors

All factors selected by us as possible motivators were recognised as such by interviewed practitioners. However the degree of agreement differed with respect to particular factors and also with respect to general and own motivation. We discuss the difference below:

The strongest motivators

Pay and benefits and *recognition* were prescribed the highest motivational power by our respondents, both generally and considering their own performance. An equally important motivator of their own performance was *opportunity for achievement*; however it was rated lower as a general motivator. This may suggest that our respondents do not seem to think that other developers are as highly motivated by this factor as they are.

Factors practitioners considered more motivating to them than others developers

Technically challenging work, opportunity for achievement, job security and *senior management support* were all considered quite strong motivators of developers' performance in general. However, developers in this study regarded themselves more motivated by these factors than developers generally.

Factors less motivating to our respondents than to developers in general

Developers perceived *work conditions* as generally quite important for motivating developers' performance, although at the same time they declared this factor to be less important in motivating their performance.

Factors with the lowest motivating impact

Company policy, autonomy, sense of ownership, technical support/supervision, increased responsibility were perceived by developers in this study as having the lowest motivating influence from the listed provided. There was no difference in our respondents' perception of how these factors motivated them and how they could motivate other developers, generally.

4.3 Indicators of project performance

Table 3: Indicators of project performance

In general… is a good indicator of how well the project is performing	During Development	At the end of a project	1 year after project completion
Cost	4	5	4
Time	5	5	5
Fault rate	5	5	5
User satisfaction	5	6	7

Median on Likert scale 1-7, N=9

Developers agree that all factors provided were good indicators of project performance, at some stage of the project. Below we discuss different levels of this agreement in respect to particular indicators.

During development
Developers agree that *time*, *fault rate* and *user satisfaction* are moderately good indicators of the projects performance during this stage. At this stage, however, developers considered *cost* as being neutral to indicating how the project is performing.

At the end of the project
Developers strongly agreed that at the time of a project's completion *user satisfaction* is a good indicator of that project's performance. The agreement was slightly less when it came to judging *cost*, *time* and *fault rate* at the same stage of project's life cycle.

1 year after project completion
The biggest variance in degrees of agreement on what is a good indicator of project performance occurred when developers judged usability of chosen factors to estimate the project's performance a year after its completion. Developers strongly agreed that *user satisfaction* is a good indicator of project performance. They also agreed, although less strongly, that time and fault rate is good indicators of project performance. Cost, during the project's development, was perceived as a neutral indicator to project success.

Cost
According to participants of our study, *cost* is the least reliable indicator of project's performance. Developers remained neutral when judging *cost* as a project indicator during project development and a year after project completion. However, they agreed that it might be a good indicator at the end of the project.

Time and fault rate
Developers agreed that in general, *time* and *fault rate* are quite good indicators of how well the project is performing, at all investigated stages of project's life cycle.

User satisfaction
The strength of agreement on *user satisfaction* appears to increase with each stage of project life. Developers agreed that this factor is a good indicator of project performance during development, strongly agreed that it is a good indicator of it at the end of work on the project, finally, very strongly agreed that it is a good indicator of project performance a year after project completion.

4.4 Impact of developers' traits on project performance

Table 4. Impact of developers' traits on project performance

Developers traits	Reduce costs	Reduce effort	Reduce faults	Improve customer satisfaction
Self-dependent	5	4	4	4
Innovative	5	4	4	5
Proactive	6	5	6	6
Driven by the work itself	5	4	4.5	4
Has high technical competence	6	6	6	6
Resolves complex problems	5	6	6	6
Has strong knowledge of problem domain	6	6	5	6
Flexible and adaptable	5.5	5	5	5.5
Eager to try new technology	4	4	4	4
Prepared to work long hours	4	3	2	3
Communicate well with stakeholders	5	5	5	6.5
Shares knowledge with the team	6	6	6	6
Adheres to process	4	5	6	5
Fully documents their work	6	6	6	6

Median on Likert scale 1-7, N=9

Table four shows perceived impact of developers' traits on project performance. In most cases, this impact was judged as positive, only rarely neutral and just once negative. We discuss the results below:

Impact of developers' traits on reducing cost
Participants of our study strongly agreed that being *proactive*, having *high technical competence*, *strong knowledge of problem domain*, *sharing knowledge with the team* and *fully documenting work* have the ability to reduce cost of the project. This agreement was slightly weaker for traits such as: *self-dependency*, *ability to resolve complex problems* and *communicating well with stakeholders*, being *driven by the work itself*, being *flexible and adaptable* and *innovative.* Influence of *eagerness to try new technology*, *adhering to the process* and being *prepared to work long hour*s was judged as neutral to reducing cost of the project.

Impact of developers' traits on reducing effort
Developers strongly agreed that traits such as *high technical competence*, *resolving complex problems*, having *strong knowledge of problem domain*, *sharing knowledge with the team* and *fully documenting wor*k reduce effort of working on the project. Being *proactive*, *flexible and adaptable*, *adhering to the process* and *communicating well with the stakeholders* was considered slightly less important in this area. Impact of *eagerness to try new technology*, *self-dependence*, being *driven by the work itself* and being *innovative* on reducing effort was judged as neutral. However, quite understandably, participants disagreed that being *prepared to work long hours* reduces effort on the project.

Impact of developers' traits on reducing faults
Developers strongly agree, that developers that are *proactive*, have *high technical competence*, *resolve complex problems*, *share knowledge with the team*, *adhere to the process* and *fully document their work* help to reduce faults in developed project. Also, though slightly less so, are developers who have *strong knowledge of problem domain*, are *flexible and adaptable*, *communicate well with stakeholders* and are *driven by the work itself.* Practitioners neither agreed nor disagreed that being *self-dependent*, *eager to try new technology* and *innovative* reduces faults of

the project. They also strongly disagreed that being *prepared to work long hours* could reduce faults of the developed project.

Developers' traits and improving customer satisfaction

Being *proactive*, having *high technical competence* and *strong knowledge of problem domain*, *resolving complex problems*, *communicating well with stakeholders* and *fully documenting work* were strongly regarded as having positive influence on improving customers satisfaction. With a slightly smaller confidence the same ability was prescribed to *adhering to the process*, being *flexible and adaptable* and being *innovative.* Developers were neutral in judging the influence of *self-dependency*, being *driven by the work itself* and *eager to try new technology* on improving customer satisfaction. Finally, again they rejected the possibility that being *prepared to work long hours* can have a beneficial influence on project performance.

The most important traits for all investigated aspects of performance

Developers strongly agree that characteristics such as *high technical competence*, *sharing knowledge with the team*, *fully documenting work*, *strong knowledge of problem domain*, *resolving complex problems* and being *proactive* reduce time, cost, effort and improve customer satisfaction. Although ability to *communicate well with stakeholders* is not in this group it seems interesting to mention that the agreement between participants of our study that this trait can improve customer satisfaction was the strongest among the set of scores for this question.

Developers' traits with the lowest importance to project performance

Developers remained neutral in their estimations of the impact on the project that such characteristics as being *self-dependent*, *driven by the wok itself* or *eager to try new technology* might have. With the exception of a minor perception that *self-dependency* and *being driven by the work itself* can reduce costs, these traits are, according to our respondents, of no significance to project performance. *Being prepared to work long hours* was the only trait in the list that was considered to have a negative influence on project performance, namely on reducing effort, faults and improving customer satisfaction.

Most diverse estimates of importance

Adherence to the process was perceived to have the most diverse influence on project performance among all the investigated traits. It was perceived neutral to reducing costs, reasonably important for reducing effort and improving customer satisfaction, finally very important for reducing faults of the product. This variation suggests a broad knowledge of advantages and disadvantages of process application, which might be a result of some sort of training related to maturity level.

5 Discussion

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

5.1 What are the characteristics of a good software developer?

Good software developers are seen as proactive, flexible and adaptable, prepared to share knowledge with team and ones who follow good practice by documenting work. In this particular high maturity company, a good software developer was one who was able to resolve complex problems, innovative and eager to try new technology.

5.2 What motivates good performance in software developers?

Most of the traditional motivators of software developers, i.e. the intrinsic factors, are confirmed in this study. So that it is not surprising to find that opportunities for achievement, technically challenging work and recognition are reported as strong motivators of developer performance. However, what is surprising is the predominance of some extrinsic factors like pay and benefits and job security, whilst some core intrinsic factors like autonomy, sense of ownership and increased responsibility are underplayed. Equally surprising is the finding that suggests that software developers on this project are not sufficiently driven by the work itself.

Also, the less than high rating for *self-dependency* could indicate collectivism. It may suggest that this particular group of respondents are team-orientated, and as such this is reflected in the strong emphasis the put on sharing knowledge. It may also indicate a fundamental characteristic of a high maturity company. We infer from this finding that software developers' need for social interaction, at least within the work context, may be changing as software developers increasing understand the need to work with others.

We suggest that these accounts may represent a small shift in the perception of software developers as to what motivates them most, but also may hint at differing motivation profiles of software developers at a time when the software industry is under some strain to prove itself. Needless to say, these are interesting observations that open up opportunities for further studies.

5.3 What are the characteristics of a good software project?

The traditional measures of cost, time and quality are guardedly endorsed as indicators of project success. This is a surprising finding because we expected an overwhelming endorsement of these factors as indicated by the literature. However, it may appear that in a high maturity organisation, these factors play a less important role in determining project success than previously assumed. User satisfaction, however, takes on a significant role as a measure of project success. Its importance as an indicator of project success, post project completion, is absolute. This particular finding is vital to our research because it indicates that there are factors that could be considered relevant indicators to project success that have not been properly explored, yet. In the wider scheme of our overall research, we are turning our attention to addressing some of these factors.

5.4 How do good software developers impact on the outcome of software projects?

High technical competence, strong knowledge of problem domain and ability to resolve complex problems impact positively on all four indicators of project success. With the exception of strong knowledge of problem area, these characteristics are also highly rated as traits of a good software developer.

Other personal characteristics like being proactive and sharing knowledge with team, impacts favourably across all four indicators of project performance. Developers indicate that the ability to communicate well with developers significantly improves customer satisfaction. Also, adherence to good processes, especially in the form of documenting work, is perceived to impact positively across all four indicators, although a *carte blanche* adherence, per se, is only judged to have a strong positive impact on fault reduction.

Overall, with the exception of strong knowledge of problem area, the above characteristics are also highly rated as traits of a good software developer. Which strongly leads one to conclude that generally, the traits of good software developers as presented by developers in this maturity company are also the straits that impact positively on software project outcomes.

We are aware that because the findings presented above have been surmised from data collected from 9 developers questions may arise concerning the validity of this data. However, we would like to emphasize that this study forms part of a larger study of this particular high maturity organization and most of the findings presented here have been supported by findings from supplementary studies carried out using focus group and one-to-one interviews.

6 Conclusions

In general, good software developers are seen as proactive, flexible and adaptable, prepared to share knowledge with team and follow good practice by documenting work. In this particular high maturity company, a good software developer was one who was able to resolve complex problems, innovative and eager to try new technology.

The biggest motivators to good performance in a high maturity organisation are pay and benefits, recognition and opportunities for achievement. These are closely followed technically challenging work, job security and senior management support. The role of pay and benefits as a strong motivator for good performance could be a reflection of the current economic environment and the state of the computer industry as a whole. This observation is refreshingly new and requires further investigation, as hygiene factors like pay and benefits have never really featured significantly in software developers' motivators.

Cost, time, product quality – in terms of reliability – and user satisfaction are all moderately good indicators of project success. User satisfaction, however, is a best indicator of all four as its importance assumes a temporal significance as a project progresses.

Technical competency, interpersonal skills and adherence to good practices impacts very favourably on software project success.

Finally, in light of the current trend towards agile software development, it would be useful to compare findings from this study to a similar study carried out in an agile environment. This forms the basis of a current proposal for which we are in the process of eliciting funding

Acknowledgements

We are sincerely grateful to the company and practitioners in this study who, for reasons of confidentiality, must remain anonymous.

References

Ahuja S, (1999). Process Improvement In A Rapidly Changing Business And Technical Environment. *Fourth Annual European Process Group Conference*. Amsterdam, Holland. c303.

Baddoo N (2001) Motivators And De-Motivators In Software Process Improvement: An Empirical Study, in *Faculty Of Engineering And Information Sciences* p 259, University Of Hertfordshire, Hatfield.

Baddoo N and Hall T (2002) Motivators Of Software Process Improvement: An Analysis Of Practitioners' Views. *Journal Of Systems & Software* **62**(2):85-96.

Baddoo N and Hall T (2003) De-motivators Of Software Process Improvement: An Analysis Of Practitioners' Views. *Journal of Systems & Software* 66(1):23-33

Baccarini, D. (1999) The Logical Framework Method for Defining Project Success. *Project Management Journal*, December, pp. 25-32.

Berdie DR and Anderson JF (1974) *Questionnaires: Design And Use*. The Scarecrow Press, Metuchen.

Black TR (1999) *Doing Quantitaive Research In The Social Sciences: An Integrated Approach To Research Design, Measurement And Statistics*. Sage.

Boehm, B., Papaccio P. (1988) Understanding and Controlling Software Costs, *IEEE Trans. Software Engineering*, 14 (10), 1462-1477

Couger JD (1988) Motivators And Demotivators In The IS Environment. *Journal Of Systems Management* 39(6):36-41.

Couger JD Adelsberger H (1988) Comparing Motivation of Programmers and Analysts in Different Socio/Political Environments: Austria Compared To The United States. *Computer-Personnel* 11 (4):13-16.

Couger JD, O'Callaghan R (1994) Comparing The Motivators Of Spanish And Finish Computer Personnel With Those Of The United States, European Journal of Information Systems 3 (4), p.258-291

Diaz M and Sligo J (1997) How Software Process Improvement Helped Motorola. *IEEE Software* September/October:75-81.

Fitz-Enz J (1978) Who Is The DP Professional? *Datamation* September:125-128.

Hall T, Wilson D (1997) "Views of software quality: a field report" *IEE Procs on Software Engineering*, April, pp111-118

Jones, C. (1995) Patterns of large software systems: failure and success, *IEEE Computer*, 28(3), p 86-87

Kaltio T, Kinnula A (2000) Deploying The Defined Software Process. *Software Process: Improvement and Practice* 5:65-83.

Likert R (1932) A Technique For The Measurement Of Attitudes. *Archives Of Psychology* **22**(140)

Linberg, K. (1999) Software developer perceptions about software project failure: a case study, The *Journal of Systems and Software*, 49, p 177-192

Mata Toledo RA and Unger EA (1985) Another Look At Motivating Data Processing Professionals. *Computer-Personnel* **10**(1):1-7.

McDermit JA and Bennett KH (1999) Software Engineering Research: A Critical Appraisal. *IEE Procs On Software Engineering* **146**(4):179-186.

Mellis W (1998) Software Quality Managment In Turbulent Times - Are There Alternatives To Process Oriented Software Quality Management? *Software Quality Journal* **7**:277-295.

Pitterman B (2000) Telcordia Technologies: The Journey To High Maturity. *IEEE Software* **17**(4):89-96.

Procaccino J. D., Verner J. M., Shelfer K. M., Gefen D. What do software practitioners really think about project success: an exploratory study, *Journal of Systems and Software, Article in Press, Corrected Proof,* Available online 13 January 2005

Turley R. T., Bieman J. M. (1995) Competencies of Exceptional and Nonexceptional Software Engineers, *Journal of Systems and Software, 28 (2)*

Warden R, Nicholson I (1995) IT Quality Initiatives At Risk. *Software Quality Management* New Year (24):24-27.

Willis RR, Rova RM, Scott MD, Johnson MI, Ryskowski JF, Moon JA, Shumate KC and Winfield TO (1998) Hughes Aircraft's Widespread Deployment OF A Continuously Improving Software Process, in, Software Engineering Institute, Canergie Mellon University.

Wilson D, Hall T (1998) "Perceptions of software quality: a pilot study" *Software Quality Journal*, Mar, vol 7(1), pp67-75

Wilson D, Hall T and Baddoo N (2000) The Software Process Improvement Paradox. *Approaches To Quality Management: Software Quality Management VIII*:97-101.

Knowledge Management in Distributed Environment

Darja Šmite, Uldis Sukovskis
Riga Information Technology Institute, Latvia
Darja.Smite@riti.lv, Uldus.Sukovskis@riti.lv

Abstract

Global market expansion creates new opportunities for cutting costs, gaining extra knowledge and resources, and speeding product time-to-market. Nevertheless, global software development projects become the source of hidden risks and new challenges. This paper offers an overview of a research on software development, involving multiple geographically distributed parties. The paper focuses on a problem of Knowledge Management in an environment, where the question of knowledge distribution between the remote partners becomes a challenge. The proposed distributed project framework describes the Knowledge Management as a component of one of the software development project process areas and characterizes it considering global appearance.

Keywords

Global software development, distributed environment, knowledge management

1 Introduction

In the era of global marketplace distributed software development has already became a well-founded trend for companies who are searching to cut costs, gain extra knowledge and resources and speed product time-to-market. The new trend has numerous names, in particular, software development outsourcing, global software development, multi-site development, software development in distributed environment. All these concepts consider software development by geographically distributed teams.

While global software development seems to be a significant way to gain benefits, it also raises new challenges and risks, by this opening a wide area for research. Jae-Nam Lee notes that information technology outsourcing has long played an important role in the field, yet outsourcing trends are little understood [3]. People and process geographical distribution puts new demands on project coordination and management approaches due to global risks, such as troubled communication, lack of trust and teamness, lack of personal contact, time zone differences, cultural and organizational differences, etc.. To overcome the problem of distance in GSD, various managers are experimenting and quickly adjusting their tactical approaches [1]. Despite its popularity, no research could determine the exact recipe for effective outsourcing performance [4]. According to Jae-Nam Lee, strategic view focuses on competitive advantage, without considering how relationships between an organization and its external environment are managed [3].

Considering global distribution and lack of experience in distributed environment, the authors emphasize the importance of knowledge management across the borders. While companies get involved in global projects and then separate after the product is delivered, knowledge and experience have to be somehow accumulated for further projects. Where and how to do it? What are the components of knowledge management in distributed environment? These are the question to be answered.

The background of this research is a survey on global software development process improvement in a software development company in Latvia. The research focuses on the software development project peculiarities brought by distance and aims to develop a framework for better performance.

The paper is organized as follows. The next chapter gives an overview of the entire research on distributed software development, including the framework for global software development. Then the concept of knowledge management in a distributed environment is described. The following chapter describes a case study of knowledge management system implementation in a distributed project involving partners from Latvia and the UK. Finally, conclusions and further research plans are given.

2 Research Overview

2.1 Research approach

The aim of the research is to develop a framework for global projects, which could identify specific risks of distributed software development and accumulate practices to address these risks. The authors work in an R&D institute for the biggest SD company in Latvia, which is participating in global software development market since early 90s. This company is used as the research case study.

The research can be divided into several steps.

1. Preliminary investigation – aims to observe the current situation in global project management in the company by means of documentation analysis and inquiry [6]. Finished in June 2004.

2. Global project risk analysis – aims to clarify major global risks, their magnitude and frequency of occurrence by interviewing experienced project managers [7]. Actual since October 2004.

3. Framework development – aims to build a framework for distributed project management in order to improve software processes addressing global specifics. Actual since April 2005.

The question of knowledge management in distributed environment discussed in this paper is explored by means of interviews with the software development collaboration stakeholders from the supplier company. The next chapter describes the model for distributed projects in detail.

2.2 Building a Framework for Distributed Projects

The main objective of the research is to organize all specific practices for distributed software development in a special framework. The following figure shows the basic structure of the framework.

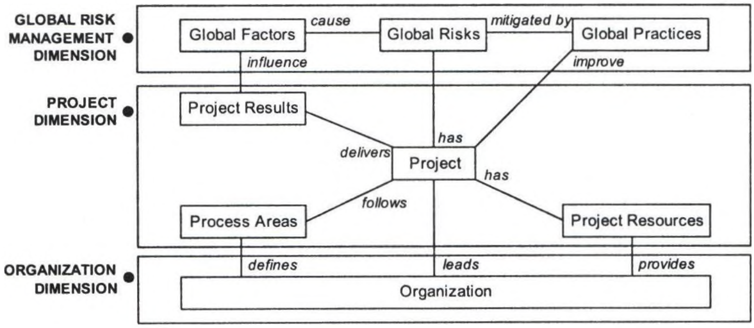


Figure 1: Framework for distributed software development projects

The framework offers three dimensions – organization dimension, project dimension, and global risk management dimension.

A distinguish element for distributed projects is that there are several organizations involved in a global project. Accordingly, process areas are defined and project resources are provided jointly by organizations, involved in the global project. The framework will specify guidelines for organizational changes in order to support global projects effectively.

Project dimension combines project resources provided by organization(s), process areas that are defined for the project and project results that are delivered during the project. The framework will use CMMI process areas as a basis for process observation and improvement. Influenced by global factors project management will use the framework for global risk identification and will be thereafter improved by using global practices. Noteworthy is the fact, that project improvements are meaningless without organizational change necessary for software development globalization.

The framework for distributed software development projects will be represented by a defined set of global risks that are caused by global factors and a set of practices that are derived from the research. The model aims to clarify how geographical distribution, cultural differences, time zone differences and other specific factors met in the global projects influence project and organization dimensions. Adequate practices can be derived by developing a set of specific factors and a set of risks caused by these factors.

$$\text{Global Factors} = \bigcup_{i=1}^{l} f_i \text{ ; Global Risks} = \bigcup_{i=1}^{m} r_i \text{ ; Global Practices} = \bigcup_{i=1}^{n} p_i \text{ ;}$$

One of the most important components, that help to provide effective software process improvement, is knowledge management. The authors propose to add this component to the Process Area component. The further chapters will describe Knowledge Management in Distributed Environment and emphasize the major challenges.

3 Knowledge Management in Distributed Environment

3.1 Why Knowledge Management Is Important?

According to GartnerGroup definition Knowledge Management is a discipline that promotes a collaborative and integrated approach to the creation, capture, organization, access and use of an enterprise's information assets. This includes databases, documents and, most importantly, the un-captured, tacit expertise and experience of individual workers [2]. India Infoline [5] derives the following value chain in order to emphasize the importance of knowledge share: DATA → INFORMATION → KNOWLEDGE → WISDOM → VALUE.

Ignoring the process of knowledge management, organization precludes effective software process improvement. Nevertheless, data gathering, information transformation into knowledge, providing organization with wisdom and value in global software development projects is troubled due to process and resource distribution. This establishes a very specific environment where knowledge management becomes a challenge.

3.2 Knowledge Management in Distributed Projects

Knowledge Management cannot be found in the list of CMMI Process Areas as an independent item. Nevertheless, Knowledge Management supports various project processes as Project Monitoring and

Control, Measurement and Analysis, Organizational Process Focus, Organizational Process Definition, Organizational Training, Risk Management, Decision Analysis and Resolution, Organizational Process Performance, Organizational Innovation and Deployment, Causal Analysis and Resolution (CMMI Process Areas [8]). It is a supportive process on the way of software process improvement and therefore has to be paid adequate attention.

Distributed project model describes Knowledge Management as one of Process Area components (see Figure 2).

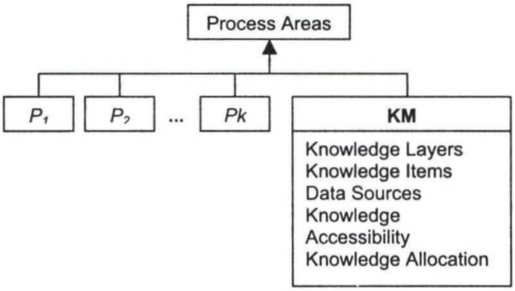


Figure 2: Knowledge Management as a Process Area component

Knowledge Management has several properties specific for software development projects in distributed environment (see Table 2).

Table 1: Knowledge Management Properties

KM Property	Description
Knowledge Layers	Describes the engaging entity and supplier entities' knowledge layers
Knowledge Items	Defines specific knowledge items and their adding value for each layer
Data Sources	Clarifies the allocation of each data source for knowledge accumulation
Knowledge Accessibility	Defines access level to knowledge items for each involved party
Knowledge Allocation	Describes how to share knowledge after the end of the project

These properties characterize knowledge management as a process of global project influenced by geographical distribution, and show the necessity of adequate definition for knowledge items between the involved parties.

3.3 Knowledge Layers in Distributed Environment

The way of developing software, involving geographically distributed teams, transforms the common way of things. Different teams are put together in a project for a short-term cooperation thereby churning different knowledge, experience and culture. When the project is finished, the overall team falls apart splitting the flow of experience. Hence, knowledge management in distributed environment becomes a challenge.

Considering geographical dispersion and increasing number of the development teams involved in global software development projects, the following two knowledge layers can be derived (see Fig. 3).

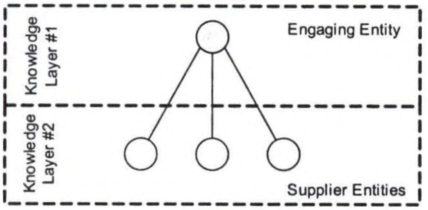


Figure 3: Knowledge Layers in Distributed Environment

The engaging entity in Knowledge Layer #1 is a company, who involves a geographically distributed supplier or suppliers in the software development project. This entity can represent an end customer or a mediating party. Accordingly, the first knowledge layer focuses on the practices used by an engaging entity and aims to answer the question "How to engage and monitor software development suppliers, and control their performance?"

The second knowledge layer accumulates practices used by outsourcing service providers or supplier entities, and aims to answer the question "How to provide software development services for a remote engaging entity and how to collaborate with other possible software development suppliers involved in the distributed project?"

In reality the distributed engagement can be increased by the complexity of each supplier entity. Suppliers are frequently searching for or extra resources, knowledge or ways of cutting cost elsewhere by engaging subcontractors. As a result these suppliers become engaging entities for those who work in the subsidiary. Hence knowledge layers become shared (see Figure 4).

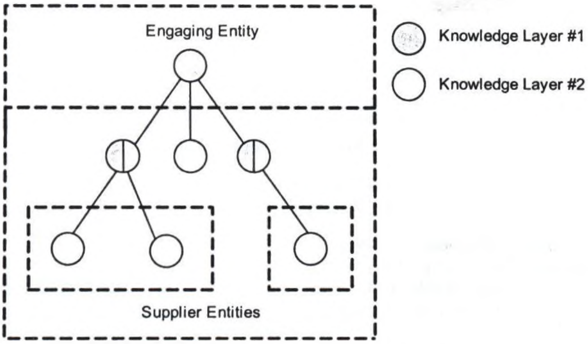


Figure 4: An example of a distributed engagement

Defining knowledge layers helps to structure knowledge assets, sources for data gathering, and necessary measurements.

3.4 Challenges in Knowledge Management in Distributed Projects

Due to geographical distance and multiple team distribution, knowledge management appears to be not a trivial thing. The authors recommend to pay careful attention for planning knowledge management in a distributed project by answering the following questions:

- What are the purposes of knowledge management in the distributed project?
- What are the parties interested in knowledge accumulation?

- Which are the knowledge assets to be accumulated?
- Which party is interested in each knowledge assets?
- How can these knowledge assets add value to each involved party?
- How can the necessary data be gathered across the borders?
- Who will have access to each knowledge asset?
- How will the knowledge be shared between the involved parties after the end of the project?

The answers help to evaluate each property of Knowledge Management component in distributed project model and settle the ground for further knowledge collection and share.

3.5 Knowledge Management Success Factors

This chapter describes an overview of Knowledge Management success factors, derived from a case study. SD Company mentioned above works with various customers and collaboration partners in the EU and Scandinavia, providing software development services. The Company participated in various middle and large distributed projects, in some cases having a geographically distributed mediating party and a customer, and in some cases involving additional subcontractors too.

Performing the interviews with project managers, the authors mapped Knowledge Management objectives on the Company's existing experience. Considering Knowledge Management System (KMS) implementation, the following list of success factors was derived (see Table 2).

Table 2: Project success factors

Success Factor	Description
KMS should integrate with existing business support applications	In order to successfully implement the Knowledge Management system, it should be integrated with the existing business support applications. Moreover, the KM system should follow determined guidelines in order to comply with the future business applications. E.g. i) asset management; ii) human resource management; iii) research; iv) product development, etc.
KMS should be built on a messaging infrastructure	Knowledge Management system might gain the superior usability in the company just in case it will be seamlessly integrated with the existing collaboration tools. Existing messaging infrastructure might be the best solution to start with.
KMS should support multiple partners types	In order to achieve the best results, the information should be exchangeable between the wide range of applications and partners. The major Knowledge management functions should be available via the web, hence being fully independent from the platform and operating system.
KMS should scale from team to enterprise	The functionality of the KM system should be the same for different sizes of the user groups. The architecture of the KMS should be designed in a manner, allowing to increase the amount of users up to several thousands or more. The open architectures and standards should be used as much as possible.
KMS should support disconnected use	By keeping in mind the busy people, being most of their time on the way, the off-site KM should be available too. Therefore, the KM system should have functions allowing the employees to use the accumulated knowledge even off-site.

KMS should provide security and roles	One of major issues considering KMS features in distributed environment is security. It is evident, that information should be accessible only by the persons authorised for certain requests considering knowledge layer division. Therefore, several role-based security models should be implemented.

4 Conclusions

Software development process and resource distribution among geographically dispersed participants brings changes in usual software development processes. Distributed projects differ from inland projects and have to be managed adequately. Considering new risks and challenges new practices for software development have to be adjusted.

One of the most important processes on the way to continuous software process improvement is knowledge management. It helps to accumulate project knowledge and turn it into valuable experience shared across the organization. Knowledge Management supports various processes defined by the Capability Maturity Model Integration as Project Monitoring and Control, Measurement and Analysis, Risk Management, Decision Analysis and Resolution, etc. Addressing distributed project specifics, Knowledge Management appears to be a great challenge.

The proposed model for effective distributed project performance improvement characterizes Knowledge Management in distributed environment as follows:

- Knowledge Management is divided between the involved parties and, therefore, has to be decomposed considering two major knowledge layers – engaging entity knowledge layer and supplier entity knowledge layer.
- The following properties shall be defined for Knowledge Management in distributed projects:
    - Knowledge layers for valuable knowledge breakdown between the project participants;
    - Knowledge assets for each project participating party;
    - Data sources and metrics to be accumulated;
    - Knowledge accessibility in order to prevent unauthorized access;
    - Knowledge allocation after the end of the project.
- It's essential to define added value for parties involved in the project, considering knowledge items;
- The following success factors need to be minded during Knowledge Management implementation:
    - Integration with existent business support applications;
    - Ensuring messaging infrastructure;
    - Multiple client type support;
    - Off-line support;
    - Providing security and roles.

The further research steps will aim to analyze other framework components in detail, describing the impact of multi-site development and geographical distribution on a software development project.

5 Acknowledgement

This research is partly supported by European Social Fund and the Latvian Council of Science project Nr. 02.2002 "Latvian Informatics Production Unit Support Program in the Area of Engineering, Computer Networks and Signal Processing".

6 Literature

1. Carmel, E., Agarwal, R. "Tactical Approaches for Alleviating Distance in Global Software Development", IEEE Software, March/April, 2001, pp.22-29
2. Harris, K. "GartnerGroup's Knowledge Management Glossary", Gartner Group, 1998
3. Lee, J., Huynh, M.Q., Kwok, C.W. and Pi S.M. "IT Outsourcing Evolution – Past, Present, and Future". Communications of the ACM, May 2003/Vol.46, No.5, pp. 84-89.
4. Loh, L. and Venkatraman, N. "An empirical study of information technology outsourcing: Benefits, risks, and performance implications". Proceedings of the 16th International Conference on Information Systems, Dec. 10-13, 1995, Amsterdam, the Netherlands, pp. 277-288
5. Varma N., Narayan S. "Managing Knowledge Perspectives and Dimensions", 10.08.2001, http://www.indiainfoline.com/bics/kmpr.html
6. Smite, D. "Global Software Development Project Management – Distance Overcoming. Software Process Improvement", The Proceedings of the 11th European Conference, EuroSPI, Trondheim, Norway, 2004
7. Smite, D. "A Case Study: Coordination Practices in Global Software Development", accepted for publication in the Proceedings of the Profes conferences, June, 2005.
8. CMMI-SE/SW/IPPD/SS, v1.1 Staged Representation, Carnegie Mellon Software Engineering Institute, 2002.

7 Author CVs

Darja Šmite

Darja is an IT consultant and auditor at the Riga Information Technology Institute, and an academic member of staff at the University of Latvia. She is also currently working towards her Ph.D. degree at the University of Latvia. Her major research interests are related to software development and software process improvement in distributed environment. Darja has 5 years of industrial experience as a Software Engineer, Systems Analyst and Project Manager. Therefore, her research has been taken in close relationship with practice.

Uldis Sukovskis

Uldis Sukovskis is IT consulting and audit director of the Riga Information Technology Institute. He has been working in the information technology industry for about 20 years in various positions, including software development project manager, head of system modelling research laboratory, outsourcing manager and IS auditor. He holds doctor degree in computer science and is a professor at the Riga Technical University. He is Certified Information Systems Auditor (CISA). Sukovskis' research interests focus on software development outsourcing, software engineering, system modelling and vocational IT education. He also has experience with information systems auditing, control and security procedures.

The Role of Knowledge Management Supporters in Software Development Companies

Péter Fehér[1]

[1] Corvinus University of Budapest, Deparment of Information Systems
1053 Budapest, Veres Pálné u. 36.
pfeher@informatika.bke.hu
http://informatika.bke.hu

Abstract. Knowledge as strategic resource and support factor has higher and higher importance in software development processes, and in software process improvement projects. Therefore the conscious management of this resource, and the efficient support of knowledge management processes are vital for organisations. In this paper the possible knowledge related processes and the use of the possible support factors are researched and analysed. As result of the research an integrated model of support factors has been developed.

1 Introduction

In the last years, software process improvement approaches were more and more popular and both managers and users recognise its necessity for success in software development organisations. These organisations require the development of the software processes, and there is a strong faith is the success of these projects.

The goal of software process improvement projects is the improvement of the results of processes (product and services), and increased efficiency of processes themselves (Bíró and Messnarz, 2000). Software process improvement is based on the TQM approach (Total Quality Management; summary: Sila and Ebrahimpour, 2002), that is the philosophy of continuous improvement and development. Among the improvement approaches, the most popular are the process-oriented quality based ISO9001, the CMM (Capability Maturity Model; Paulk et al, 1993) that evaluates process maturity, or Bootstrap (Messnarz et al, 1994; Kuvaja et al, 1994). The SPICE (Software Process Improvement and Capability dEtermination; Dorling, 1993) evaluation approach combines several methodologies since 1998 as ISO-15504 Standard. Process capability is the property of processes, by what these processes are able to reach previously defined goals (Siakas and Georgiadou, 1999). These models examine the controllability, manageability and processes, and service providing.

Another approach that has important role in software processes is knowledge management. Since the mid-1990s it became more and more popular among both practitioners and researchers. The first knowledge management projects explored the tools and possibilities of information technology, and due to the fast development of technological solutions, the idea and practice of knowledge management have spread quickly. However, knowledge management is not only a simple information management topic, but the researchers of other disciplines (general management, organisational theories, strategic management, human resources management, sociology, psychology, philosophy, etc.) are working in this area, and enriching this area with new and different ideas, approaches and interpretations. Therefore knowledge management is a frontier between different scientific areas.

In this paper, the role of knowledge management and knowledge processes are examined in software development companies, and in software improvement processes. The goal of this research is to explore this demanded and only partly explored area: What kind of tools, solutions are able to support knowledge management activities in order to increase efficiency? Further goal of this research is to identify the supportive factors of knowledge management, and to set the conditions of their usage.

2 Research methodology

The targeted research area of this research is the behaviour of software development companies. Software development requires knowledge that embeds in products (software), and knowledge that describe organisational processes. In this case the area of activity by itself justifies the requirement of conscious management of knowledge. The processes and tasks of this industry are knowledge-intensive, because of the following conditions (Starbuck, 1992; Sveiby, 1992; Apostolou-Mentzas, 1999):

- Organisational value-creating processes are not able to be fully standardised: although software development processes could be standardised at high level by methodologies (feasibility study, planning, physical/logical modelling, programming, testing, etc.), but in each task creativity and adaptation is required.
- Activities require high-level of group-work, cooperation is required, that are performed in projects;
- Employees are well-trained, creative. Quality of products and services are determined by the capabilities of employees.

In the research 72 cases – software development organisations – were examined. The source of the cases is the European Software Process Improvement database, in which the researched cases were identified. The research is focusing on knowledge management solutions in software development processes, that are based on the own reports of the organisations. The deep, detailed description provided the possibility of examining relationships, that could be ignored by using only statistical analysis. Through quantification of cases, statistical analyses were also possible. Because of the immature nature of the researched area, the research is explorative, therefore self-feedback of findings continuously refines the research questions.

3 Research model of KM supporters

Based on the literature review of existing holistic models of knowledge management, a research model has been developed, that is the collection of research expectations. By accepting the necessity of system thinking approach (Rubenstein-Montano et al, 2001), the model contains the role of organisational strategy in order to provide wider context of interpretation. In the organisational strategy – based on the knowledge-based theory of the firm – there is an important role of managing, handling of organisational knowledge, as strategic resource, that is the basis of competitive advantage (Grant, 1996). Sub-strategies of organisations (e.g. HRM strategy, IT strategy), that support organisational goals, can determine the conditions of knowledge management processes.

The connection between organisational strategy and knowledge management processes is provided by knowledge management strategy: it supports organisational goals, and define knowledge management processes and supporters. Knowledge management strategies are vital of successful knowledge management practices.

For successful competition, based on the knowledge assets of an organisation, the conscious management of this strategic resource is required. Knowledge management strategy of organisations defines the utilisation of knowledge required for supporting and accomplishing organisational strategy. It includes the goals of knowledge management together with the tools, methods and approaches to accomplish it.

Developing knowledge management practice of an organisation can be based on either the threats of external factors (increasing competition, better practice of competitors, etc.) that pressurise the behaviour of the organisation or internal initiatives that provides opportunities to be the best in the market, or to prepare proactively for future threats. Knowledge management strategy consists of either external factors or the internal possibilities of organisations (Zack, 2000). Nonaka and Toyama (2003) argue that a knowledge management strategy is the reflection for the internal possibilities and external position of an organisation.

Knowledge management processes can characterised by their purpose: *evaluating and mapping knowledge assets*, *knowledge leverage* (sharing, transfer and use) or *developing knowledge* (creating new knowledge and organisational learning).

Knowledge management strategies can be developed based on different approaches: existing knowledge of organisation can be exploited or new knowledge can be acquired and developed (von Krogh et al, 1994); personalisation or codification (Hansen et al, 1999); conservative or aggressive strategies (Zack, 1999). Independently of the used approach, knowledge management strategy of an organisation has to appear in organisational and technical architecture.

Technological solutions provide the possibility of effective management of codified (store, process, transfer) knowledge. Information and communication technologies (ICT) decrease the barriers of knowledge sharing and transfer. Although IT solutions (knowledge management support systems - KMSS) have a key role of supporting KM practices, management understanding of their possibilities and limits is also required. There are several possibilities to support KM processes: creating knowledge by data-mining systems; discovering knowledge by intelligent agents; supporting cooperation, coordination and communication; using knowledge repositories or applying expert systems (Lawton, 2001).

The several possible supportive factors of organisational solutions are widely discussed in the general management, and KM literature. Researches presented, that even each of the general management factors can have affect to KM practices: Organisational structure can be a barrier of effective use of KM efforts, therefore conscious reorganisation, new structures and new roles are required (Spender, 1996). Human resource management activities have to face the more important selection of valuable and appropriate employees, the different motivation factors of knowledge workers, and also the increased requirements of the management. The supportive culture can have the highest impact on KM, which is mostly based on trust, and heavily emphasised by leadership styles.

In order to measure the effectiveness of any practice or process, a well defined measurement system is required, with feedback possibilities (Ahmed et al, 1999). Measurement can include the monitoring of performance indicator, analysis of process effectiveness, questioning workers in the KM system.

The research model is used as a basis, as a basic expectation of useable knowledge management supporters. During research, this model was extended (Figure 1).

4 Research and findings

Role of knowledge in software development organisations

Among the formally determined goals of the researched organisations to improve their software development processes (that is highly depends on the knowledge of the organisations), clearly defined approach for knowledge management is rare, but is several cases in several cases, the general definition of the improvement activities (quality improvement, process development, keeping the deadlines, clients' satisfaction) covers development, introduction or execution of knowledge related activities. These projects attempt to achieve a higher level of some maturity model (CMM, Bootstrap), and the required changes were executed based on this approach.

By now, knowledge management is considered to be essential part of software process improvement models (Meehan and Richardson, 2002). Process improvement approaches emphasise well-defined and standardised use of processes, which requires the disposability of the required knowledge. In process improvement models, the demand for knowledge sharing is also important, in order the decrease the dependency on employees who are single owners of critical knowledge, therefore knowledge management have vital, either direct or indirect role in software process improvement projects (Meehan and Richardson, 2002).

Related to the management goals of knowledge, in the case of the researched organisations, six kind of activities can be distinguished (Table 1): Bigger part of these activities concentrates to knowledge transfer, and the remaining concentrates to creating new or developing existing knowledge. Organisations and projects attempted to achieve several knowledge related goals simultaneously, and neither of the knowledge related project goals can exclude other project goals. "Sharing and reusing existing experiences" and "Developing and creating knowledge about processes" correlates negatively (-0,358), although in 10% of the cases these two activities are performed simultaneously. Most popular project goals are sharing and reusing experience and process knowledge.

Knowledge activities	Rate of companies	Explanation, examples
Transferring and leveraging existing knowledge		
Sharing and reusing existing experience	65,3 %	Recording, processing of personal and organisational experience, arisen in any software development project, for future sharing and reuse. *Example: AMN company the AMN company stores and makes the employees' experiences obtained from the different projects available via an Internet-based portal, which is easily accessible and utilisable for the employees working in different geographical locations.*
Sharing and using of knowledge arisen in processes (Knowledge in processes)	40,3 %	Collection of information and knowledge required for tasks within one project instead of the general project experiences. *Example: At MED experiences are documented, which arise in the certain phases of the development activity to make them be available for the execution of the connecting tasks.*
Sharing knowledge about processes (Process knowledge)	31,9 %	Definition of standardised, best practice processes, giving access to these definitions and transferring this knowledge to the employees. *Example: In favour of making all employees work with the expected effectiveness, MOC company centred on that during the detail trainings everybody could acquire and utilise the processes. This knowledge helps to increase and maintain the effectiveness of the processes and the ensuring the quality of the products.*
Extracting and sharing customer knowledge	18,1 %	Extract, store and communicate information and knowledge about customers and customer requirements to the developers during the development process. *Example: In order to meet the demands of its customers better, the CCL company attempted to develop the communication with the clients and store the knowledge about the clients (expectations).*
Creating and developing knowledge		

Developing and creating knowledge about processes	26,4%	Deeper examination and exploration of actual software development process of an organisation, monitoring the process, and based on the results, determination of strengths and weaknesses, which could be the base of later process improvement approaches. *Example: the MNO company collects the data typifying its activity, and then analyses them by statistical instruments. The created knowledge typifying the processes helps in discovering the weak point of the processes and establishing the process improvement activity.*
Acquiring and integrating external knowledge	8,3%	Developing and expanding organisational knowledge by integration of external knowledge through employees, which can be other's experience, new solutions, technologies or process organisational knowledge. *Example: in favour of developing its activity the SOC company integrates the experiences provided by its partners as the member of a cooperation group.*

Table 1. Knowledge activities in change projects

Knowledge management strategies

The strategic approach, which is widely concerned in the knowledge management literature, is the question of choosing between the codification and personalisation strategies, i.e. the choice between technological and organisational solutions (Hansen et al, 1999).

In the case if software development organisations, knowledge and internal support of technological solutions can be considered as given: employees do not feel aversion to apply IT solutions, use of PC can be considered as evident and even managers do not feel aversion for development and use the technological solutions for internal purposes. It is a question of the research, whether these organisations are able to use non-technological solutions in order to provide the efficiency of their processes?

Various solutions of the codification approach were used in 78,3% of the organisation, and in further 15,9% of the companies applied technological solutions to supplement other approaches. Based on these results, most of the examined organisations attempt to codify their knowledge and support it by technological solutions, according to their field of activity. This approach emphasizes the object character of knowledge, and it attempts to manage explicit knowledge. In the process of knowledge storing, knowledge is detached from persons, and it becomes available for everybody. This approach facilitates the transfer and sharing knowledge via IT solutions.

The personalisation approach – which is the emphasised usage of organisational solutions –, is considerably wide-spread, although the rate of use of personalisation solutions does not achieve the level of codification approaches. At more than the half of the examined companies (50,7%) the usage of strong personalisation solutions can be found, and there are less intensive, less characteristic personalisation solutions in further 21,7% of the organisations. However, at more than a quarter of the organisations (27,5 %) there is no use of personalisation solutions at all. In the case of personalisation approaches, knowledge is shared through personal interactions, meetings, conversations, in which personal communication, expertise and experience have important role. In these cases, the subjective and personal character of knowledge is dominant, and during personal interaction tacit knowledge can be transferred.

The research of Hansen et al (1999) among consultancies – that can be considered as one of the most typical knowledge-intensive activities – concludes, that organisations must choose between codification and personalisation approaches, otherwise they endanger their efficiency and successfulness.

Based on the results of the present research, strong use of both personalisation and codification approach is typical at almost a third (30,77%) of the organisation that introduced knowledge management processes *successfully*. Furthermore almost two-thirds (65,38%) of the successful companies applied some sort of combined approach of personalisation and codification. Applying a successful pure strategy meant typically the codification strategy at almost a third of the successful organisations (30,77 %), while pure personalisation approach can be found only at 2 companies (3,85 %) (Table 2). However the rate of the organisations that use pure personalisation approach is also low (4,35%), and the rate of pure codification strategy is rather high (27,54%) in the whole sample.

Based on the results of this research, the use of pure personalisation of codification knowledge management approaches is not the precondition of successfulness. Furthermore, the compulsory option between the two knowledge management strategic approaches is not required for every knowledge-intensive organisation. This result is confirmed by the opinion of other researchers (Wiig, 1999; Adelmann and Jashapara, 2003), who believe, that the compulsory option is a uniqueness of consultancies. However another research suggests that consultancies moved from the pure strategies towards combined strategies, because of the problems of their traditional approaches (Fehér, 2004). Furthermore even the research experience of Truch and Bridger (2002) can not be justifiable, who conducted their research among non-consultancies, and concluded that the precondition of successfulness is the use of combined strategies.

Based on the results of this research, examined organisations typically attempt to use their existing, internal knowledge, thereby they accomplish conservative strategies. Development and enlargement of knowledge, even significant innovative activities are not priorised approaches, therefore accomplishing conservative strategies is not typical.

	Personalisation is not typical	**Weak personalisation**	**Strong personalisation**	*Totally*
Codification is not typical	0%	0%	3,85%	***3,85%***
Weak codification	0%	1,92%	15,38%	***17,31%***
Strong codification	30,77%	17,31%	30,77%	***78,85%***
Total	***30,77%***	***19,23%***	***50,00%***	***100,00%***

Table 2: Use of codification and personalization strategies in case of successful activity

Technological Issues

Technological supporting tools usually mean the usage of IT solutions, that are exist in almost every organisational process. However, concentrating only on IT solutions would exclude the research of communication tools, such as telephone, mobile phone or videoconference.

Therefore the research was conducted on the field of wide technology (including IT and telecommunication technologies). Technological solutions provide the environment and infrastructure for executing processes. Widely accepted and used technological solutions, like communication devices (telephone, mobile phone, fax), internet access, e-mail are excluded from the analysis. These tools are the basic part of the daily operations of organisations.

At 62,5% of the researched organisations clear technological support of knowledge management is identifiable. Regarding to the knowledge value chain technological support mostly used for creating, storing and sharing knowledge, but except these knowledge processes, others were no so popular. Therefore only a close circle of the possible technological solutions were used, but complex solution, that support every organisation activity (Lawton, 2001) have not been introduced. (example: TTC used the most technological solutions to support storing and using existing experience: Qualitative and quantitative information stored in conventional databases; "Best Practice" based general tools; expert system; Knowledge map of employees having useful experience; Detailed description of processes based on the environment).

In the field of office-automation, for communication, widely accepted, basic solutions were mostly used. For coordination activities, the usability of specific solutions (workflow) is rather low. Mostly cooperation activities were supported by document management, modelling, CASE tools that supports that widely accepted theory, that the area of cooperation has the biggest impact.

Based on the statistical analysis the reason of using technological solutions was primarily supporting the cooperation activities (between departments and employees), and providing communication channels. However beside technological solutions, personal interactions were de-emphasized (e.g. mentoring).

Form of technological support	Rate	Main correlations
Technological support overall	62,5%	+ co-operation between several departments is necessary (,333) + Codification strategy (,400) - Personalization strategy (-,367) - mentoring, coaching (-,238) - empowerment (-,353) - ask for the opinion of the employees, agreeing (-,417)
Document management / knowledge repository	38,89%	+ Codification strategy (,353) + strategy focusing inwards (,262) - empowerment (-,238)
CASE tool, modelling	15,28%	+ psychological motivation (,290) + kind of culture (,247) + support the co-operation (,499) + drawing in via the trainings (,302) - Personalization strategy (-,268)
Workflow solution	5,56%	Not applicable
Knowledge map	5,56%	Not applicable
Data analysing tool	22,22%	- mentoring, coaching (-,238) + managerial role (,308) + necessity of managerial support (,273)
Expert system	1,39%	Not applicable
Other supporting tool	9,72%	-
Support the document handling	-	+ codification strategy (,851) + change of culture (,261) - Personalization strategy (-,358) - integration of new knowledge (-,383) - training during work (-,390)

Table 2: Summary of technological supporting tools

Technological solutions can ensure the more independent work of employees with wider decision making possibilities with less risk. However, the experiences show the opposite: empowerment is typically used at organisations that do not emphasise the role of technologies. Technological solutions can provide higher possibilities of controlling activities that could be beneficial for managers, therefore it can support centralised decision making. This theory is supported by the phenomena, that during introduction of technological solutions, the opinion of employees was not emphasised.

To summarise, the usage of technological solutions is typical at organisations that use mainly codification KM strategy approach, and less typical in the case of personalisation strategies. However technological solutions are able to support even the personalisation strategy. Knowledge map solutions that describe the knowledge, expertise and skills of employees, and communication solutions are able to support personalisation goals.

Organisational Issues

In the case of analysing organisational issues – based on the research model – overlapping categories, factors can be recognised, and these factors can affect each others in several occasions.

The application of the organisational factors has a significant role not only regarding to the provision of the activities, but also for the purpose of the execution of the changes. Change management deals prominently with the role of the managers, the involvement of the employees as well as the question of change of the organisational culture.

In most of the organisations the new values, new solutions were built into the daily operations and behaviour, therefore the culture has been successfully changed. Representative cultures were the knowledge networks (informal knowledge sharing, mentoring, learning-by-doing), and knowledge flare (management example and support). The change of culture and behaviour did not base on status or compensation.

For success the formal and informal channels of personal communication have vital role that is mostly the characteristic of organisations with personalisation strategies. In the researched organisations, significant change of the organistional structure was not experienced, organistions were project based, only new roles or organistional units were appeared related to knowledge management activities.

Measurement and feed-back

In the case of the researched organisations, measurement and feedback activities were experienced on two levels. On the top level, mostly benchmarking, or software process levels approaches (Bootstrap, CMM) were used that resulted a level of software process maturity. It is necessary to measure these levels not only before and after changes, but also in regular periods, in order to provide continuous control, and evaluation of tendencies. This approach was realised in the majority of the examined organisations (61,1%), and the results of this benchmarking approaches is comparable with the practice of other organisations. This approach concentrates to process development and development, therefore these measures are partly adequate for assessing the knowledge-related activities.

A deeper, operative level of measurement activities is the continuous controlling of performance indicators that could be the basis of further process improvement activities. This approach provides a more direct feedback, and indicates areas in which development is required, or explores problematic or risky areas.

Measurement approaches examine the adequacy of processes, explore the roots and causes of problems, and support problem solving. Based on these measurements, the organisation of activities, the selection of supportive tools or organisational solutions can be arranged or changed. Based on operative measurement, the continuous, incremental development of supportive solutions was necessary. Measures did not analyse the effectiveness of single supportive factors, and deep cause-result analysis was not possible. During measurement the effectiveness of knowledge-processes and supportive tools were analysed together.

Based on a third kind of method, performance was measured in separate departments of organisations, and summarised indicators were analysed. These indicators can be: costs, number of faults, time-to-market ratio. These indicators are adequate for comparing the situations before and after changes.

5 Conclusions

Every supporter in the research model has a role in the examined cases, although their importance is variable. The basic findings are the following:

- Knowledge management strategies have impact both on KM processes and on supporters. KM strategies connect organisational strategy with KM practice.
- The use of technological solutions is widely accepted. They have a role not only in the case of codification strategies, but in these cases they have a vital role.
- Knowledge oriented organisational culture and support of communication has critical role for success in the area of organisational solutions. The structure of organisations has not changed dramatically, therefore the evaluation of the role of organisational structures is partly missing. Incentives have not a critical role, especially of using cash-incentives. Lack of management support is more a barrier, but definitely not a success factor
- Performance evaluations and metrics are present during KM practices, but the scope is slightly different, and rarely KM specific. Metrics are often measure the performance of a department, or a project.

As result of the research the basic research model has been extended. This model includes both technological and organisational solutions, support factors, that are able to increase the efficiency of knowledge management practices in software development organisations, and therefore the efficiency of organisational processes. (Figure 1.) Based on the conducted research that for conscious management of organisational knowledge in software development organisations the simultaneous use of both factors are necessary, based on a well-grounded knowledge management strategy and with the support of continuous feedback and performance evaluation.

6 Acknowledgements

I have to thank to International Software Consulting Network[1], for the possibility of conducting this research, as well for Varpex Plc. (Depozit) for supporting it. I also would like to thank for the unknown reviewers for reading and commenting the paper.

[1] http://www.eurospi.net/

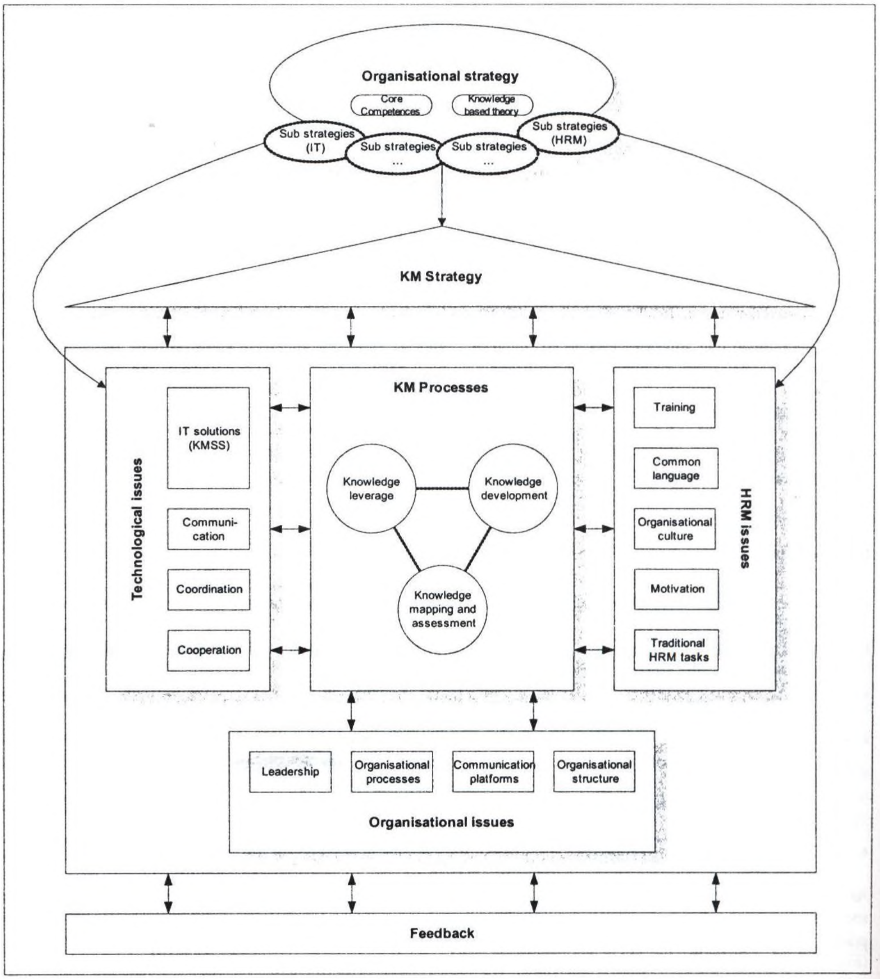


Figure 1: Research validated framework of knowledge management and supporters in software development organizations.

7 References

1. Adelmann, H. – Jashapara, A. (2003) Embedding knowledge management into business processes: The use of threaded discussion forums and knowledge objects at AstraZeneca, in: Proceedings of the Fourth European Conference on Knowledge Management (McGrath, F., Remenyi D., Eds.), Management Centre International Limited, Reading, pp. 13-23.
2. Bíró M – Messnarz R. (2000) Key success factors for business based improvement, in: Software Quality Professional, Vol. 2., No. 2., pp 20-31.
3. Dorling, A. (1993) SPICE: Software Process Improvement and Capability dEtermination, in: Software Quality Journal, Vol. 2, No. 4, pp. 209-224.
4. Eisenhardt, K.M. – Martin, J.A. (2000) Dynamic capabilities: What are they? In: Strategic Management Journal, Vol. 21. No. 10-11 Winter Special Issue, pp. 1105-1121.

5. Fehér, Péter (2004) Combining Knowledge and Change Management at Consultancies, in: Electronic Journal of Knowledge Management, Vol. 2. No. 1, pp. 19-32.
6. Grant, R.M. (1996) Toward a Knowledge Based Theory of the Firm, in: Strategic Management Journal, Vol. 17, Winter Special Issue, pp. 109-122.
7. Hansen, M. – Nohria, N. – Tierney, T. (1999) What's Your Strategy for Managing Knowledge? in: Harvard Business Review, Mar-Apr, pp. 106-116.
8. Kuvaja, P. - Simila, J. - Krzanik, L. - Bicego, A. – Koch, G. - Saukonen, S. (1994) Software Process Assessment and Improvement: the BOOTSTRAP approach, Blackwell Publishers, Oxford
9. Lawton, G. (2001) Knowledge management: ready for prime time?, in: IEEE Computer, Vol. 34, No. 2, pp. 12-14.
10. Meehan, B. – Richardson, I. (2002) Identification of Software Process Knowledge Management, in: Software Process Improvement and Practice, Vol. 7, No 2, pp. 47-55.
11. Messnarz R., et. al. (1994) BOOTSTRAP: Fine Tuning Process Assessment, in: IEEE Software, July, pp. 25-35
12. Nonaka, I. – Toyama, R. (2003) The knowledge-creating theory revisited: knowledge creation as a synthesizing process. Knowledge Management Research & Practice Vol 1, No 1, pp. 2–10.
13. Paulk, M.C. – Curtis, B. – Chrissis, M.B. – Weber, C.V. (1993) Capability Maturity Model, Version 1.1, in: IEEE Software, Vol. 10, No. 4, pp. 18-27.
14. Rubenstein-Montano, B. – Liebowitz, J. – Buchwalter, J. – McCaw, D. – Newman, B. – Rebeck, K., The Knowledge Management Methodology Team (2001) A systems thinking framework for knowledge management, in: Decision Support Systems, Vol. 31, No. 1. pp. 5-16.
15. Siakas K.V. - Georgiadou, E. (1999) Process Improvement: The Societal Iceberg, in: European Software Process Improvement Conference (EuroSPI '99), Pori, Finland, 25-27. November 1999.
16. Sila, I. – Ebrahimpour, M. (2002) An investigation of the total quality management survey based research published between 1989 and 2000: A literature review, in: International Journal of Quality & Reliability Management, Vol. 19, No. 7, pp. 902- 970.
17. Tampoe, M. (1993) Motivating Knowledge Workers – The Challenge for the 1990s, in: Long Range Planning, Vol. 26. No. 3. pp. 49-55.
18. Truch, E. – Bridger, D. (2002) The Importance Of Strategic Fit In Knowledge Management organisation, in: Proceedings of the Xth European Conference on Information Systems: ECIS 2002. June 6-8, Gdansk, Poland (Ed: Stanislaw Wrycza), University of Gdansk, pp. 905-918.
19. von Krogh, G. – Ichijo, K. – Nonaka, I. (2000) Enabling Knowledge Creation, Oxford University Press, New York
20. Wiig, K.M. (1999) What future knowledge management users may expect, in: Journal of Knowledge Management, Vol. 3, No. 2, pp. 155-165.
21. Zack, M.H. (2000) Developing a Knowledge Strategy: Epilogue, in: The Strategic Management of Intellectual Capital and Organisational Knowledge: A Collection of Readings (Bontis, N. and Choo, C.W., eds), Oxford University Press
22. Spender, J.C. (1996) Making Knowledge the Basis of a Dynamic Theory of the Firm, in: Strategic Management Journal, Vol. 17. Winter Special Issue, pp. 45-62.

Less is More in Software Process Improvement

Andre Heijstek[1], Hans van Vliet[2]
SEI Europe, An der Welle 4, 60322, Frankfurt, Germany
andreh@sei.cmu.edu
Vrije Universiteit, De Boelelaan 1081a, 1081 HV, Amsterdam, The Netherlands
hans@few.vu.nl

Abstract

Many software process improvement (SPI) programs fail to deliver the benefits they promise. This is usually due not to bad new processes being created, but to a lack of adoption of the new processes. Many articles have been written about reasons for lack of adoption. One element that is missing in these articles is the capacity of organizations to adopt more changes. Often organizations implement many changes in parallel, hoping to achieve an accumulated benefit. We conjecture that in many cases less is more – concentrating on implementing a few changes well yields more benefits than implementing a lot of changes less well. This article presents a model to analyze and measure adoption of processes and some practical techniques to support adoption of processes.

Keywords

Software Process Improvement, Adoption, Infusion, Diffusion, Absorption Capacity

1 Short description / Stage setting

Many software process improvement (SPI) programs fail. Practitioners in the field of SPI claim that about 80% of those programs fail to achieve their promised benefits. Unfortunately there is no published data to support this statement, for an obvious reason: people do not publish failures. Studies show that 67% of SPI managers want guidance on *how* to implement SPI activities, rather than *which* SPI activities to implement [10], which can be taken as some evidence confirming the high failure rate claim. Even programs that do not fail as such often fail to deliver the promised returns. It appears to be very difficult to truly change the behavior of managers and engineers involved in software or systems development. Failure can be complete, where nothing of what was planned is achieved. But mostly, failure is much more subtle—some new processes are adopted to some extent, but not to the extent needed to achieve any real benefits. Improved processes do not bring any benefits if they aren't used. We conjecture that the lack of adoption is often caused by the desire to do too much at the same time. The support activities needed for successful adoption are too diluted over all changes and lose their effectiveness.

There are many consequences of failed adoptions:

- Organizations still do not achieve their development project goals, and projects often are too late, over budget, deliver lower quality than planned
- The investment in the improvement program is fully or partially wasted
- The improvement model (e.g., CMMI, Six Sigma, ISO 9000) is perceived by the affected organization as a bad model that does not deliver what is promised

It is currently difficult to analyze if improved processes have been fully adopted. For SPI programs based on the CMMI model [19] there is a well-defined appraisal method [20]. However, even with this rigorous method it is almost impossible to gauge if the processes have really changed the hearts and the minds of the practitioners.

This paper attempts to address these issues and indicate directions for future research, focusing on understanding adoption to the level that we can influence and measure it, and on the absorptive capacity of an organization, the extent to which it can adopt more changes. This paper first elaborates on the research questions, and then reviews the most applicable prior literature, derives methods and models for adoption and concludes with suggestions for further research.

2 Key Questions / challenges that we address

To support SPI programs to achieve better results one not only needs to introduce the right technologies and processes but the following topics need to be addressed as well: transition mechanisms, adoption measurement and absorptive capacity. When current SPI programs fall short in achieving adoption of processes, we need to offer systematic mechanisms to achieve adoption. We call these transition mechanisms, as they support an organization transition along the stages of the adoption curve: contact, awareness, understanding, trial use, adoption, institutionalization and internalization [5], [9]. Taking an organization step by step through these phases strongly enhances the likeliness of full adoption. Measurements of adoption make the effectiveness of these transition mechanisms transparent, allowing corrective action in cases where progress stalls. As–in our view–absorptive capacity is one of the most prominent limiting factors in achieving full adoption, we devote special attention to this topic.

2.1 Transition Mechanisms

It is widely recognized that process change or technology change occurs through several distinct stages [5], [15], [7, 8]. Prospective users of new processes first get in *contact* with the new process, become *aware* of what this process could do for them and obtain a deeper *understanding* of the technology itself. When these stages leave a positive impression users will perform a *trial usage* of the process before deciding on full *adoption*. The new processes become *institutionalized*, embedded in the culture, and after sufficient passage of time they become *internalized*; users cannot think they would ever perform the process in the old way again. Recognizing these stages, we could use them actively to support adoption in an SPI program. The design of the SPI program should identify and plan mechanisms to support the process users to proceed along these stages. This paper gives suggestions for identification of transition mechanisms and relates these mechanisms to adoption measurement.

2.2 Adoption Measurement

To make any founded claims about the lack of adoption of new processes, we should be able to measure adoption. To what extent do people apply a process? And how many people from our target population do apply the new processes. The following example illustrates the difference between partial and full adoption. An organization has developed a new estimation process to improve the accuracy of the development project estimates. The new process includes a spreadsheet which requires several attributes to be given (number of screens, number of interfaces, etc.) and then suggests an estimate for software development effort. The estimator is allowed to differ from this suggestion, but he then needs to document why.

A partial adoption of the new estimation process could have the following indicators:

- People use the new estimation spreadsheet.
- They compare the model outputs with what their gut-feeling estimate tells them.
- They then tune their inputs into the model to make both numbers match.
- When estimates turn out to be wrong, they blame the model.

A full adoption of the new estimation process might look like this:

- People use the new template and estimation model.
- They also compare model results with their gut feeling estimates.
- They document where there are differences between the two and involve the estimation process owner in case of significant deviations.
- During project execution they compare their estimates to the actual numbers for effort and time for each of the tasks performed.
- Based on this the estimation model parameters are tuned to better match reality.

Key differences between real and partial adoption:

- The estimation method is used according to its intentions.
- The estimation results are integrated with the project tracking process. More generally speaking, the results of one new process are integrated with other processes, in ways that the process author may not have foreseen.
- The organization uses feedback to become a learning organization.

2.3 Absorptive capacity

SPI involves change, changes to the work practices of managers and engineers. Is there a limit to the amount of change people and organizations can handle?

If we implement too many changes at the same time the resistance against the changes will increase, the amount of support and coaching for each change will be diluted, leading to a less than optimal implementation. On the other hand, if we change too little we may not achieve the goals we strive for. Can we find an optimal amount of change? There may be synergy between these multiple improvement programs, but the opposite, dysergy, is a definite possibility as well. On top of what an SPI initiative launches as changes, the target group of our SPI program is subject to much more change than that what is initiated by the SPI program: reorganizations, new products, new technology, changes in personnel and management. We expect that absorptive capacity is to process improvement what Brook's law is to project management [3]. Brooks stated that adding more resources to a late project would only make it later. We conjecture that adding more improvements to a suboptimal organization only makes it more suboptimal. We want to define, measure and analyze the concept of absorptive capacity, and find ways to influence it.

3 Review of Existing Literature

Changes to processes should become changes to work practices, changes to behavior. Ajzen [1] describes in the theory of reasoned action (TRA) how intentions and beliefs may or may not lead to actual changed actions. According to TRA, a person's performance of a specified behavior is determined by his or her behavioral intention (BI) to perform the behavior, and BI is jointly determined by the person's attitude (A) and subjective norm (SN) concerning the behavior in question. As TRA describes behavior in a general sense, this model does also apply to behavior in the context of SPI. TRA is a descriptive model; it describes how behavior is driven by attitudes and subjective norms. We conjecture that absorptive capacity is missing in this model. TRA assumes that when one has the intention to adopt a certain behavior, this behavior will actually be performed. In reality people are typically constrained to adopt a new behavior by lack of time or lack of training.

Gallivan [7] describes many relevant aspects of process adoption. Based on his analysis of core frameworks on innovation adoption from the social psychology [1], [15], [6] he concludes that these models neglect the realities of implementing technology in organizations, especially when adoption decisions are made at the organization, division, or workgroup levels, rather than at the individual level. In these cases adoption follows two decisions: the primary adoption decision by management to adopt a new technology, and a secondary adoption decision by the workforce to use it in their daily work. Following the primary adoption decision, management could proceed by three fundamentally different paths to ensure secondary adoption: (1) they can mandate that the innovation be adopted throughout the organization at once; (2) they can provide the necessary infrastructure and support for employees to adopt the innovation allowing a more voluntary diffusion; or (3) they may target specific pilot projects within the firm, observe the processes and outcomes that unfold, and decide whether to implement the innovation more broadly later on. This confirms our conjecture that if organizations attempt to implement too much, they won't have the time and resources to ensure a thorough secondary adoption takes place.

Gallivan's work focuses mainly on the adoption of 'hard' technologies, for example the adoption of client-server computing. In our work we focus on 'soft' technologies or process technologies, where the adoption decision is not binary (adopt or not adopt), but continuous (to what extent). Gallivan focuses his research on when and how adoption occurs; we are mostly interested in how well adoption occurs. A similar interest is found with Saga and Zmud [17] who identify three different facets of infusion: more extensive use of the innovation (e.g. using more technology features); more integrative use (using technology to create new workflow linkages among tasks); and emergent use (using technology to perform tasks not previously considered possible).

A study has recently been published that applies several social psychology models to the world of SPI. Umarji [21] tries to predict the acceptance of SPI programs on the basis of TRA and the technology acceptance model [1], [6], extending these models because of the fact that SPI is intangible, often more intrusive and judgmental than hard technologies. She suggests adding four groups of factors to the model for SPI:

- Organizational issues: visibility, transparency, and reward structure/incentives.

- Personal issues: fear of adverse consequences, communication, self-efficacy, and degree of control.
- SPI-related issues: amount of learning required, compatibility of work practices, champions/advocates.
- Factors from social psychology: perceived usefulness, attitude, perceived behavioral control, subjective norm, ease-of-use.
- We agree with the reasons to adapt the model, however we conjecture that most of the factors she describes are specializations of the attitude and subjective norms in TRA.

A model that has become quite popular to support SPI programs comes from the family therapy domain [18], [22]. Satir describes a process of change as follows: we begin with the *status quo* of the organization (termed *old status quo* by Weinberg), a situation where people have learned to cope with the fallacies in the current system. Then a *foreign element* is introduced, in the context of SPI that would be the SPI initiative. The foreign element causes *chaos*, people who have long worked in the old status quo see all kinds of comfortable practices change, and the people and the system moves into a state of disequilibrium. At some point in time a *transforming idea* is found that drives the system into *integration*, *practice*, and *new status quo*. The description here is sequential, however practice often is not. At each point in the process there is the possibility to fall back to the old status quo. Weinberg [22] suggests that introducing new changes during the chaos phase should be avoided at all times, and mostly during the integration phase as well. Adding new changes during chaos and integration just increases the risk of falling back to old status quo. The optimum time for a new change is during the practice phase, shortly before new status quo. In that phase the chaos is over, so organizations are capable of adopting something new again, but not yet so fixed into a new status quo that new changes will lead to a major chaos again. We suggest using the transition mechanisms as a means to support the organization to find effective transforming ideas quickly. The notion of not adding new changes during transition supports our concept of a limited absorption capability.

4 Approach taken

We are taking the Action Research [14] approach to this investigation and will develop new theory on the basis of observations in both practice and previous literature.

We created a theoretical model about adoption based on the literature review and our industrial experience, see Fig. 2.

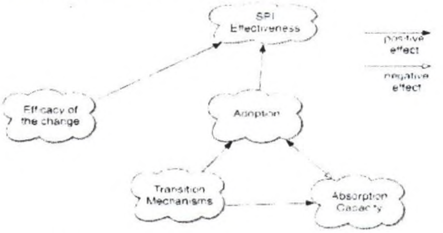


Fig. 1. **Modeling the effectiveness of SPI as a function of efficacy and adoption**

The model describes how the effectiveness of an SPI program depends on the efficacy of the change and the adoption of the change. Some authors combine efficacy and adoption into one concept; for example Wynekoop as quoted in [7] defines infusion as "the extent to which an innovation is used completely and effectively and improves the organization's performance". However, we claim that separating the orthogonal constructs efficacy–the content of the improvement program–and adoption–the method of change in the improvement program–gives us more analytic power. The efficacy of the change is a key successfactor. Have we implemented the right thing? Do the new processes solve the problems of the old situation? John Rost [16] gives some excellent examples of implementing the wrong set of processes in certain situations. For this study we assume that a useful approach has been taken. Our focus is on the adoption of the change. We contend that this depends on two main factors: the transition mechanisms used, and the absorption capacity of the organization.

Another way to look at the adoption part of the model would be to adopt the TRA model [1], using absorption capacity as a new construct, see Fig. 2.

QuickTime™ and a
TIFF (Uncompressed) decompressor
are needed to see this picture.

Fig. 2. **Theory of Reasoned Action model augmented with Absorption Capacity**

The absorptive capacity is a new construct that limits the Behavioral Intention of the target group of new processes to adopt it.

4.1 Transition Mechanisms

Adoption of change typically follows a seven-phase adoption curve [5]. The phases are contact, awareness, understanding, trial use, adoption, institutionalization, and internalization. When organizations conduct a process improvement program, they should apply mechanisms to help the organization transition from one phase to the other. Mechanisms to bring people into the contact phase could be e-mails or posters about the new technology. Mechanisms to bring people from contact to awareness could be information presentations on the new technology. The more effective the mechanisms are, the more effectively and efficiently will the organizations adopt the new technology. There is not, and cannot be, any single set of transition mechanisms. Organizations have different backgrounds, cultures, and structures, so the set of mechanisms should be defined for each organization. Effective mechanisms from the past are of course a very good source for new efforts. Table 1 gives several examples of transition mechanisms for each phase, obtained from a customer engagement and from work by Garcia [9].

Table 1. Transition Mechanisms

Stage	Transition Mechanisms
Contact	Information emails, Posters, Brown bag lunches
Awareness	"Elevator speech" Magazine articles and conference briefings Flash cards with objectives, benefits, URL, etc. Web site, FAQ Successful ROI stories, case studies
Understanding	Training sessions, communication Detailed case studies Identify and authorize champions

	Identify stakeholder roles and responsibilities
Trial use	Coaching by process experts Small working group to support pilots Special authorities for pilots Documented pilot results
Adoption	Availability of process artefacts Strong set of incentives; rewards and consequences Refined guidance on process usage choices Education - mature courses, modularized for JIT delivery In-Process Aids
Institutionalization	Inclusion in quality audits Integration with other processes Emergent use in unanticipated situations Fully realized curriculum of training for different users New employee training/orientation Continuous improvement to adoption artifacts (guides, etc.)
Internalization	-

4.2 Absorption Capacity

There is no literature on the concept of absorption capacity except from the remarks Weinberg [22] makes about what a good time is to introduce new change.

Independent of how good the transition mechanisms are, the capacity of the organization to absorb change is a limits the adoption of new changes. Organizations trying to adopt many changes at the same time are likely to fail in several of them. Both individuals and organizations have a limited capacity for change and will resist more changes once their limit is reached. What determines this adoption capacity? At what point do people and organizations reach their limits? Can we identify mechanisms that enlarge absorption capacity? Can we measure absorption capacity and use that measure to predict when limits are likely to be hit? Probable factors that influence the absorption capacity are: the amount of changes at a given point in time, individual personalities and traits [15], [11], passage of time (it makes a difference if the last change was a month or a year ago), top-down or bottom-up changes, group effects, learning organization characteristics. We plan to explore this phenomenon in future research.

Brooks' law [3] states that the efficiency of teams reduces when new team members are added due to increased communication between the team members. By analogy we suggest that the effectiveness of SPI reduces when more change is introduced due to diluted attention to each of the SPI initiatives. Effective SPI requires that sufficient support be given to the target group [2], [12], [13], training, coaching, motivation. This support should be given both my management and by a process group. The generic practices in the CMMI model [19] are included in the model to ensure the new processes are institutionalized. The generic practices include: training, resources, measurement, and verification by quality assurance and management. These support activities should be performed for each process in the organization, with specific attention to new processes that still need to be institutionalized. The more changes we have, the more diluted the support activities will be, leading eventually to a level of support that falls below a threshold where it has any effect.

If the number of process improvement activities is P, and the total capacity to support process improvement is S, then the average support given to each improvement activity is S / P. If the effective contribution to an organization's effectiveness for each improvement is $e_{i,}$, then the overall effectiveness E of all improvements is:

$$E = \sum_{1}^{P} \frac{e_i}{P} \qquad (1)$$

Formula 1 does not account for synergies or dysergies between improvement activities, nor for a threshold value for support. The total effectiveness E is the return we get on process improvement. This is strongly reduced by conducting many improvements; the investment increases with every process improvement, making the return on investment (ROI) very negative when many improvements are carried out.

4.3 Adoption Measurement

Process adoption is generally characterized as consisting of two orthogonal dimensions [9], [7]: infusion and diffusion. Infusion or depth of adoption describes how well the new process has been adopted by the target population. Is there a full or a partial adoption of the new processes? Diffusion or breadth of adoption describes how broadly the target population has adopted the new process. Do they all apply the new processes? It has been suggested by Kaputo and Garcia [4], [9] to use these dimensions as measurements for adoptions. An additional benefit of measuring adoption is that it provides a leading indicator for the return on investment (ROI) of process improvement. Assuming the right new processes have been developed, a high adoption measure tells us that a return on investment is to be expected.

An example of how an adoption measurement could be presented is given in Fig. 3. This shows a hypothetical organization that has introduced three process improvements. The inspections are performed very well by a small group of people; most people perform estimations poorly and testing is performed well by most.

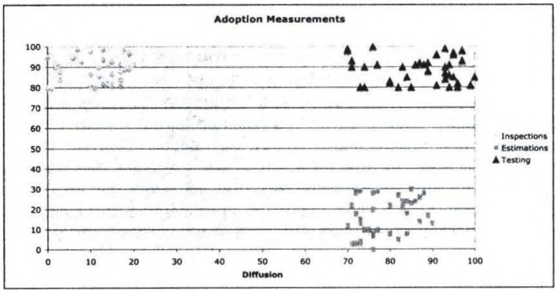


Fig. 3. **Graph of adoption as measured by diffusion and infusion**

4.3.1 Measuring Diffusion

To measure diffusion, we should measure how far the target population has come along the stages of the commitment curve [5]. When transition mechanisms have been defined in an improvement project, these mechanisms can be used as proxies for adoption commitment. Transition mechanisms are the means that have been used to get the target population through the stages of contact, awareness, understanding, trial use, adoption, institutionalization and internalization. Examples of transition mechanisms from some recent customer engagements can be found in **Table 2.** Suggested measurements for each stage in the adoption curve**Table 2**

Table 2. **Suggested measurements for each stage in the adoption curve**

Stage	**Measurement**
Contact	
Awareness	Number of attendees
Understanding	Number of attendees, course completion tests
Trial Use	Number of pilot projects, number of people reporting usage
Adoption	Number of projects/people reporting usage, practice implementation indicators, training attendees
Institutionalization	Process change requests
Internalization	

Table 3. **Transition Measurements**

To compute a composite diffusion measurement D, we count the number of people P_s who are in a certain stage S of the adoption curve, and give weights to each stage W_s, internalization gets the highest weight, contact the lowest.

$$D = \sum \frac{P_s}{W_s} \qquad (2)$$

4.3.2 Measuring Infusion

To measure infusion we need to gauge how well the people within the target population use the new processes. The process developer should identify all roles that should work with this process, and for each role define how well the knowledge of the process should be and what the typical tasks would be when the process is fully carried out. The infusion could be measured along the infusion facets: ordinary use, extensive use, integrative use, and emergent use. The following table uses an inspection process as an example.

Table 4. **Using infusion facets as means to measure infusion**

Role	**Tasks**	**Ordinary use**	**Extensive use**	**Integrative use**	**Emergent use**
Meeting moderator	Send invitations, moderate the meeting, follow-up that rework is performed	Inspection form with indication that rework has been checked	Not just asking for rework completeness but go through all changes together	Using checklists from the producing process (e.g. design)	Using inspection data as input to predictive failure models
Author	Inform	Time	Having	Using the	Use

	moderator that work is ready for inspection. Perform rework	spent on rework after the meeting	criteria defined to determine when a document is ready for inspection	inspection status of documents as progress indicators in project tracking	inspection effort as cost driver in a Cost of Quality model
Reviewer	Review material in advance of the meeting. Comment during the meeting	Preparation time	Using a checklist in review preparation	Ensure all stakeholders are involved in the review	Use inspection outcomes in risk management

5 Expected or Achieved Results

This paper describes a model around adoption of software process improvement based on literature and on the experience of the author. Some elements have already been applied in practice: we have identified transition mechanisms, infusion and diffusion measures with several customers in an engagement to start an improvement program. The program is currently too young to see any practical results; it is too early to gather data based on the measurements defined.

So, the model will be further validated using action research. Some initial ideas on further work include:

- Study an improved process that has already been well adopted according to an organization, retrospectively define the infusion and diffusion measurements, and measure whether the intuitive understanding of adoption matches with measurement data, thus validating the concepts of measuring infusion and diffusion.
- Refine the methods to establish infusion and diffusion measurements, allowing taking accurate measurements at low cost.
- Perform a controlled experiment in a university setting to validate the absorption capacity construct in the adapted theory of reasoned action model.
- Validate absorption capacity comparing companies with many changes against companies with few changes.

6 Conclusion

Rogers has identified that most research on innovation diffusion has a clear pro-innovation bias, and gives some directions for research in other directions to strengthen the analytic power of the research. This paper, with its critical position towards performing too many improvements in parallel, shows a balanced position to innovation. We regard innovation as not a bad thing in itself, but suggest it should be treated with care and limits. We contend that 'less is more' in software process improvement.

7 Literature

1. Ajzen, I., From Intentions to Action: A Theory of Planned Behavior, Springer-Verlag, New York, 1985

2. Baddo, N. and Hall, T., Motivators of Software Process Improvement: an analysis of practitioners' views, The Journal of Systems and Software, 2001, (85-96)

3. Brooks, F. P., The Mythical Man-Month: Essays on Software Engineering, 20th Anniversary Edition, Addison-Wesley Professional, 1995

4. Caputo, K., CMM Implementation Guide: Choreographing Software Process Improvement, Addison-Wesley Professional, 1998

5. Conner, D., R. and Patterson, R., W., Building Commitment to Organizational Change, Training and Development Journal, 1982, (18-30)

6. Davis, F. D., Bagozzi, R. P. and Warshaw, P. R., User Acceptance of Computer Technology: A Comparison of Two Theoretical Models, Management Science, 1989, (982-1003)

7. Gallivan, M. J., Organizational Adoption and Assimilation of Complex Technological Innovations: Development and Application of a New Framework, Database for Advances in Information Systems, 2001, (51-85)

8. Gallivan, M. J., Strategies for Implementing new Software Processes: an Evaluation of a Contingency Framework, SIGCPR/SIGMIS, 1996, (313-325)

9. Garcia, S., Are you prepared for CMMI? 2002

10. Herbsleb, J. D. and Goldenson, D. R., A systematic survey of CMM experience and results., Proceedings of the 18th International Conference on Software Engineering, 1996, (323-330)

11. Moore, G., Crossing the Chasm: Marketing and Selling Technology Products to Mainstream Customers, Harper Business, New York, NY, 1991

12. Niazi, M., Wilson, D. and Zowghi, D., A maturity model for the implementation of software process improvement: an empirical study, The Journal of Systems and Software, 2003, (155-172)

13. Rainer, A. and Hall, T., Key success factors for implementing software process improvement: a maturity-based analysis, The Journal of Systems and Software, 2002, (71-84)

14. Remenyi, D. et al., Doing Research in Business and Management, Sage Publications, 2000

15. Rogers, E. M., Diffusion of Innovations, The Free Press, 1995

16. Rost, J., Software Engineering Theory in Practice, IEEE Software, 2005, (94-95)

17. Saga, V. L. and Zmud, R. W., The Nature and Determinants of IT Acceptance, Routinization, and Infusion, Diffusion, transfer, and implementation of information technology: proceedings of the IFIP TC8 Working Conference on Diffusion, Transfer and Implementation of Information Technology, Pittsburgh, PA. U.S.A., 11-13 October 1993, 1994,

18. Satir, V., Banmen, J., Gerber, J. and Gomori, M., The Satir Model - Family Therapy and Beyond, Science and Behavior Books, Inc., Palo Alto, CA, 1991

19. SEI, Capability Maturity Model Integration (CMMI), Version 1.1, Software Engineering Institute, 2002

20. SEI, Standard CMMI Appraisal Method for Process Improvement (SCAMPI), Version 1.1: Method Definition Document, Software Engineering Institute, 2001

21. Umarji, M. and Seaman, C., Predicting Acceptance of Software Process Improvement, St. Louis, Missouri, USA, 2005

22. Weinberg, G., Quality Software Management: Anticipating Change, Dorset House, 1997

8 Author CVs

Vliet, Hans van

Hans van Vliet is a Professor in Software Engineering at the Vrije Universiteit. His research interests include software architecture and software measurement. Before joining the Vrije Universiteit, he worked as a researcher at the Centrum voor Wiskunde en Informatica (Amsterdam) and he spent a year as a visiting researcher at the IBM Almaden Research Center in San Jose, California. Hans has an M.Sc. in Computer Science from the Vrije Universiteit and a Ph.D. in Computer Science from the University of Amsterdam. He has co-authored over 80 refereed articles, and is the author of "Software Engineering: Principles and Practices, published by Wiley (2nd Edition, 2000). He is the chairman of Jacquard, the Dutch national research program in software engineering.

Heijstek, Andre

Andre Heijstek has some 15 years experience in software development and software process improvement. He has been responsible for coordinating the Ericsson global CMM program, has lead the SEPG at Ericsson Netherlands. After his Ericsson career Andre supported various companies across Europe implementing SPI programs, in the automotive, telecoms and financial domains. He currently works as visiting scientist for SEI Europe. Andre's research interests include technology change management and measurement.

Process Improvement Through Evaluation of Operational Feasibility of Strategic Release Plans

Joseph Momoh[1)2)] & Guenther Ruhe[1)]

[1)] Laboratory for Software Engineering Decision Support
University of Calgary
2500 University Drive NW, Calgary, Alberta, Canada T2N 1N4
ruhe@ucalgary.ca

[2)] Trema Laboratories Inc.
Suite 200, 5970 Centre Street SE
Calgary, Alberta, Canada T2H 0C1
Joseph.momoh@trema.com

Abstract

Improvements in software processes can improve software quality, cost, and on-time delivery. Supplementing strategic software release planning with fine grain operational planning provides project managers the tools and abilities to make timely decisions affecting the project. Strategic planning of software releases is accomplished by the assignment of requirements to releases on a strategic level where effort, finance and risk constraints are considered to determine the strategic release plans. Strategic planning however needs to be supplemented by more fine-grained planning typically performed in project management known as operational release planning. Operational planning as a refinement of strategic planning is performed for the immediate next release.

In this paper, we present the improvements in software development process and product by performing the evaluation of the operational feasibility of strategic software planning. We also present the use of research prototype ReleasePlanner™ in a real-world situation through a case study at Trema Laboratories Inc and the improvements achieved.

Keywords

Release Planning, Strategic Planning, Operational Planning, Decision Support, Stakeholder, Process Improvement, ReleasePlanner, Case Study.

1 Introduction

Improving software processes can improve software quality, cost, and on-time delivery [Raffo 93]. In a worldwide survey of more than 100 global software companies and 450 top executives, [Hoch et al 00] presents a summary of some of the strategies used by successful software companies. These include:

- The right product at the right time is important. But the management's ability to learn from mistakes and make the right decisions is what decides between making and breaking it.
- It's the Development process that can either crash a company or boost its productivity. Software companies that have excellent processes such as very clear team structures, extensive stakeholder involvement, daily builds, and software reuse, largely ease frustration for developers thereby making them more productive. The processes make work more enjoyable. Boring rework and bug detection are reduced. At the same time product quality increases and time to market improves.

An approach that improves timely decision making and the development process is therefore crucial to success. The concept of continuous process improvement which has been long established in manufacturing industry is necessary in the software industry. Investment by any organization in process improvement has had significant benefits including improvements to product quality, reduction in the time to market of the product, and improvements in productivity [Zahran 98], increased organizational flexibility and stakeholder satisfaction [Florac et al 97]. As reported in the study by [Abrahamsson 01], "A 1996 report commissioned by The Data & Analysis Center for Software (DACS) reported that successful SPI programs have reduced the number of defects delivered to customers by 95%, reduced software development schedules by 71%, and increased productivity in terms of lines-of-code or function points per day by 222%. Additionally, SEI (Software Engineering Institute) reported an average return of 5:1 investments in successful SPI programs"

In software project management, the project manager has the most control in the release planning and implementation phases and the benefit of good decisions to the success of the project are highest in these phases. The committed cost of the project is still relatively low during these planning phases in the life of the project [Jurison 99]. The better the decisions taken during these phases, the higher the economic benefits realized from the project and the likelihood of project success.

Release planning can be done in very different ways [Bagnall et al. 01], [Penny 02], [Greer & Ruhe03], [Karlsson et al. 04], [Ruhe& Ngo-The 04]. Distinguishing parameters in approaching release planning are: Considered time-horizon of planning, objects of planning, degree of formality in the planning procedure, inclusion of additional constraints as part of the planning process, degree of stakeholders involvement, and finally the actual solution technique that is applied to obtain release plans.

In its easiest way, release planning is done in an ad-hoc manner, eventually supported by spreadsheet computation. In this paper, we will consider the improvement by applying a more sophisticated approach that is based on formal problem description and its solution using specialized optimization procedures called EVOLVE^{ext} which is based on the interplay between strategic and operational planning. The difference is in terms of the planning time horizon, the objects to be planned and the granularity of planning. EVOLVE^{ext} also provides a more flexible system of stakeholder voting. The new scheme encompasses groups of requirements and allows flexible assignment of stakeholders to requirements or groups of requirements.

Following this introduction, Section 2 studies the strategic release planning at Trema Laboratories Inc. and the required extensions of ReleasePlanner^{TM} to fully cover the interplay between strategic and operational planning and establishes the baseline on how ad-hoc release planning was done. Section 3 discusses the improvements achieved through the use of the intelligent decision support tool ReleasePlanner^{TM}. Section 4 addresses the evaluation of the improvements. Finally, Section 5 summarizes the paper and presents conclusions for future work.

2 Ad-hoc Planning at Trema Laboratories Inc.

2.1 Background

Trema Group is a provider of strategic software solutions for the financial industry. The software development focus is on providing a fully integrated cash and treasury management product suite designed to support front to back office treasury operations, as well as specific applications for cash management and accounting. Requirements are added to the product incrementally. So, planning for future releases becomes extremely important for the business.

The requirements for the next releases of the suite of products come from different sources such as contractual commitments, market positioning, technology opportunities, sales analysis, customer needs and existing functionality enhancements. The planning for the next releases is also tied to the strategic priorities which could change from release to release. The strategic priorities include:

- Marketability,
- performance,
- customer functionality,
- stability, and
- quality,
- scalability,
- technology,
- contractual commitments

To better understand existing challenges in planning of releases, a series of questionnaire based interviews were performed. The 23 participants were project managers (8 participants), product manager (5), development manager (6), and functional architects (4). The key messages to emerge from this survey and interviews are summarized in Table 1 below. The result was computed on the basis of the percentage of respondents ranking any of the factors as the greatest challenge.

Challenging Factors	% of Response
No easy way to get good overview of dependencies of requirements	26%
Stakeholders participation difficult especially in a distributed environment	22%
Difficulty in re-planning due to late breaking requirements or changing stakeholder priorities	17%
Managing resource constraints and their effect on a release is tedious	13%
Difficulty of translating corporate strategy to the product through the plan	9%
Assigning the right resources and skills to tasks to be performed in each requirement	4%
Visibility of the required skills and roles to implement each requirement.	4%
Adequate interaction among the stakeholders during the selection of requirements for a release	4%
No easy way to obtain stakeholder votes for requirements	0%
No easy way to generate reports showing the overview of stakeholder voting for each requirement	0%

Table 1: Ranked priority of challenging factors in release planning at Trema

The questionnaire contained ten key challenges facing the industry and participants were asked to rank them in order of magnitude of the problem. The sample for the survey is small but care was taken in drawing conclusions from the results through follow-up interviews with respondents to explore their understanding of some of the terms and issues in the survey in more depth.

2.2 Baseline

Current deficits in release planning and requirements prioritization are reported in [Lehtola et al. 04] and [Ruhe & Ngo-The 04]. As with most software organizations, the planning process is currently mostly ad-hoc with use of several spreadsheets through endless meetings with stakeholders in several different locations to arrive at an acceptable content of the releases.

The planning process is performed in two stages. The first stage is the development of the longer term roadmap also known as the strategic release plan. This usually spans three or more releases over one or more years. The roadmap is used as a strategy and planning document for the organization. The ultimate goal is to provide an optimal longer term release plan that will help fulfill the product's mission based on all the requirements that have been received. The roadmap is generally taken as a living document which evolves over time and becomes more concrete and offer more details as one gets closer to each of the releases on the roadmap. This strategic plan is also revised based on the lessons learnt from each of the releases as they are scheduled and executed. From the roadmap, the focus is usually on the near-term release and on how to achieve successful scheduling and execution of that release.

The inventory of requirements from which the roadmap or strategic release plan is developed from will usually contain over 500 items that are requested to be in future releases. Some of these items are dependent on other items in the inventory. There are five or more stakeholders with varying interests in each of the requirements participating in prioritizing these items. The stakeholders are located in different places around the world. The process of ranking and agreeing the content of the releases takes several days using spreadsheets to calculate and collate the results. Some of the problems with the ad-hoc approach include:

- There is usually no way to know all the alternative solutions considering all the constraints and dependencies between the requirements. The end result is the Non-feasibility of the generated solutions.
- Getting all the stakeholders together for meetings and go through all the requirements takes time.
- Every stakeholder has a different opinion of what is important and the process of negotiation among the stakeholders can take too long.
- Requirements are continuously changing and evolving. The impact of the changes on the strategic plan is not immediately obvious.

3 Use of Releaseplanner^{TM}

3.1 Improvement

Trema Laboratories Inc. found through the initial trial that ReleasePlanner^{TM} provided very good ways to manage the top four challenges in the release planning process identified in the survey results above. The time to generate release plans and to alternate release plans because of changing requirements was drastically reduced to clicks of buttons from a process that took several days. Figures 1- 3 below show some of the screens in the process of defining the above roadmap using ReleasePlanner^{TM}.

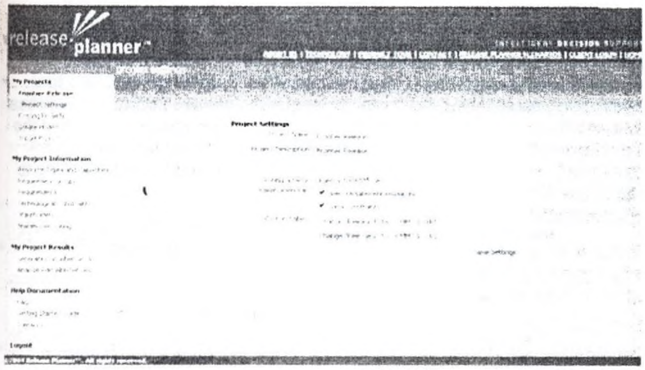
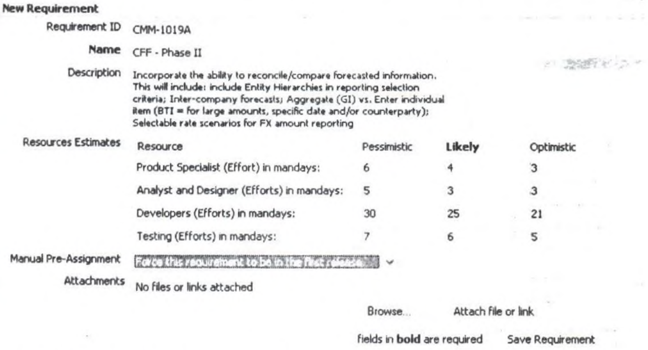


Figure 1: Defining the project settings.

The efforts for the release is measured in person-days and the totals available to be allocated to each of the releases have been determined based on the availability of the team members as shown in Figure 2 below.

The requirements are ranked based on the urgency model.



New Requirement

Requirement ID	CMM-1019A
Name	CFF - Phase II
Description	Incorporate the ability to reconcile/compare forecasted information. This will include: include Entity Hierarchies in reporting selection criteria; Inter-company forecasts; Aggregate (GI) vs. Enter individual item (BTI = for large amounts, specific date and/or counterparty); Selectable rate scenarios for FX amount reporting

Resources Estimates

Resource	Pessimistic	**Likely**	Optimistic
Product Specialist (Effort) in mandays:	6	4	3
Analyst and Designer (Efforts) in mandays:	5	3	3
Developers (Efforts) in mandays:	30	25	21
Testing (Efforts) in mandays:	7	6	5

Manual Pre-Assignment: Force this requirement to be in the first release

Attachments: No files or links attached

Browse... Attach file or link

Fields in **bold** are required Save Requirement

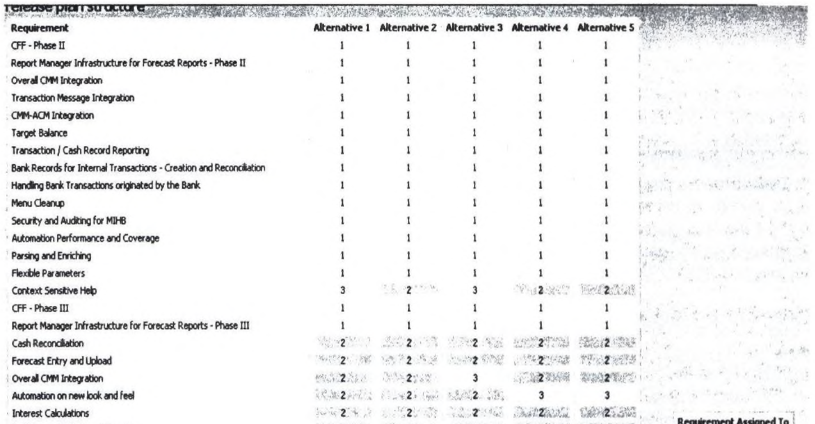
Figure 2: Capturing of the requirements.

The roadmap had a total of 49 requirements to be released over a period of one year. Certain requirements such as the building of the framework for future requirements are required to be in the early or first release in the roadmap. Some of the requirements must precede other requirements in the roadmap. In addition, we have six coupling constraints.

There are six stakeholders involved in voting on the roadmap. At the end of the voting exercise by all stakeholders in the project, release plan alternatives were generated by ReleasePlanner^{TM} as shown in Figure 3 below.

At the end of the trial of using the intelligent decision support for both strategic and operational release planning at Trema Laboratories Inc., the general agreement is that ReleasePlanner^{TM} provides benefits to the organization. Some of these benefits are tangible and some are intangible. The intangible benefits are difficult to realistically evaluate but the good thing in this study is that most of the intangible benefits are generally positive. In summary the benefits include:

- Improvement in pre-planning: Pulling together the roadmap and generating the release plans takes 60% of a full time job for a product manager and a release manager. This effort spent for this activity depends on the size of the organization and the complexity and size of the product. Bigger organizations tend to expend more effort expended for this activity. Keeping an overall view of all the requirements and the dependencies between the requirements makes this extremely difficult especially when dealing with hundreds of requirements. Also, gathering the efforts to implement each requirement and establishing the capacity versus the required effort for a release is a tedious process. ReleasePlanner^{TM} provides a quick overview of these requirements and the generation of alternative release plan becomes trivial through the push of a button on the keyboard. During the case study we experienced a reduction of the time spent in pre-planning to less than 20% compared to the 60% using an ad-hoc approach.



release plan structure

Requirement	Alternative 1	Alternative 2	Alternative 3	Alternative 4	Alternative 5
CFF - Phase II	1	1	1	1	1
Report Manager Infrastructure for Forecast Reports - Phase II	1	1	1	1	1
Overall CMM Integration	1	1	1	1	1
Transaction Message Integration	1	1	1	1	1
CMM-ACM Integration	1	1	1	1	1
Target Balance	1	1	1	1	1
Transaction / Cash Record Reporting	1	1	1	1	1
Bank Records for Internal Transactions - Creation and Reconciliation	1	1	1	1	1
Handling Bank Transactions originated by the Bank	1	1	1	1	1
Menu Cleanup	1	1	1	1	1
Security and Auditing for MIHB	1	1	1	1	1
Automation Performance and Coverage	1	1	1	1	1
Parsing and Enriching	1	1	1	1	1
Flexible Parameters	1	1	1	1	1
Context Sensitive Help	3	2	3	2	2
CFF - Phase III	1	1	1	1	1
Report Manager Infrastructure for Forecast Reports - Phase III	1	1	1	1	1
Cash Reconciliation	2	2	2	2	2
Forecast Entry and Upload	2	2	2	2	2
Overall CMM Integration	2	2	3	2	2
Automation on new look and feel	2	2	2	3	3
Interest Calculations	2	2	2	2	2
Bank Account Statement Report	2	2	2	2	2
Validation at Bank Statement Import	2	2	2	2	2
Intercompany Direct Debits	2	2	2	2	2
Credit Lines support - commitment and drawing fees	2	2	2	2	2
Tolerance Enhancements	2	2	2	2	2
MIHB Settlement	2	2	2	2	2
Many to many reconciliation	2	2	2	2	2
Routing Rules for External Bank Transactions / Receivables	2	2	2	2	2
User Configurable Routing Rules	2	2	2	2	2
DC Specific Interfaces	2	3	2	2	2
CFF - Phase IV	2	2	2	2	2
Report Manager Infrastructure for Forecast Reports - Phase IV	2	2	2	2	2
Dashboard	2	2	2	3	3
Manual entry of bank statements	3	2	2	2	3
Balances - Derivation and Display	3	3	3	3	3
Multiple Account Numbers for Payment Files and Bank Statements	3	3	3	2	2
Bank Account Statement Export	3	3	3	3	2
MIHB Transaction Routing	3	3	2	3	2
Bank Holidays Enhancement	3	3	2	3	3
Locking Bank Statements	3	3	3	2	3
Automatic Reconciliation and Tolerance Enhancements	3	3	3	3	3
Add TRM MT and LT forecasting functionality to CMM	3	2	3	3	3
User Configurable Transaction Approval Levels	3	3	3	3	3
Instant Messaging/Event Messaging	3	3	3	3	3
Charting	3	3	3	3	3
User Definable Daily Activity Timetable	2	3	3	3	3
FX Position Report	3	3	3	3	3

Requirement Assigned To	
1	CMM7.1.0-R1
2	CMM7.1.0-R2
3	Postponed

Figure 3: Structure of qualified release plan alternatives generated by ReleasePlanner^{TM}. Each column represents an alternative. Final selection is done by the human expert.

- Improved and quick re-planning capabilities: The re-planning process takes even further time and efforts. Whenever there is a change in a requirement, in the estimated size of a requirement or a change in stakeholder priority there is always a need to revise the plan to accommodate the change. This process consumes about 30% of a release manager's job including the time of other stakeholders involved in the release. It is estimated that the other stakeholders (6 in this case study) spend an average of 3 hours a week for re-planning during the release. The findings in the case study indicate that a saving of 75% was achieved through the use of ReleasePlanner™.

- Increased participation of stakeholders in the planning process: Through the ad-hoc approach, stakeholders will usually need to meet in a location to discuss, prioritize and finalize the content of the release. In distributed organizations where the stakeholders are based in different locations, this becomes very difficult and expensive considering travel costs. With ReleasePlanner™ stakeholders are able to participate from wherever they are in the prioritization process for as long as they have access to the internet. This saves travel cost and increases the level of participation. The added cost of not participating early in the process is in re-planning.
- Improved stakeholder satisfaction through active participation: It is established through the case study that stakeholders feel a sense of ownership of the content of the release because of their active involvement in the requirement selection and prioritization. This leads to stakeholders having better satisfaction with the release.
- Provides open, collaborative atmosphere among stakeholders: Stakeholders are able to interact directly with the system. This also leads to a higher level of satisfaction in the process.
- Provides ease of setting priorities and decision making: Priority setting by stakeholders is improved and allows for quick decision making by management on the content of the release. There is a more focused approach to delivering the requirements in the release by the project and development teams.
- Improves project planning, product quality and quick product wins thorough the incremental requirement selection and focus on the optimal selection of requirements.
- Improves time to market of required features: Since the appropriate selection of requirements are delivered in the release, the product gets to market with these required features. This potentially leads to increased market share through customer satisfaction with the product. During the course of the case study, revenue increased by $1,750,000.00. All of this increase cannot be attributed to the use of the new release planning system but attributing 1% of this increase to the system was deemed acceptable.

4 Evaluation of Improvements

4.1 Introductions

At the start of the case study, the release planning methodology at Trema Laboratories was mostly carried out ad-hoc. Many of the personnel involved in the planning process have built extensive experience with this ad-hoc approach to strategic and operational release planning. We have been involved in the release planning process in the organization for more than 4 years. This provided a good basis to measure the impact of introducing a new approach using ReleasePlanner™.

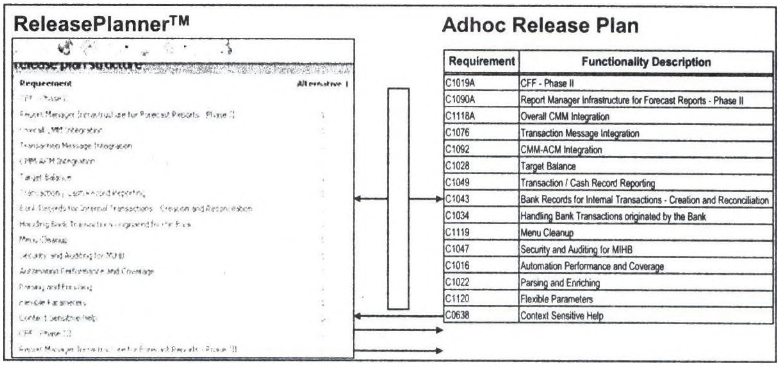
Most organizations expect decision support system to be a silver bullet that will kill the problems of operational release planning, the stakeholder cost of participation, quality of the product, and more. Decision support system for planning a release can have positive impacts in many areas, and this was demonstrated during the case study at Trema Laboratories, Inc. to save money and solve some of the planning problems highlighted during the survey and interviews with the industry.

This chapter also provides some practical guidance for understanding and computing cost and benefits from using ReleasePlanner™. There are many other factors that should be considered with the use of an intelligent decision support system such as ReleasePlanner™. It has some impacts on the organization in areas such as the planning approaches, resource utilization, resource scheduling, tasks granularity, product features, and the release dates or time to market of the product. It is therefore useful to understand the impact, potential benefits or value added to the business by the use of an intelligent decision support system.

Based on the case study at Trema Laboratories Inc., the impact of ReleasePlanner^{TM} will be assessed from three perspectives. The first is on the effect on the generated release plans, and secondly, the impact on the release planning process, and finally the impact on the product.

4.2 Impact on Generated Release Plans

Items not on the release plan: There were some obvious differences in the release plan alternatives generated by the ReleasePlanner^{TM} and the one produced using the ad-hoc approach with the same set of data. The benefit of using ReleasePlanner^{TM} is that the decision maker is presented with five possible alternatives as shown in figure 3 above to select the most feasible solution from.



ReleasePlanner^{TM}

Release plan structure

Requirement	Alternative 1
CFF - Phase II	
Report Manager Infrastructure for Forecast Reports - Phase II	
Overall CMM Integration	
Transaction Message Integration	
CMM-ACM Integration	
Target Balance	
Transaction / Cash Record Reporting	
Bank Records for Internal Transactions - Creation and Reconciliation	
Handling Bank Transactions originated by the Bank	
Menu Cleanup	
Security and Auditing for MIHB	
Automation Performance and Coverage	
Parsing and Enriching	
Flexible Parameters	
Context Sensitive Help	
CFF - Phase III	
Report Manager Infrastructure for Forecast Reports - Phase III	

Adhoc Release Plan

Requirement	Functionality Description
C1019A	CFF - Phase II
C1090A	Report Manager Infrastructure for Forecast Reports - Phase II
C1118A	Overall CMM Integration
C1076	Transaction Message Integration
C1092	CMM-ACM Integration
C1028	Target Balance
C1049	Transaction / Cash Record Reporting
C1043	Bank Records for Internal Transactions - Creation and Reconciliation
C1034	Handling Bank Transactions originated by the Bank
C1119	Menu Cleanup
C1047	Security and Auditing for MIHB
C1016	Automation Performance and Coverage
C1022	Parsing and Enriching
C1120	Flexible Parameters
C0638	Context Sensitive Help

Figure 4: Comparing ad-hoc versus ReleasePlanner^{TM} generated release plans

For the case study, the alternatives are similar in the assignment of requirements to release 1 except for the 'context sensitive help' requirement which goes into either release 2 or 3 in some of the alternatives. On further examination, the distinguishing differences between the alternatives are the assignments of requirements to releases 2 and 3 as shown in figure 4.

Alternative 1 is selected for further analysis because it most closely compares with the ad-hoc plan that was used during the case study. As can be seen in figure 4, there were two requirements suggested for release R1 by ReleasePlanner^{TM} that are not in the ad-hoc plan and one requirement in the ad-hoc plan that was suggested by ReleasePlanner^{TM} to be included in future release. The ad-hoc planning took into consideration that certain resources can only work on certain requirements regardless of the availability of similar resource types. This is 'Resource Specialization'. Requirements C1019A, C1090A in release R1 and C1019A, C1090A in R2 must be worked on by same resource and therefore cannot belong to the same release.

Maximization of efforts: ReleasePlanner^{TM} enabled the maximization of efforts available to the release which is difficult to achieve using ad-hoc planning approach.

4.3 Impact on Release Planning Process

Stakeholder involvement: There was a noticeable greater involvement of stakeholders in the planning process because the use of ReleasePlanner^{TM} eliminated the traveling overhead involved for stakeholders based in other locations since they are all able to log on to the tool and provide their voting on each requirement.

Frequency of planning meeting: The frequency and length of meetings was extensively reduced. There is however need for stakeholders to be able to provide comments for each requirement when they vote on them.

Less effort on planning and more on features: With the number and frequency of meetings reduced, the time saved from these is now channeled and focused on the delivery process of more requirements for the releases. The quality of the product also receives better focus and emphasis.

4.4 Impact on the Product

More focused delivery: During the case study, we found that ReleasePlanner^{TM} provided a good means of communicating the roadmap to customers. The impact of changes to requirements becomes quick to assess and present. More efforts are then devoted to delivering the requirements on the plan.

Time to market of features: There was the perception that there was an improvement in the time to market of features. I found this difficult to evaluate but could be subject of further study.

5 Summary and Conclusions

Determining the most appropriate requirements is one of the hardest and one of the most crucial problems of software development [Lehtola et al. 04]. The complexity of planning and scheduling a release is greatly increased when there are several requirements in the release to be scheduled. A schedule that satisfies all the scheduling constraints is said to be a feasible solution. A general observation is that many companies still rely on ad-hoc approach to prioritize requirements [Lehtola et al. 04] and determine the feasibility of a release. The major finding in this paper is that the use of intelligent decision support system for these scheduling activities removes the major risks and challenges associated with the release scheduling process. This was validated through a case study of real life release planning and scheduling process at Trema Laboratories, Inc.

The main contribution of this paper is the real-world evaluation of the intelligent decision support tool ReleasePlanner^{TM} and the benefits realized by the business from the use of the tool. This study shows that there is significant value added to the business compared to the use of ad-hoc approach to release planning.

Future work will be devoted to implementation, to the conceptual ideas for evaluating operational feasibility into the tool suite and to evaluate the overall performance of the approach EVOLVE^{ext}.

Acknowledgement

The authors would like to thank the Alberta Informatics Circle of Research Excellence (iCORE) for its financial support of this research. Also, thanks to Trema Laboratories, Inc. for providing access to data and other information for the research. The reviewer's comments helped to make the paper more readable.

6 Literature

[Abrahamsson 01]

Abrahamsson P.: "Commitment development in software process improvement: critical misconceptions". 23rd international conference on Software engineering (ICSE2001), Toronto, Canada, pp. 71-80, 2001.

[Bagnall et al. 01]

Bagnall, A. J., Rayward-Smith, V. J., and Whittley, I. M.: "The Next Release Problem", Information and Software Technology, 43 (14), pp. 883-890, 2001.

[Florac et al 97]

Florac W.A., Park R.E., Carleton A.D.: "Practical Software Measurement: Measuring for Process Management and improvement". CMU/SEI-97-HB-003, The Software Engineering Institution, Pittsburgh, 1997.

[Greer & Ruhe 04]

Greer, D. and Ruhe, G.: "Software Release Planning: An Evolutionary and Iterative Approach", Information and Software Technology, 46(4), pp. 243-253, 2004.

[Hoch et al 00]

Hoch, D., Roeding, C., Purkert, G., Lindner, S., Müller, R.: "Secrets of Software Success: Management Insights from 100 Software Firms Around the World", McKinsey & Co., 2000.

[Jurison 99]

Jurison J.: "Software Project Management: The Manager's view". Communications of the Associations for Information Systems, Vol. 2 No. 17, 1999.

[Karlsson et al. 04]

Karlsson L., Berander P., Regnell B., och Wohlin C.: "Requirements Prioritization: An Experiment on Exhaustive Pair-Wise Comparisons versus Planning Game Partitioning". Proceedings of Empirical Assessment in Software Engineering (EASE2004), Edinbourgh, Scotland, 2004.

[Lehtola et al. 04]

Lehtola, L., Kauppinen, M., Kujala, S.: "Requirements Prioritization Challenges in Practice". In Proceedings of 5th International Conference on Product Focused Software Process Improvement, Springer-Verlag, pp. 497-508, 2004.

[Momoh 04]

Momoh, J.A.: "Applying Intelligent Decision Support to Determine Operational Feasibility of Strategic Software Release Plans". MSc Thesis, University of Calgary, Department of Electrical and Computer Engineering, 2004.

[Penny 02]

Penny D. A.: "An Estimation-Based Management Framework for Enhancive Maintenance in Commercial Software Products," In Proceedings of International Conference on Software Maintenance, pp. 122-130, 2002.

[Raffo 93]

Raffo D.: "Evaluating the Impact of Process Improvements Quantitatively using Process Modeling". Proceedings of CASCON'93, pp. 290-313, 1993.

[Ruhe & Ngo-The 04]

Ruhe, G., Ngo-The, A.: "Hybrid Intelligence in Software Release Planning". International Journal of Hybrid Intelligent Systems, Volume 1, No. 2, pp. 99-110, 2004.

[Zahran 98]

Zahran S.: "Software process improvement: practical guidelines for business success". Addison-Wesley Publishing Co., Reading, Mass., 1998.

7 Author CVs

Joseph Momoh

Joseph Momoh is a manager in the Research and Development department of Trema Laboratories Inc. He manages the software projects to develop new releases of the suite of products. He has over 18 years of experience with the management of IT projects and the organization of IT processes. He has led a large number of IT-process assessments and improvement projects. He is also a graduate of the University of Calgary, Canada and holds a masters degree in software engineering.

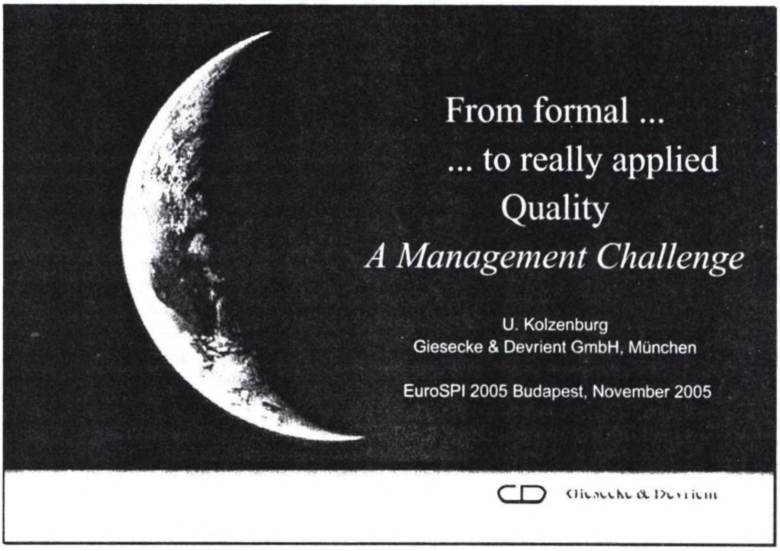
Guenther Ruhe

Dr Ruhe holds an Industrial Research Chair in Software Engineering at University of Calgary and is an iCORE Professor since July 2001. Dr. Ruhe's main results and publications are in software engineering decision support, software release planning, requirements and COTS selection, measurement, simulation and empirical research. From 1996 until 2001 he was deputy director of the Fraunhofer Institute for Experimental Software Engineering. He is the author of two books, several book chapters and more than 120 publications. Dr. Ruhe is a member of the ACM, the IEEE Computer Society and the German Computer Society GI.

From formal to really applied quality - a management challenge

Titel	**From formal to really applied quality - a management challenge**
Referent	***Ulrich Kolzenburg, Director Project- & Processmanagement, Giesecke & Devrient GmbH, München***
Statements	An increasing complexity of the technical solutions drives the increased demand for formal requirements management processes and defined development processes. However, more formalism does not automatically increase the productivity, it could also overload documentation and hide the really important requirements (more documents and less interaction is the wrong way). In practice the success in such areas largely depends on the right balance between really required formalism and communicative iterative and interactive development concepts. To implement this „right balancing" approach support from both, the top management and the development team, is required.
Abstract	In many high tech industries the number of requirements and the size of technical specifications are dramatically increasing. This is driven by the increasing technical complexity, concerning interfaces and, interoperability, and the quality standards (ISO 15504, CMMI, etc.) . All manufacturers now invest into these quality standards to be able to handle this high project management complexity. However, these models very much base on the concept of increased formal models and formal documentation and miss (although possible) the dynamic systems approach. We agree that specification work is important. However, real life projects illustrate that quality processes alone are no guarantee for a project success. For instance, 17000 pages of specification did not help when the road pricing system in Germany has been delayed. Too much formal documentation especially leads to inconsistencies when they get that large (17000 !) that inside the documents there are contradictions, dependencies, etc. Thus it is important to implement communicative iterative and flexible development processes and keep the quality documentation lean by still covering the quality aspects. Giesecke & Devrient (G&D) is the world leading company in the security field and is known for ist high product quality. Products are - Smard Cards im Banking, Mobile phones, where specifications are influenced by many stake holders and many norms of security have to be applied. This paper/presentation will outline how G&D manages this balance between formalism and dynamic development processes.

Slide 1



From formal ...

... to really applied

Quality

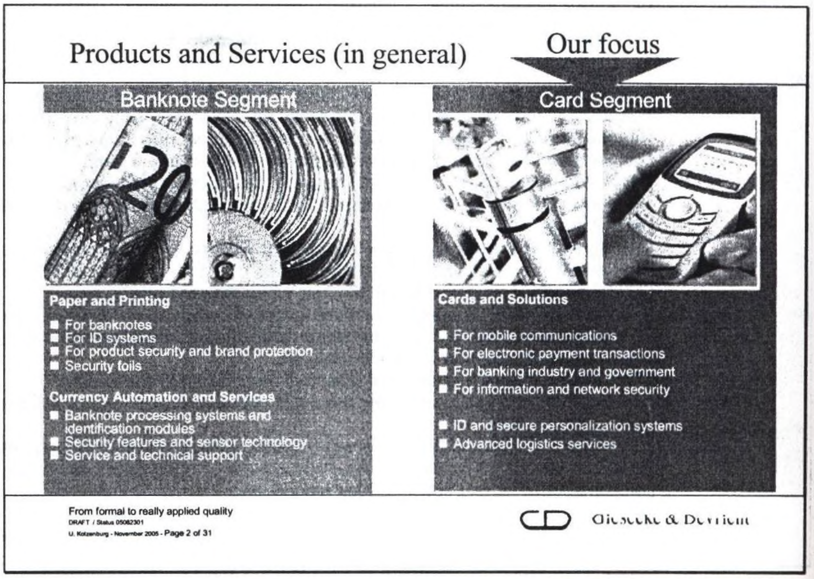
A Management Challenge

U. Kolzenburg

Giesecke & Devrient GmbH, München

EuroSPI 2005 Budapest, November 2005

Slide 2



Products and Services (in general)

Our focus

Banknote Segment

Paper and Printing

- For banknotes
- For ID systems
- For product security and brand protection
- Security foils

Currency Automation and Services

- Banknote processing systems and identification modules
- Security features and sensor technology
- Service and technical support

Card Segment

Cards and Solutions

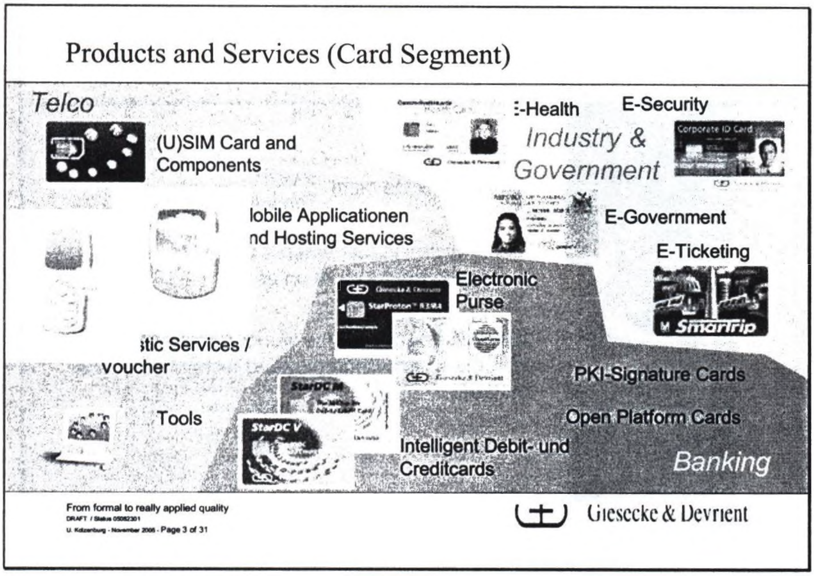
- For mobile communications
- For electronic payment transactions
- For banking industry and government
- For information and network security
- ID and secure personalization systems
- Advanced logistics services

From formal to really applied quality

DRAFT / Status 05082301

U. Kolzenburg - November 2005 - Page 2 of 31

Slide 3



Products and Services (Card Segment)

Telco

(U)SIM Card and Components

lobile Applicationen nd Hosting Services

ɜtic Services / voucher

Tools

:-Health

E-Security

Industry & Government

E-Government

E-Ticketing

Electronic Purse

PKI-Signature Cards

Open Platform Cards

Intelligent Debit- und Creditcards

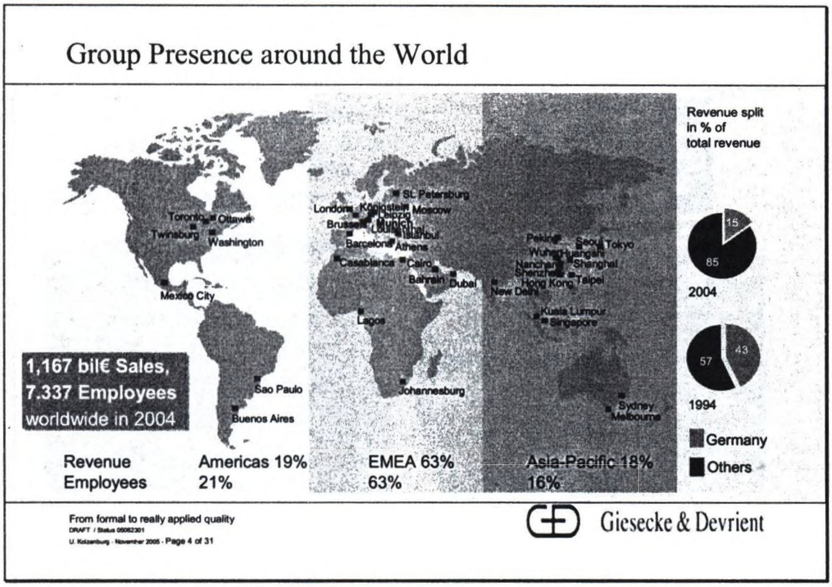
Banking

Corporate ID Card
StarProton
SmarTrip
StarDC M
StarDC V

From formal to really applied quality
DRAFT / Status 05082301
U. Kolzenburg - November 2005 - Page 3 of 31

Giesecke & Devrient

ide 4



Group Presence around the World

1,167 bil€ Sales,
7.337 Employees
worldwide in 2004

Revenue	Americas 19%	EMEA 63%	Asia-Pacific 18%
Employees	21%	63%	16%

Revenue split in % of total revenue

2004

1994

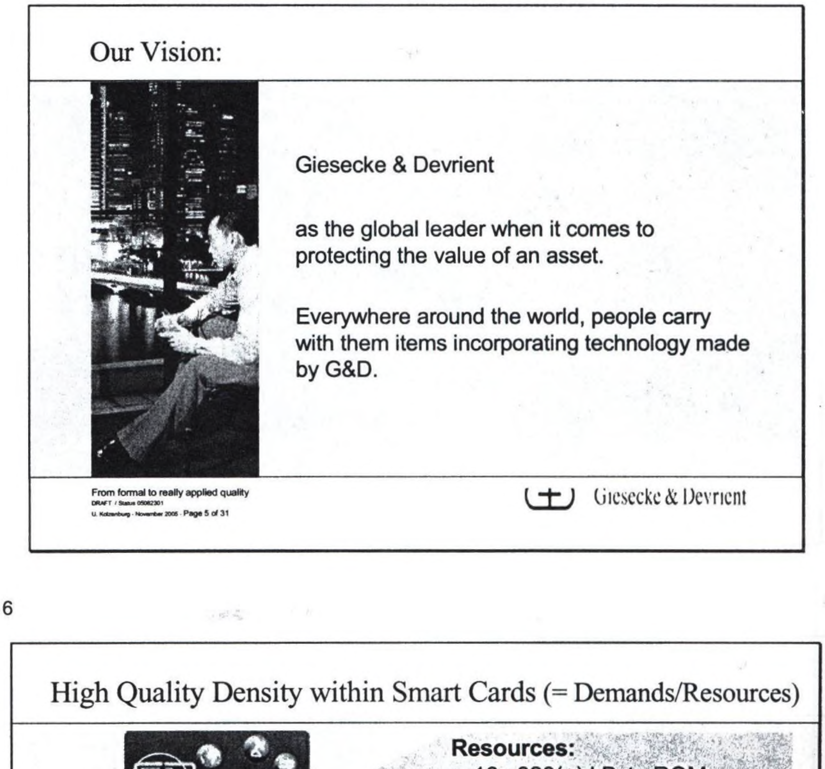
▨ Germany

■ Others

From formal to really applied quality
DRAFT / Status 05082301
U. Kolzenburg - November 2005 - Page 4 of 31

Giesecke & Devrient

Slide 5



Our Vision:

Giesecke & Devrient

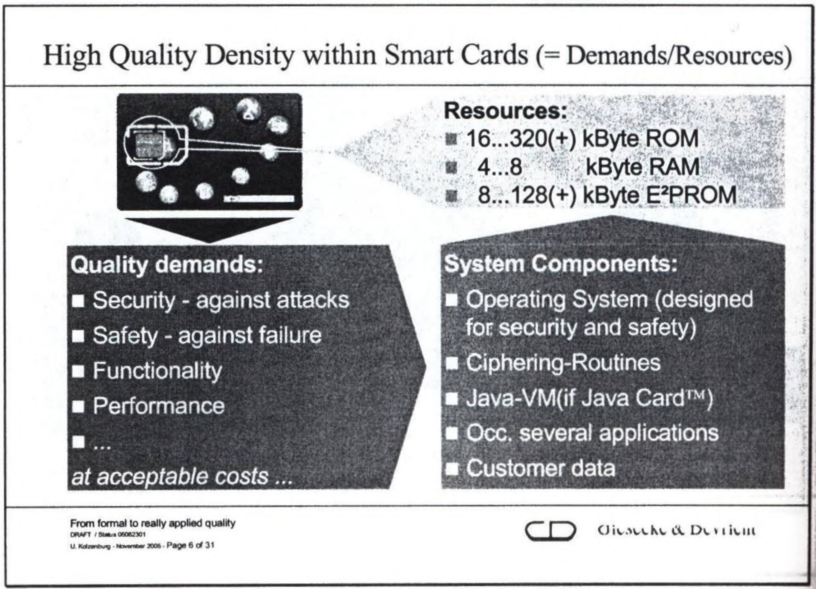
as the global leader when it comes to protecting the value of an asset.

Everywhere around the world, people carry with them items incorporating technology made by G&D.

From formal to really applied quality
DRAFT / Status 05082301
U. Kolzenburg · November 2005 · Page 5 of 31

Giesecke & Devrient

Slide 6



High Quality Density within Smart Cards (= Demands/Resources)

Resources:

- 16...320(+) kByte ROM
- 4...8 kByte RAM
- 8...128(+) kByte E²PROM

Quality demands:

- Security - against attacks
- Safety - against failure
- Functionality
- Performance
- ...

at acceptable costs ...

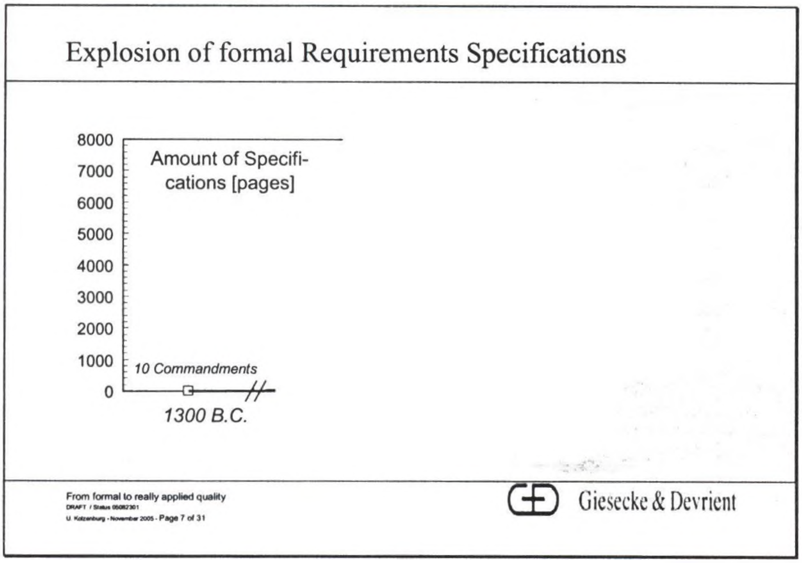
System Components:

- Operating System (designed for security and safety)
- Ciphering-Routines
- Java-VM(if Java Card™)
- Occ. several applications
- Customer data

From formal to really applied quality
DRAFT / Status 05082301
U. Kolzenburg · November 2005 · Page 6 of 31

Giesecke & Devrient

Slide 7

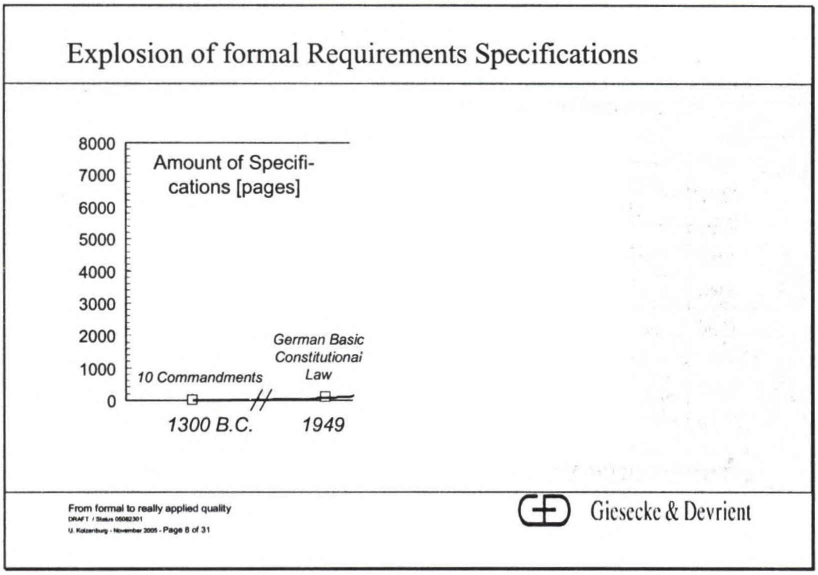


Explosion of formal Requirements Specifications

From formal to really applied quality
DRAFT / Status 05082301
U. Kolzenburg - November 2005 - Page 7 of 31

Giesecke & Devrient

Slide 8

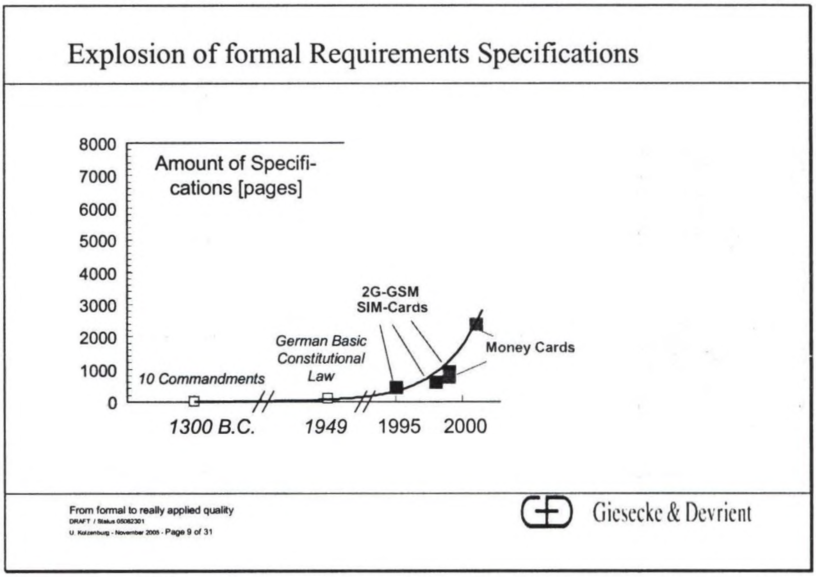


Explosion of formal Requirements Specifications

From formal to really applied quality
DRAFT / Status 05082301
U. Kolzenburg - November 2005 - Page 8 of 31

Giesecke & Devrient

Slide 9

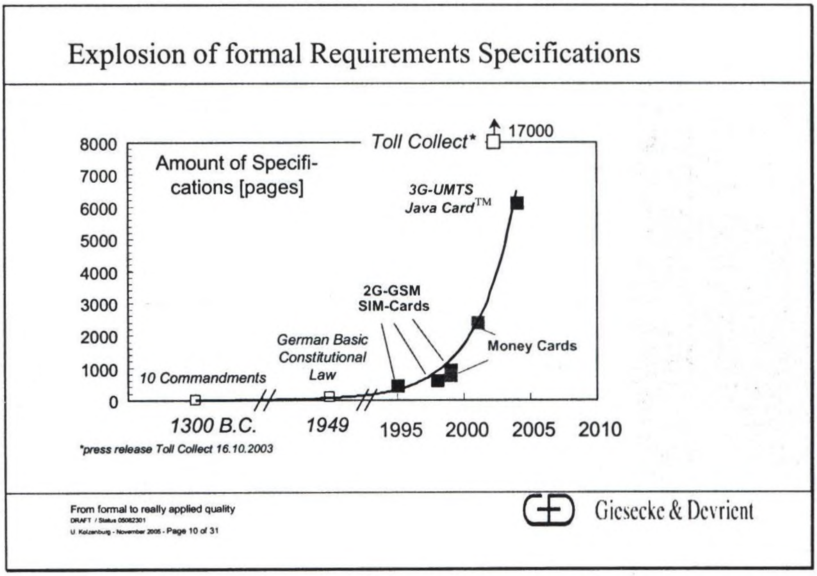


Explosion of formal Requirements Specifications

From formal to really applied quality
DRAFT / Status 05082301
U. Kolzenburg - November 2005 - Page 9 of 31

Giesecke & Devrient

Slide 10



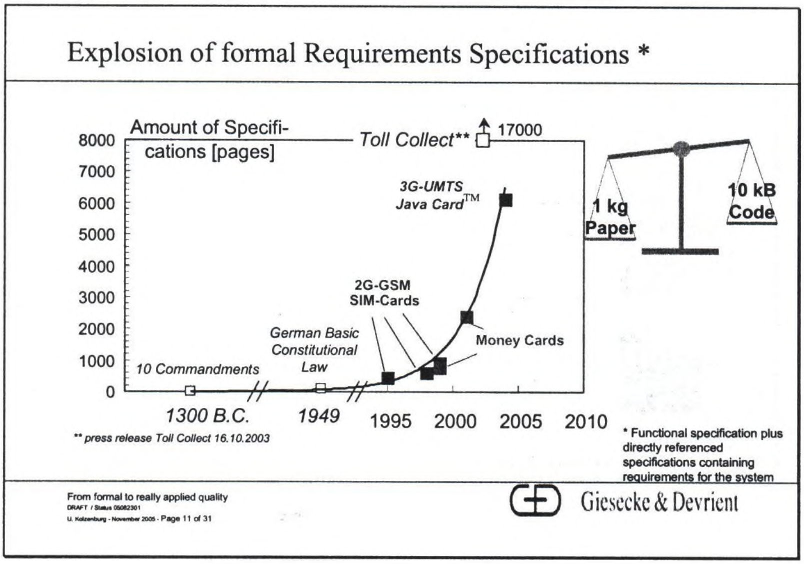
Explosion of formal Requirements Specifications

*press release Toll Collect 16.10.2003

From formal to really applied quality
DRAFT / Status 05082301
U. Kolzenburg - November 2005 - Page 10 of 31

Giesecke & Devrient

Slide 11



Explosion of formal Requirements Specifications *

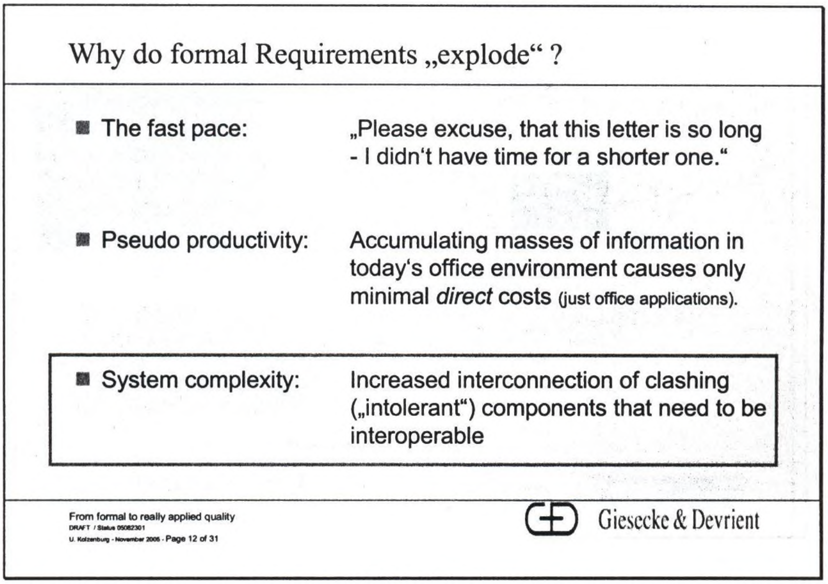
** press release Toll Collect 16.10.2003

* Functional specification plus directly referenced specifications containing requirements for the system

From formal to really applied quality
DRAFT / Status 05082301
U. Kolzenburg - November 2005 - Page 11 of 31

Giesecke & Devrient

Slide 12



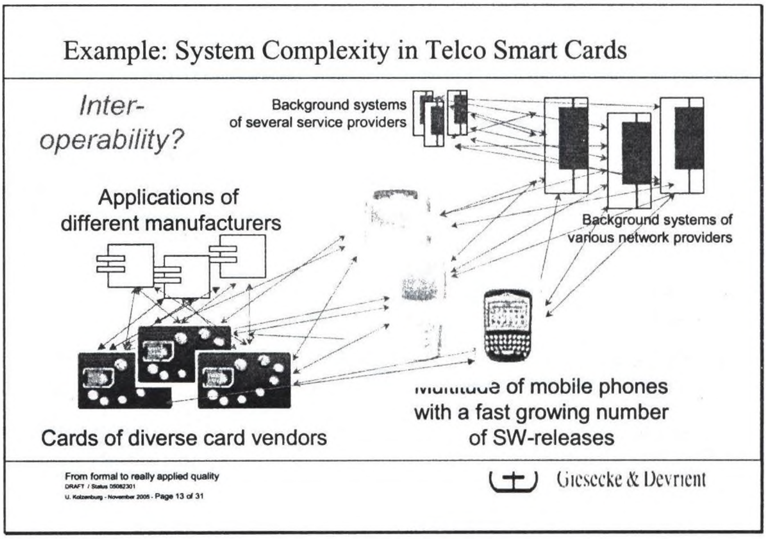
Why do formal Requirements „explode“ ?

- The fast pace: „Please excuse, that this letter is so long - I didn‘t have time for a shorter one.“
- Pseudo productivity: Accumulating masses of information in today‘s office environment causes only minimal *direct* costs (just office applications).
- System complexity: Increased interconnection of clashing („intolerant“) components that need to be interoperable

From formal to really applied quality
DRAFT / Status 05082301
U. Kolzenburg - November 2005 - Page 12 of 31

Giesecke & Devrient

Slide 13



Example: System Complexity in Telco Smart Cards

Inter-operability?

Background systems of several service providers

Applications of different manufacturers

Background systems of various network providers

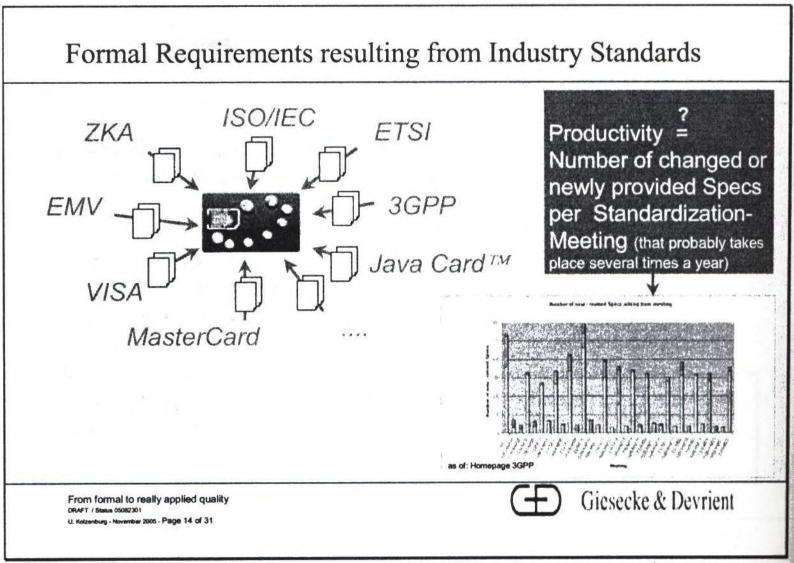
Cards of diverse card vendors

Multitude of mobile phones with a fast growing number of SW-releases

From formal to really applied quality
DRAFT / Status 05082301
U. Kolzenburg - November 2005 - Page 13 of 31

Giesecke & Devrient

Slide 14



Formal Requirements resulting from Industry Standards

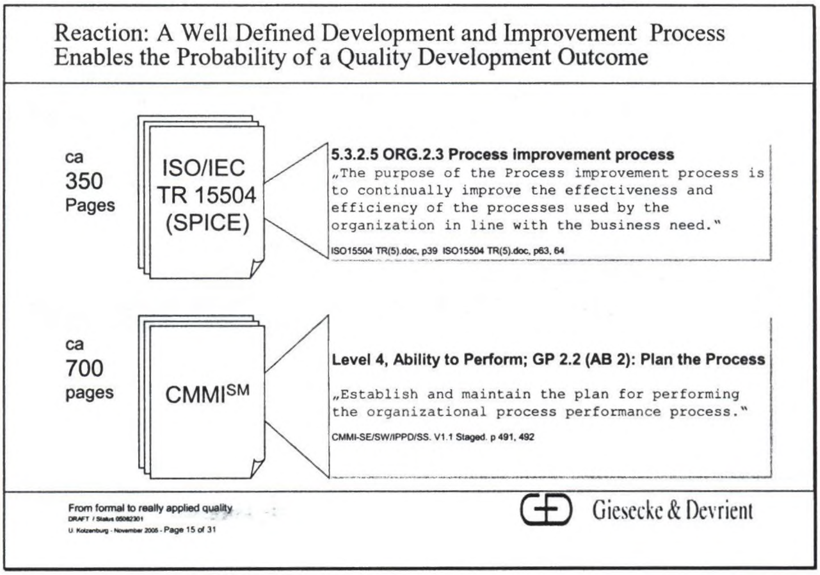
Productivity $\overset{?}{=}$ Number of changed or newly provided Specs per Standardization-Meeting (that probably takes place several times a year)

as of: Homepage 3GPP

From formal to really applied quality
DRAFT / Status 05082301
U. Kolzenburg - November 2005 - Page 14 of 31

Giesecke & Devrient

Slide 15



Reaction: A Well Defined Development and Improvement Process Enables the Probability of a Quality Development Outcome

ca
350
Pages

ISO/IEC
TR 15504
(SPICE)

5.3.2.5 ORG.2.3 Process improvement process

„The purpose of the Process improvement process is to continually improve the effectiveness and efficiency of the processes used by the organization in line with the business need."

ISO15504 TR(5).doc, p39 ISO15504 TR(5).doc, p63, 64

ca
700
pages

CMMI^{SM}

Level 4, Ability to Perform; GP 2.2 (AB 2): Plan the Process

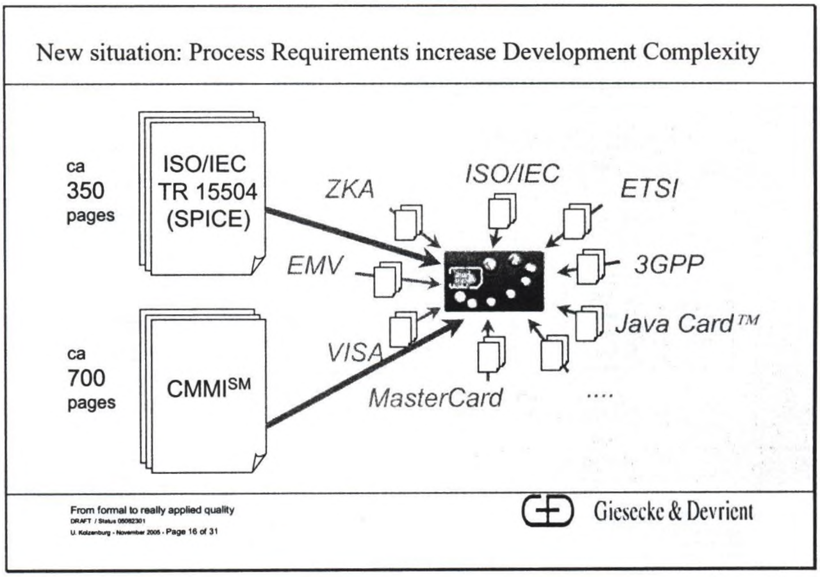
„Establish and maintain the plan for performing the organizational process performance process."

CMMI-SE/SW/IPPD/SS. V1.1 Staged. p 491, 492

From formal to really applied quality
DRAFT / Status 05082301
U. Kolzenburg - November 2005 - Page 15 of 31

Giesecke & Devrient

Slide 16



New situation: Process Requirements increase Development Complexity

ca
350
pages

ISO/IEC
TR 15504
(SPICE)

ca
700
pages

CMMI^{SM}

ZKA

ISO/IEC

ETSI

EMV

3GPP

VISA

Java Card^{TM}

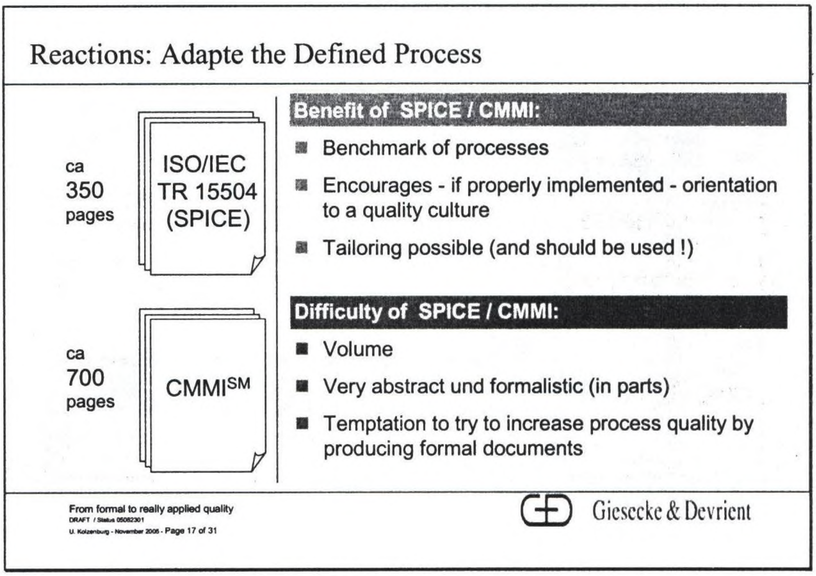
MasterCard

....

From formal to really applied quality
DRAFT / Status 05082301
U. Kolzenburg - November 2005 - Page 16 of 31

Giesecke & Devrient

Slide 17



Reactions: Adapte the Defined Process

ca
350
pages

ISO/IEC
TR 15504
(SPICE)

ca
700
pages

CMMI^{SM}

Benefit of SPICE / CMMI:

- Benchmark of processes
- Encourages - if properly implemented - orientation to a quality culture
- Tailoring possible (and should be used !)

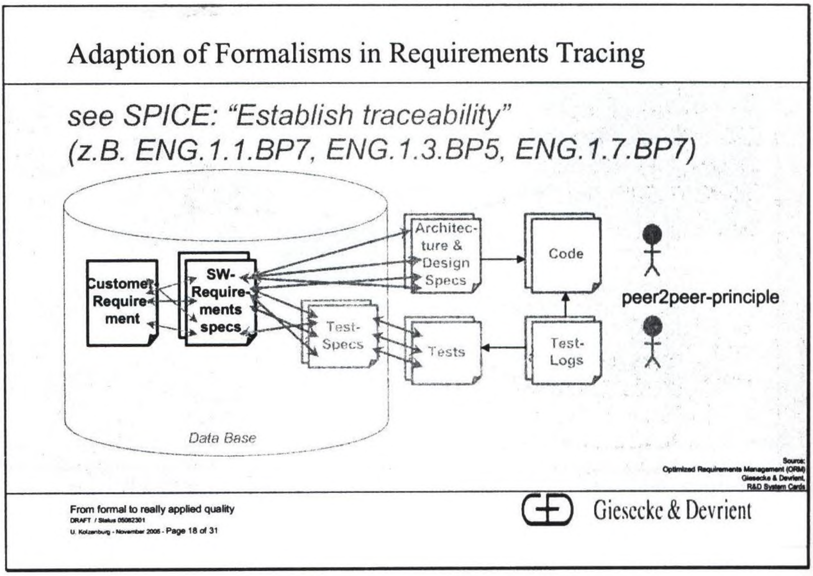
Difficulty of SPICE / CMMI:

- Volume
- Very abstract und formalistic (in parts)
- Temptation to try to increase process quality by producing formal documents

From formal to really applied quality
DRAFT / Status 05082301
U. Kolzenburg - November 2005 - Page 17 of 31

Giesecke & Devrient

Slide 18



Adaption of Formalisms in Requirements Tracing

see SPICE: "Establish traceability"
(z.B. ENG.1.1.BP7, ENG.1.3.BP5, ENG.1.7.BP7)

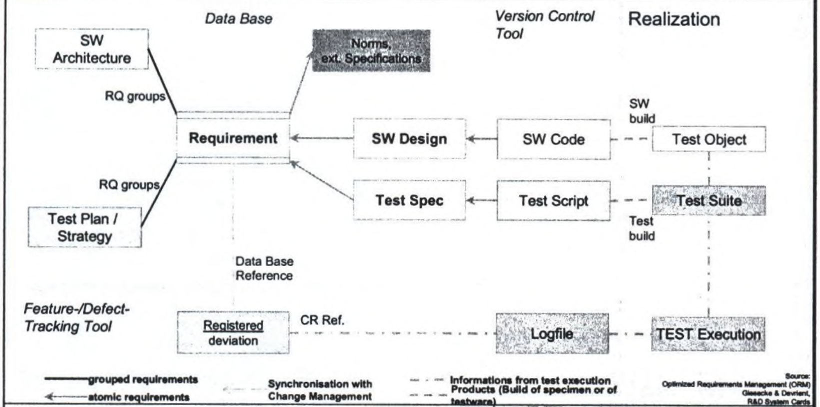
Source:
Optimized Requirements Management (ORM)
Giesecke & Devrient,
R&D System Cards

From formal to really applied quality
DRAFT / Status 05082301
U. Kolzenburg - November 2005 - Page 18 of 31

Giesecke & Devrient

Slide 19

Realization of Formalisms in Requirements Tracing



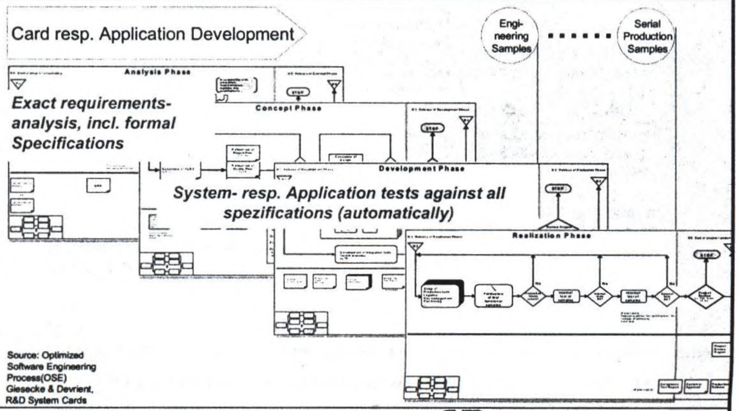
Source: Optimized Requirements Management (ORM) Giesecke & Devrient, R&D System Cards

From formal to really applied quality
DRAFT / Status 05082301
U. Kolzenburg - November 2005 - Page 19 of 31

Giesecke & Devrient

Slide 20

Essential: Systematic SW development proces



Exact requirements-analysis, incl. formal Specifications

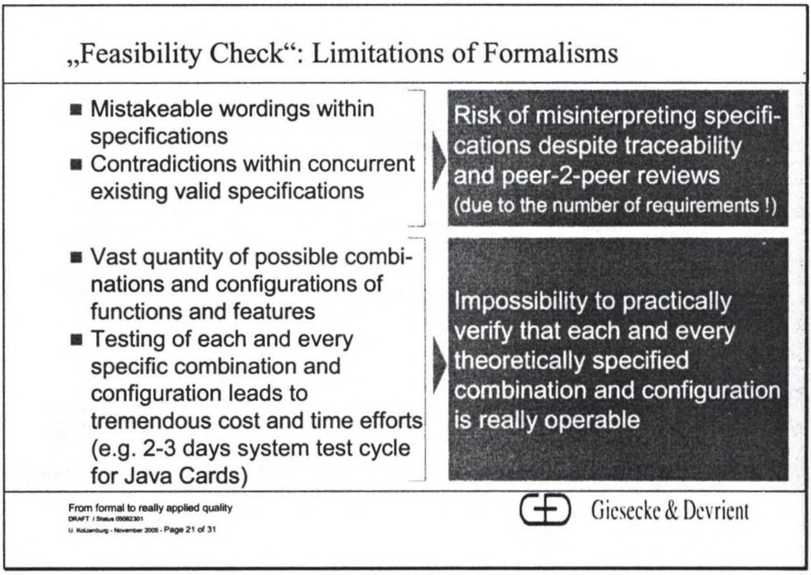
System- resp. Application tests against all spezifications (automatically)

Source: Optimized Software Engineering Process(OSE) Giesecke & Devrient, R&D System Cards

From formal to really applied quality
DRAFT / Status 05082301
U. Kolzenburg - November 2005 - Page 20 of 31

Giesecke & Devrient

Slide 21



„Feasibility Check“: Limitations of Formalisms

- Mistakeable wordings within specifications
- Contradictions within concurrent existing valid specifications

Risk of misinterpreting specifications despite traceability and peer-2-peer reviews (due to the number of requirements !)

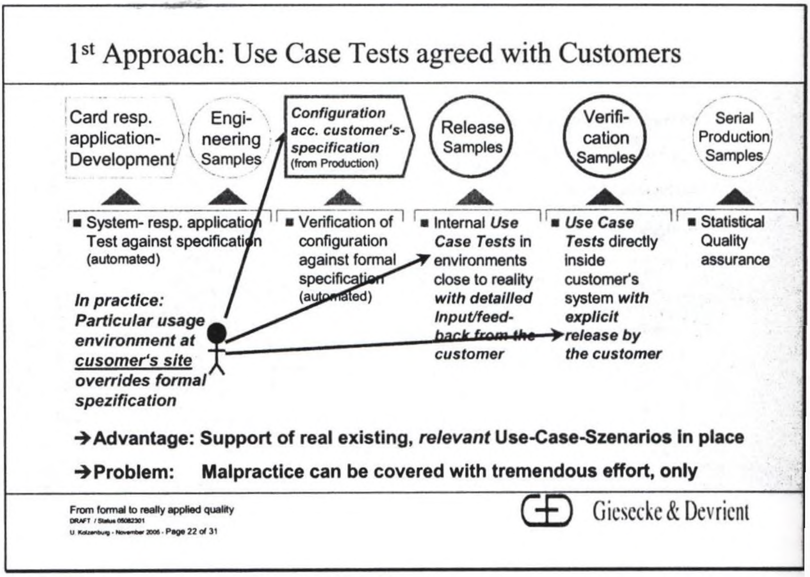
- Vast quantity of possible combinations and configurations of functions and features
- Testing of each and every specific combination and configuration leads to tremendous cost and time efforts (e.g. 2-3 days system test cycle for Java Cards)

Impossibility to practically verify that each and every theoretically specified combination and configuration is really operable

From formal to really applied quality
DRAFT / Status 05082301
U. Kolzenburg - November 2005 - Page 21 of 31

Giesecke & Devrient

Slide 22



1st Approach: Use Case Tests agreed with Customers

Card resp. application-Development

Engineering Samples

Configuration acc. customer's-specification (from Production)

Release Samples

Verification Samples

Serial Production Samples

- System- resp. application Test against specification (automated)

- Verification of configuration against formal specification (automated)

- Internal *Use Case Tests* in environments close to reality *with detailled Input/feed-back from the customer*

- *Use Case Tests* directly inside customer's system *with explicit release by the customer*

- Statistical Quality assurance

In practice: Particular usage environment at <u>cusomer's site</u> overrides formal spezification

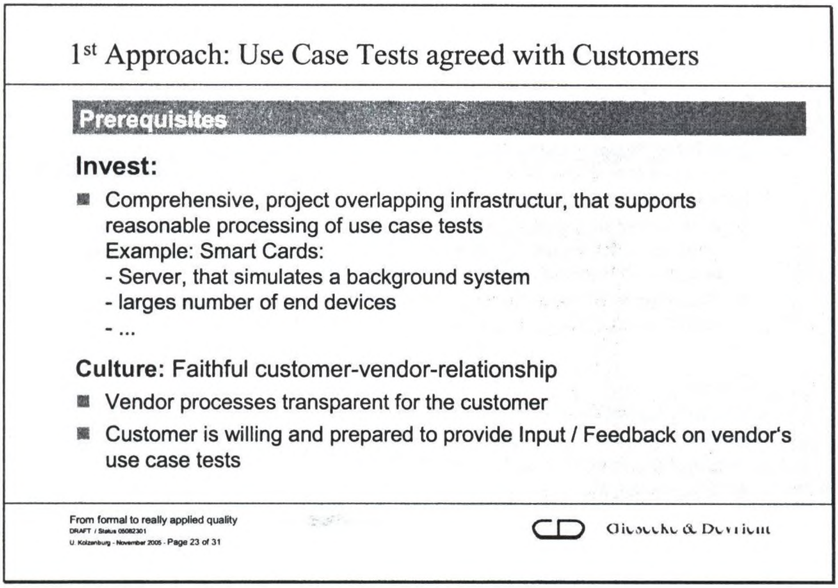
→**Advantage: Support of real existing, *relevant* Use-Case-Szenarios in place**

→**Problem: Malpractice can be covered with tremendous effort, only**

From formal to really applied quality
DRAFT / Status 05082301
U. Kolzenburg - November 2005 - Page 22 of 31

Giesecke & Devrient

Slide 23



1st Approach: Use Case Tests agreed with Customers

Prerequisites

Invest:

- Comprehensive, project overlapping infrastructur, that supports reasonable processing of use case tests
  Example: Smart Cards:
  - Server, that simulates a background system
  - larges number of end devices
  - ...

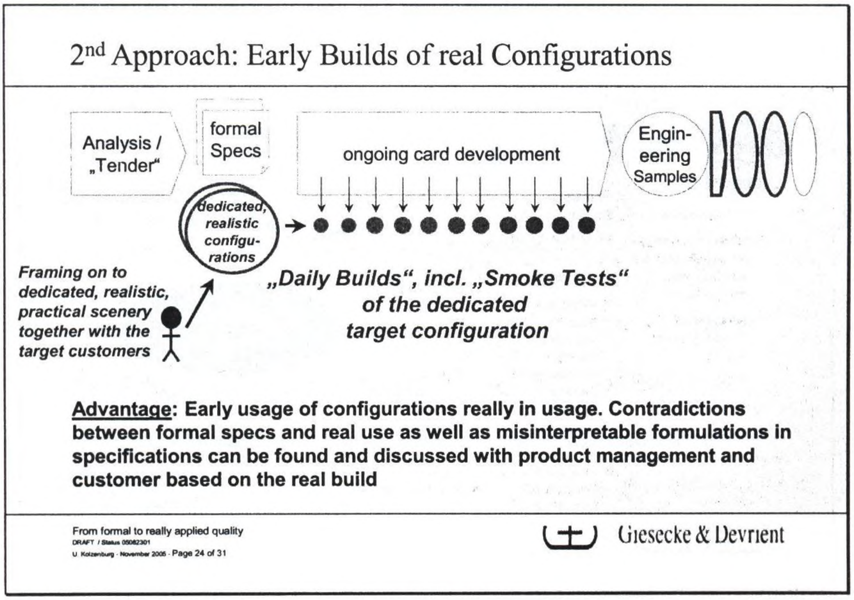
Culture: Faithful customer-vendor-relationship

- Vendor processes transparent for the customer
- Customer is willing and prepared to provide Input / Feedback on vendor's use case tests

From formal to really applied quality
DRAFT / Status 05082301
U. Kolzenburg - November 2005 - Page 23 of 31

Giesecke & Devrient

Slide 24



2nd Approach: Early Builds of real Configurations

Analysis / „Tender"

formal Specs

ongoing card development

Engin-eering Samples

dedicated, realistic configu-rations

Framing on to dedicated, realistic, practical scenery together with the target customers

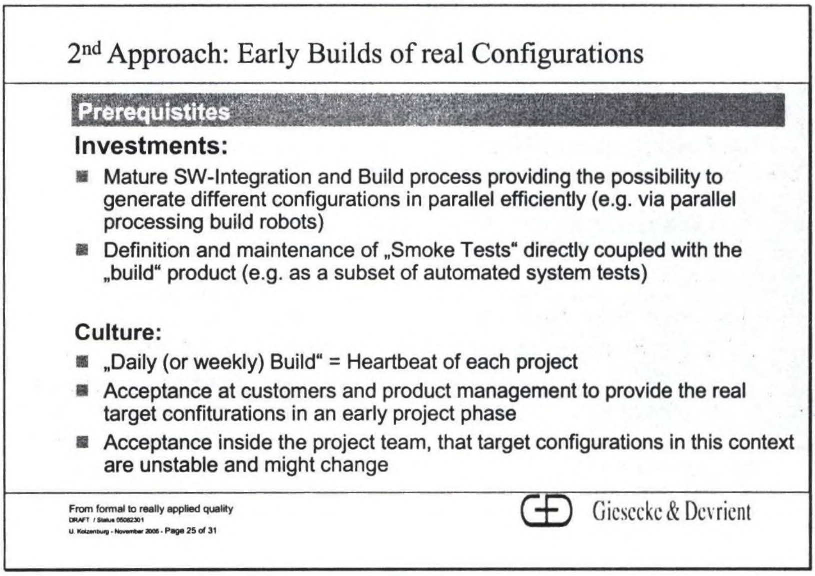
„Daily Builds", incl. „Smoke Tests" of the dedicated target configuration

<u>Advantage</u>: Early usage of configurations really in usage. Contradictions between formal specs and real use as well as misinterpretable formulations in specifications can be found and discussed with product management and customer based on the real build

From formal to really applied quality
DRAFT / Status 05082301
U. Kolzenburg - November 2005 - Page 24 of 31

Giesecke & Devrient

Slide 25



2nd Approach: Early Builds of real Configurations

Prerequistites

Investments:

- Mature SW-Integration and Build process providing the possibility to generate different configurations in parallel efficiently (e.g. via parallel processing build robots)
- Definition and maintenance of „Smoke Tests" directly coupled with the „build" product (e.g. as a subset of automated system tests)

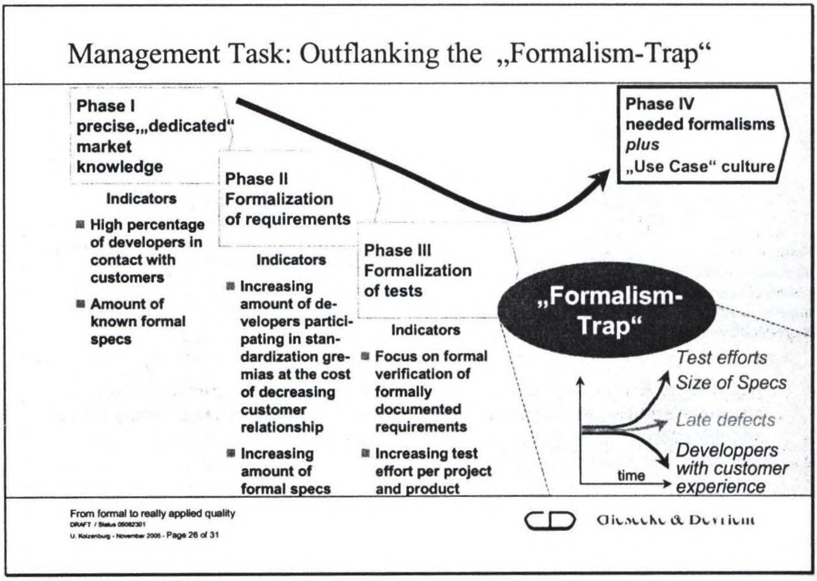
Culture:

- „Daily (or weekly) Build" = Heartbeat of each project
- Acceptance at customers and product management to provide the real target confiturations in an early project phase
- Acceptance inside the project team, that target configurations in this context are unstable and might change

From formal to really applied quality
DRAFT / Status 05082301
U. Koizenburg - November 2005 - Page 25 of 31

Giesecke & Devrient

Slide 26



Management Task: Outflanking the „Formalism-Trap"

Phase I
precise„dedicated" market knowledge

Indicators

- **High percentage of developers in contact with customers**
- **Amount of known formal specs**

Phase II
Formalization of requirements

Indicators

- **Increasing amount of developers participating in standardization gremias at the cost of decreasing customer relationship**
- **Increasing amount of formal specs**

Phase III
Formalization of tests

Indicators

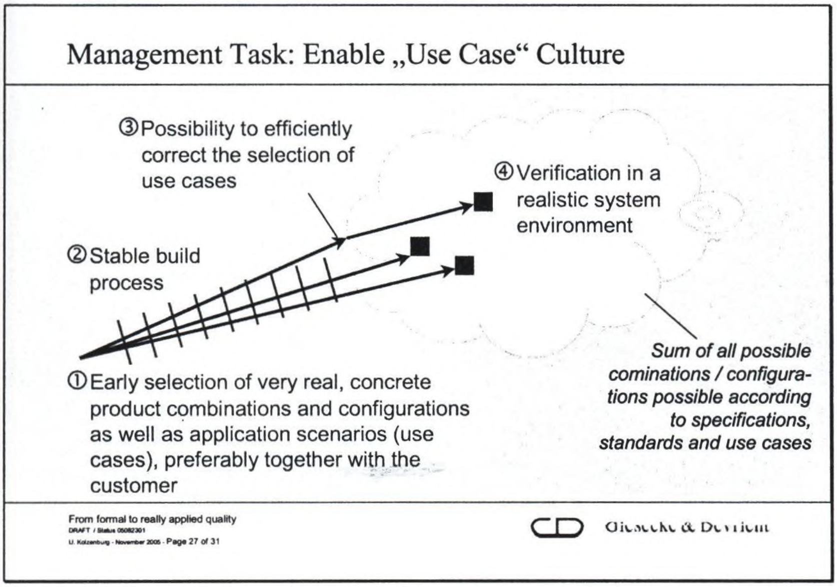
- **Focus on formal verification of formally documented requirements**
- **Increasing test effort per project and product**

Phase IV
needed formalisms
plus
„Use Case" culture

From formal to really applied quality
DRAFT / Status 05082301
U. Koizenburg - November 2005 - Page 26 of 31

Giesecke & Devrient

Slide 27



Management Task: Enable „Use Case“ Culture

③Possibility to efficiently correct the selection of use cases

④Verification in a realistic system environment

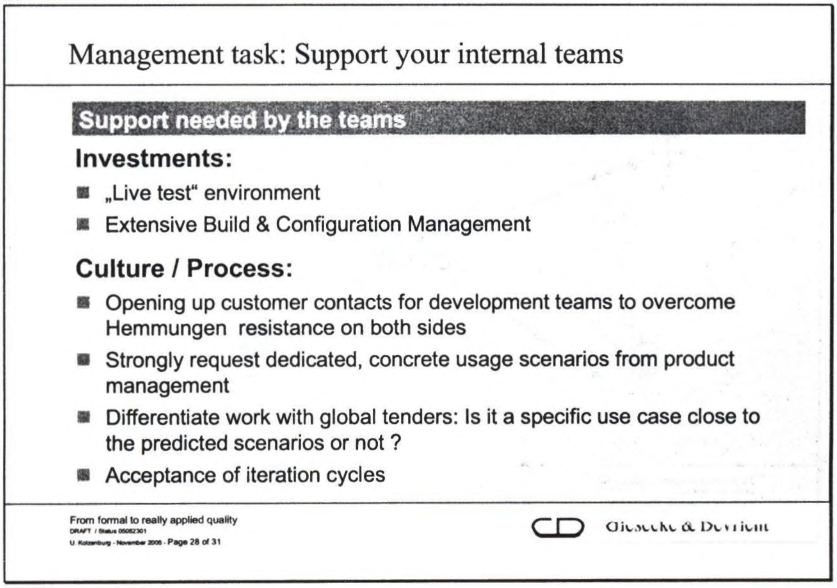
②Stable build process

Sum of all possible cominations / configurations possible according to specifications, standards and use cases

①Early selection of very real, concrete product combinations and configurations as well as application scenarios (use cases), preferably together with the customer

From formal to really applied quality
DRAFT / Status 05082301
U. Kolzenburg - November 2005 - Page 27 of 31

Slide 28



Management task: Support your internal teams

Support needed by the teams

Investments:

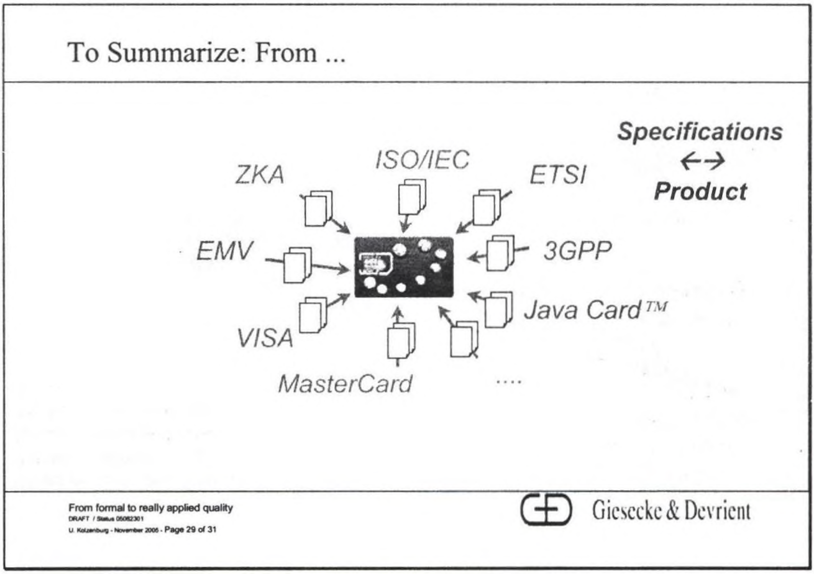
- „Live test“ environment
- Extensive Build & Configuration Management

Culture / Process:

- Opening up customer contacts for development teams to overcome Hemmungen resistance on both sides
- Strongly request dedicated, concrete usage scenarios from product management
- Differentiate work with global tenders: Is it a specific use case close to the predicted scenarios or not ?
- Acceptance of iteration cycles

From formal to really applied quality
DRAFT / Status 05082301
U. Kolzenburg - November 2005 - Page 28 of 31

Slide 29



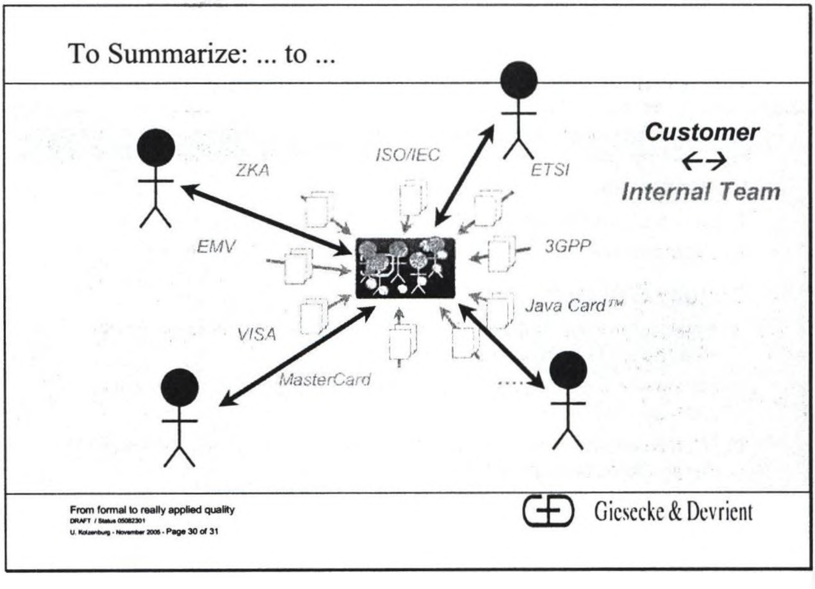
To Summarize: From ...

Specifications ←→ Product

From formal to really applied quality
DRAFT / Status 05082301
U. Kolzenburg - November 2005 - Page 29 of 31

Giesecke & Devrient

Slide 30



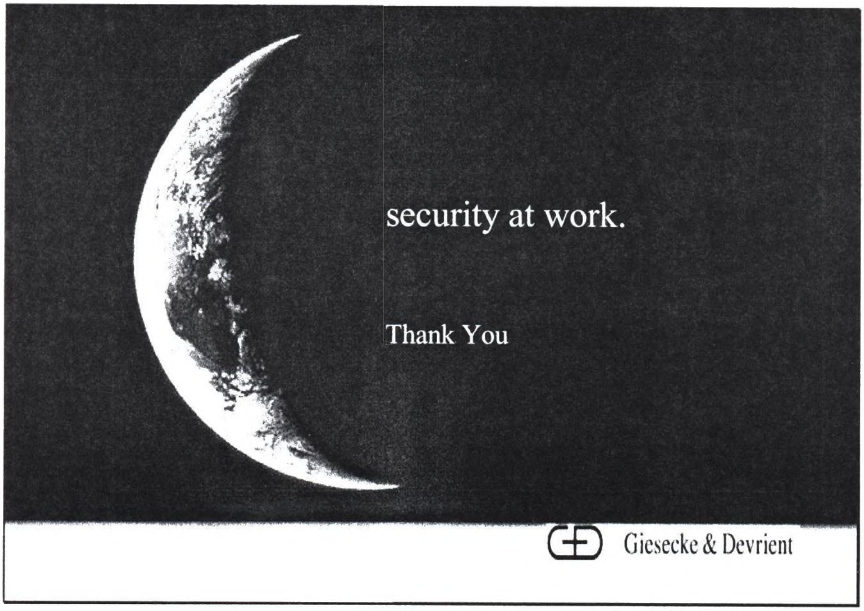
To Summarize: ... to ...

Customer ←→ Internal Team

From formal to really applied quality
DRAFT / Status 05082301
U. Kolzenburg - November 2005 - Page 30 of 31

Giesecke & Devrient

Slide 31



security at work.

Thank You

Giesecke & Devrient

A Method for Modelling a Mature Process

Justin Kelleher,
University of Cape Town,
Cape Town
jkr@cs.uct.ac.za

Abstract

This paper describes the creation and assessment of a software traceability process. The project is part of a larger research project at the University of Cape Town. The purpose of this paper is threefold. Firstly, we describe a method to model a software process. Secondly we describe a method for assessing the capability of this process using the ISO 15504 standard. Thirdly, we compare the process capabilities of our process to that of the Rational Unified Process.

We describe the modelling of a process metamodel called **TRA**ceability **M**etamodel (**TRAM)**. TRAM provides a language for the definition of the elements of the **TRA**ceability **P**rocess (**TRAP**). The language used is based on the UML Software Process Engineering Metamodel (SPEM) specification which is defined by the Object Management Group. The goal of TRAP is to describe the implementation of software traceability across the product lifecycle. We describe the TRAP process which is a synthesis of best practices collected and developed in the course of the research programs activities which incorporates the best available requirements traceability techniques for telecommunications software projects.

We describe how we used the ISO 15504 framework in the assessment of the TRAP process. While ISO 15504 delineates a list of activities that should occur it does not to stipulate the order in which such activities should be carried out. This paper therefore proposes a process for modelling and assessing any software process.

We conclude with a comparison of the capabilities of the TRAP process to the capabilities of the Rational Unified Process.

Keywords

Traceability, ISO 15504, Rational Unified Process

1 Introduction

The goal of software engineering is "to build a software product or to enhance an existing one" [18]. It is the disciplined approach and application of engineering, science, and mathematical principles, methods and tools to the economical production of quality software [6]. The IEEE defines a process as "a sequence of steps performed for a given purpose" [7] The Software Engineering Institute [12] states that "An essential aspect of software engineering is the discipline it requires for a group of people to work together cooperatively to solve a common problem. Defined processes set the bounds for each person's roles and responsibilities so that the collaboration is a successful and efficient one" Rational defines a process as "a set of partially ordered steps intended to reach a goal" [10]. In this paper, we introduce the **TRAP** (**TRA**ceability **P**rocess) which describes the roles and work activities for the implementation of software traceability across the product development lifecycle.

Lee Osterweil wrote in 1987: "Software processes are software, too." [15]. This notion has become accurate given that over the past two decades, visual modelling has developed as an essential discipline in software engineering.

A metamodel is a precise definition of the constructs and rules needed for creating semantic model elements at a high level of abstraction. Current literature describes a metamodel as an architectural blueprint. The process metamodel developed in this project are abstract descriptions of the implementation of a software process which has traceability as a core best practice. The Unified Modelling Language (UML) is an Object Management Group (OMG) standard used for creating a metamodel(s) of software artefacts and processes. We combine the UML and the Software Process Engineering Metamodel (SPEM) specification which is also defined by Object Management Group (OMG) to describe the TRAM and the RUP metamodel. The SPEM describes a metamodel "as a concrete software development process or a family of related software development processes" [16]. To engage in successful product development, businesses must rigorously assess their processes. The ISO 15504 standard, formly known as SPICE (Software Process Improvement and Capability dEtermination) is a "framework for the assessment of software processes" [9] and was developed by the International Organization for Standardization (ISO). The TRAP software process capability describes the range of expected results that can be achieved by following the process. We propose a capability for the TRAP and the RUP processes. We then proceed to analyse the capability of the process against the ISO target process profile with the aim of identifying which of the processes has the better capability. The basic hypothesis of this paper is: "can we model software processes using UML and SPEM, create a process from the models and assess the capability of the process using the ISO 15504 standard?" In summary the purpose of this paper is to describe an effective process modelling and assessment approach for the software traceability domain.

2 Motivation For TRAceability Process (TRAP)

Analysing the current body of literature on software traceability, Marco Leon [14] noted that traceability usage is rare, stating, "It is a very valuable but seldom used technique in today's development processes. Traceability analysis is rarer still in the internet development industry, where it is even more essential" Scott Ambler (1999), states "It's rare to find a software project team that can honestly claim full requirements traceability throughout a project, especially if the team uses object-orientated technology"

The IEEE has defined traceability as the identification and documentation of derivation paths (upward) and allocation or flow down paths (downward) of work products in the work product hierarchy [8] Gotel and Finkelstein have described traceability (1994) [5] as "The ability to describe and follow the life of a requirement, in both a forward and backward direction, i.e. from its origins, through its development and specification, to its subsequent deployment and use, and through periods of on-going refinement and iteration in any of these phases". Edwards and Howell (1991) [4] defined traceability as well, commenting that it is a technique used to "provide a relationship between the requirements, the design, and the final implementation of the system". Palmer (1997) [17] has noted that "traceability gives essential assistance in understanding the relationships that exist within and across software requirements, design and implementation". These relationships allow designers to demonstrate that the design meets the requirements as well as aiding as aiding in early recognition of those requirements not satisfied by the design. Palmer states that traceability sets out to show "how and why system development products satisfy stakeholder requirements".

Two of the problems motivating the creation of a traceability process are, firstly there is no process in existence which focuses on software traceability across the entire product lifecycle and secondly many of the standards that mandate traceability do not provide a comprehensive guide explaining how to implement this best practice. For example, the standards governing the development of systems for the U.S. Government (e.g., MIL-STD-2167-A and MIL-STD-498 which replaces it (DoD, 1988)) require the development of requirements traceability documents, but don't mandate how to do so. Overall, the practices and usefulness of traceability vary considerably across systems development efforts, ranging from very simplistic practices aimed simply at satisfying the mandates to very comprehensive traceability schemes used as an important tool for managing the systems development process.

TRAP is a process which can be adapted for any project. It describes the work products (traceability items or artefacts), the roles involved in creating the work products and their traceability responsibilities. The TRAP contains workflows conveying the development time for the product as a sequence, the traceability best practices and traceability guidelines, traceability patterns, and the range of traceability tools for seamless implementation of traceability in an organisation. The backbone to this process is the list of traceability items and their corresponding traceability matrixes.

3 *The Research Inputs*

3.1 Rational Unified Process (RUP)

The Rational Unified Process provides a process that can be customized to any software development organization's needs. A major characteristic of RUP is that it provides a disciplined approach to assigning tasks and responsibilities. This characteristic of the process proved to be very useful when we were assigning traceability tasks to the different roles. We decided to use RUP for two reasons. Firstly, it is a process framework with many traceability activities defined which is supported by an integrated tool suite. RequisitePro is a dedicated requirement management tool integrated with supporting tools and process workflows. The desired result is an easy-to-use process. Secondly, the RUP process model could be configured and adapted to satisfy our customers needs.

3.2 Software Process Engineering Metamodel (SPEM)

SPEM is object-oriented specification which describes how to model a software process. UML is used as the notation. The SPEM is a metamodel for defining processes and their components. A tool based on SPEM is a tool for process authoring and customization.

3.3 Architecture of integrated Information Systems (ARIS)

The ARIS Toolset integrates new and existing modelling methods for modelling processes and providing the functionality for creating and evaluating our modelled processes. The ARIS architecture is the basis for the ARIS Toolset. [20] It also serves as an orientation framework for complex development projects due to the fact that in its structuring elements it contains an implicit procedural model for the development of integrated information systems. The result is a highly complex UML metamodel, integrating the view of processes, knowledge processing, organisational structures and information systems. We evaluated a number of commercial process modelling tools but accepted the ARIS Toolset because it has the following process modelling features:

- Object Process Modelling: Represent the static, structural and data-related aspects of a process.
- Dynamic Process Modelling: Illustrate the software development lifecycle in both time and behaviour. The sequence of operations is described by mapping the sequences of events.
- Functional Process Modelling: Clarify the transient and functional aspects of the process i.e. roles mapped to responsibilities.

3.4 Spice: ISO 15504

Several models of varying quality were studied: McCall [13], Boehm [1], FURPS , ISO 9126 [19], Dromey [3] , ISO 15504 and CMM [11] with the intention of identifying those which possess aspects deemed to be important in a Systems Quality Model. James A. McCall [13] described the problems encountered when defining software quality and the best technique for establishing a framework for the measurement of software quality.

The ISO 15504 document suite has a set of categories in which the assessors can place the data that they collect during their assessment. The result is that the assessors can give an overall determination of the organisation's capabilities, which in this project is the capability to implement traceability in the product lifecycle.

We selected ISO 15504 for the following reasons:

- It encourages self-assessment.
- It produces a set of process ratings (a process profile) rather than a pass/fail result. This is essential when comparing two processes.
- It addresses the adequacy of the management of the assessed processes;
- It takes into account the context in which the assessed processes operate;
- It is appropriate across all application domains and sizes of organization.

One of the main reasons we used the ISO 15504 was because of its international recognition and acceptance as a process standard. ISO 15504 does not conflict with social, cultural or legislative expectations and requirements. The actual standards document for ISO 15504 is divided into 9 parts [9]. The ISO 15504 framework defines the process practices for software engineering organisations as well as the measurement criteria to determine process capability. It assists the software development organisation in planning, managing, monitoring, controlling and improving the acquisition, supply, development, operation, evolution and support of software. We utilize the framework by assessing the process capability of TRAP and RUP and comparing the results.

4 The Process Modelling Method

In this section we describe the method we followed to model and assess the TRAP and the RUP processes. Figure 1 illustrates the methodology that we followed.

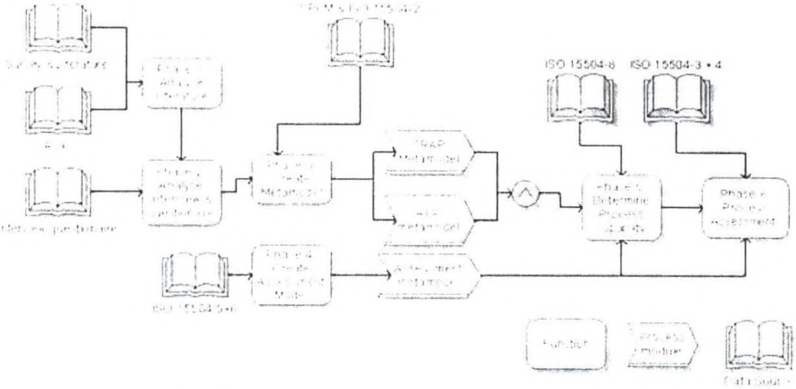


Figure 1: The Methodology

Table 1 describes briefly each phase, the inputs to each phase and the related outputs. During Phase 1 and 2 we created an encyclopaedia of software traceability practices. We designed and implemented a random survey by sending out questionnaires and interviewing experts in the telecommunications industry. The survey gave the process team a broader picture of the traceability practices being performed. During the interviews, the participants were asked to describe their localised traceability practices with a special focus on the relationships between the work products produced.

Development Stage	Data Inputs	Outputs
Phase 1 Review Literature	ACM, IEEE, Academic research focus groups, experience documentation, ISO, Carnegie Mellon etc	Repository of reusable literature or best practices for TRAP
Phase 2 Interview Questionnaires	Review the interview and questionnaire from focus groups on process modelling and software traceability	Identify critical traceability problems and create document list for TRAP process blueprint
Phase 3 Create TRAP RUP Metamodel	The RUP TRAP process metamodel ISO TR 15504 Software Process Engineering Metamodel (SPEM)	RUP, and TRAP metamodel (work products, activities, guidelines, work flows, best practices)
Phase 4 Create Assessment Model	ISO 15504 Software Process Engineering Metamodel (SPEM)	TRAP and RUP process capability and maturity assessment report
Phase 5	Questionnaire	TRAP and RUP process capability and maturity assessment report
Phase 6	Review and validation of work with focus group	Approval report

Table 1: The Phases in the Method

During phase 3 we model the TRAP and RUP metamodel. Figure 2 depicts the three layered abstract modelling architecture defined by SPEM. The three layers are described as:

- Level M2: The separate metamodel(s) of TRAP and RUP. The metamodel(s) al level M2 is compatible with the reference model defined in 15504-2 and the metamodel in SPEM, so that a common basis for judgment was employed.

- Level M1: The process definition. For example, RUP is defined at level M1. We adapt and configure a level M1 process for a process enactment at level M0.
- Level M0: A process enactment or instance, i.e. a process in production in a specific project.

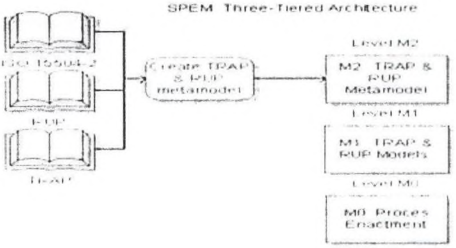


Figure 2: SPEM 3 Layered Architecture

5 The Assessment Method

This section presents the assessment methodology we followed. Figure 3 below illustrates the TRAP and RUP assessment steps:

- Review the assessment input
- Select the process instances
- Determine the actual ratings
- Determine derived ratings
- Validate the ratings

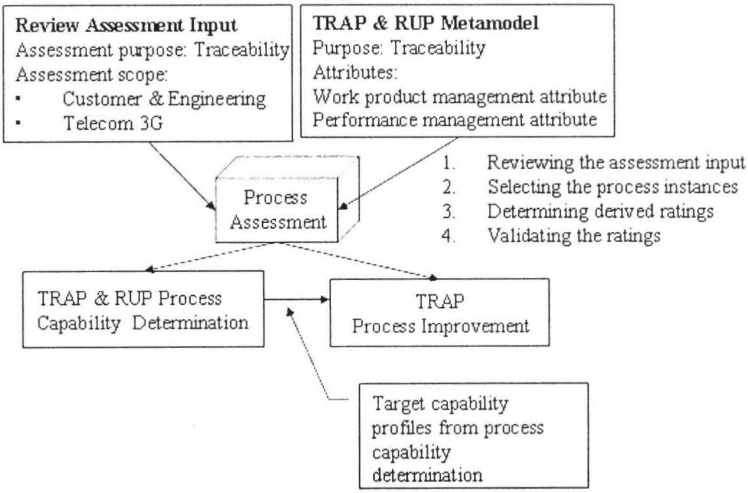

Review Assessment Input
Assessment purpose: Traceability
Assessment scope:
▪ Customer & Engineering
▪ Telecom 3G
TRAP & RUP Metamodel
Purpose: Traceability
Attributes:
Work product management attribute
Performance management attribute
Process Assessment
1. Reviewing the assessment input
2. Selecting the process instances
3. Determining derived ratings
4. Validating the ratings
TRAP & RUP Process Capability Determination
TRAP Process Improvement
Target capability profiles from process capability determination

Figure 3: The Assessment Method

5.1 Review the assessment input

Purpose: We designed and documented a unique assessment process for this research project. The main purpose was to assess the capability of the TRAP process and compare its capability to the RUP.

Scope: TRAP has traceability as the core engineering practices. Therefore we decided to limit the scope of the TRAM metamodel and corresponding TRAP process to the primary lifecycle process categories:

- *The Customer-Supplier* process category consists of processes that directly impact the customer, support development and transition of the software to the customer, and provide for the correct operation and use of the software product. The survey and interview results revealed that traceability to customer requirements is an important practice.
- *The Engineering* process category consists of processes that directly specify, implement or maintain the software product, its traceability to the system and its customer documentation.

A base practice is an activity that addresses the purpose of a particular process. For example tracing test cases to customer requirements is a base practice in the customer compliance process. Consistently performing the tracing practice associated with the compliance process will help to consistently achieve customer compliance. In Table 2 a coherent set of base practices is associated with each process in the process dimension. Management practices relate to the process attributes defined in the process capability dimension of the reference model. Evidence of their effective performance supports the judgement of the degree of achievement of the attribute. Management practices are the principal indicators of process capability. The set of management practices is intended to be applicable to all processes in the process dimension of the model. Evidence of the performance of the defined management practices can be derived from the practice performance characteristics.

Process Category	Base Practice
Customer-Supplier: CUS.1.1 Acquisition Preparation Process CUS.1.4 Customer Acceptance Process CUS.3 Requirements Elicitation Process	CUS.1.1.BP1: Prepare and negotiate contract. CUS.1.1.BP4: Define acceptance criteria. CUS.1.4.BP1: Evaluate the delivered product CUS.1.4.BP2 : Accept the delivered product CUS.3.BP1: Obtain customer requirements and requests. CUS.3.BP2: Agree on requirements. CUS.3.BP3: Establish customer requirements baseline. CUS.3.BP4: Manage customer requirements changes.
ENG.1 Develop system requirements & design ENG.2 Develop software requirements ENG.3 Develop software design ENG.4 Implement software design ENG.5 Integrate and test software ENG.6 Integrate and test system ENG.7 Maintain system and software	ENG. 1.1 BP Specify system requirements ENG. 1.2 BP Describe system architecture ENG. 1.3 BP Allocate requirements ENG.1.4 BP Determine release strategy ENG.1.1 BP7 : Establish traceability. ENG. 2.1 BP Determine software requirements ENG. 2.2 BP Analyse software requirements ENG. 2.3 BP Determine operating environment impact ENG. 2.4 BP Evaluate requirements with customer ENG.2.5 BP Update requirements for next iteration ENG. 3.1 BP Develop software architectural design ENG. 3.2 BP Design interfaces at top level ENG. 3.3 BP Develop detailed design ENG. 3.4 BP Establish traceability ENG. 4.1 BP Develop software units ENG. 4.2 BP Develop unit verification procedures ENG. 4.3 BP Verify the software units ENG. 5.2 BP: Build aggregates of software units ENG. 5.5 BP Develop tests for software ENG. 5.6 BP Test integrated software

Table 2: TRAP Process Dimension and Base Practices

5.2 Select the process instances

We identified the process instances for the assessment using ISO 15504-3. The TRAP software process model is an abstract representation of the way people work. Because different projects have varying levels of adherence to the model, its specific realisations on each project are called process instances. In order to provide a consistent basis for assessment, part two of the ISO 15504 document suite establishes a process model that is representative of the software process as a whole.

5.3 Determine the actual ratings

In addition to reviewing the scales and actual results of the assessment, this section shows how the actual rating was performed. The assessment was implemented as workshop sessions. We determined the capability of TRAP and RUP against the reference model described in ISO/IEC 15504-2. Processes in the reference model are grouped according to the type of activity they address. Each process has a defined purpose describing the high-level objectives that the process should achieve. The purpose statements describe what to do, but do not prescribe how the process should achieve its objectives. Although the reference model contained in ISO 15504-2 covers a range of processes applicable to the software process, we evaluated the capabilities of TRAP and RUP only for the processes related to traceability. We determined the process capability in a systematic assessment and analysis of the TRAP and RUP software processes, carried out with the aim of identifying the strengths, weaknesses and risks associated with deploying the traceability process. The output of a process capability determination is the process capability report. It summarizes, for each key process included within the target capability statement, strengths and weaknesses expressed in terms of process attribute gaps, and the risks associated with each.

We defined the process attributes, their rating scale, and the process capabilities levels. The process attributes were used to determine whether a process has reached a given capability. Each attribute measures a particular aspect of the process capability. The attributes are themselves measured on a percentage scale and therefore provide a more detailed insight into the specific aspects of process capability required to support process improvement and capability determination. The rating scale is a percentage scale from zero to one hundred percent that represents the extent of achievement of the attribute.

The ratings are as follows:

- **N** Not achieved: 0% to 15% - There is little or no evidence of achievement of the defined attribute in the assessed process.
- **P** Partially achieved: 16% to 50% - There is evidence of a sound systematic approach to and achievement of the defined attribute in the assessed process. Some aspects of achievement may be unpredictable.
- **L** Largely achieved: 51% to 85% - There is evidence of a sound systematic approach to and significant achievement of the defined attribute in the assessed process. Performance of the process may vary in some areas or work units.
- **F** Fully achieved: 86% to 100% - There is evidence of a complete and systematic approach to and full achievement of the defined attribute in the assessed process. No significant weaknesses exist across the defined organizational unit.

In order to rate a process one must decide what the process indicators are. An indicator is defined as an objective attribute or characteristic of a practice or work product that supports the judgement of the performance or capability of an implemented process. We defined *Traceability indicators* which confirmed that certain traceability practices were performed. The existence of base practices, work products, and work product characteristics, provide evidence of the performance of the processes associated with them. Similarly, the existence of management practices provides evidence of process capability.

Management practices relate to the process attributes defined in the process capability dimension of the reference model. Evidence of their effective performance supports the judgement of the degree of achievement of the attribute. Management practices are the principal indicators of process capability.

For example TRAP has work product management as one of its process attributes. The indicators for this managed practice is to maintain the traceability of functional, non-functional and quality requirements, maintain work products under configuration management and baseline copies of the work product for the process correspond to the project's current development status.

We established the TRAP and RUP ratings as follows. Firstly, adequacy ratings (F, L, P or N) were determined for all base practices and for all generic practices with respect to each base practice. Then the ratings were converted to percentages by dividing for each capability level (1 to 5) the amount of ratings on each adequacy level (F, L, P or N) with the amount of ratings within that capability level. The resulting percentages were then used to create diagrams and to derive further ratings. Figure 4 shows the determined capability results of TRAP and RUP.

5.4 Validating the ratings

We used self assessment of the TRAP and RUP process. We are currently validating our results using an independent assessor. The TRAP process metamodel has been configured for process enactment and is currently being tested by a number of local companies in Cape Town.

6 Findings of Assessment

We classified TRAP and RUP into similar categories as ISO 15504 to simplify the assessment results. The following are the TRAP processes capabilities:

Level 1 *The software design and implementation processes*. The TRAP process performance attribute is that the process transforms the identifiable input work products to produce identifiable output work products. The traceability relationships are difficult to identify and represent due to the complexity of the design models and implemented code. The TRAP managed practices ensured that the work products are produced. We conclude that traceability is difficult to describe at the design and implementation levels.

Level 2 *The customer acquisition and preparation process and the engineering process for the integration and testing* of the software were determined at the capability level 2. The TRAP process described the requirement (functional and non-functional) work products, how to document and control these work products, the traceability dependencies among the work products and how to control changes to the requirements.

Level 3 *The requirement elicitation process*, the *architectural requirement process* and the *software requirements process* were determined as Level 3. These processes satisfied the work product management attribute but also the process resource attribute. TRAP described the roles involved in software traceability, their corresponding responsibilities and competencies required for performing the traceability process and the process infrastructure required for performing the traceability process was identified.

The results for the RUP capability determination were:

Level 1 *The customer acquisition and preparation process* and the *software design and implementation processes*. The customer acquisition process is poorly defined in RUP. However, the process performance attribute that the process transforms the identifiable input work products to produce identifiable output work products was true. Like in TRAP the traceability relationships were difficult to identify and represent due to the complexity of the design models and implemented code.

Level 2 *The customer requirements elicitation process*, the *system and software requirements process and the integration and test process* were determined to have a level 2 rating. We determined that the requirements and testing discipline are the two most mature process disciplines in RUP. The integration between the requirement management and test management environments was taken into consideration in its process rating.

Traceability is a poorly defined practice in RUP. For example, RUP describes the management of traceability dependencies in the requirements discipline but omits this practice in the business modelling discipline.

TRAP Process Dimension	Capability Level 1	Capability Level 2	Capability Level 3
Cus 1: Customer acquisition preparation process	■	■	
Cus. 2: Customer acquisition process	■	■	
Cus 3: Customer requirements elicitation process	■	■	■
Eng. 1:Develop system requirements and design	■	■	■
ENG.2 Develop software requirements	■	■	■
ENG.3 Develop software design	■		
ENG.4 Implement software design	■		
ENG.5 Integrate and test software	■	■	

RUP Process Dimension	Capability Level 1	Capability Level 2	Capability Level 3
Cus 1: Customer acquisition preparation process	■		
Cus. 2: Customer acquisition process	■		
Cus 3: Customer requirements elicitation process	■	■	
Eng. 1:Develop system requirements and design	■	■	
ENG.2 Develop software requirements	■	■	
ENG.3 Develop software design	■		
ENG.4 Implement software design	■		
ENG.5 Integrate and test software	■	■	

Figure 4: TRAP Process Dimension and Base Practices

7 TRAP Process Outcome

TRAP is intended to provide project managers and requirements engineers with a standardised and efficient means of handling requirements through the full product development life cycle. In particular the process is intended to encourage and support the handling of traceability in an iterative development context. TRAP incorporates the best available requirements traceability techniques for telecommunications software projects in any open systems domain. TRAP is structured as a sequence of high level workflows. Each workflow is broken into discrete steps supported by descriptions of traceability activities, roles and artefacts associated with each step. As the workflows and steps are organised in chronological sequence within the product development life cycle, the process should ideally be read in sequence. Workflow diagrams and descriptions are extensively enabled with hyperlinks to also facilitate nonlinear navigation within the process. In Figure 5 below we see an example taken from TRAP of the requirement engineering traceability activities, the input artefacts, the output artefacts and the tool environment for a local enactment.

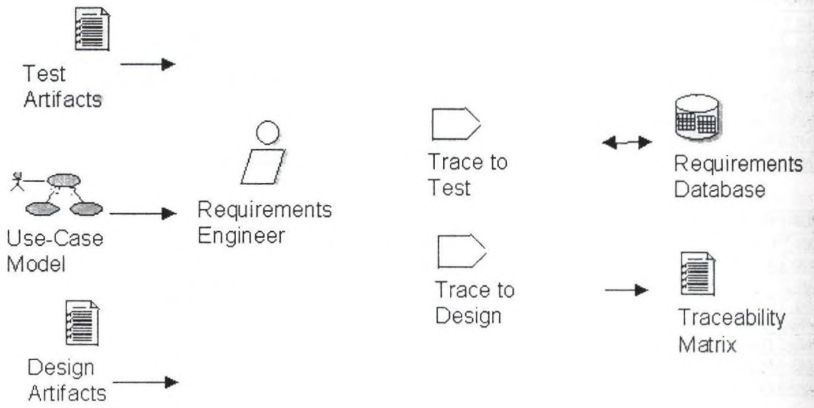


Fig. 6. TRAP Traceability Process Enactment

8 Conclusion

This paper is part of a larger research project on requirement traceability in the product lifecycle, with the research being undertaken in the University of Cape Town and the process enactments occurring in small and medium enterprises in Cape Town. The overall goals of this project and TRAP is to encourage organizations interested in improving their processes and in particular their traceability practices to employ reliable methods for creating and assessing their process. This paper establishes a common framework for expressing the process capability ratings for a 15504-conformant assessment and it provides models and methods for building a traceability process.

To overcome the problems of process reusability we recommend that organisations begin by creating a configurable process metamodel. The Object Management Group (OMG) defines a three-layered architecture for process modelling. Our research proposes that the Object Managements Group specification SPEM is a suitable best of breed process modelling specification. The TRAM process metamodel we generated provides a language for describing the elements of the process. We supported the metamodel with a process authoring tool which publishes product lifecycle process configurations as a web site for practitioners to access.

We defined TRAP to have traceability as its core base practice. TRAP was evaluated to have a high capability level. We assessed RUP under similar conditions. While RUP has evolved into a rich family of integrated software-engineering process products we conclude that it does not support traceability consistently across the entire software development lifecycle. For example, RUP describes the activities associated with traceability in the requirements discipline but does not discuss traceability during the deployment phase. One of the outcomes of our assessment of RUP is that we deem many of its core practices asymmetrical and inconsistent. IBM have made few changes to the RUP workflows and activities over the past few years. We recommend that RUP undergo an independent ISO 15504 assessment and that IBM publish the results or make the necessary process improvements to maintain its position as a leading commercial process framework.

Overall we encountered many problems in interpreting the ISO 15504 models, due to the volume and complexity of the document suite. Our intention was to create and evaluate a simple traceability process and while ISO 15504 is a complete framework we conclude that the effort we put in using the framework is not reflected in the results we obtained . The most interesting and valid data came from the observations and interviews with industrial experts. We therefore argue that the ISO 15504 framework needs to be streamlined and abridged for more agile approaches.

9 Literature

1. Boehm B. W., Brown J. R., Lipow M. Quantitative Evaluation of Software Quality. In Proceedings of the 2nd international conference on Software engineering San Francisco, California, United States. Pages: 592 - 605 (1976)

2. Boehm Barry W., Kevin J. Sullivan. Software economics: a roadmap. In International Conference on Software Engineering Proceedings of the Conference on The Future of Software Engineering. ACM Press New York.

3. Dromey R. Geoff. Cornering the Chimera. IEEE Software Volume 13 , Issue 1 (January 1996) Pages: 33 - 43. IEEE Computer Society Press Los Alamitos (1996).

4. Edwards M. and S. Howell. A Methodology for System Requirements Specification and Traceability for Large Real-Time Complex Systems. technical report, U.S. Naval Surface Warfare CenterDahlgren Division, Dahlgren, Va., 1991.

5. Gotel, O. & Finkelstein, A. (1994). An Analysis of the Requirements Traceability Problem. 1st International Conference on Requirements Engineering (ICRE'94), Colorado Springs, April 1994, pp. 94-101.

6. Humphrey W. S. The software engineering process: definition and scope. In Proceedings of the 4th international software process workshop on Representing and enacting the software process.(Devon, United Kingdom 1988) Pages: 82 - 83 ACMPress, New York, 1989

7. IEEE-STD-610 ANSI/IEEE Std 610.12-1990.IEEE Standard Glossary of Software Engineering Terminology. February 1991

8. IEEE Guide for Information Technology, IEEE-STD 1362-1998, Available: http://standards.ieee.org/catalog/olis/se.html

9. ISO/IEC TR 15504. International Standard for Software Process Assessment. Available: http://isospice.com/standard/tr15504.htm

10. Jacobson I., Booch G. & Rumbaugh J., Unified Software Development Process, Addison-Wesley, 1999

11. Lawlis, P.K., Flowe, R.M., and Thordahl, J.B. A correlational study of the CMM and software development performance. CrossTal/e (Sept. 1995), 21-25.

12. Linda M. Northrop. A Framework for Software Product Line Practice Version 4.2. Software Engineering Institute. Available: http://www.sei.cmu.edu/productlines/frame report/process def.htm

13. McCall James A., Joseph P. Cavano. A framework for the measurement of software quality. ACM SIGSOFT Software Engineering Notes Volume 3 ,Issue 5 (November 1978) Pages: 133 - 139. ACM Press New York(1978).

14. Marco Leon, Intelligent Enterprise Magazine, July 17, 2000, Vol 3 Number 12 http://www.intelligententerprise.com/000908/ebusiness.jhtml

15. Osterweil L. Software processes are software too. In Proceedings of the 9th international conference on Software Engineering (Monterey, California, United States 1987) Pages: 2 - 13. IEEE Computer Society Press Los Alamitos 1987

16. OMG. Software Process Engineering Metamodel (SPEM). Available: http://www.omg.org/docs/formal/05-01-06.pdf

17. Palmer J.D., Traceability. Software Requirements Eng. R.H. Thayer and M. Dorfman, eds., pp. 364-374, 1997.

18. Philippe Kruchten, The Rational Unified Process-An Introduction, 2nd ed, Addison- Wesley-Longman, Reading, MA (2000)

19. Shahid Nazir Bhatti. Why quality?: ISO 9126 software quality metrics (Functionality) support by UML suite. ACM SIGSOFT Software Engineering Notes ACM Press New York, NY

20. Zachman J.A., 'A framework for information systems architecture', in: IBM Systems Journal, 1987, vol. 26, nr. 3, pp. 276-292.

10 Author CVs

Justin Kelleher

Justin Kelleher graduated with an Engineering Degree from University College Galway, Ireland in 1995 and a Software Design and Development postgraduate degree in 1997. He worked as a project mentor and senior technical trainer for Ericsson between 1997-1999. He was employed as a process prime during the Joint Development Initiative project between Ericsson and Rational for the evolution of TDMA product line release in the R&D center, Montreal. He also worked as a system architect and project manager for a number of 3G projects before starting a PhD in 2004 at the University of Cape Town in the Data Network Architecture Research Lab.

Concurrent Software Process Modeling

Oktay Türetken and Onur Demirors

Abstract

Majority of current methods for process modeling utilize a centralized approach in which group of experts (process engineers) work with the agents to analyze, understand, model and improve organization's processes. This paper proposes a method for organizations where the responsibility of understanding, modeling and improving software processes is delegated to the agents (individuals or teams) that actually perform the processes. Proposed method is demonstrated through a sample process.

Keywords

Process Modeling, Business Process Modelling, Software Process Modeling, Process Modeling Method

1 Introduction

We have successfully utilized process models to better understand, analyze and communicate organizational knowledge. In many fields of organizational life such as creating process scripts for workers to follow, establishing quality manuals, assessing and identifying added value, establishing control mechanisms, automating workflow and identifying software requirements we observe process modeling as a core activity. As a result, process models are viewed as one of the most valuable assets of organizations.

The growth of information society increased the significance of knowledge which in turn increased the importance of process models. Information society has also increased a wider distribution of knowledge and expertise within the organization and society as a whole [7]. The results of wider distribution of knowledge also enables (and requires at the same time) organizations to change more frequently and to change much faster. The process model infrastructure of the organizations of information society therefore should enable frequent and rapid changes.

In the society, where knowledge is the primary resource for individuals and for the economy, many researchers agree that the traditional structures of organizations are not appropriate for creating products and services that require knowledge work and its integration [7], [3], [17]. According to Senge [17], the unified premise of quality movement is "to make continual learning a way of organizational life, especially improving the performance of the organization as a total system". Senge acknowledges that this can be achievable if traditional authoritarian, command-and-control hierarchy -where the top thinks and the local acts- is broken. Merging thinking and acting at all levels is necessary. One of the first prerequisites of this achievement is removing impediments disempowering the workers. In knowledge based environments the greatest knowledge is at the bottom where it is created. Software engineers feel that they require some autonomy in planning, executing and controlling their work and applying their knowledge without close supervision [19]. The studies by Sommerville [19] show that these professionals require some control over their work activities and strongly resent particular work practices imposed by the organization unless they participate in the design of that process. Armour identified many problems that might arise when process is developed by people who do not actually employ it [2].

Process modeling traditionally performed by a group of experts (process group, or process engineers [2]), who work with the individuals -actually performing the activities- one by one in order to analyze, understand, define and improve organization's processes. It usually takes months to model an organization's processes from scratch and once processes are considered stable it is not desired to change them frequently. We need new approaches that enable processes to be modeled in the order of days. In fact, ideally, not only modeling but the whole process improvement cycle should be performed in days. Such a rapidly changing process infrastructure also require new methods for decision making, process enactment as well as application of traditional tools with a new insight.

Agent-based concurrent business process modeling (ABC-BPM) is an approach where the responsibility of understanding, modeling and improving the processes is delegated to individuals or teams that actually perform the processes. We view organizations as a set of autonomous, interacting and collaborating units (agents), which own their tasks, information and resources involved in the process. They perform their processes concurrently and interact when needed. The individual process models form the organization's process network. This is more suited for organizations where knowledge workers are integrated and collaborate for production. They interact and collaborate among themselves and with computerized tools. Therefore, software engineering organizations are excellent candidates.

We believe that a method that enables agents model their business processes gives a better reflection of the actual processes. Since process modeling is performed concurrently by all agents, it would require less time to develop and 'maintain' the entire organization's process model. It would provide a better mechanism for improvement initiated at the bottom and concurrently spread over the organization.

In this paper, we propose a method that will provide perception and consciousness of an agent over its activities, and awareness for identifying the part it is representing in the puzzle. By modeling the interface (dependency requirements) between its suppliers and consumers of the outcome of its activities, it acquires a better understanding of its goals and objectives of performing such processes. This approach takes the responsibility of process improvement from process engineers to all agents of an organization which enables improvement to be owned and performed continuously and totally rather than discretely and in partial fashion as in traditional approaches.

The remainder of this paper is structured as follows: In section **Fehler! Verweisquelle konnte nicht gefunden werden.**, we discuss the related research. After introducing the method (in section **Fehler! Verweisquelle konnte nicht gefunden werden.**), in section **Fehler! Verweisquelle konnte nicht gefunden werden.** we give a short exemplar study demonstrating how we can utilize the model with a process modeling notation.

2 Related Work

The concept of distributed process modeling is studied for two distinct purposes during the last decade: Process modeling and enactment and requirements elicitation.

Most of the existing process modeling and enactment approaches assume (explicitly or implicitly) central specification and execution of processes (PADM [5], SPADE [4], Process Weaver [12], MARVEL [11], etc.). In general, a central server provides functionality for process modeling, and a process enactment engine runs the models, records execution, supports task automation and tool integration where necessary.

RAD (roles activity diagram) approach introduced by Ould [16] is one of the few approaches that places the role concept at the center of the modeling. The Riva method (formerly STRIM) [16] proposes a number of steps to be followed for process modeling using RADs. The approach focuses on the interaction between roles which ease the modeling of concurrent engineering processes [14]. However, due to the type of constructs it uses, pure RADs are not fully applicable to our approach.

The idea of agents modeling their activities in a decentralized manner is proposed by Demirors as the "Horizontal Change Approach (HOC-A)" [6] to manage change in software development organizations. In HOC-A, process modeling and change are performed in a decentralized manner concurrently by all the members of the organizations. In this sense, it is similar to neural networks in which the overall goal is achieved collectively without direction or structuring effort at any specific organizational level.

ABC-PBM shares common points on the methods and the tools that are utilized in 'ViewPoints' [8], [15] or 'view-based approaches' employed in requirements engineering and process elicitation research areas. 'Each ViewPoint encapsulates partial knowledge about a system and its domain - expressed in a suitable representation scheme - together with partial knowledge about the overall process of development' [15]. The framework is based on the idea that, the construction of a complex description or model involves many agents who have different perspectives or views of the artifact or system they are trying to describe [9].

CORE (Controlled Requirements Expression) [13] method is one of the earliest requirements analysis and specification method that provides prescriptive guidelines on specifying and analyzing system requirements based on viewpoints. The phases of the method comprise the definition of the problem, viewpoint identification and gathering and documenting information about each viewpoint. Each viewpoint is described with a tabular collection diagram, where the sources of inputs and the destination of outputs to each action performed by each viewpoint are identified and inconsistencies are identified based on these interactions. However, in tabular collections diagrams, behavioral representation of the process, which is a requirement in our approach, is limited.

Verlage is one of the first that introduced a formalization of core requirements of view based process elicitation [20]. He proposed the following steps for eliciting a model from different views: independent modeling of views, detecting similarities between views, detecting inconsistencies between views, and merging views.

Turgeon and Madhavji [21] have proposed a prototype tool called V-Elicit that helps to elicit process models from multiple sources or views. Works by Verlage and Turgeon & Madhavji are in parallel in many points. However, Turgeon & Madhavji's work on multi view process elicitation is one of the most complete approaches with an available tool support. V-elicit tool responses a larger set of requirements. However, there are still issues, such as common element identification, that require further enhancements.

The critical difference between ABC-BPM and view based approaches lies in the way they perceive the process of process modeling. In view-based approaches, process modeling is still performed centrally by a process engineering group who is responsible to understand, elicit, model, maintain and improve the processes. Conflicts are generally not communicated and this crucial knowledge is not disseminated to the organization. ABC-BPM's strength lies in the idea of delegating these responsibilities to the agent that enact their processes. It is only then; agents communicate, surface implicit assumptions and resolve conflicts. These characteristics pose unique challenges and necessitate unique requirements that should be answered with a framework. Although the tools and the methods we discussed above meet certain requirements, there are unique constraints that are yet unfulfilled.

3 A Method for ABC-BPM

The main contribution of this paper is the proposed method that guides an organization in performing agent-based concurrent business process modeling. Tool support for automating and supporting some of the activities of the method proposed is necessary. It supports the agents in representing their activities with a suitable process modeling notation.

We assume knowledge workers to have knowledge and organizational environment that provide empowerment and motivation to continuously improve their processes. Therefore we consider modeling to be the very basic part of their responsibilities rather than an additional burden loaded on daily work activities. From agents' perspective, it is an enabler for them to think, understand, define and improve their own processes.

3.1 Agents and Processes

An agent is an object in an organization which has activities assigned as responsibilities in a business context. It can be an individual, a group, a team, a functional unit, board, or an application system, that carries out a set of activities (participates in a process) in order to accomplish one or more business objectives.

We may also talk about *information sources*, which might represent a library, an application system or any other information source that provides necessary information an agent requires and stores for further needs. It is conceptually comparable to 'data stores' used in data flow diagrams (DFD) [10].

In ABC-BPM, agents are responsible for three distinct roles; modeling, change and enactment. With respect to responsibility, we can distinguish three types of agents:

- Active agents are development agents who are responsible for concurrently modeling, improving and enacting their own processes.
- Inactive agents are agents who interact with active agents but left out of the scope of process modeling and definition procedure due to some certain reasons. An inactive agent might be an external role to an organization (customer, supplier, etc.) or an internal function whose processes cannot be controlled or are of no interest.
- Process agents primarily have two important responsibilities. First they represent the voice of the inactive agents and information sources during consistency management. The interactions between inactive agents and active agents (which are modeled by active agents) should also be checked and validated. This is also true for the interactions between information sources and active agents. Second, they are the owners of the meta-process; where they are responsible from

the successful application of this method. They guide active agents mainly in process modeling and maintaining the process network.

3.2 Phases of the Method

Figure 1 illustrates activities of the method and data flow among them. Although the phases are presented sequentially, modeling is an iterative process.

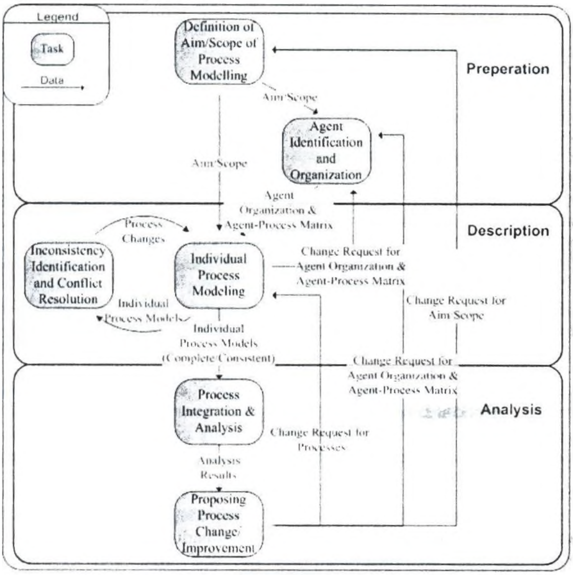

Legend
Task
Data
Definition of Aim/Scope of Process Modelling
Preperation
Aim/Scope
Agent Identification and Organization
Aim/Scope
Agent Organization & Agent-Process Matrix
Process Changes
Description
Inconsistency Identification and Conflict Resolution
Individual Process Modeling
Change Request for Agent Organization & Agent-Process Matrix
Individual Process Models
Change Request for Aim Scope
Individual Process Models (Complete/Consistent)
Change Request for Agent Organization & Agent-Process Matrix
Process Integration & Analysis
Analysis
Change Request for Processes
Analysis Results
Proposing Process Change/ Improvement

Figure 1. A Method for ABC-BPM

3.2.1 Preparation.

This stage sets up the organization for ABC-BPM. This is the phase where a 'process agents' team' plays the key role. The goal is to achieve a structural frame of the organization in terms of the high level process network, related agents (roles) and their organization.

Definition of Aim and Scope. An organization initiates the method by defining the aim and scope of process modeling. First, process agents' team and other stakeholders (which may include active agents) set organization's intentions to model its processes. These intentions should be aligned with the vision and mission of the company and they should serve one or more goals. As in any other similar undertakings in an organization, the support from upper level management is critical. Therefore, all stakeholders and related functional units should have a clear understanding and consensus about the aim and scope of this initiative. The scope of the process modeling helps identifying the boundaries of the processes to be modeled and the agents that will participate in modeling.

Agent Identification and Organization. With the scope determined in prior step, process agents' team identifies the active and inactive agents as well as their relationship (in structural terms). In order to represent this structure we use diagrams similar to the organizational charts.

One of the two fundamental products of this step is the 'agent (role)-process matrix' which represents the identified agents and the list of processes they participate with. Each participant should have a consensus over the information depicted on diagrams. The outputs of these steps are subject to changes and improvements as active agents begin individual process modeling. Change proposals by active agents are communicated with the team of process agents and if accepted reflected on the models (organization diagram, agent-process matrix) and communicated though the organization by process agents. If necessary, active agents are subject to training by process agents before they proceed to modeling.

3.2.2 Process Description.

This stage forms the hearth of the method. As the aim and scope of process modeling are established and agent organization and their associated processes are identified, concurrent process modeling commences. This stage is a continuous activity specifically when the main objective for process modeling is process improvement.

Individual Process Modeling. All participants, as being one or more agents, begin to model their activities in parallel. They use the notation for describing the behavior of the processes that they are associated with in the first phase. The notation is based on eEPCs (extended Event Driven Process Chain) [18] and tabular collection diagrams used in CORE [13], [15]. eEPCs are semi-formal and widely accepted in practice. The main constructs are functions and events. An event can trigger a function or a function can produce an event so combinations of events and functions in a sequence produce event-process chains. In order to represent control relationship between triggering events and functions, logical operators are used. Data and organization view of business processes are also represented in eEPCs. Similar to the concept in tabular collection diagrams, we wanted agents to represent the sources of inputs and the destination of outputs to and from each action they perform and used eEPCs accordingly.

Inconsistency Identification and Conflict Resolution. In case of conflicts, which manifest themselves as inconsistencies among individual process models, agents communicate and share related issues. If necessary, they negotiate and resolve conflicts. Our primary concern in consistency management is the inconsistencies that might arise between agent's process models (inter-consistency).

Our approach expects each agent to model its processes and the expectations from neighbor agents. The models are consistent if all interacting agents fulfill their expectations from each other. We presume that there is an inconsistency between two individual process models if the interaction modeled by one of the agents is incompliant with the one modeled by the other. The interaction refers to the information (message) received or sent between two agents or an activity performed collaboratively by two or more agents. The method does not offer a protocol for inter agent communication and negotiation, but, a tool, first, can provide the necessary functions for identifying inconsistencies between individual process models; second can provide functionalities (messaging, chat, etc.) that facilitate the negotiation to resolve conflicts. With an agreement, agents change (re-model) their activities in order to reflect the solution and establish the consistency among models. Automated conflict resolution is by no means applicable in these cases.

3.2.3 Analysis.

This task comprises a set of activities performed in order to gain insight into the way the organization works, pinpoint problems and inefficiencies, identify improvement opportunities, and finally recommend changes and improvements. Analysis activities can be performed at any time after individual process models are complete and consistent.

Process Integration and Analysis. Process integration comprises the activities that combine 'consistent' individual process models to form the process network and visualize the interactions. It is performed to obtain the big picture of the activities performed by all agents. Agent dependencies can also be analyzed to understand the interaction and dependencies within agents in order to uncover the effects of any change or improvement proposal. These analyses are used to identify problems and drawbacks of current execution of processes and to identify improvement opportunities. A tool that automates process integration and the generation of dependency diagrams are of great value to agents.

Proposing Process Change/Improvement. Change and improvement suggestions from agents are communicated and negotiated between related agents and reflected in individual process models. Process-agents are responsible for understanding the need for analysis and improvement, and supplying and improving the techniques to agents. All suggestions are feed back to 'description' and 'preparation' phases as recommended modifications.

4 An Example –Review Process

In this section, we demonstrate the use of the proposed method (and the notation) by going through a review process of a small software organization. The primary objective of presenting this example is assessing the applicability and usability of the method rather than providing a validation of the framework. As a limitation, concurrent process modeling could not be achieved completely in desired manner since individual process modeling is performed asynchronously with one of the authors of the paper. However, this very first example shed light both on the applicability of the method and on the usability of the notation. We used ARIS Toolset [1] for modeling and representing processes. We have been developing an add-on to the tool for identifying inconsistencies and automated process generation (process integration, dependency diagrams, etc.).

Preparation: We model this example process with the aim of representing it on a model that will facilitate agents understanding and communication. Participating agents will share a common representational format and the model will guide agents in future performances of the process. Figure 2 presents the agent-process matrix where the identified agents are mapped with the processes. For this example, since only one process is under consideration, the extent of information provided on this diagram is limited.

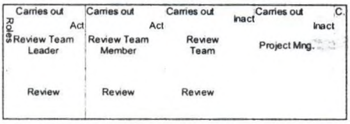


Figure 2. Agent-Process Matrix (Review Process)

Agents participating in the process are identified and are depicted in an organization diagram (Figure 3). In this scope, there are the project manager; as an inactive agent; and the review team agent composed of two active agents (role).

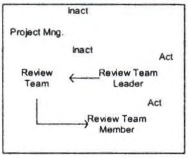


Figure 3. Agent Organization Diagram (Review Process)

Process Description: Figure 4 presents a fragment of the individual process model of the Review Team Leader agent. Likewise in the figure, all agents identified in previous phase represent their activities in such models.

In this example, as noted above, one of the authors of the paper acted as the agents participating in this process. In order to decrease the bias associated, we left three days between modeling of each agent's activities. The activities of each agent are represented with the notation where interaction expectations from other agents are also modeled. For example, as the review team leader agent, the activities performed and ingoing/outgoing information from these activities are represented. Where applicable, the sources or destinations (agents or information sources) of these information containers (documents, messages, etc.) are indicated. While doing this, the expectations, if available, modeled by other agents (in this case only 'reviewer team member') are also taken into consideration in order to come up with a set of inter-consistent process models. In real cases, where there is a conflicting expectation, agents communicate and solve the issue and reflect the change onto their models. The modeling tool helps the agents to review the expectations modeled by other agents by aggregating related information from other agents' individual process models (e.g. information that is consumed or demanded from and information that is supplied to related agent, activities that are performed together.

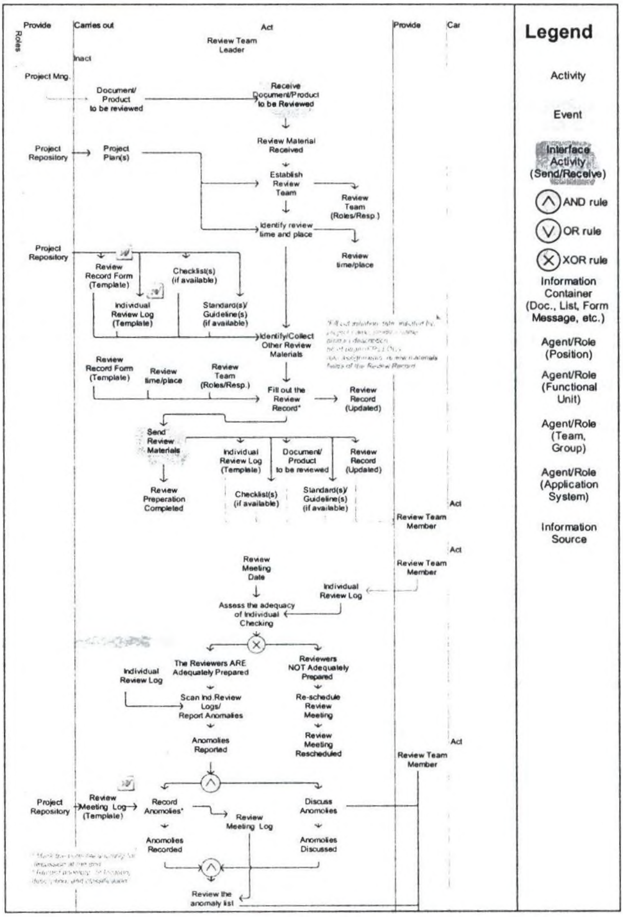

Provide
Carries out
Act
Roles
Review Team Leader
Inact
Project Mng.
Document/ Product to be reviewed
Receive Document/Product to be Reviewed
Review Material Received
Project Repository
Project Plan(s)
Establish Review Team
Review Team (Roles/Resp.)
Identify review time and place
Review time/place
Project Repository
Review Record Form (Template)
Checklist(s) (if available)
Individual Review Log (Template)
Standard(s)/ Guideline(s) (if available)
Identify/Collect Other Review Materials
Review Record Form (Template)
Review time/place
Review Team (Roles/Resp.)
Fill out the Review Record*
Review Record (Updated)
Send Review Materials
Individual Review Log (Template)
Document/ Product to be reviewed
Review Record (Updated)
Review Preperation Completed
Checklist(s) (if available)
Standard(s)/ Guideline(s) (if available)
Provide
Car
Act
Review Team Member
Act
Review Team Member
Review Meeting Date
Individual Review Log
Assess the adequacy of Individual Checking
The Reviewers ARE Adequately Prepared
Reviewers NOT Adequately Prepared
Individual Review Log
Scan Ind.Review Logs/ Report Anomalies
Re-schedule Review Meeting
Anomolies Reported
Review Meeting Rescheduled
Act
Review Team Member
Project Repository
Review Meeting Log (Template)
Record Anomolies*
Review Meeting Log
Discuss Anomolies
Anomolies Recorded
Anomolies Discussed
Review the anomaly list
Legend
Activity
Event
Interface Activity (Send/Receive)
AND rule
OR rule
XOR rule
Information Container (Doc., List, Form Message, etc.)
Agent/Role (Position)
Agent/Role (Functional Unit)
Agent/Role (Team, Group)
Agent/Role (Application System)
Information Source

Figure 4. A Fragment of the process model of "Review Team Leader" (Review Process)

There were also cases where, in an agent's model, an information container that is expected to be input to another agent's activities is not present in the destination agent's model. In general, this might be caused due to several reasons. In case of simple mistakes (naming, representation, etc.), an inconsistency can be handled without communication taking place between agents. However, if both sides insist on their position, which results a conceptual disagreement, it becomes a conflict [15]. In this case, they should communicate and negotiate in order to solve the issue.

Due to the extent of the activities involved, modeling the activities of review team leader's took relatively longer. If we assume that agents model their activities concurrently, total time for process description phase is the time for modeling editor's activities and the time devoted for solving inconsistencies. In real cases, if necessary, one should also consider the time devoted for training agents for learning the formalism.

Analysis: Having agents modeled their activities and came up with a complete and consistent individual process models, we integrated them to demonstrate the entire model of the process. Figure 5 displays a fragment of the integrated model. In order to facilitate understanding, we also generated agent dependency diagrams (Figure 6). As seen on the figure dependency diagrams can be generated with different levels based on the relationships modeled on the organization diagrams.

Although this was not the case for this example for now, both the integration of models and generation of dependency diagrams can be automated once individual process models are complete and consistent. Interaction points of the individual models are clear (information items sent/received, and activities performed together). In the integrated model, an additional swim-lane -belonging to both agents- is added to represent the activities performed collaboratively.

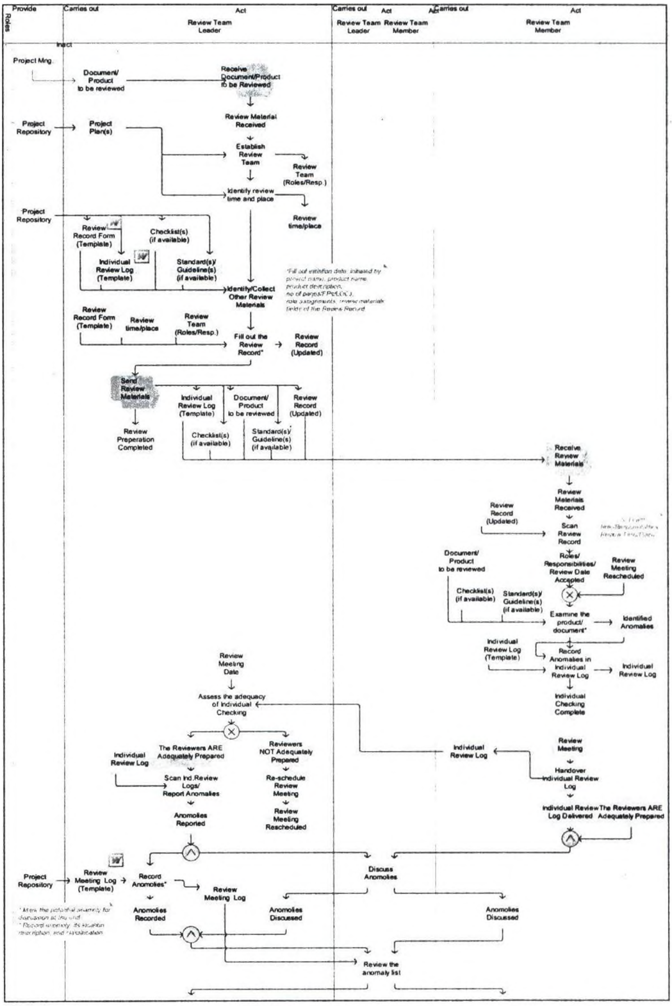

Provide
Roles
Carries out
Act
Review Team Leader
Carries out
Act
Review Team Leader
Review Team Member
Act
Carries out
Act
Review Team Member
Inact
Project Mng.
Document/ Product to be reviewed
Receive Document/Product to be Reviewed
Review Material Received
Project Repository
Project Plan(s)
Establish Review Team
Review Team (Roles/Resp.)
Identify review time and place
Review time/place
Project Repository
Review Record Form (Template)
Checklist(s) (if available)
Individual Review Log (Template)
Standard(s)/ Guideline(s) (if available)
Identify/Collect Other Review Materials
*Fill out initiation date, initiated by project name, product name, product description, no of pages/F Pt/LOC, role assignments, review materials fields of the Review Record
Review Record Form (Template)
Review time/place
Review Team (Roles/Resp.)
Fill out the Review Record*
Review Record (Updated)
Send Review Materials
Individual Review Log (Template)
Document/ Product to be reviewed
Review Record (Updated)
Review Preperation Completed
Checklist(s) (if available)
Standard(s)/ Guideline(s) (if available)
Receive Review Materials
Review Materials Received
Review Record (Updated)
Scan Review Record
*Confirm Roles/Responsibilities Review Time/Place
Document/ Product to be reviewed
Roles/ Responsibilities/ Review Date Accepted
Review Meeting Rescheduled
Checklist(s) (if available)
Standard(s)/ Guideline(s) (if available)
Examine the product/ document*
Identified Anomalies
Individual Review Log (Template)
Record Anomalies in Individual Review Log
Individual Review Log
Individual Checking Complete
Review Meeting Date
Assess the adequacy of Individual Checking
The Reviewers ARE Adequately Prepared
Reviewers NOT Adequately Prepared
Individual Review Log
Scan Ind.Review Logs/ Report Anomalies
Re-schedule Review Meeting
Anomolies Reported
Review Meeting Rescheduled
Individual Review Log
Review Meeting
Handover Individual Review Log
Individual Review Log Delivered
The Reviewers ARE Adequately Prepared
Project Repository
Review Meeting Log (Template)
Record Anomolies*
Review Meeting Log
Discuss Anomolies
* Mark the potential anomaly for discussion at the end
* Record anomaly, its location description, and classification
Anomolies Recorded
Anomolies Discussed
Anomolies Discussed
Review the anomaly list

Figure 5. A Fragment of the Integrated Process Model (Review Process)

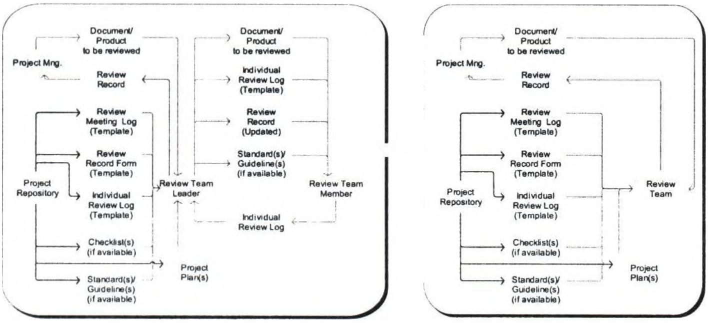


Figure 6. Agent Dependency Diagrams (Review Process)

5 Conclusion and Future Work

In this paper, we focused on a guiding method for organizations to perform agent-based concurrent process modeling. Utilizing such a method has number of advantages. First of all, due to the concurrency in process modeling, it would eventually require less time to develop and 'maintain' the organization process model. It covers all agents and their activities participating in processes and considering their viewpoints in processes would decrease the chance of incompleteness in the models. It would provide a better mechanism for handling changes in the processes (traceability on a process change is inherently tackled). Once models are complete and consistent, it provides process analysis to be performed from multiple on different abstraction levels. The approach will not only direct agents to process thinking but also will facilitate the communication among agents in the organization. This will serve to uncover implicit assumptions and misunderstandings, and will provide faster knowledge spread over the organization.

An organization that decides to perform agent-based concurrent process modeling should be concerned about the challenges that will be faced off and the assumptions of the method that it will utilize. In order the method to be successful, first, agents participating in the processes should have adequate knowledge of how they perform their activities. This requirement, in essence, can also be considered as one of the fundamental characteristics a knowledge based organization should possess. The success of the method is also determined by the ability of the agents to demonstrate their process knowledge in a process model using the defined notation. The motivation of an agent to participate and communicate with other agents to resolve conflicts would also have significant effects on the desired level of achievement. These two challenges are areas in which process agents play crucial roles. They should be able to control the process modeling activities throughout the whole cycle. They guide and motivate agents and they act as the primary means of catalysts among agents. The degree of involvement of upper management is also important in this context as it is in almost all organizational undertakings. We presume that agents promote the mission and vision of their organization. Process agents also have a significant role of maintaining the top view of the organization in order to understand the big picture and capture high level improvement opportunities. Process agents may guide and direct agents towards improving their processes with respect to this type of organizational change requests.

Tool support for automating and supporting some of the activities of the methodology is desired. It will ease the modeling of individual processes using the methodology and its integral notation. It should mainly provide functionalities that enable agents to model their processes using the process modeling notation; identify and highlight inconsistencies between individual models; provide functions for inter agent communication and negotiation; and, finally provide agents with a number of process analysis instruments (model integration, semantic check, generation of dependency diagrams, etc.) to verify the models and identify improvement opportunities. Having defined its top level requirements, we developed the parts satisfying the most critical requirements of the tool. We are enhancing the tool also by using it on a real life case study. As a future work, we will extend and enhance the method by performing case studies in several organizations.

Literature

1. ARIS Method. ARIS 6 Collaborative Suite, IDS Scheer AG. (2004)
2. Armour, Philip G.: The Laws of Software Process: A New Model for the Production and Management of Software. Auerbach Publications (2004)
3. Argyris, C.: Good Communication That Blocks Learning. Harvard Business Review, (July-August 1994)
4. Bandinelli, S., M.Braga, A.Fuggetta, and L.Lavazza: The architecture of the SPADE-1 Process-Centered SEE. B.C. Warboys, editor, Proceeding of the Third European Workshop on Software Process Technology (EWSPT'94). Villard-De-Lans, France, Springer- Verlag (1994)
5. Bruynooghe, R.F., R.M. Greenwood, I. Robertson, J. Sa, R.A. Snowdon, and B.C. Warboys: PADM: Towards a Total Process Modelling System. In A. Finkelstein, J. Kramer, and B. Nuseibeh, editors, Software Process Modelling and Technology. Research Studies Press (1994) 293-334
6. Demirörs, O.: A Horizontal Reflective Process Modeling Approach for Managing Change in Software Development Organizations. Ph.D. Thesis. School of Engineering and Applied Science, Southern Methodist University (1995)
7. Drucker, P.F.: The New Society of Organizations. Harvard Business Review, (1992) 95-104
8. Finkelstein, A., J. Kramer, B. Nuseibeh, L. Finkelstein, M. Goedicke: Viewpoints: A Framework for Integrating Multiple Perspectives in System Development. International Journal of Software Engineering and Knowledge Engineering, Vol. 2, No. 1, World Scientific Publishing Co. (1992) 31-58.
9. Finkelstein, A. and I. Sommerville: The Viewpoints FAQ. Software Engineering Journal, Vol. 11, (1996) 2 – 4
10. Jeffrey L.W., Lonnie D.B.,Kevin C.D.: System Analysis and Design Methods. 6th Ed. The McGraw Hill, NY (2004)
11. Kaiser, G. E. "Rule-based modelling of the software development process". Proceedings of the 4th international software process workshop on Representing and enacting the software process, Devon, United Kingdom. Pages: 84 - 86. 1988.
12. Le Brasseur, Mryse, Gerald Perdreau: PROCESS WEAVER: from CASE to Workflow Applications. IEE Colloquium on CSCW (Computer Supported Co-operative Working) and the Software Process (Digest No. 1995/036) (1995) 7/1 -7/5
13. Mullery, G.: CORE - a method for controlled requirements expression. Proceedings of 4th International Conference on Software Engineering (ICSE-4), IEEE Computer Society Press (1979) 126-135
14. Murdoch, J., J. A. McDermid, P. Wilkinson: Modelling Engineering Design Processes with Role Activity Diagrams. Trans. Soc. for Design and Process Science, 4 (2) (2000) 44-65
15. Nuseibeh, B.: A Multi-Perspective framework for Method Integration. PhD Thesis, Department of Computing, Imperial College, London (October 1994)
16. Ould, M.A.: Designing a re-engineering proof process architecture. Business Process Management Journal, Vol. 3 No. 3, pp. 232-247, MCB University Press (1997)
17. Senge, P.: It's the Learning: The Real Lesson of the Quality Movement. Journal for Quality & Participation, Nov/Dec99, Vol. 22, Issue 6. (1999)
18. Scheer, W.A.: ARIS- Business Process Frameworks. 3rd Ed., Springer-Verlag Berlin (1999)
19. Sommerville, I., T. Rodden: Human, Social and Organisational Influences on the Software Process. Technical Report: CSEG/2/1995, CSEG, Computing Department, Lancaster University (1995)
20. Verlage, M., "Multi-View Modeling of Software Processes", In Proceedings of EWSPT3, pp. 123-127, Springer-Verlag, Grenoble, France, 1994.
21. Turgeon, J.: A View-Based System for Eliciting Software Process Models. Ph.D. thesis, McGill University, September (1999)

6 Author CVs

Oktay Türetken

Oktay Türetken has M.Sc. degree in Information Systems and B.Sc. degree in Industrial Engineering. He is a doctorate student and research assistant in information systems mainly focused on software engineering. He participated in a number of ERP (enterprise resource planning) system implementation projects as a consultant, and is working as a software engineer in software intensive system specification/acquisition projects. His research interests include; business process modeling, software process improvement, software quality management and requirements engineering.

Onur Demirors

Onur Demirors has Ph.D. and M.Sc. degrees in Computer Science and B.Sc. degree in Computer Engineering. He has been working in the domain of software engineering as a consultant, academician, researcher and entrepreneur for the last 15 years. His work focuses on software process improvement, software project management, software engineering education, software engineering standards, and organizational change management. He worked as a consultant for a number of software developing companies to improve their processes based on ISO 9001, ISO 15504 and CMM. He managed a number of research and development projects on software process improvement, business process modeling and large scale software intensive system specification/acquisition. He has over 40 papers published in various journals and conferences.

From Process Improvement to Learning Organisations

Richard MESSNARZ[1], Ted O'Keeffe[2], Gearoid O'Suilleabhain and Ray Coughlan[3]
[1]*ISCN GesmbH, Schieszstattgasse 4, A-8010 Graz, Austria*
Tel: +43 316 811198, Fax: + 43 316 811312, Email: rmess@iscn.com
[2]*WIT, Waterford Institute of Technology, Ireland*
[3]*DEIS/CIT, Cork Institute of Technology, Cork, Ireland*

Abstract

This paper outlines a case study where process improvement has been combined with innovation management strategies. This approach is a key success factor in industry because process improvement without regarding product innovation would in the long term stabilise existing cash cow productions but would not unleash new product ideas. However, new product concepts are necessary to win a competitive advantage on the market.

Concerning process improvement the case study based on ISO 15504. Concerning innovation skills and processes the case study based on the results of three former projects : (1) CREDIT (MM 1032 , 1998 – 2001) developing a multi-user Intra/Internet based assessment platform for skills and process assessments, (2) ORGANIC (Leonardo da Vinci, 2003 – 2006) developing a training program for required skills in innovation management, and (3) SOQRATES (originally Bavarian state funded in 2003 and later funded by a group of leading firms in central Europe) comprising task forces shared across companies to exchange process improvement knowledge and jointly collaborate on synergies and improvements.

Keywords

Process Improvement, Innovation, Networking, Web Based Learning Environment

1 The Approach

ISO 15504 Process **Requirements Elicitation** . The exemplar assessment model ISO 15504-5 expects that requirements scenarios are analysed, different product platform scenarios are analysed, and requirements are assigned to releases and future product platforms. This creates a pro-active focus towards future products and their functionality.

The innovation process CRM (**Customer Relationship Management**) looks at the same from a more holistic approach (not only process related) including CRM processes, participative models including customers, knowledge management strategies for collecting and sharing product ideas/requirements , and establishing long term mutually beneficial partnership / service strategies.

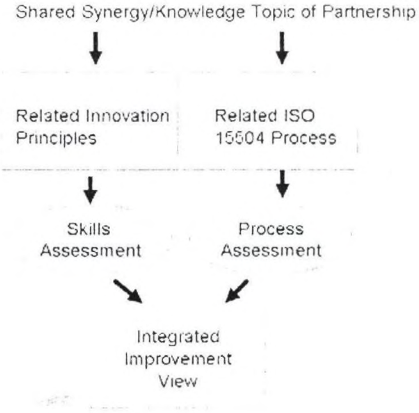

Shared Synergy/Knowledge Topic of Partnership
Related Innovation Principles
Related ISO 15504 Process
Skills Assessment
Process Assessment
Integrated Improvement View

Figure 1: Model for Combining Innovation Aspects and Process Improvement

The innovation process **Learning Culture** emphasises that learning in teams, sharing knowledge across teams, establishing a culture of collaboration and joined mission, is a human lever to become successful and act quicker and more efficient.

The approach is illustrated in Figure 1 above. With regard to the Learning Culture cross-company task forces were established to jointly collaborate on SPI and innovation aspects. With regard to the Requirements Elicitation the task forces shared knowledge about the best practices for Requirements Elicitation. With regard to the innovation aspects virtual knowledge spaces and online training platforms have been used to virtually collaborate about how to consider innovation aspects in CRM in the Requirements Elicitation.

1.1 How it is Implemented

The task force used two existing systems:

- SOQRATES (http://portl.soqrates.de, http://www.soqrates.de) established a group of leading companies who formed cross-company task forces to jointly collaborate and share knowledge in important SPI and innovation fields. This included a web based portal system for ISO 15504 based process assessments

- ORGANIC (http://www.innovationmanager.org and some predecessor projects) developed an innovation manager skills card, a skills assessment portal and related educational materials, plus a certificate for innovation managers.
- EPI (successor of CREDIT) (http://www.iscn.com/projects/epi_skill_portal/) integrated a web based learning platform with the existing skills and process assessment portal systems.

In ORGANIC there is a learning objective called "Customer Relationship Management" , which focuses on (= Related Innovation Principles in Figure 1) -

- Learning from customers.
- Capturing customer innovation plans for ears in advance to adapt early enough to required innovations.
- Establishing knowledge based communication means between customers and suppliers to provide idea sharing and knowledge sharing for future innovation.
- Establishing a mutual feeling of trust with customers so that ideas of innovation are exchanged and not hidden.
- Capturing requirements even if they are not relevant for the current projects and to assign them to potential future releases (idea pool and later exploitation) .
- Segmenting the customers so to establish innovation plans per customer segment.
- Etc.

In ISO 15504 there is a process "Requirements Elicitation" which focuses on

- Gathering the requirements and analysing different scenarios of implementation.
- Deeply analysing requirements and planning customer releases in advance.
- Understanding customer expectations and tracing the requests and controlling changes.
- Etc.

Both definitions are similar. However, innovation adds skills factors such as innovation forecasting, relationship strategies, and knowledge gathering and exploitation.

You could actually have the best defined process, but if you fail to understand the innovation strategy of your customer, you will miss the direction and you do not get the follow up contract on the next generation of products.

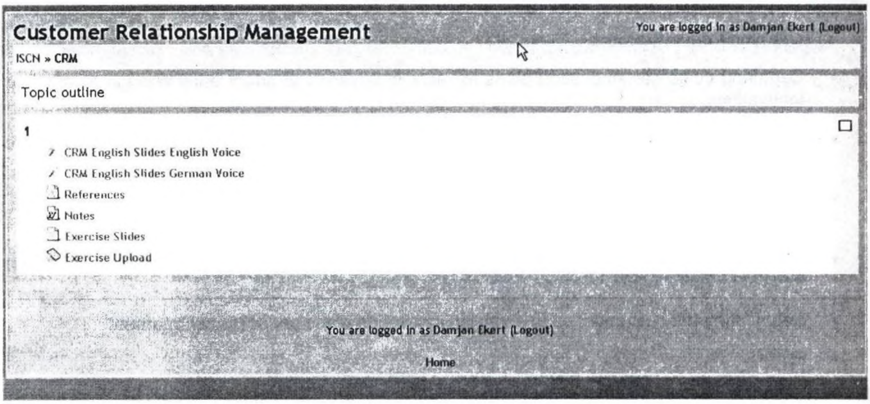
To implement this in the group we created virtual learning teams working as follows:

1. The partners can access the portal for process assessment and assess themselves against e.g. requirements elicitation, and share materials. (http://portal.sogrates.de)
2. The moderator of the learning team benchmarks the results and identifies synergy leaders (best in team on process level)
3. The partners access the skill portal for innovation managers (https://www.iscn.com/projects/piconew_skill_cards/index.html) and self assess themselves against CRM (Customer Relationship management Skills).
4. The moderator of the learning team benchmarks the results and identifies synergy leaders (best individual results of team members)
5. All partners access a web based training area (integrated with skills portal) to attend a course online (presentation with sound, homework, chat, Internet based tel-conferencing) and leran

about additional innovation criteria

6. All partners do a homework and upload materials for review to the moderator
7. The moderator reviews the materials and uploads recommendations for joint improvement actions covering both aspects, processes and innovation skills.
8. In an online conference (chat based with all materials available , and with a sub-group having Internet phone support) a proposed action list is discussed.

After the 9 steps it is up to the partners how they implement this in their own firm. The implementation experiences are exchanged via the task forces meetings.



Customer Relationship Management

You are logged in as Damjan Ekert (Logout)

ISCN » CRM

Topic outline

1 □

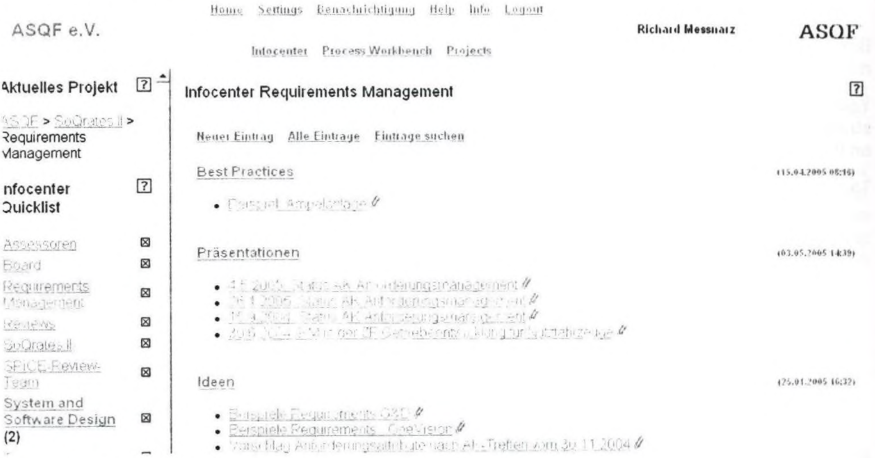
- CRM English Slides English Voice
- CRM English Slides German Voice
- References
- Notes
- Exercise Slides
- Exercise Upload

You are logged in as Damjan Ekert (Logout)

Home

Figure 2: Course Management Area to Listen to Presentations in English or German

The whole discussions, the homework of partners, the reviews, the proposed action lists and the results of actions inside the organisations are then shared in a knowledge pool through a jointly shared e-working area.



Home Settings Benachrichtigung Help Info Logout

ASQF e.V. Richard Messnarz ASQF

Infocenter Process Workbench Projects

Aktuelles Projekt [?]

ASQF > SoQrates II > Requirements Management

Infocenter Quicklist [?]

Assessoren	☒
Board	☒
Requirements Management	☒
Reviews	☒
SoQrates II	☒
SPICE-Review-Team	☒
System and Software Design (2)	☒

Infocenter Requirements Management [?]

Neuer Eintrag Alle Einträge Einträge suchen

Best Practices (15.04.2005 08:16)

- Beispiel_Ampelanlage

Präsentationen (03.05.2005 14:39)

- 4.5.2005_Status AK Anforderungsmanagement
- 26.1.2005_Status AK Anforderungsmanagement
- 17.9.2004_Status AK Anforderungsmanagement
- 20.5.2004 RM in der ZF Getriebeentwicklung für Nutzfahrzeuge

Ideen (25.01.2005 16:32)

- Beispiele Requirements G3D
- Beispiele Requirements_GeoVision
- Vorschlag Anforderungsattribute nach AK-Treffen vom 30.11.2004

Figure 3: Information and Team Sharing pool (in German)

2 *Background about the Underlying Systems*

2.1 Multi-user Assessment Portals (CREDIT) and Learning Systems

The CREDIT platform offers the management and performance of assessments, evaluation and reporting. It offers online browsing, online self assessment, online formal assessment, and capability profiles.

What makes the system (it is called Capability Adviser) special is the fact that

- Beside supporting ISO 15504 or CMMI assessment of processes (portal.soqrates.de) it also offers skills assessment of job roles (e.g. assessing skills of innovation managers). This means that with one system you can catch two views, the process view and the individual skills.
- It works as a multi-user portal so that many assessors are logged in at the same time and can share their views and the ideas and comments and knowledge from all assessors are kept in a knowledge SQL database. This is different to the boring (information loosing) assessment approach where only one set of evaluations and comments (by the lead assessor) are kept.
- It can be integrated with an online learning management system (www.moodle.com) so that if projects are weak in certain processes (e.g. requirements elicitation) they can attend online courses and join online experience forums.

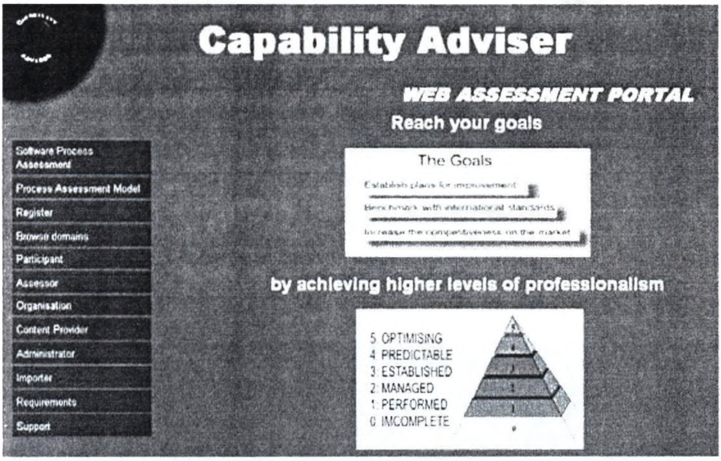
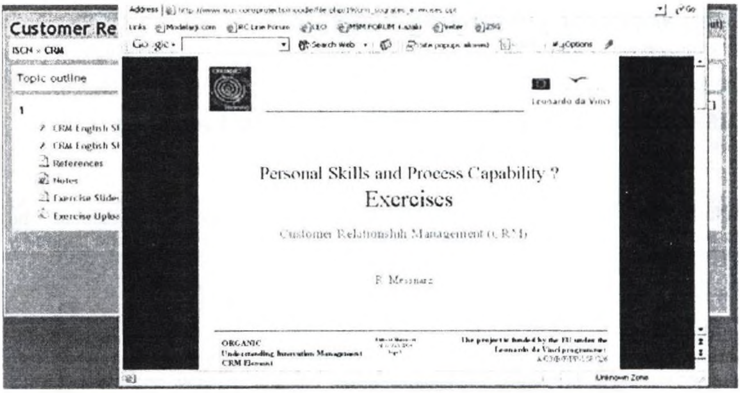


Figure 4: Home Page of Assessment Portal



Customer Re

ISCN » CRM

Topic outline

1

CRM English Sl

CRM English Sl

References

Notes

Exercise Slides

Exercise Uplo

Leonardo da Vinci

Personal Skills and Process Capability ?

Exercises

Customer Relationship Management (CRM)

R. Messnarz

ORGANIC
Understanding Innovation Management
CRM Element

The project is funded by the EU under the Leonardo da Vinci programme

Figure 5: Example Online course with Flash/Sound Exercises Etc.

Administrator

An administrator can create organisation accounts and maintain an assessor pool.

Organisation

An organisation administers projects, assessments, an assessor pool, and can tailor assessment models (to their organisational needs).

Content Provider

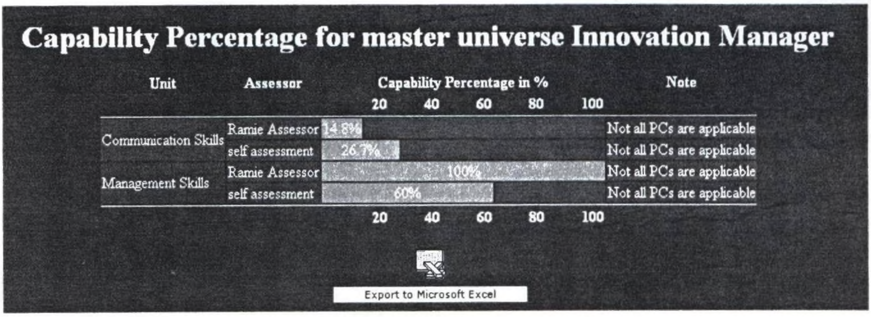
A content provider maintains the underlying content of the assessment models and the link to the integrated LMS (learning management system).

Assessor

An assessor sees all assigned assessments in his work bench, assesses processes and/or skills, provides notes, evaluates evidences provided, share knowledge with other assessors, prints profiles and assessment records.

Participant

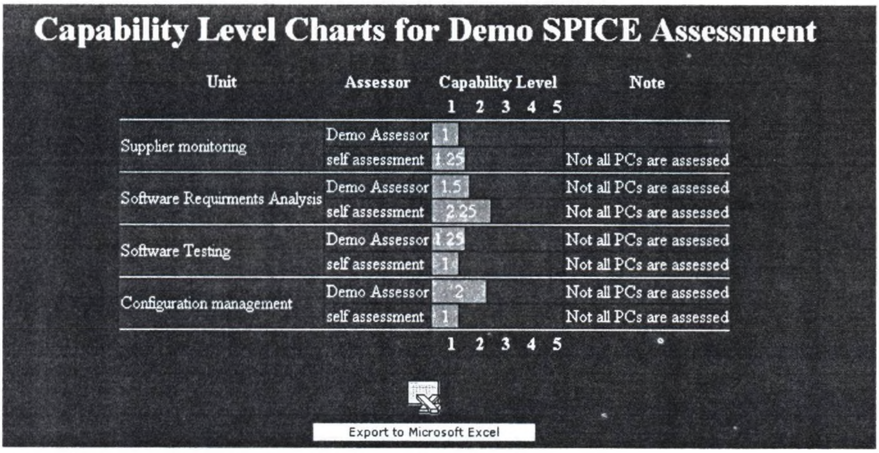
A participant can do a self assessment, upload evidences and link them to specific processes and/or skills, and can attend learning courses and experience forums connected to specific process areas.



Capability Percentage for master universe Innovation Manager

Unit	Assessor	Capability Percentage in %	Note
Communication Skills	Ramie Assessor	14.8%	Not all PCs are applicable
	self assessment	26.7%	Not all PCs are applicable
Management Skills	Ramie Assessor	100%	Not all PCs are applicable
	self assessment	60%	Not all PCs are applicable

Figure 6: Example Skills Profile for 2 Selected Innovation Skills Units



Capability Level Charts for Demo SPICE Assessment

Unit	Assessor	Capability Level 1 2 3 4 5	Note
Supplier monitoring	Demo Assessor	1	
	self assessment	1.25	Not all PCs are assessed
Software Requirments Analysis	Demo Assessor	1.5	Not all PCs are assessed
	self assessment	2.25	Not all PCs are assessed
Software Testing	Demo Assessor	1.25	Not all PCs are assessed
	self assessment	1	Not all PCs are assessed
Configuration management	Demo Assessor	2	Not all PCs are assessed
	self assessment	1	Not all PCs are assessed
		1 2 3 4 5	

Figure 7: Example Capability Level Profile for Selected Processes

Web sites using the portals and available in public are

http://portal.sograted.de, http://www.iscn.com/projects/epi_skill_portal/

The system is used by major companies in automotive, finances and security and in the telecom field with between 2000 and 95000 employees at the moment.

2.2 Innovation Manager Skills (ORGANIC)

ORGANIC (www.innvationmanager.org) based on 3 Europe wide studies, such as -

- 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]).
- 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.
- 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].

ORGANIC developed a skills et, self assessment portal for managers, and a training programme for 20 learning objectives of innovation managers.

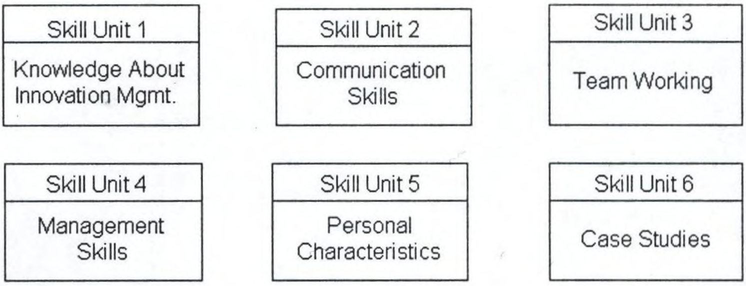


Figure 8: Skill areas for Innovation Managers

Innovation Management Learning elements are

Skill Unit 1	Knowledge Management, Skills Management, Customer Relationships Management, Networking, and Market Research
Skill Unit 2	Literacy in E-Skills, Reporting Skills
Skill Unit 3	Team Communication, Conflict Management, Distributed Team Mgmt., Motivation building, and Cross-cultural factors
Skill Unit 4	Corporate Innovation Mgmt., Innovation factors in project management, in process management, and in risk management
Skill Unit 5	Personal Characteristics, Learning Culture, Cross-cultural learning factors
Skill Unit 6	Knowledge Pool and Case Studies on Innovation Success

All materials have already been developed, a Europe wide course programme has been established.

2.3 SPI Knowledge and Synergy Sharing Initiative (SOQRATES)

SOQRATES (www.soqrates.de) has been originally formed as a group if Bavarian firms. From 2004 onwards firms fro all other German countries joined. Leading firms such as Continental TEMIC, ZF, Panasonic Automotive, G&D, etc. contribute to the joined task forces.

The initiative started by coaching ISO 15504 assessments in a group of 16 companies who jointly shared their experiences.

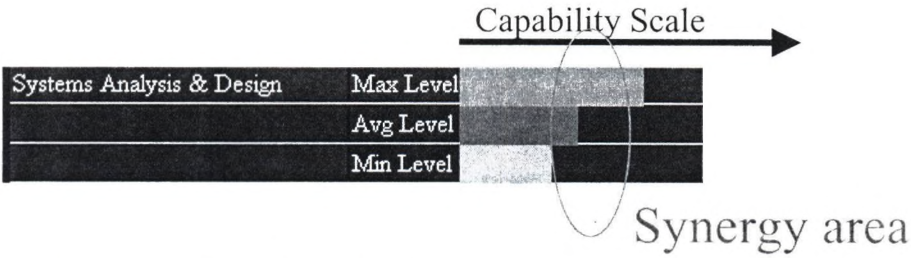


Figure 9: Benchmarking as a Tool to Identify Synergy Leaders

Assessment results have been compared, synergy leaders have been identified, and in areas of joint interest cross-company experience teams have been set up.

Three major areas of common interest have been identified and task forces are operative since 1993-

- System Design
- Requirements analysis and traceability
- Testing

From 2005 onwards we added

- Safety (ISO 61588)

And we started to form innovation teams for sharing knowledge on specific topics and to widen the scope of ISO 15504 to include the success criteria outlined in ORGANIC as well.

3 Results

The initiative is still going on.

- The workshop (online) resulted in a set of agreed best practices to implement innovation principles in Requirements Elicitation.
- Companies then analysed their firm and published heir proposed concepts.
- At the moment the exchange of ideas and knowledge is being moderated to agree (gathering the ideas of cross-company expertise) a best practice way to implement
- During the implementation the team will still stick together and learn fro each other to be better than competitors (acting as a knowledge group)

The detailed ideas of what the team agreed as being the nest practice cannot be published because it's the IPR of the task force.

What the paper wants to emphasise is that

- Cross-company learning is possible
- Cross-company sharing of best practices is a competitive factor
- Innovation s a field where through cross-company ideas new concepts can be unleashed

If you are capable of German, and you want to join this initiative you my contact the moderator, Dr Richard Messnarz, rmess@iscn.com .

4 Literature

[1] M. Biro, R. Messnarz, A. Davison (2002) The Impact of National Cultures on the Ef-fectiveness of Improvement methods - The Third Dimension, in Software Quality Professional, Volume Four, Issue Four, American Society for Quality, Sep-tember 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, Wash-ington, 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'Keeffe, 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

5 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 RegionalTechnology 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 (Ameri-can 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.mdisq.de) initiative, and he is the technical moderator of the SOQRATES initiative (www.soqrates.de).

6 Picture of Latest Team Meeting – May 2005

Understanding IEC 61508 Through a Semantic Web Ontology Language

Micheal Mac an Airchinnigh[1]
[1]ISCN LTD, Bray, Dublin, Ireland
Tel: +353 1 205 0020, Fax: + 353 1 205 0021, Email: mmaa@eircom.net

Abstract

ISO 61508 is the standard-to-be for safety systems based on the concept of Safety Integrity Levels (SILs 1-4). With increasing SIL level the probability of occurrence of a dangerous failure must be decreasing. The standard describes (highly) recommended practices for each SIL level and many of them at higher SIL levels (>=3) describe the need of formal methods. This tutorial will use 18 years of formal methods experiences in different fields to investigate: What is the way of thinking when moving from coding views to formally proved design views? How do we relate Product Safety Terminologies to Evidence for Certification on the Semantic Web? and Why is the Semantic Web Ontology Language (for Web Services) (OWL-S) a practical way forward? World-Wide ISO 61508 compliance will be greatly facilitated by knowledge representation (KR) of relevant concepts of safety and corresponding product properties (in Description Logic of OWL) on the Semantic Web. In all safety-critical fields such as automotive, aerospace, medical, etc. there is an increasing demand for ease of compliance, driven by the manufacturers. Practical examples to assist will be given using Protégé OWL from Stanford University..

Keywords

Process Improvement, Semantic Web, IEC 61508

EuroSPI 2005 Proceedngs 11/05
ISBN 963 8431 94 6
John von Neumann Computer Society
Not-for-profit Professional Organisation for the Information Society - URL: www.njszt.hu