EuroSPI

European Software Process Improvement

2002

18th – 20th September
Nuremberg · Germany

European Conference
on Software
Process Improvement

18th–20th September 2002

PROCEEDINGS

University of Applied Sciences

GEORG-SIMON-OHM
FACHHOCHSCHULE
NÜRNBERG

GI

Software
Stützpunkt
Region Nürnberg

SOFTWARE
OFFENSIVE
BAYERN

EuroSPI 2002

— Proceedings —

September 18 – 20, 2002, Nuremberg, Germany

The papers in this book compromise the proceedings of the conference mentioned on the cover and title page. 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 the publisher.

EuroSPI
c/o ISCN Coordination Office
Florence House, 1 Florence Villas
Bray, Co. Wicklow,
Ireland

Printed in Germany

ISBN 3-00-010074-1

Table of Contents

Session E-IV

Session E-V

Session E-VI

Session E-VII

Session E-VIII

Session E-IX

Session E-X

Committees

Conference Chair

Richard Messnarz, ISCN IRL & ISCN A

Programme Committee

Micheal Mac an Airchinnigh, Trinity College & ISCN, IRL
Gualtiero Bazzana, Onion Technologies, It
Miklos Biro, Budapest University of Economic Sciences and Public Administration, HU
Mads Christiansen, Delta, DK
Howard Duncan, Dublin City University, IRL
Taz Daughtrey, Chief Editor of the Software Quality Professional Magazine, USA
John Elliott, DERA, UK
Tim Hind, AXA-Sunlife UK
Bernd Hindel, ASQF e.V., D
Theresa Hunt, Programs Chair of the ASQ Software Division, USA
Ioana Ene, Onion Technologies, It
Jorn Johansen, Delta, DK
Carsten Jorgensen, Delta, DK
Karl Heinz Kautz, Copenhagen Business School, DK
Patricia McQuaid, Americas Program Committee Chair for the Second (2000) and Third (2005) World Congresses on Software Quality, USA
Nils Brede Moe, Sintef, No
Messnarz Richard, ISCN IRL & ISCN A
Risto Nevalainen, STTF, Fin
Carsten Nøkleby, Grundfos A/S
Poul Grav Petersen, Novo A/S
Dag Sjoberg, University of Oslo, No
Terttu Orci, Royal Inst. of Technology & Stockholm University, S
Tor Stalhane, Norwegian University of Science and Technology, No
David Teague, British Telecom, UK
Timo Varkoi, Pori School of Technology, Fin
Yingxu Wang, University of Calgary, Canada

Welcome Addresses

Bernd Hindel
President ASQF e.V.

Richard Messnarz
EuroSPI Chairman

Helmut Gierse
President of Siemens Group Automation and Drives

Herbert Eichele
Rector Fachhochschule Nuremberg

Prof. Dr. Bernd Hindel
President ASQF e.V.

Welcome Addresses

Today, modern upper-class cars are equipped with more software than the Saturn rocket used in the Apollo -11 Mission. 33 years after its flight to the moon, software is of vital importance not only to spacemen. We encounter it in various applications and appliances in the household, at the job or during recreation.

A large portion of our everyday-life is dependent on programs that function. As a result, requirements on the quality of software are constantly rising. The demand for software quality solutions and IT professionals with thorough knowledge and experience in the area of software-process-improvement will therefore also rapidly continue to increase. A reason why the ASQF e.V. has been committed to further education since it was founded.

In IT to be up-to-date is of more importance than in any other branch of the economy. Since 1997 CONQUEST has been pushing forward the exchange of experience and information on quality engineering in software technology. EuroSPI, which has been taking place since 1994, aims at the same direction. This year EuroSPI and CONQUEST have united their strengths, combining the leading European conference in the area of software process improvement with the major German conference on software quality. This double conference, "CONQUEST/EuroSPI 2002" in Nuremberg, is a very special highlight for the international IT industry.

"CONQUEST/EuroSPI 2002" offers a greatly upgraded program, the latest information from the IT scene and many opportunities to exchange experience and establish new contacts: Twelve tutorials, numerous sessions, keynotes, industrial sessions, a tool exhibition, as well as a social event during the following days. In addition the ASQF®-Certified-Tester Certificate-Holder-Group will be founded on the first day.

Finally I'd like to wish you all a pleasant stay in Nuremberg at "CONQUEST/EuroSPI 2002". Make use of this double conference as a platform for further discussions, new ideas and profitable experiences.

B. Hindel

Prof. Dr. Bernd Hindel
ASQF President

Dr. Richard Messnarz
EuroSPI Chairman

Welcome Addresses

On behalf of EuroSPI I welcome all participants to our annual conference in Nuremberg.

EuroSPI is a partnership of large Scandinavian research companies and experience networks (SINTEF, DELTA, STTF), QinetiQ as Europe's largest research center, the ASQF as a large German quality association, the American Society for Quality, and ISCN as the coordinating 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 9th 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.

Conferences have been held so far 1994 in Dublin (Ireland), 1995 in Vienna (Austria), 1996 in Brighton (UK), 1997 in Budapest (Hungary), 1998 in Gothenburg (Sweden), 1999 in Pori (Finland), 2000 in Copenhagen (Denmark), 2001 in Limerick (Ireland), 2002 in Nuremberg (Germany). EuroSPI is planned well in advance with 2003 in Birmingham (UK), and 2004 in Trondheim (Norway).

This year we joined forces with the CONQUEST 2002 conference to organise a major European event focussing on practical experiences with quality management and process improvement.

Recently EuroSPI established an on-line experience library with 170 well elaborated and peer reviewed experience reports. It is based on the EuroSPI proceedings from 1998, 1999, 2000, 2001 and 2002. It offers experiences for major industry sectors such as telecom, automotive, aerospace, and research, and groups the experiences into 18 major topics.

We hope that you will benefit from this conference by receiving and exchanging experiences which can be fruitful for your own organisation.

Dr. Richard Messnarz

EuroSPI Chairman

Helmut Gierse
President of Siemens Group Automation and Drives

Welcome Addresses

Today, the benefits to the customer and the level to which many items of equipment and systems are actually accepted are greatly dependent on software. This is true in the areas of communication and information as well in industry, medical engineering and transport.

And the same applies to automation systems. Here at Siemens Automation and Drives, more than two thirds of our expenditure on R&D is accounted for by software, the main examples being engineering tools, operator control and monitoring systems as well as simulation and runtime software. But, to an increasingly greater extent, hardware functions are also being replaced by intelligent software solutions - in mechatronic systems, for example. For Siemens Automation and Drives, software and IT are and will continue to be key factors in the acquisition of business, which presently amounts to 9 billion euros.

The speed of software innovation continues to be unabated. More than ever before, it is the motivated, qualified and knowledge-hungry staff that are determining who actually succeeds in the face of world-wide competition. As a market leader in the field of automation, we promote activities which strengthen and expand the software expertise of both our staff and managers. And there are no better ways to do this than by exchanging experience, best practice sharing, comprehensive knowledge management and continuous professional training.

The ASQF (Working Party for Software Quality in Franconia), in which many Siemens employees are also involved, has been doing exemplary work here for many years now. With CONQUEST, it has set an example which has been perceived far beyond the region and which is internationally recognized for its outstanding achievements.

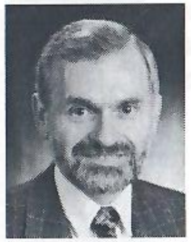
I am firmly convinced that CONQUEST 2002 will continue the successes of preceding events and will lay the foundations for further leaps ahead in the area of innovation.

As patron of this conference, I hope that all the participants will make interesting new contacts, encounter stimulating new ideas and leave with lasting impressions.

Helmut Gierse

Helmut Gierse

President of Siemens Group
Automation and Drives



Prof. Dr. H. Eichele
Rector
Georg-Simon-Ohm-Fachhochschule
University of Applied Sciences

Welcome Addresses

I am pleased to welcome all participants and organisers of CONQUEST 2002 and EuroSPI in the premises of Georg-Simon-Ohm-Fachhochschule Nürnberg – University of Applied Sciences. Meanwhile it is the 6th time we host CONQUEST in our 179 years old university.

Our university offers the full spectrum of engineering degree programs and expects to have 8000 students enrolled by Oct. 1, 2002. The field of information and communication technologies is covered by bachelor-, diplom- and master-degree programs in Computer Science, Computer Science in Business Applications, Data- and Information-Technologies, Microelectronics, Telecommunications, Software-Engineering and Media Technology. Of course, aspects of dependability, software quality and software process improvement are appropriately covered. This is a must as faulty software systems not only cause monetary losses but also endanger safety and life.

On October 1 the new semester will commence. With more than 7400 applications for admission we again have an increase of more than 10% compared to the previous year. Only part of these young people actually may commence their studies this autumn. Four out of five applicants in Computer Science and in Computer Science in Business Applications must be rejected despite the university's past and ongoing big effort to increase the corresponding teaching resources, mostly by shifting resources internally and by a private-public-partnership project called I.C.S. International Cooperative Studies running special bachelor degree programs in IT and Computer Sciences.

However, the IT-labour market resembles a roller coaster. The boom has followed a crash landing for many companies and many IT-experts lost their job already. IT-job advertisements dropped by 70% in the first half of this year. Only under optimistic assumptions the Council of European Professional Informatics Societies Cepis sees 1.5 million additional IT-jobs being created until 2005. We shall see, but should be prepared ...

Software influences the world. Universities and job experienced must accept their responsibility. Initiatives like CONQUEST and EuroSPI are invaluable. Congratulation and many thanks to the initiators, sponsors and organisers.

I wish you all a pleasant stay at Georg-Simon-Ohm-Fachhochschule Nuernberg – University of Applied Sciences, valuable presentations and interesting discussions.

Prof. Dr. Herbert Eichele
Rector

Session E-I

Experiences with Software Process Assessments

Towards a Framework for Managing Virtual Offices

Jenny Coady B.Sc. Email: jcoady@wit.ie
Dr. Larry Stapleton Email: lstapleton@wit.ie
I.S.O.L.
Waterford Institute of Technology, Rep. Of Ireland

Brian Foley Email: brian.foley@tecnet.ie
TECNet

Abstract:

Advanced information and communication technologies have enabled new options in organizational structure and processes, e.g., global "virtual" teams, work-at-home options, inter-organizational teams or alliances as in the "virtual" organization. The word virtual has become a potent buzzword and as such, is freely applied to many situations, with many meanings. As a result, like other buzzwords, it is in danger of meaning nothing (Watson-Manheim, Crowston & Chudoba (2002)). To compete effectively in today's fast paced, rapidly changing business environment, firms must organise people and processes to enhance responsiveness and flexibility. Researchers have begun to realise the critical importance of delivering information technology solutions, which favour distributed work scenarios. Research in CSCW has accomplished a great deal in this area but, besides email and notes, many workflow support solutions are not widely taken up by industry. This paper proposes an approach which promotes a role-scenario based view of organisational information processing, as opposed to the mechanistic, functional approach of traditional IT methodology and concomitant IT technical solutions. It outlines a combined methodological/technical solution, which has been developed and shows how this solution can resolve many of the problems faced by virtual work communities, enabling them to achieve significant benefits in terms of productivity, motivation and costs and reports its findings for the SPI network.

1 Introduction

The word "virtual" describes work that spans one or more discontinuities. The term has been applied to work where people are in discontinuous physical work locations, where work is done in discontinuous time frames, where people have discontinuous organizational affiliations, and others (Watson-Manheim, Crowston & Chudoba (2002)).

Community is an important aspect of life for most people. Cooley (1983) states that all normal humans have a natural affinity for community. He suggests that the primary factor inhibiting the formation of communities, no matter what their scale, is that they are difficult to organise. Extending the moral ideas inherent in nearly all individuals to the notion of community requires a system or institutional framework. The development and maintenance of such institutions sap the energy of the members of the would-be community and confuse the moral ideals inherent in the notion of community with the project of the institution itself (Fernback & Thompson (1995)).

As new communication technologies are developed, it is worth examining the new relations and their potential for new or renewed relationships. The online world of computer-mediated communication (CMC) is one of those new ways for humans to relate to one another, and it has been growing rapidly for some years (Fernback & Thompson (1995)).

Alongside these developments researchers have been advocating the need for a fundamental revision of software technology support in organisations for some time (Wood & Wood-Harper (1993)). However, few have proposed a basis upon which this progress can be made. This paper recognises the need for new technological forms, which recognise that organisational members act in fluid ways, that organisations are moving towards flexible forms in virtual space. It thus attempts to address this gap in the academic literature, as well as in common practice. The paradigm proposed here is based upon a model of IT, which views organisations in terms of roles operating in particular settings or ‘scenarios’. The remainder of this paper briefly reviews a new solution based upon this role-scenario paradigm.

2 The differences between remote and traditional working environments

Over the years researchers have highlighted the need for an alternative approach to support technology for organisational information processing. Indeed, the relationship between organisations and the technologies they use to process information has been, and continues to be, a major problem for researchers (Stapleton (2002)). In the nineteen eighties this problem was recognised as being driven by the complexities and ambiguities inherent in organisational decision making, and this remains as difficult to understand as ever (Markus & Robey (1988), Stapleton (2000)). In response to the various difficulties associated with the software and hardware paradigms inherent in information technology since its inception, some authors have recognised the need for radical new approaches to the way we think about information systems in organisations. Hedberg & Jonsson (1978) remind us that organisations are highly fluid and need IT support which helps them to evolve and adapt, rather than be provided with functional systems based upon a mechanistic view of organisations in which people operate according to strict procedures and guidelines. Although Hedberg & Jonsson (1978) threw down an important gauntlet for researchers, few took up the challenge. In the 1990’s some researchers returned to these fundamental questions of IT support and reviewed many of the basic principles that underlie IT provision. For example, Wood & Wood-Harper (1993) argued for a ‘radical rethink’ of IT support and suggested that software researchers needed to create completely new paradigms for information technology. Again, this call to arms largely fell upon deaf ears.

Scholars such as McLuhan (1964) have noted that the development of electronic communication technologies has essentially abrogated space and time so that we live in a boundless “global village”. Boorstin (1978) argued that communication technology creates ties; bind nations into a new type of community, which he terms the “Republic of Technology”. This community is one of shared utopian experiences; he states

> “with crushing inevitability, the advance of technology brings nations together and narrows the differences between the experiences of their people”
>
> (p. 6)

The most recent communication technology development within the post-industrial era is CMC. Comprised of different systems such as Electronic mail, bulletin board systems, and real-time chat services, CMC is both an interpersonal, one-to-one medium of communication as a one-to-many or even many-to-many form of mass communication. With an estimated 25 million CMC users worldwide, and this still growing (Calem (1992)) computer mediated communications have the potential to affect the nature of social life in terms of both interpersonal relationships and the character of the community. Virtual communities encompass the economic, political, social and cultural dimensions of community postulated by Van Vliet & Burgers (1987).

Meyrowitz (1985) asserts that community has been affected by electronic media's undermining of the relationship between location and access to information. He states:

> "Many categories of people …. were once "naturally" restricted from much social information be being isolated in particular places. The identity and cohesion of many groupings and associations were fostered by the fact that members were "isolated together" in the same or similar locations…. Now however, electronic messages…. democratise and homogenise places by allowing people to experience and interact with others in spite of physical isolation. As a result, physical location now creates only one type of information-system, only one type of shared but special group experience"
>
> (p 143-144)

Mehrabian (1971) argues that a considerable percentage of peoples' intent is conveyed by tone of voice and facial expression. Ordinarily when people communicate they are not simply exchanging information, but projecting an image of themselves. Technology reduces the fear of appearing foolish to others by removing the reminder of a critical audience (Stapleton (2001)). Electronic group dynamics are not the same as face-to-face and tend to have unpredictable consequences for those attempting to streamline organisational communication. Electronic groups often find it more difficult to reach consensus and have to work harder to achieve it.

3 Roles & Scenarios

Most modern organisations cannot function effectively without information systems support, but they struggle to live with them in a harmonious relationship (Mc Bride (1999)). They experience a series of gaps, or mismatches, associated with the assumptions, which underpin information systems practice as regards organisational structure, dynamics and the various realities with which organisations must grapple. There are also conflicts associated with the system models, the resulting technological artefacts and many other aspects of the IT domain. Indeed, it has been shown on many occasions that our approach to eliciting and understanding technological and system requirements is typically at variance with the social environment the systems are intended to support (Stapleton (2000)).

Early IS thinking was concerned only with the interactions of the parts of the things to be explained. Systems' thinking is similarly concerned, but it is additionally occupied with the interactions of that thing with other things within its environment, and with its environment. Systems' thinking is also concerned with the functional interaction of the parts of a system. This orientation derives from the preoccupation of systems thinking with the design and redesign of systems. In systems design, system components identified by an analysis of the functions to be performed by the whole are intended to fit each other so as to work together harmoniously as well as efficiently and effectively (Hamilton (1997)). However, when it comes to the interaction between users and the technology, this is rarely the case (Davenport (1998), Stapleton (2001)). Researchers have argues that the mechanistic approaches which typify information systems methods are based upon a functionally rationalistic paradigm which needs to be reviewed and revised. This paper proposes the adoption of roles & scenarios (i.e. a model of the social/task setting in which roles are played out) as basic modelling constructs. These constructs are then used to determine system functionality in terms of an organisation of documentation and contracts. This concept drives information systems thinking in a new direction and fundamentally plants both the design process and the technology itself in the organisational space, both in terms of process and language. So, whilst in itself role-based methods and scenario modelling (as separate approaches) are not new, the unique combination adopted here, and the configuration of the methodology with the technology is highly innovative and improves process management.

Scenarios can be understood in a number of ways. They can be a sequence of activities, or a more or less richly branded structure of such sequences. They can represent parallels or alternatives, or various intermediate options (Alexander (1999)). Branches can represent alternatives or parallels, or various intermediate options.

Scenarios can assist in all phases of the systems life cycle. They can clarify systems scope; help to identify stakeholders, situations, and needs. They can also aid in the organisation of requirements, guide design and coding and also provide scripts for testing. But throughout the cycle their main contribution is perhaps simply to improve project communications between all the groups of people involved (Alexander (1999)).

3.1 Modelling Roles & Scenarios in a Virtual Environment

The model is a common language: the users who are providing the information about the system readily understand it. Using working models, users and analyst's work together to reach an identical understanding of the requirements. Once the model is agreed, the system is implemented by building a real-world version of the model (Kerth (2001)). In Systems and Software Engineering, there are perhaps more schools of thought concerning IS notations and method that on any other topic (Alexander (1999)). One key function of scenarios is to help people communicate about the systems they are involved in. The choice of representation would ideally be a pragmatic one - people selecting the best for their situation.

This paper proposes a role-centred approach for primarily pragmatic reasons. Firstly, people identify themselves better with roles than functions and thus feel more able to understand the system and, hopefully, helping them to focus upon to role ownership. Evidence for this appears in (Warboys, B., Kawalek, P., Robertson, I., & Greenwood, M. (1999)). Role models design roles and identify explicitly information flows between them. This information flow based architecture is needed for later implementation in an remote working organisation (such as over the intranet or internet), which is the essence of a virtual work setting. Workflow based structures can be easily derived from role models following a defined set of rules which can be pre-determined and are widely used in industry (Warboys, B., Kawalek, P., Robertson, I., & Greenwood, M. (1999))

A scenario workshop can represent a sequence of tasks directly by role-play. A task can be described verbally by a player, or acted out in a variety of ways. The execution path can be symbolised by passing a token, such a juggling a ball to the player whose turn it is next. Both singly- and multi-threaded scenarios can be simulated in this way. Stakeholders see, often for the first time, how their tasks relate to the overall process, and many misconceptions and confusions can be banished in a playful workshop (Alexander (1999)). A similar approach was adopted in object oriented BPRE as laid out in Graham (1994), but has received little attention in the context of roles as specified here.

One of the best-known representations is the idea of Use Cases and UML (Jacobson (1992)). This supposes that the world of business is divided into neatly addressable cases where a user interacts with a system. Graphically a stick person connected to an elliptical bubble depicts it. Each bubble depicting a use case, and a number can be arranged into a list to form a diagram. The norm is to arrange them vertically, which does not imply sequence and this can visually be misleading. There has been much debate about whether a use case is simply another synonym for scenario (Alexander (1999)). The question could be answered by obtaining a precise definition of both terms, were it not for the elasticity of both terms. Hence UML notation can be helpful, but currently only goes part of the way towards resolving modelling differences.

4 The TeamWork Project:

The TeamWork initiative grew out of developments in the automotive industry in the early 1990's (Messnarz et. al. (1999)) Large firms in this sector realised that in supply chains they work together in kinds of virtual organisations and that borders could not be easily drawn between enterprises. This mirrored experiences elsewhere in industry as outlined in Darnton & Giacoletto (1992), Davidow & Malone (1993), Drucker (1988, 1993) and elsewhere.

The TeamWork research project involves three main axes, as shown in figure 1. This paper focuses upon the information systems development support aspects of the project, and therefore addresses itself to the Methodology and Technology components of TeamWork. It recognises the importance of new organisational forms and is targeted at communities of workers whose members come together for specific projects, but may not be part of the same organisation or inhabit the same geographical space. The power of such communities, and the risks involved in such organisational forms, are outlined elsewhere (Sproull & Kiesler (1992)).

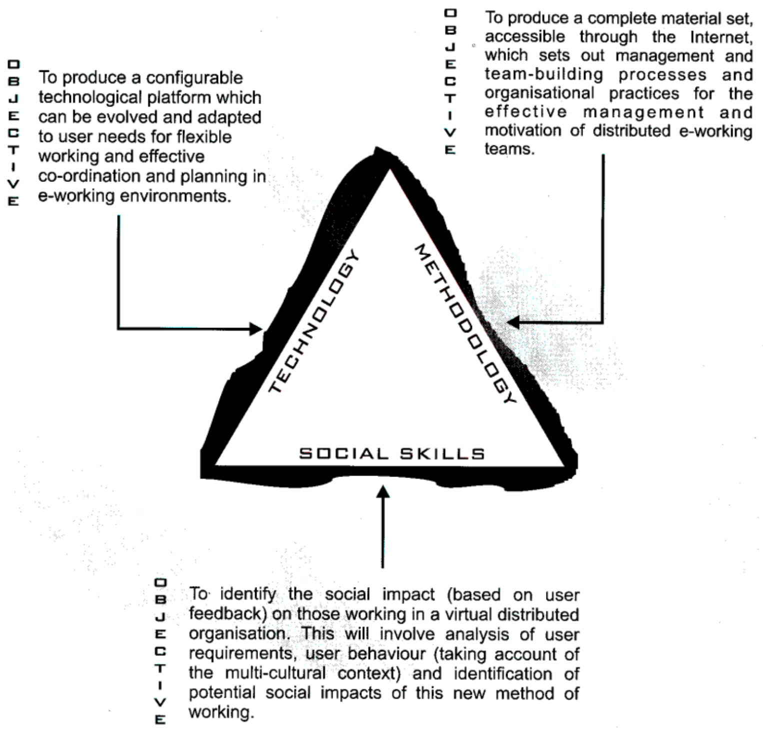


Figure 1: The TeamWork Research Project – Major Axes

TeamWork delivers a holistic solution based upon the concepts embodied in a role/scenario organisational perspective. This holistic solution comprises BESTREGIT as a methodological component and NQA as an information and coordination technology based upon organisations as comprising roles and scenarios.

The basic advantages of this approach to information systems development and organisational support include the following:

1. People will be more aware of their responsibilities and clearly understand communications interfaces to other virtual organisation members. Role-based solutions are specifically adopted with this in mind.
2. New staff can be more easily integrated through clear role assignments, skill acquisition for the roles and clear appreciation for information and communication flows within the communi-

ty. Sproull & Kiesler (1993) show that this is a specific opportunity presented by CSCW systems which support virtual communities.

3. Information technologies can be deployed which support virtual community communications, documenting of activities and configuration of results in a highly effective manner. This is primarily achieved by making the communications interfaces highly visible, a central deficiency of many current CSCW and ISD approaches which focus on highly functionally rationalistic views of organisational behaviour (Stapleton (2000), Stapleton & Murphy (2002)).

Studies examining the effectiveness of this paradigm suggest that these advantages do exist where such an approach is adopted. Indeed preliminary studies of software development projects conducted in Ireland, Austria, Spain and Germany show potential benefits to firms, which adopt technologies based upon this approach (Messnarz & Tully (2000); Messnarz, Stubenrauch, Melcher & Bernard (1999)). In particular, the studies identified the following specific benefits to firms investigated:

1. 50% increase in the effectiveness of new staff integration processes
2. 67% increase in team motivation in ISO9001 projects
3. 67% reduction in maintenance and 50% rise in productivity because of role decomposition in virtual teams. The software development projects in virtual communities showed that the allocation and clarification of highly visible roles
   a. allowed for the effective distribution of tasks (both asynchronous and concurrent task activities)
   b. reduced the propensity towards monolithic program architectures where everyone was responsible for the same software without a clear distinction of interfaces and modules

Technologically TeamWork is constructed using Hyperwave's Knowledge Management solution. The key innovation is in the organisational coordination utilities supplied via a combination of NQA technology and the BESTREGIT Methodology.

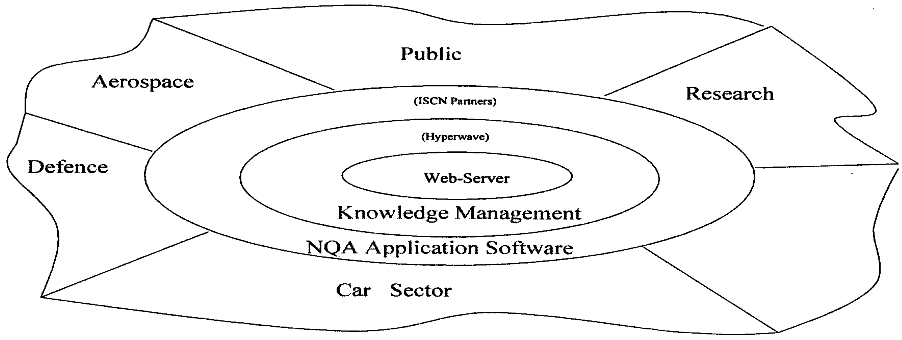


Figure 2. The Basic Technology Platform Architecture

4.1 The Development of BESTREGIT:

The importance of the Best Regional Innovation Transfer (BESTREGIT) for the modelling organisation is that it can act as a trigger to become an Innovative company through a life long learning philosophy. The research approach adopted in the TeamWork Project (2001) had to do with formally examining the work scenarios behind an organisations business fields. Through this analysis TeamWork came to a greater understanding of the roles people in organisations play, how those roles interact to produce results, and how TeamWork could track the progress of these results. Once a team view, defined to be a commonly understood mission and goals (TeamWork (2001)), has been achieved, then the next logical step is to improve the work scenarios so that those goals can be met. In the current day work goals are as such that work can no more be done alone, by a single individual (TeamWork (2001)). Communication and work coordination is needed, and scenarios are designed aiming at productivity.

A major success factor in the transformation from goal trees into TeamWork-based processes is to set the right priorities for the organisation (TeamWork (2001)). In real business cases, unlike the BESTREGIT research, it is often not possible to model all work scenarios, often due to time demands and limited resources. However if a business field with a typical work scenario is selected then there is a certain "re-use factor". Modelling of these fields proves most fruitful in an organisation as they are many times re-used, shared and multiplied amongst a group of people.

Each organisation will consist of a set of work scenarios, typical examples of these include customer handling, service delivery, etc. A work scenario is therefore a description of the best way to conduct a certain business case in the organisation. In BESTREGIT work scenarios are described with two complimentary views. Firstly as role models, where role centred models base on roles, which are played by individuals. One person can play many roles as well as many people playing just one role. Roles exchange information and work results. This information flow between the roles forms the role model. Secondly are the workflow models, which consist of a network of work steps. These produce results that can be used by other work steps, each step requiring resources.

BESTREGIT uses an integration of both of these views. Firstly the role models are analysed and designed. Secondly the role models are transformed into workflow views. Then both models are integrated so that a work scenario according to BESTREGIT can be defined as whereby (TeamWork (2001)):

> "People are assigned to roles, roles are assigned to activities, activities are part of a network of work-steps, activities produce results, and roles use resources to perform the activities."
>
> (P 3.4)

These relationships are then defined for a certain business case of the organisation to have a description of the best way to perform the business case. Once process have been modelled using BESTREGIT it becomes possible to think about optimally organising all their elements for greater effectiveness and efficiency.

5 NQA – A New Departure in CSCW:

NQA is a computer-support cooperative work system (CSCW) based upon a role-scenario view of organisational behaviour. The initial concepts, which are reified in NQA, are the basic building blocks upon which BESTREGIT was constructed. NQA comprises important concepts in the context of information systems development through configuration:

1. Development by Configuration
2. Re-Use Pool - Organisational Function Based Configuration

5.1 Development By Configuration

This paradigm takes data independence objectives as its underlying principle with the added notion that data can be assigned with functionality by the user through specific configuration techniques. Such an approach overcomes problems often associated with component based approaches to software development, particularly those associated with the object oriented paradigm in which configuration of data (attributes) and methods (functions) is inseparable because both data and functions are encapsulated into a single, inviolable object (Graham (1991), Atzeni et. Al. (2000)). The Object Oriented paradigmatic approach has often been mooted for advanced coordination systems and methodologies for CSCW, Business Process Re-Engineering (BPRE) and other approaches associated with organisational transformation (Malone (1992), Shelton (1994)). However, this research present recognises that, whilst superficially the purely object-oriented paradigm may seem attractive, it does not reflect basic organisational information processing as it occurs in virtual communities. This fit between organisational information processing (as well as other organisational properties) has been shown to be important elsewhere, particular for groupware-type systems (Kraut et. al. (1994), Rogers (1994)). NQA must maintain the power of object-orientation by storing document objects etc. but a paradigmatic shift towards data independence assumptions is also required in order for the system to adequately reflect organisational behaviour in virtual spaces. This is an important theoretical shift which is required in order to effectively place NQA as a solution for virtual communities.

In role-based views, the data and function in organisational information processing are fundamentally separated and this must be reflected in the supporting technological architecture. In this approach users can insert and maintain document or result templates and adapt the system to their own specific documentation requirements without any change or customisation of code (just by configuration of data). Whilst documents are stored in an object-based technological solution (ensuring that the power of objects is harnessed), NQA also provides for a data independence rationality, which is usually seen as fundamentally opposed to object orientation. The reconciliation of these two rationalities is fundamental to NQA.

Re-Use Pool Concept (Organisational Function Based Configuration) - The Re-use pool contains a series of objects that can be reused in order to configure NQAs virtual office and enable the user to tailor the system. At the moment the following basic elements can be configured to which the above functionality is generated.

Tasks - A submission of a document and/or report automatically selects roles (role based configuration) which should receive this information and creates tasks for the users playing these roles. These tasks can then be traced.

Documents - There is a standard user interface for offering documents in the virtual office. Documents fall under the configuration management utility, and they can be linked with reports and other documents, can be downloaded, edited, and uploaded, and submitted to a team.

Reports - Reports usually are forms to be filled in, linked to a document, and submitted to a team.

Linked Reports – these are treated the same as regular reports, plus the link report is automatically linked by its creation backward and forward to a predefined set of reports and documents.

Depending on the user needs the elements are configured within a project administration structure. For example, Feature Requests (FR) can be linked with User Requirements Documents (URD), so that an URD is automatically created by the links to accepted FRs. Further basic elements are under consideration for insertion into the NQA configuration pool in later releases. This is possible because of the relatively easy extensibility provided by the object-oriented structure in NQA.

6 Future Developments

This paper presents a new technology which has been developed to prototype stage, and is based on an alternative view of organisational information processing. The next stage is to perform a full system test of the technology, and its underlying methodology. To this end, the research team have organised a full test with naïve users. This test will involve a scenario in which several research partners develop a research proposal across several countries. Participants in the test include research agencies as well as researchers in technical universities across Europe. Specifically, the countries involved are Ireland, Austria, Slovenia and Spain. This test is organised and will get underway shortly. The success of this test will not only be measured by user feedback, but also by the successful completion of a research project to proposal (and hopefully funding) stage. Thus the test process utilises NQA in a real-life scenario, and involves major stakes for those involved. It is intended that such an approach will yield the best possible usability data and reflects recent thinking in information systems development as regards training and learning (Stapleton (2000), O'Keeffe (2001)).

7 Conclusion

The Teamwork approach combines role and scenario based approaches into a coherent methodology. This methodology is then reified in a new virtual office technology called NQA. The virtual office environment and, in particular, the social relationships and processes which make such an environment work, are strongly reflected and supported by such an approach. Consequently, NQA is a very promising system with regard to highly dispersed organisations. Preliminary studies show that there is empirical evidence for merit in such an approach, although the evidence still remains sparse. Consequently it is important for this research to progress to a full test in a user environment. This full testing phase is scheduled and will be conducted for a remote work environment where a complex set of social interactions must be supported. With certain caveats, evidence suggest that the basic methodology and technology will be successful although it is likely that modifications will be necessary for the prototype NQA system. This paper presents the results of a research study which has culminated in a working prototype based upon a new IT view of organisational behaviour, a view which is far closer to the reality of virtual work than previous systems or approaches. Further research must be carried out in order to obtain substantial test results across a large community of users. Furthermore, other researchers must build upon role-scenario based approaches. One area of potential development is the inclusion of intelligent agent technologies, which interpret and update user profiles, thus obviating the extensive tree searches, which are required in goal tree-type systems. Whatever the case, it is readily apparent that NQA represents a radical new approach to IT support in remote working environments. This approach is both innovative and potentially opens an array of new possibilities – which are the subject for future research.

Acknowledgements:

The Commission as part of the IST programme of FP5 supports the TEAMwork Project (IST-2000-28162).

8 References:

Alexander, I. (1999), Migrating Towards Co-operative Requirements Engineering. CCEJ, 1999, 9, (1), pp. 17-22 Scenario Plus User Guide and Reference Manual, http://www.scenarioplus.org.uk.

Boorstin, D. (1978), *The Republic of Technology: reflections on our future community,* Harper & Row, USA.

Calem, R. (1992), The network of all networks, New York Times, December 6th, p. 12.

Cooley, C. (1983), *Social Organization: A Study of the Larger Mind*, Transaction Books, New Brunswick, NJ, USA.

Darnton, G. & Giacoletto, S. (1992)*, Information in the Enterprise: It's More than Technology*, Digital Press: MA.

Davidow, W. & Malone, M. (1993), *The Virtual Corporation: Structuring and Revitalising the Corporation for the 21st Century*, HarperBusiness: NY.

Davenport, T (1998). Putting the Enterprise back into the Enterprise System.' Harvard Business Review, August, 76, 4, pp. 121-131.

Drucker, P. (1988). 'The Coming of the New Organisation', Harvard Business Review, Jan-Feb pp.45-53.

Drucker, P. (1993). *Post-Capitalist Society*, Butterworth-Heinemann Ltd: Oxford.

Fernback, J. & Thompson, B. (1995), Computer Mediated Communication and the American Collectivity: The Dimensions of Community Within Cyberspace, presented at the annual convention of the International Communication Association, Albuquerque, New Mexico.

Graham, I. (1994). 'Business Process Re-engineering With SOMA', in Proceedings of Object Expo Europe Conference, SIGS: NY, pp. 131-144

Hamilton, A. (1997), *Management by Projects: Achieving Success in a Changing World,* Oak Tree Press, Dublin.

Hedberg, B. & Jonsson, S. (1978). 'Designing Semi-Confusing Information Systems for Organisations in Changing Environments', Accounting, Organisations & Society, 3/1 pp. 47-64.

Jacobson, I., et al (1992), *Object-Oriented Software Engineering: A Use Case Driven Approach*, Addison-Wesley.

Jones, Q. (1997), Virtual Communities, virtual settlements and cyber-archaeology: a theoretical outline, Journal of Computer Mediated Communication, 3(3).

Kerth, N. (2001), *Project Retrospectives: A Handbook for Team Reviews*, Dorset House, NY, USA.

King, M. (1996), On Modelling in the Context of I, JTCI Workshop on Standards for the use of Models that Define the Data & Processes of Information Systems.

Markus, L. & Robey, D. (1988). 'Information Technology and Organisational Change: Causal Structure in Theory and Research', Management Science, 34/5, May, pp. 583-598.

Mc Bride, N. (1999), Chaos Theory and Information Systems Track 4: Research Methodology and philosophical aspects of IS research, www.cms.dmu.ac.uk/~nkm/CHAOS.htm

McLuhan, M. (1964), *Understanding Media: The Extensions of Man*, McGraw-Hill, NY, USA.

Mehrabian, A. (1971). *Silent messages*. Wadsworth, Belmont, California

Meyrowitz, J. (1985), *No Sense of Place: The Impact of Electronic Media on Social Behavior*, Oxford University Press, NY, USA.

Nocera, J. (1998), Proceedings Cultural Attitudes Towards Communication and Technology, University of Sydney, Australia, 193-195.

Rheingold, H. (1993), *The Virtual Community: Homesteading on the electronic frontier*, Addison-Wesley, Reading, MA.

Sproull, L. & Kiesler, S. (1992), *Connections*, MIT Press: MA.

Stapleton, L. (1999). 'Positivism, Interpretivism & Post-Phenomenology - Views of an Organisation in Flux: An Exploration of Experiences in a Large Manufacturing Firm', in Proceedings of the European Colloquium of Organisational Studies (EGOS 1999), University of Warwick: UK.

Stapleton, L. (2000). 'From Information Systems in Social Settings to Information Systems as Social Settings', Proceedings of the 7th IFAC Symposium on Automation Based on Human Skill: Joint Design of Technology and Organisation, Brandt, D. & Cerenetic, J. (eds), VDI/VDE-Gesellschaft Mess- und Automatisierrungstechnik (GMA): Dusseldorf, pp. 25-29.

Stapleton, L. (2001). Information Systems Development: An Empirical Study of Irish Manufacturing Firms, Ph.D. Thesis, National University of Ireland.

Stapleton, L. & Murphy, C. (2002). 'Examining Non-Representation in Engineering Notations: Empirical Evidence For The Ontological Incompleteness of The Functionally-Rational Modelling Paradigm', in Proceedings of the International Federation of automation and Control World Congress, Forthcoming.

Van Vliet, W. & Burgers, J. (1987). Communities in transition: From the industrial to the post-industrial era. In I. Altman & A. Wandersman (Eds.), *In Neighborhood and Community Environments*. New York: Plenum Press.

Warboys, B., Kawalek, P., Robertson, I., & Greenwood, M. (1999), *Business Information Systems: A Process Approach*, McGrath Hill, UK.

Watson-Manheim, M. B., Crowston, K. & Chudoba, K. M. (2002), A new perspective on "virtual": Analysing discontinuities in the work environment. Presented at *HICSS-2002*, Kona, Hawaii, USA.

Wood J.R.G. & Wood-Harper, T. (1993). 'Information Technology in support of individual decision making', Journal of Information Systems, 3/2, April, pp. 85-102.

Results from an E-Working Trial in Different Domains Across Europe

G. Nadasi, ISCN, Austria, R. Messnarz, ISCN, Ireland, R. Bernhard, D. Ekert, ISCN, Austria, N. O'Shea, Tecnet, Ireland, R. Moyano Mateo, GMV, Spain, J. Elliott, QinetiQ, UK, A. Fenz, APS, Austria, R. Vajde-Horvath, University of Maribor, Slovenia

Abstract

This paper discusses the experiences gathered in the establishment of a quality and teamwork process driven e-working infrastructure in different European domains.

In a project TEAMWORK (supported as IST-2000-28162) a configurable platform system for team-working over the Inter(Intra)net has been used to set up a number of defined teamworking processes which allow people from different domains to collaborate in distributed joint projects.

Experiences have been gathered in three aspects: (1) Teamwork Processes, (2) European Demands for a generic Platform Concept, (3) Social Barriers.

Keywords

Best Practice, Teamwork, E-Work

1 Background

1.1 Why is TEAMWORK Needed

1.1.1 Industrial Demand

The idea of this initiative goes back into the beginning 90s when big manufacturing firms realised that in supply chains they work together in kind of virtual organisations. You could not draw exactly the border between firms, and teams were interdisciplinary with members from different firms working together on sophisticated technical solutions.

1.1.2 Process Improvement Demand

If an organisation moves from capability level 2 to a level 3 a common working platform is needed which supports across all projects the agreed standard/tailored processes. This should be done by a working platform which clearly supports roles, document and report types, document flows, version management and control, and queries which automate metrics and measurement. Such a system prepares the ground for collecting corporate knowledge and provides later interfaces for project oversight, measurement and calibration which is needed to ever reach a higher capability level than 3.

1.1.3 Critical Mass Demand

Especially in the fields of European research strategies the idea was created to analyse concepts and systems which enable that researchers and developers can jointly work on key topics to form critical masses of intelligence.

1.2 What is TEAMWORK

TEAMwork developed and applies a novel, integrated solution for distributed e-working organisations through the innovative and creative combination of Technology, Methodology and a Skills solution / Social analysis. The solution combines (see Figure 2 below):

- A methodology developed 1996 – 1999 to establish role based TEAMWORK environments, which was tested in trials in innovation centres across Europe. [1], [2]
- NQA (Network based Quality Assurance) as the technology platform developed 1996 – 2001 which represents an application on top of a knowledge management server to run quality assurance through virtual offices [3], [4].
- A set of social studies in the IT sector represented by organisations who did such social studies for Europe [5], [6], [7], [8], [9], [10].

Figure 1 illustrates the integration of methodology, technology, and additional skills into an overall e-working infrastructure.

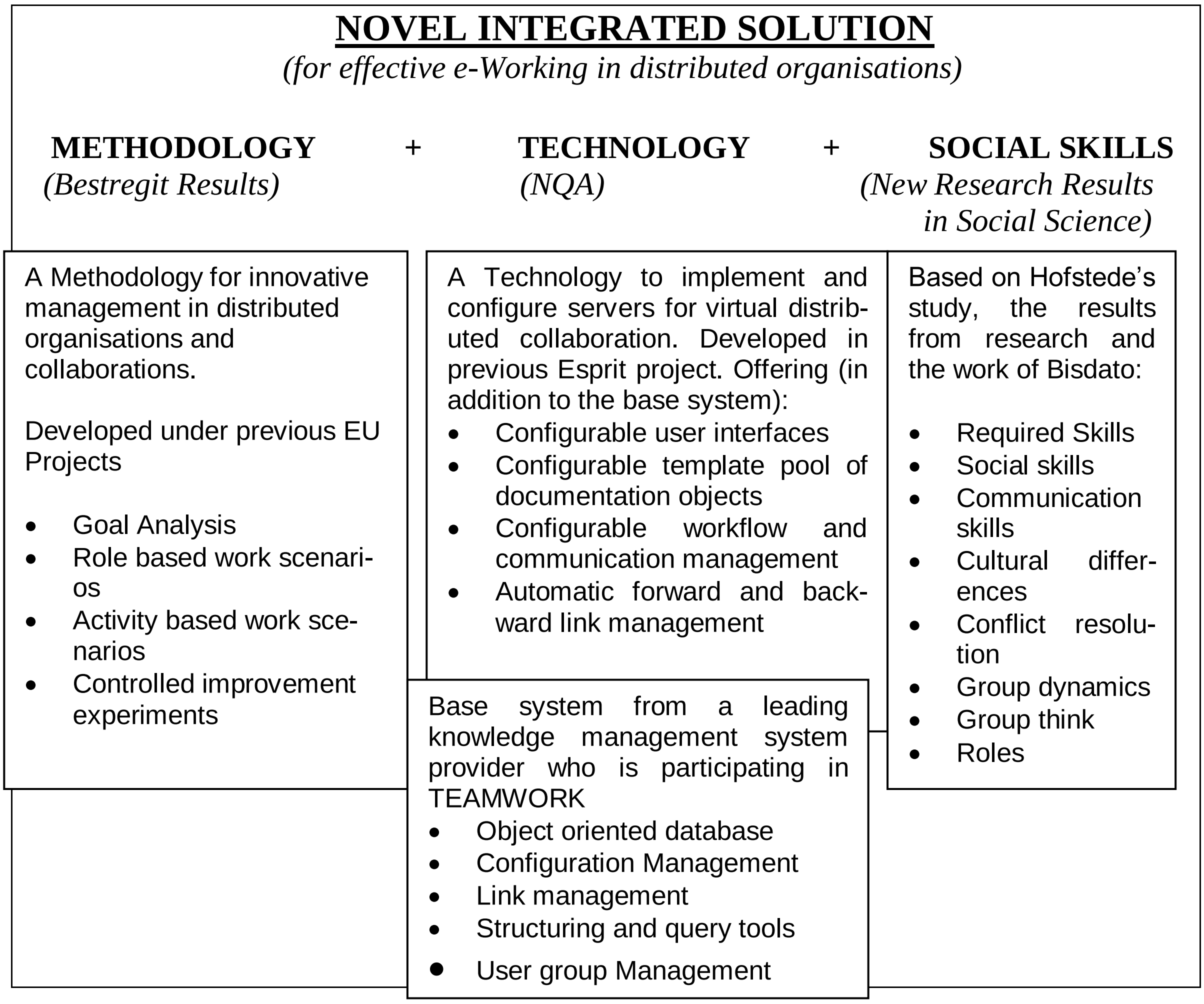


Figure 1: Integrated View of the System Architecture

1.3 The Scope of TEAMWORK

The following team-work scenarios have been further refined and configured, delivered and tested on servers in Europe.

For the Software Engineering Domain

- Team-work scenarios for processes for safety critical projects in defence environments.
- Teamwork scenarios to support the ESA ECSS (new European Space Agency) Standards.

For the Research Network Domain

- Teamwork scenarios for facilitating the proposal writing under Enterprise Ireland's Partnership Innovation Programme.
- Teamwork scenarios to professionally manage an EU funded project.
- Team-work scenarios for a project collaboration for a research topic on which researchers from Ireland, Austria, and Slovenia will co-operate.
- Team-work scenarios to optimise the work for Socrates network co-operation in Europe.

For the General (Public) Service Domain:

- Teamwork scenarios for collaboration in Leonardo programmes.
- Teamwork scenarios for co-operation between the EU Innovation Relay Centres.

2 Experiences with the TEAMWORK Methodology

The underlying methodology to analyse and draw teamwork based processes has been developed in the BESTREGIT project (1996 – 1999). As stated by Austrian, Irish and Spanish national innovation centers : "*The highest value of that methodology is that it is written in a language which can be understood and applied by non-technical people*".

Originally the project started with a purely technical (workflow driven) view and had to be applied by decision makers in national educational, innovation, and semi-public government institutions. However, these people did not understand purely technical views and needed some model that develops towards a technical effective infrastructure but bases more on the human integration and development.

Therefore the purely technical workflow was exchanged by a role based teamwork process view in which people identify with roles, roles produce work products which are exchanged with other roles according to defined role communication models. This then was accepted by all people (also non-technical) and brought the following good results:

- Because people see roles with which they can identify, they identify with the overall process.
- Especially in a distributed e-working environment beautifully drawn work-flows (with steps, and result flows) are worthless if they do not clearly show the roles, interfaces and support the communication.
- The role communication models make it much clearer how to interact in a team.
- Roles can be distributed across the world, with an e-working platform in the middle supporting the defined role-communication driven teamwork models.
- The integration of new staff becomes much easier. They get roles and immediately know how to be integrated into the overall team.

Measured results were:

- Staff Integration: Especially in fields where companies grow very fast or have a certain extent of changing staff the team-working scenarios with clear roles and interfaces helped in the integration of (new) staff. Best values achieved, measured in tutoring effort required, where a saving of about 66% of integration effort in average per staff member.
- Knowledge Sharing / Corporate Knowledge: When using a system like NQA all data are accessible to team members in certain protected project areas so that knowledge is submitted between roles, can be annotated and linked, and is accessible. This helped a lot in team-work effectiveness. Best values achieved were that before using the approach if a person (if someone became sick or left the company or was on travel to meetings) was not available the other staff could not continue working on the material of this person. After the introduction 100% confirmed they could continue.
- Usage Breakthrough: Using team-working and role based identification of people with specific duties, plus an integrated work place solution helped a lot in the acceptance of staff and partners to apply specific standards. Best values achieved where a usage breakthrough from 17% (before) to 67% (after 6 months) in the usage of software engineering standards.

The outcome of a team-work analysis based on Bestregit are role based team-working scenarios. Figure 2 illustrates the scenario "Project Management" according to the new European Space Agency Standard ECSS.

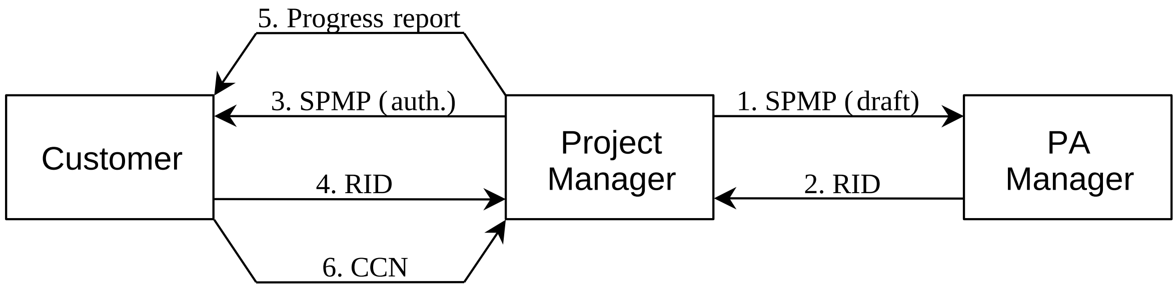


Figure 2: The ESA ECSS Project Management Scenario

SPMP stands for Software Project Management Plan. RID stands for Review Item Discrepancy report. CCN stands for a Contract Change Notice. PA Manager id the Project Assurance Manager.

The use of the methodology has major effects on how staff see a process:

- They are assigned to a role and thus take ownership of results to be produced. E.g. If the arrow "SPMP draft" goes from the Project Manager to the PA Manager, and Mr X plays the role Project Manager, he is the one who has to produce this result.
- Beside the above mentioned ownership of roles also the interfaces between roles are clarified, and for each result template results have to be used.

The combined use of the methodology with an e-working system has major effects on how staff apply the standards:

- By creating a project of e.g. the pre-configured type "ECSS" this automatically creates a work space for a team with all scenarios and documents and reports necessary to apply this standard.
- By assignment of roles to team members and document flows to documents and report objects, the flows described in Figure 2 can be initiated and controlled.
- By using the system over the Internet projects with many different suppliers can be controlled and managed through a virtual office architecture.
- By having all knowledge stored on the server all team members can draw from the knowledge and corporate learning becomes possible.
- Etc.

For people with computer science background we must note that the BESTREGIT guide also defines a methodology to convert the role based view into a traditional work-flow. However, we do not use that in most of the cases as domains (see previous chapter about the project scope) which are non-technical usually do not accept this pure technical (and not humanised) view.

2.1 Modelling Experiences – Good/Bad

In the project we trained people from very different domains in the usage of the BESTREGIT methodology. These domains had to deliver the team-working processes which have then be set up in distributed servers across Europe. By applying this we identified two major faults which organisations usually make when entering the e-working field.

Findings:

1. If according to BESTREGIT a scenario is defined it is assumed that a defined team will be assigned to that scenario. The team assignment also then defines access rights. In many cases when it came to non-software-engineering but rather general service and political domains the us-

ers mixed different types of teams (students and university staff, government and researchers, developers and customers, etc.) into just one scenario which would have caused privacy problems.
2. If according to BESTREGIT a scenario is defined it should be grouped into a „business section/department" of a firm which later becomes a project type in e-working system. So it happened that one scenario covered two different business aspects and teams so that they had to be split and clearly assigned to well defined project types.

Conclusions:

1. There is a huge difference in the organisational structure when you move a traditional firm to an e-working concept. Projects are no more just in-house, they are partly outside and inside the own controlled environment. To not make serious mistakes this requires a thorough analysis of teams, groups, and privacy.
2. It showed us that still firms struggle to have well defined business units and business goals to which specific project types and scenarios of collaboration could be assigned. In this case before starting e-working a thorough business analysis would be required, otherwise conflicts of interest between business segments might happen.

Observations:

A major observation was how the users selected specific project types and working scenarios to be supported. They selected project types by their MULTIPLICATION factor (also a rule posted by the BESTREGIT methodology).
This means that organisations analyse their business fields and identify re-usable project types and scenarios which can be applied many times. Only if the MULTIPLICATION factor was high the scenarios have been established and implemented in the e-working infrastructure.

A further observation was that specifically technical domain users (aerospace and defence) combined the MULTIPLICATION with the BUSINESS factor. Project types and scenarios were selected which can be multiplied across the organisation and contribute directly to the business success of lead projects (e.g. GPS development using the new ECSS standard as a lead project).

3 Experiences with the Generic Platform Concept

A system is called **generic** if the same solution can handle many different applications with just some adaptation of data and functional configurations. A system is called a **platform** if there is a computer supported mechanism which allows to generate such applications for customers based on defined configuration principles.

Why is such a generic platform useful ?

A major feature to make such a virtual approach applicable for different domains (also non-technical ones) is that such a system must be kept completely configurable. The menu, the data, the functions, the document/information flows can be configured for different user scenarios and this high configurability is the major feature of an NQA virtual office.

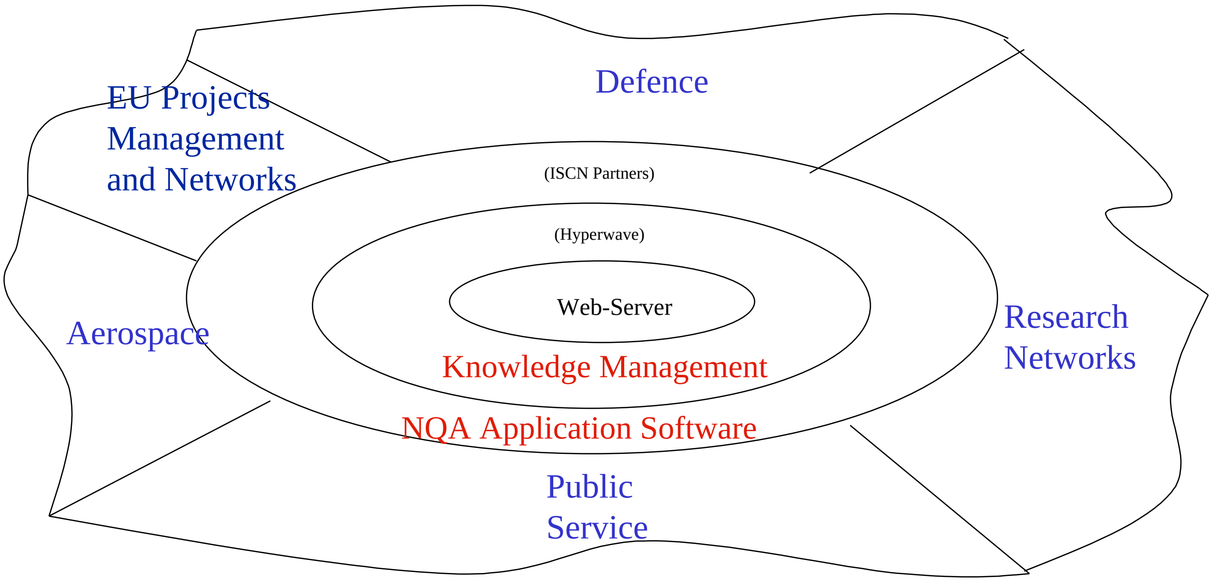


Figure 2 : The Basic Technology Platform Architecture

3.1 The Technology Platform

The below list describes basic elements which have been configured for each of the domains (using the NQA platform):

Project Type - A project type contains a number of working scenarios. Usually on one NQA platform you configure a number of project types (e.g. ESA projects, Defence Projects, etc.) which are then offered to the NQA quality administrator. By creating a project of that type automatically the right structure is created. For each project a team can be assigned with accounts on the system.

Working Scenario - Each project type contains a number of working scenarios (working phases). Each scenario contains documents and reports to be managed. Roles can be assigned to team members of the project, and document flows can be configured.

Roles - Depending on the working scenarios you designed, roles can be configured. The roles are then offered through a menu so that team members can be assigned to roles.

Template Pool - Documents or reports are created from templates which the NQA quality administrator publishes into a template pool.

Tasks – A submission of a document and/or report automatically selects roles (role based configuration!) who should receive this information and creates tasks for the users playing these roles. These tasks can then be traced.

Documents - There is a standard user interface to offer documents. Documents fall under configuration management, can be linked with reports and other documents, can be down-loaded, edited, and uploaded, and submitted to a team.

Reports - Reports usually are forms to be filled in, linked to a document, and submitted to a team.

Linked Reports - same as with reports, plus the report is automatically linked by its creation backward and forward to a predefined set of reports and documents.

Query Function - Each document or report type receives an acronym describing the type. This info is stored as an attribute of documents and reports. The system can be configured for these types to run queries such as: want to see all Review Reports made in the projects to which I have access.

There are a number of additional functions to run project control which are a standard part of the system, independent of specific configurations. This includes

- Control of the test status of documentation
- Log of all document/report submissions
- Control of performance status of tasks of persons in projects

Below you can find a few typical figures illustrating the above element descriptions.

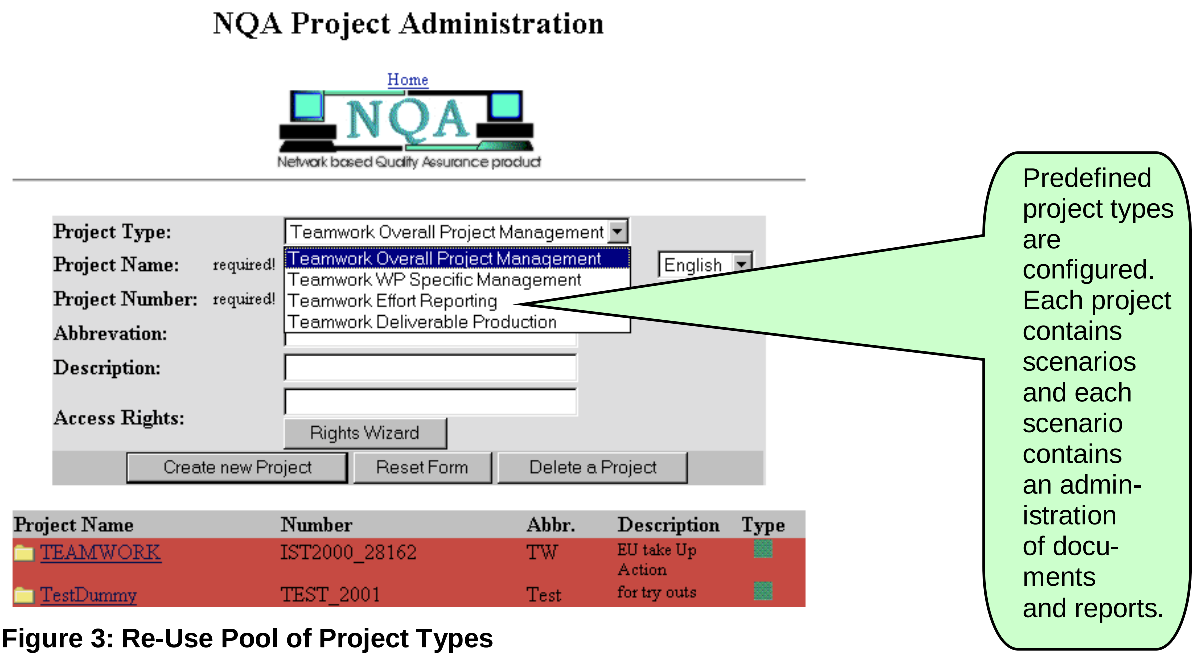


Figure 3: Re-Use Pool of Project Types

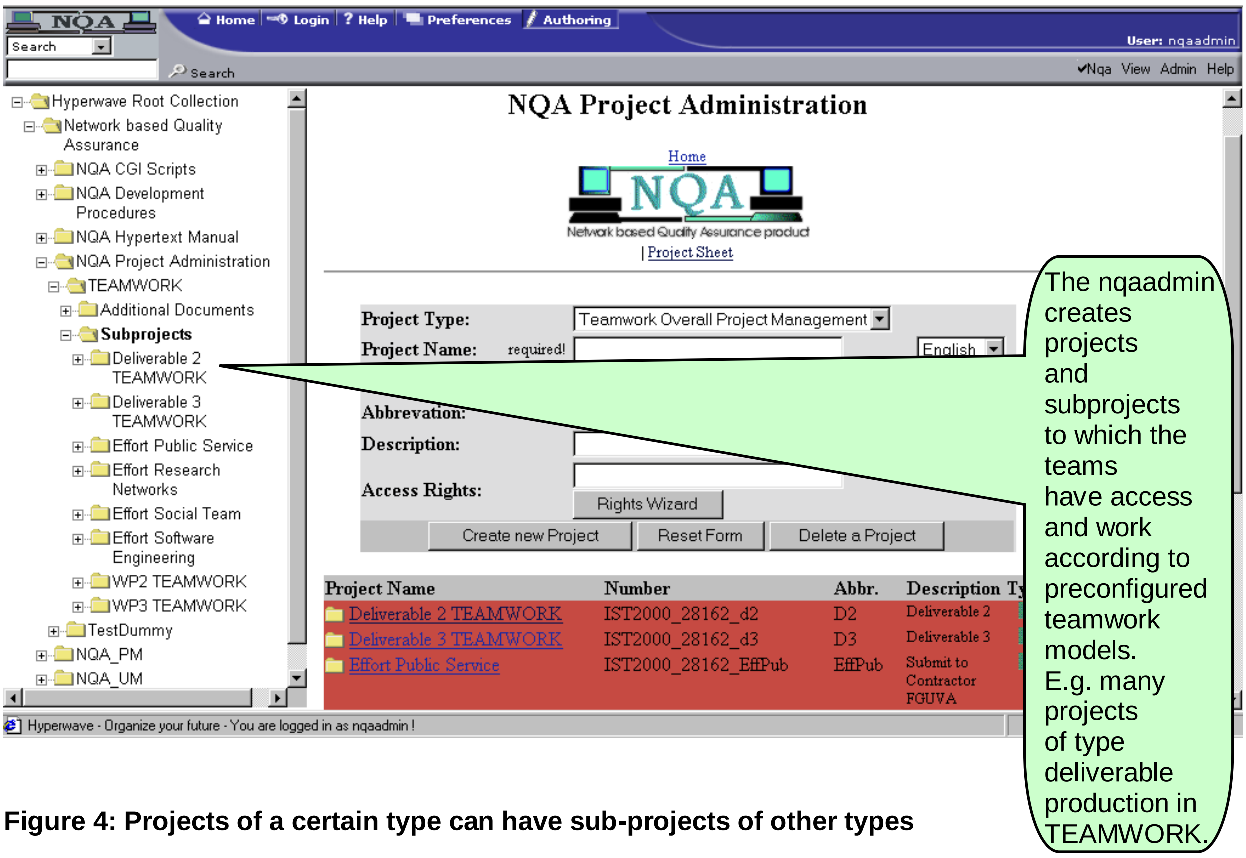


Figure 4: Projects of a certain type can have sub-projects of other types

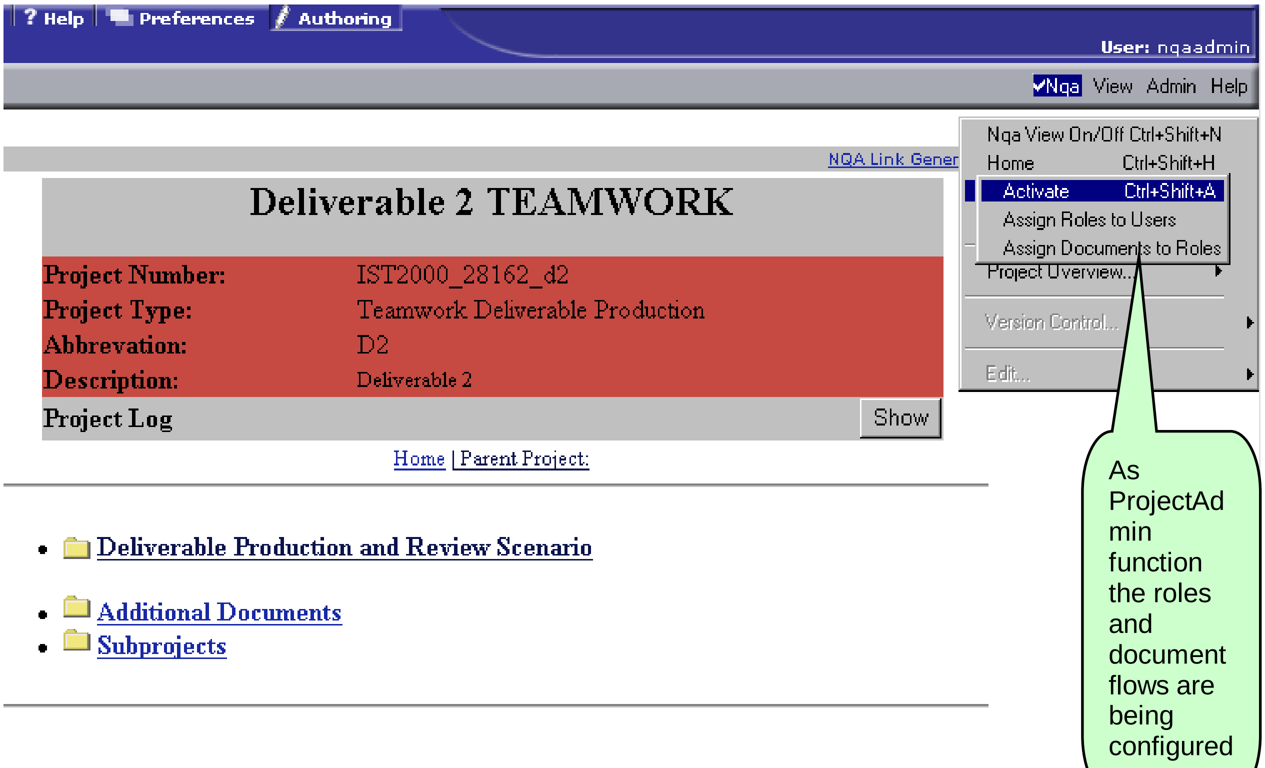


Figure 5: Projects contain scenarios and the NQA admin can assign team members to

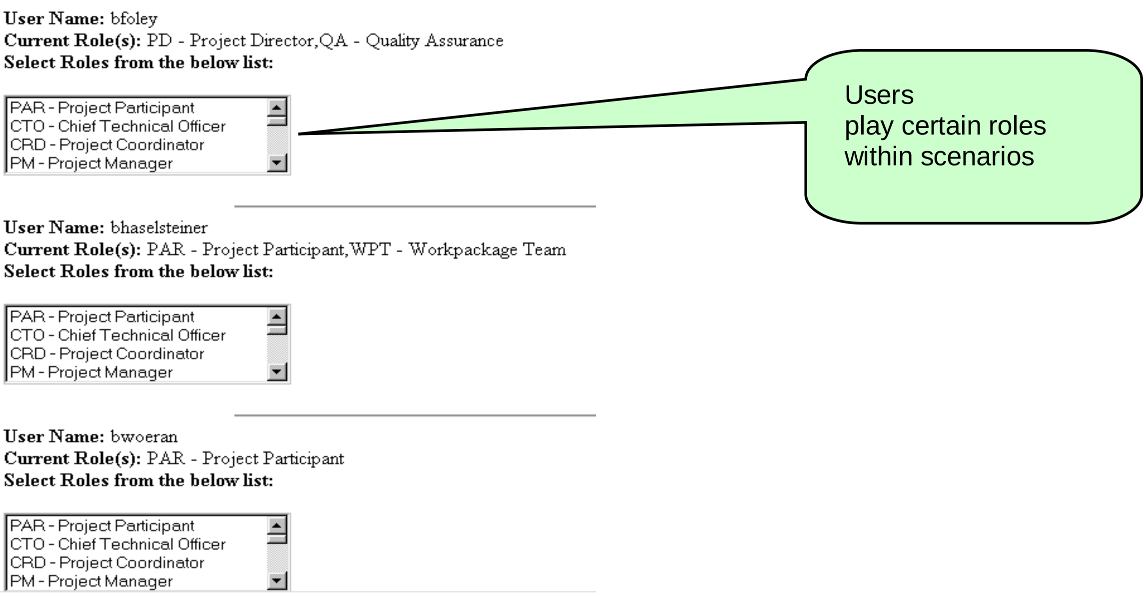


Figure 6: Assignment of roles to team members

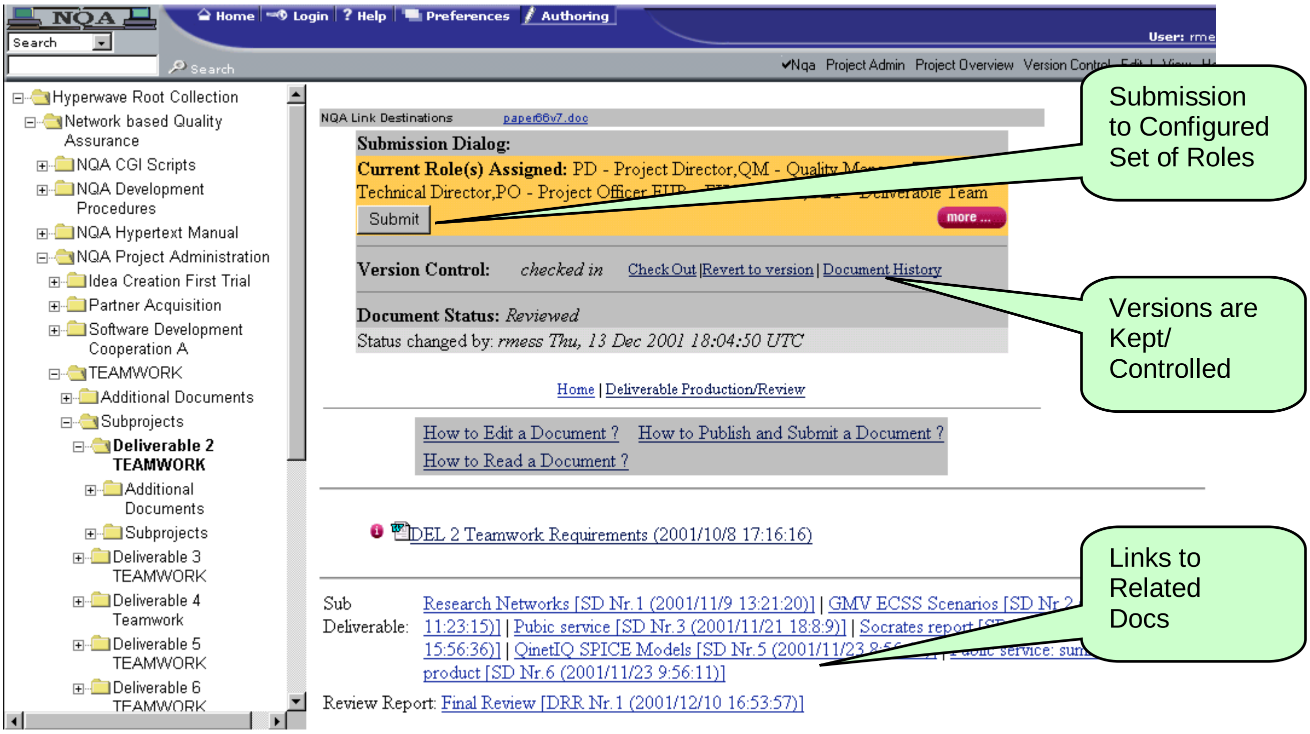


Figure 7: Submission to a defined set of roles

3.2 Technology Experiences – Good/Bad

In the project, after we had set up the e-working servers we identified that still the methodology is more important than the technology. The major reason why they used the e-working platform was because they first could model with the methodology their expected way of collaboration and secondly they saw the configurable e-working platform just as a tool that facilitates what they had agreed.

The advantage of a re-use pool based platform which can be configured for the different domains is that the complexity of the technology matter is decreased (you take already tested functional elements from a function pool and configure just user interface scripts to reflect the actual project types, scenarios, document flows, etc.) so that people can concentrate on the teamwork processes, the quality of results, and the collaboration models/strategies.

This in fact is a major difference of TEAMWORK from other e-working projects which focus much more on technical solutions and leave too less time for analysing the required collaboration processes. However, it must be noted that it is a long process to get such a configurable e-working platform with NQA which bases on Hyperwave (development since 1985 as spin off from Austrian research, and won the EU IT price in 1997) and NQA as the e-working platform pool (development since 1997 on top of Hyperwave).

Still the platform approach is not fully elaborated as users wished to have wizard like functions so that not NQA administrators (super users of the e-working nodes in the distributed network) are managing project type configurations but they can be done by every user in the world through a simple to handle function.

4 *General Experiences – Good/Bad*

A virtual organisation is a strategy to

- Build an infrastructure which allows different organisations to collaborate on certain projects through a (distributed) server concept.
- Build a set of processes and quality criteria which are supported by the infrastructure and help to control the performance and quality of the distributed team.
- Build on team-work and communication concepts which support distributed work.
- Build information retrieval mechanisms which allow a quality manager of the virtual organisations to extract the status of the project electronically at any given time.

Typical mistakes are

- Organisations buy a technology infrastructure and think that this is already it !
- Organisations establish processes on the infrastructure but leave out the quality issue !
- Organisations establish processes, quality and infrastructure correctly but do not use concepts that favour distributed team-work !
- Organisations do all correct but forget the social factors and the acceptance of staff !

The advantage of the TEAMWORK approach is

- People know their responsibilities better and know their communication interfaces to other members in the team.
- New staff can easily be integrated (assign a role, learn the skills required to play the role, follow the communication flows in the team)
- Information technologies like NQA (because the communication interfaces become visible) can be used to support the team communication, documentation, and configuration of results.

And if you want to survive with e-working in non-technical domains

- Focus more on human aspects, roles, and integration in teams.
- Accept that methodology and the understanding of key decision makers is much more important than the technology itself.
- Try reducing the technical complexity and focus on collaboration effectiveness and models.

5 Literature

[1] 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

[2] The Perception of Quality Based on Different Cultural Factors and Value Systems, in: Proceedings of the EuroSPI 2000 Conference, 7-9 November 2000, Copenhagen, Denmark, ISBN: 952-9607-29-6

[3] Messnarz R., Tully C. (eds.), The PICO - Book: Better Software Practice for Business Benefit - Principles and Experience, IEEE Computer Society Press, in publication

[4] 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

[5] M. Biro, R. Messnarz, Key Success Factors for Business Based Improvement, in Software Quality Professional, Volume Two, Issue two, American Society for Quality, March 2000

[6] Calori, Roland. A European management model : beyond diversity. New York ; London : Prentice Hall, 1994

[7] Tom DeMarco, T. Lister, Peopleware – Productive Projects and Teams, 2nd edition, Dorset House Publishing, New York, 1999

[8] Hofstede Geert: Cultures and Organisations, Intercultural co-operation and its importance for survival, Software of the mind, McGraw-Hill UK, 1994

[9] Siakas Kerstin V. The Effect of Cultural Factors on the Implementation of Software Quality Management Systems, 5th Software Quality Conference, Dundee, 9-10 July, -96

[10] Siakas Kerstin V., Process Improvement – The Societal Iceberg, in: in: Proceedings of the EuroSPI 1999 Conference, 25-27 October 1999, Pori, Finland

[11] ISO 9001. Quality Systems. Model for Quality Assurance in Design/ Development, Production, Installation and Servicing. International Organisation for Standardisation, Geneva (1987)

[12] 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

[13] Merkle, M., Virtuelle Organisationen – Ihr Erfolgspotential: eine integrative Informationsinfrastruktur. Institutsbericht des IFI der Universität Zürich, 1996.

[14] Mertens, P., Faisst, W., Virtuelle Unternehmen, eine Organisationsstruktur für die Zukunft?“. Technologie & Management, 44 2/1995, p.61ff.

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

6 Author CVs

Dr Richard Messnarz

Dr Richard Messnarz made his MSc at the University of Technology in Graz in Austria. In 1995 he received a PhD from the same university for the work on a QUES - "A Quantitative Quality Evaluation System".
He has participated in a number of European research and industry projects since 1990. He was also the project leader of the Comett expert exchange project ISCN in 1993 which led to the foundation of ISCN itself in 1994.
He is the Executive Director of ISCN. He was the technical co-ordinator in previous process improvement initiatives such as PICO (Process Improvement Combined apprOach), Victory (Virtual Enterprise for Process Improvement), and the co-ordinator of a software engineering group within the pilot project for building a prototype of a virtual university. He is the editor of a book published in 1999 from IEEE "Better Software Practice for Business Benefit - Principles and Experience". He is currently the technical director of a large e-working initiative TEAMWORK, and manages in the initiative MediaISF the technical development for an e-working solution for news agencies collaborating on Eastern European enhancement topics.
He is a chairman and main organiser of the EuroSPI conference series (the former ISCN series).
He is a SPICE and BOOTSTRAP lead assessor.

Gabor Nadasi

Mr Nadasi made a MSc at the University of Veszprem in Hungary. In 1997 he worked at research institutes in Graz, Austria on a contract basis and started with ISCN as an Internet software engineer in 1998. Since then he specialised in the development of the NQA system, and application development on Hyperwave basis. He participated in a number of European research projects, such as SELECT (new protocol for selective browsing), Victory (Virtual Enterprise for Process Improvement), and recently Teamwork (Applying NQA in different domains),
Since 2001 he is combining his work at ISCN with a dissertation in the fields of e-working and integrated work place solutions for the future.

Ricardo Moyano Mateo

GMV S.A. is one of the three subsidiaries of the holding company Grupo GMV, a Spanish privately owned company founded in 1984. GMV S.A. was founded in 1992 and has inherited all the traditional activities of the parent company in the space, defense, and air transport sectors. On October 27th 2000 GMV, together with AENA, CASA, Hispasat, INDRA and Sener, constituted the joint society Galileo Sistemas y Servicios (GSS), of which GMV owns a 14%. The objective of this society is to promote the development, the operation and the commercial exploitation of applications and services based on the future European satellite navigation system GALILEO.
Mr Moyano Mateo made a Bsc. degree in Computer Science (1988) from the Politechnical University of Madrid. He has participated in and lately then also managed a large number of projects at GMV since 1989. Soem of his most recent projects are:
1999-2001: Project Manager of EOSS.
EOSS stands for Earth Observation Simulation System, aiming to provide a generic framework able to support the simulations of one, several or all stages involved in the image production processing chain. In addition, EOSS is a Web application, managed and run by users remotely through the Internet.
The project is formed by a two company consortium, GMV and ACRI, acting GMV as the prime contractor. ACRI provides its technical experience in engineering consulting and simulators for Earth observation systems.

2001-: Project Manager of SAFNWC.
This project was originated by EUMETSAT aimed to produce a Satellite Application Facility for the development of an application intended to support the short-term weather forecast. The application consists of the integration of twelve weather product generator elements (PGE) with common components used by all of them.

2001-: Project Manager of CID-MT.
CID-MT stands for Cargo Integration Database Management Tool, in charge of building a Web-oriented application to ease several users to establish the cargo management of the ATV flights to the ISS.

2001-: TEAMwork responsible.
Within TEAMwork, GMV has to provide its expertise in the software engineering field for the space domain. As a consequence, the NQA tool shall be adapted to follow the ECSS standards, which are the new SW engineering standards for the ESA. From this point, new projects shall make use of the corresponding scenarios within TEAMwork system, and user comments shall be collected.

How implementing self-assessment powered our SPI programme

Finn N. Svendsen
Morten Thy Christensen
Kurt Greve
Niels Jørgen Strøm Mortensen
Grundfos A/S, Denmark

Abstract

Experiences from the use of reduced BOOTSTRAP 3.2 [1] self-assessments at Grundfos.

In early 2001, Grundfos, a major pump producer with development centre based in Denmark, decided to use self-assessment of the software development projects as one tool to measure the progress of the in-house SPI programme.

Self-assessment is now being implemented as a process in Grundfos. All projects are subject to self-assessment. An annual plan is prepared. The self-assessment process consists of 5 stages: Planning, interview, scoring, reporting and publication of the results.

A Grundfos self-assessment includes up to 10 of the processes defined within the BOOTSTRAP model [1]. The selection of the 10 processes was carried out in cooperation with DELTA [6], a local member and licensee of BOOTSTRAP Institute [1], and included both life-cycle dependant processes as well as managing and supporting processes. The selection includes the processes we believe to have the highest value for our projects.

The learning has been versatile. This paper concludes on the learning that our projects, the SPI programme and the assessors achieved, as the result of the self-assessments.

Most important, we learned that implementing self-assessment brought new power into the SPI programme. The results have been more focus on the processes in the development project and an increased number of feedbacks from the projects.

Keywords

Self-assessment, BOOTSTRAP, SPI, Industrial experiences

1 The Grundfos SPI programme

In early 2001, Grundfos, a major pump producer with development centre based in Denmark, decided to use self-assessment of the software development projects as one of the tools to measure the progress of the in-house SPI programme.

Software development at Grundfos includes development of embedded software for motor control, communication and product control, application controllers and supporting PC tools. Development projects are often integrated product development, where software development is a subproject. The self-assessments only deal with these subprojects, the size of which vary from 6 to 200 man months.

Grundfos has a company culture of continuous improvement based on the EFQM Excellence Model [5] and the principles of The Learning Organisation. In software development terms this takes the form of an SPI programme. The SPI programme is the turning point for improvements of the software development processes and engages many software developers in a number of PITs (Process Improvement Teams). The target is to reach a BOOTSTRAP level of 3 by 2002. External assessments according to the BOOTSTRAP model [1] are used as reference milestones. Between these milestones the self-assessments are carried out. The self-assessments are also based on BOOTSTRAP, but with a reduced number of processes.

This paper will summarize the results from the first year of self-assessments.

The most important result is not the scores of the process measurements, but the impact that self-assessment has had on the SPI activities. Self-assessment simply puts focus on the relationship between SPI and the development projects and thus creates process awareness.

2 The assessment method and process

2.1 Self-assessment based on BOOTSTRAP 3.2

There are several maturity models available for assessment of software development maturity (e.g. CMM, BOOTSTRAP, SPICE). Grundfos has decided to use the BOOTSTRAP model as the best choice for our type of company. DELTA [6], a local member and licensee of BOOTSTRAP Institute [1], is supporting us with knowledge, training and external assessments according to BOOTSTRAP version 3.2 [1], which also complies with both CMM [3] and SPICE [4]. An overview of the processes included in BOOTSTRAP is shown in figure 1.

For the purpose of self-assessment the complete model is not used. We identified the software development projects as key customers for the SPI programme and the self-assessments at this stage. Due to this and also to keep the self-assessment effort at a manageable level with limited resources, a reduced number of relevant processes for the projects were selected (marked in figure 1). The selection, which was carried out in cooperation with DELTA, included:

- Life-cycle dependant processes
- Managing and supporting processes

The life-cycle dependant processes were chosen to cover the complete software development life cycle. At the beginning we chose to assess only requirement specification (ENG-3) and architectural design (ENG-4), as the processes with most impact on quality of the developed software. Soon we realized that the complete life cycle should be included in order to give a holistic view of the projects.

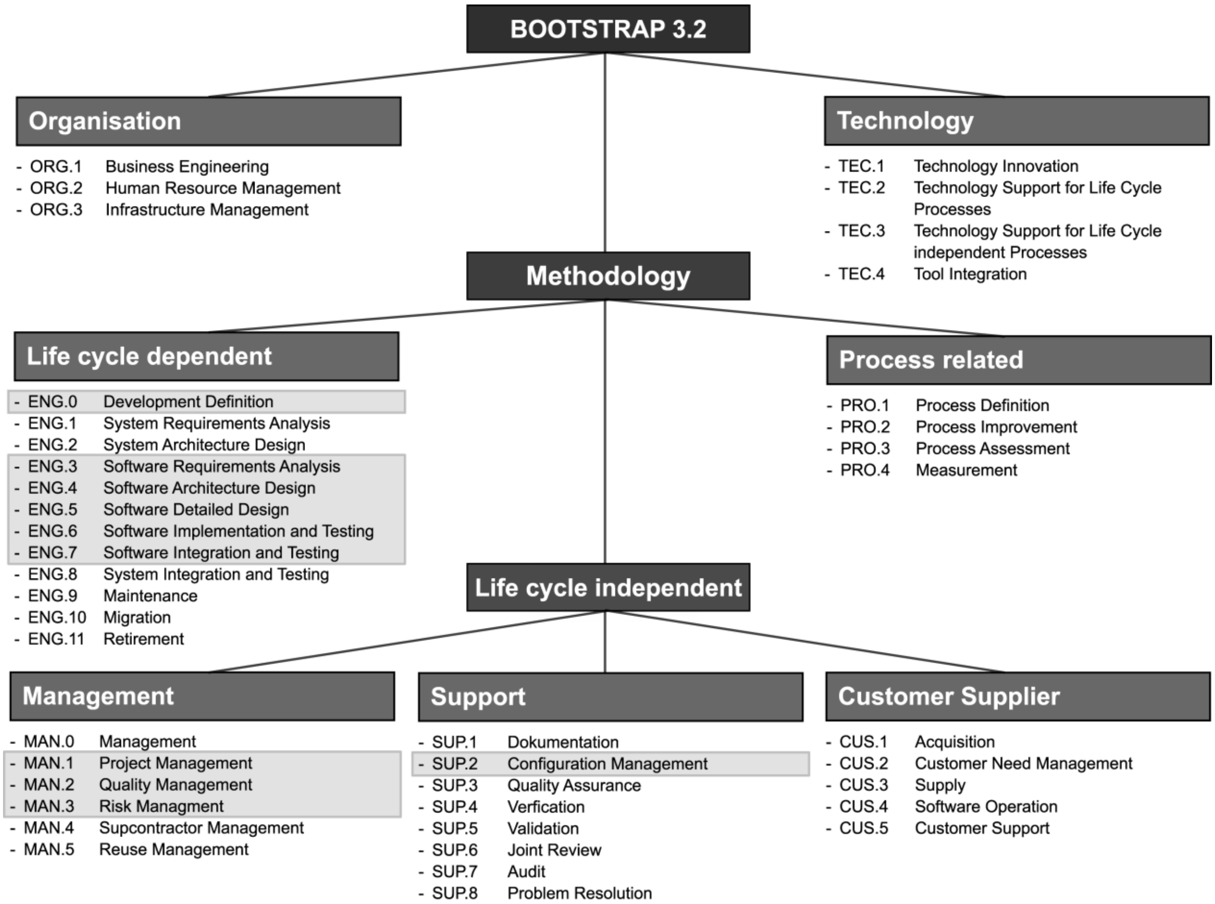


Figure 1: The BOOTSTRAP 3.2 process model [1]. Processes within the frames are selected for the Grundfos self-assessments.

At present the life cycle now covers:

- Development model (ENG-0)
- Requirement specification (ENG-3)
- Architectural design (ENG-4)
- Detailed design (ENG-5)
- Module implementation and test (ENG-6)
- Integration and software system test (ENG-7)

Management and supporting processes also had to be included, as these has a major impact on the project results. The following processes were chosen:

- Project management (MAN-1)
- Quality management (MAN-2)
- Risk management (MAN-3)
- Configuration management (SUP-2)

Quality of the end product is of the highest importance. With our present process knowledge and being at level 2, we find it more valuable to assess quality assurance (SUP-3) instead of quality management (MAN-2). This is because we have found quality assurance (SUP-3) to be more important at level 2 than quality management (MAN-2).

The present goal of the SPI programme is to reach level 3. Hence it was decided to evaluate only levels 1 to 3.

2.2 Training of the self-assessment team

Self-assessment is not an easy task to perform. The results must be valid in order to gain acceptance within the organisation. The key to achieve this is to have a trained and calibrated assessment team.

Because we wanted to use only a selected number of processes of the complete model we asked DELTA [6] to train a group of employees to be internal assessors in the selected processes. Grundfos has signed an agreement with DELTA regarding the use of the model as described in this paper.

Calibration of the team was reached by several means:

- As part of the training DELTA was mentoring the first assessments
- Teams for the single assessment are continuously mixed between the trained assessors and all assessments are carried out by a team of two (or more)
- The first assessments were carried out by an expanded team, which worked together throughout the entire assessment process
- A checklist for every process based on the BOOTSTRAP model [1] supplemented with our own implementation of the requirements in the model thus mapping BOOTSTRAP requirements to company requirements in the company "language"

2.3 The self-assessment process

Self-assessment is now being implemented as a process in Grundfos. All projects are subject to self-assessment. To enable this, an annual plan for self-assessment is prepared. The plan includes the active software development projects of the year.

An assessment consists of a number of stages:

- Planning
- Interview
- Scoring
- Reporting
- Approval and publication of the results

2.3.1 Planning

The assessment is planned together with the project team. For most projects all members of the project team participates. The assessment then becomes a common event in the project giving a common reference for improvements and understanding in the project. However, some larger projects select a smaller number of project members as representatives of all the assessed processes. In this selection, it is important, that the self-assessment team can interview more than one project member about each assessed process. The assessment is planned to last for app. ½ hour per assessed process.

2.3.2 Interview

The interview starts with:

- A short introduction to the BootStrap Model
- An introduction of the assessment process
- A discussion about the importance of the processes in the project giving input to the improvement matrix (se figure 3).

- And giving the project insight into the outcome of the self-assessment.

The interview continues with the questions related to the single processes. The assessment team divides the work between them. One acts as "lead" assessor asking the questions, the other(s) take(s) notes and ask(s) supplementary questions.

2.3.3 Scoring

Immediately following the interview, the assessment team meets to evaluate the answers and to score according to the BOOTSTRAP model [1]. Only levels 1 to 3 are assessed.

2.3.4 Reporting

A report is prepared. In the report a section for every assessed process will be found.

These sections summarise:

- The strong areas
- The improvement areas
- The improvement potential with recommended improvements to be made to raise the maturity level (the score) of the process to the next level.

The report also concludes on the project showing the score, a SPICE profile (figure 2), an improvement matrix (figure 3) and a summary in words.

The SPICE profile shows the individually assessed process attributes:

- PA1.1 – Basic activities
- PA2.1 – Performance management
- PA2.2 – Work product management
- PA3.1 – Process definition
- PA3.2 – Process resources

	ENG.0 Development Definition	ENG.3 Software Requirements Analysis	ENG.4 Software Architecture Design	ENG.5 Software Detailed Design	ENG.6 Software Implement. and Testing	ENG.7 Software Integration and Testing	MAN.1 Project Management	MAN.2 Quality Management	MAN.3 Risk Management	SUP.2 Configuration Managenemt
PA 3.2	F	F	L	F	L	L	L	L	P	L
PA 3.1	F	L	P	P	L	P	F	L	L	F
PA 2.2	L	F	L	F	F	F	F	L	P	L
PA 2.1	L	F	F	L	L	L	L	P	P	L
PA 1.1	F N	F	F	F	F	L	L	L	P	F

Figure 2: Example of a SPICE profile [4]. The process attribute scores used: N (Nothing), P (Partly), L (Largely), or F (Fully). The four marked columns and rows are examples of which conclusions can be drawn: 1. Example of a weak process in the project, 2. Example of a strong process in the project, 3. PA 2.2 is a strong process attribute in this project, 4. PA 3.1 is a weaker process attribute in this project

An example of the improvement matrix is shown in figure 3. The matrix is used by the assessed project to quickly spot the most value-adding improvement areas.

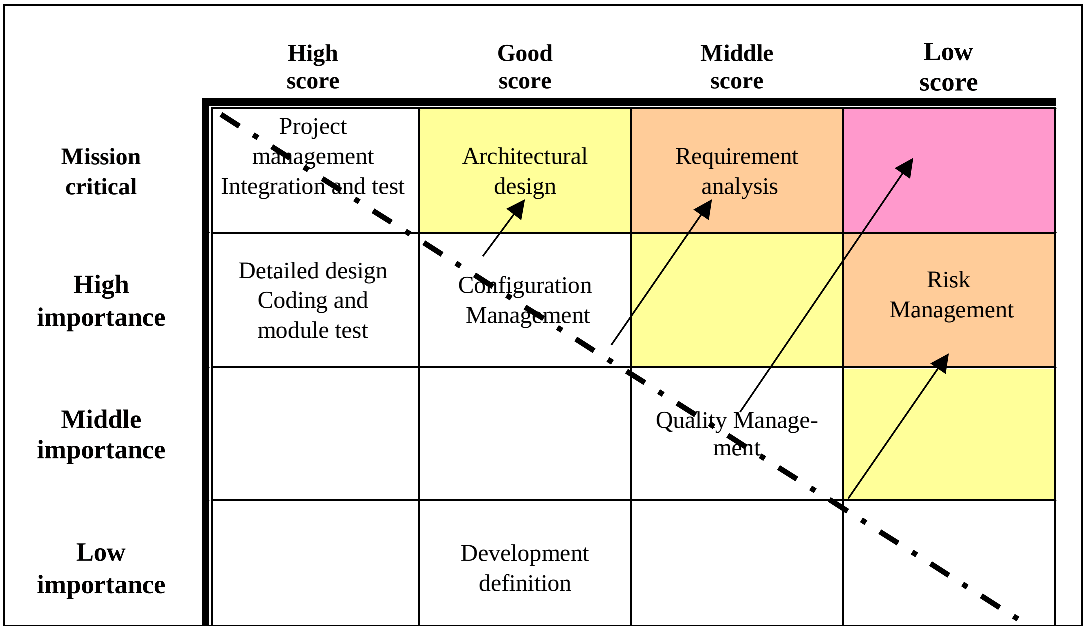


Figure 3: Example of an Improvement Matrix. The vertical axis indicating the importance of the process in the assessed project. The horizontal axis indicating the relative score of the assessed process. Processes situated in the upper right hand corner of the matrix as indicated by arrows are the processes, where the project would benefit the most from improvements.

The report is presented and discussed with the project team. After approval the report is published.

2.3.5 Publication

The publication is done internally in an information database containing reports from all finished assessments making it possible for all interested parties to view the results and learn from them. The assessment team concludes its work with the publication and does not follow up on the recommended improvements. Improvements are the sole responsibility of the project and the SPI programme.

3 Assessment results

3.1 Facts about the assessments

In 2001 and in the beginning of 2002, a total of 11 software development projects involving more than 40 project members were assessed. The projects included embedded software development and PC support tool. The size of the projects was from 6 man months to more than 200 man months. The assessed number of processes within each project was 6 to 10.

3.2 Results: Learning in the projects

The projects had their strong and week sides identified in relation to the assessed processes. This was leading directly to an improvement potential. Because some projects were assessed late in the development phase, not all improvement potentials were turned into actual improvements. By using the improvement matrix to select the improvements, the improvement effort could be focused on the vital and value adding ones at that point in time. However, the project members in new projects will most likely adopt those improvement potentials that were not turned into actual improvements.

Since the self-assessment team is not directly involved in the project work, some recommendations of general knowledge could be given as well.

The most important learning was perhaps the deeper knowledge of the applied processes obtained by the project teams. Better understanding on why the processes are applied and how the processes support the project objectives is important.

Many of the project members commented on the process learning, that the assessment had provided them with.

3.3 Results: Learning for the common SPI programme

Most important to the organisation was the identification of the strongly as well as the weakly applied processes. This directly identified areas for overall improvement and focus of the efforts put into the SPI programme. The customers of the SPI programme are the projects and the company management. This knowledge enables the SPI programme to add more value to the customers.

Secondly, the assessments gave more direct information on how the available SPI products, e.g. guidelines and templates, were used by the projects. Thus a picture of the actual demand for each of the SPI products is beginning to appear.

Finally, the assessments were an important measurement for the SPI programme, because it showed the progress of the programme and the results of the invested resources.

3.4 Results: Learning about our software processes

When the results of the three assessed levels are compared, a few common findings are observed:

- Level 1: Basic activities: The appliance of measurements are a weak spot, which also impacts the process management at level 2
- Level 2: There is more focus on product results than on process results. At present we find this is a good balance and it makes good sense to the project teams as well.
- Level 3: Knowledge about the SPI products and project feedback, which is very important in order to improve the SPI products, are inhomogeneous throughout the organization and at present depending on individuals.
- Level 3: Analysis of needs of qualification and infrastructure are not always done, however we find only minor examples of lack of qualification or infrastructure, which is a result of the company policies on education and resources.

3.5 Results: Learning done by the assessors

Software self-assessment is a new activity within Grundfos. The assessors went through initial training, but the real learning happened while the assessment programme was carried out. The assessment team performance is important for the individually results as well as for the cross project validity of the self-assessment results. The team improved the interview techniques, the reporting and the assessment time and schedule. The productivity was increased from 1 hour interview per process in the first assessment to ½ hour per process in later assessments.

The cross project validity is difficult but important. The assessment team worked together, so the team was calibrated in order to have the same evaluation across the projects.

The calibration was found to be a difficult and time-consuming activity, but also essential to the self-assessment programme. In the first assessment it took 2 days for the team to agree on the scores and prepare the report. Resent assessments have taken less than one day. A follow-up seminar together with DELTA [6] is planned.

The team also learned that it is important to use the same language as the projects. We therefore prepared the checklists with questions adapting references to our SPI products and wrote the reports in the language used by the projects. Reporting the results requires openness and an attitude to look for improvements in the assessed project. We found this attitude to be present as a result of our company culture. The carefully selected wording in the reports, followed by the discussions with the project team before publication, gave acceptance of the conclusions and proposed improvements.

3.6 Results: Pre-assessment demand

During the process of carrying out the self-assessment programme, we have found the projects to be very interested in the concepts of software processes according to the BOOTSTRAP model [1]. How does a process work? What makes a good process? A lot of questions have been asked, and some new projects are now demanding pre-assessments prior to the project start. Such a pre-assessment must serve an educational purpose by training the project members in process understanding in general. Furthermore it can help the project members getting answers to such questions as:

- How do we obtain a good process performance in our project?
- What is the difference between controlling the process performance and controlling the work product quality?

The self-assessment team welcomes this demand and has decided to take on this task as well. Within shortly, a pre-assessment process will be designed. Pre-assessment will typically take place at the project boot camp and at selected points in the lifecycle of the project.

3.7 Results: Learning about project maturity

Having a goal of being a World Class company, Grundfos tries to benchmark against competitors, customers and others. The self-assessment results gave the possibility also to benchmark the software development processes between projects by using the SPICE profiles [4] and BOOTSTRAP scores [1] for single processes. We were able to identify the examples of best performance and use the examples for improvement across projects.

Also external benchmarking has been made against the SPICE phase 2 results, as shown in figure 4. Concluding that our project has processes with maturity comparable to what has been found in the SPICE phase 2, expect for two: project management, where our practices in the best projects are better and configuration management, where our practice are weaker.

We learned from this that our software development is comparable to the companies assessed in

SPICE. In some fields we are better than the average, in some we are not, but most important of this learning is the self-confidence and the will to carry on (the boost) this gives to the projects as well as to the SPI programme.

3.8 Results: Powering the SPI programme

The self-assessment gave power to the SPI programme due to the following effects:

- Awareness about processes and process control was raised in general.
- As a consequence of the increased awareness a demand for information and training in process understanding was expressed. The answer to this was the pre-assessment, see paragraph 3.6.
- Increased demands for support for process definition and process tailoring in the projects were experienced.
- Finally the involvement in SPI and its PITs (Process Improvement Teams) was seen as direct participation in the teams, but also as feedback to the PITs about good experiences and examples of practices in the projects. As a result we how see the Software projects pulling information from the SPI programme instead of the SPI programme pushing information to the projects.
- Awareness of the maturity level of our company compared to other companies due to the bench-marking possibility (illustrated by figure 4).

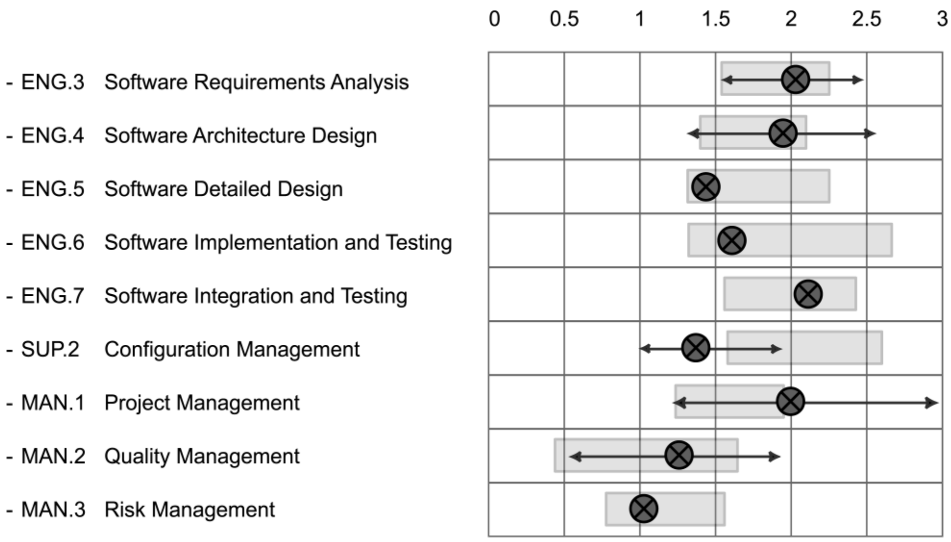


Figure 4: Selected SPICE phase 2 results (ref. [2]) compared to Grundfos project results. The SPICE phase 2 results shown by bars. The Grundfos results shown: The crossed circle is the mean of the measured process. The double arrow shows the range (min. and max.) of measured values.

4 Conclusions and recommendations

4.1 Self-assessment adds value

- In the projects, by setting focus on the needs for process adjustments to fulfil the project objectives and increase the process knowledge and awareness among the project members, as they learn from the assessment.
- For the company SPI programme, by identifying the weak processes and improvement potentials. Also the need for new SPI products can be found. The need of training in specific areas can be identified and the project members or the organisation in general can be trained. The assessments also put attention to the fact that in order to obtain BOOTSTRAP level 3 it is crucial to have an SPI programme, because level 3 demands interaction between the projects and the organisation.
- For the assessors, by giving deeper in-sight into practises and project knowledge. The assessors can be promoted to process consultants
- To the development processes, which can be adjusted, based on input from the observations made in self-assessments. The observations, being calibrated between the projects, and the process risks (failures and mistakes, the human factor) are valuable learning.

4.2 Recommendations on self-assessment

We warmly recommend the use of self-assessment as one method of evaluation your SPI programme. However, there are some conditions for success which should be observed:

- The organisation should have a certain maturity and culture of continuous improvement, where self-assessments are seen as one source of information to the improvements.
- The SPI experiences should be used for improvements of the processes in the projects. It is important to market the self-assessment actively within the organisation.
- Assessors must be trained in assessment by a competent trainer, which also should be able to provide assistance in the first assessments. Assessors should have practical project experiences and should also be involved in the SPI work in general. Interview and presentation techniques could be additional training of the assessors
- And finally two small pit falls for the process:
    - In integrated product development projects including mechanics, hardware and software subprojects; it can be difficult to evaluate e.g. project management, because product level and software subproject level has a tendency to mix up.
    - It is important to report back as fast as possible. If the elapsed time between the interview and the presentation of results is too long, it may change the view of the project members as the project proceed, i.e. they no longer agree with the results.

4.3 Recommendations on the process models

We have selected BOOTSTRAP 3.2 to be the best model for us. CMM has been evaluated too, but we found that the flexibility and European way of thinking in BOOTSTRAP 3.2 was a better choice for us. Concluding on the process models we can recommend:

- It is important to use a common accepted process model to assure both the validity of the assessments and the possibility to benchmark the in-house results with results of external origin.
- The trends in the process maturities over time requires a stable model or backward compability. Updating the model to a newer version, e.g. CMMI or SPICE2002 could result in a loss of the possibility to document the improvement effort.
- The model should be flexible in the meaning, that it shall be possible to select a part of the total model and still be able to do assessments.
- The model should promote the identification of improvement areas, as a direct result of the scoring after the interviews.
- The documentation of the models should be short and easy to use for a trained assessor. Unfortunately the documentation of both SPICE and CMMi are very large documents.
- The model should evaluate per process and per attribute of that process. We have had much benefit from the use of the SPICE profile of the assessment. Furthermore the scoring should be divided into many steps (like the quarter of a point in Bootstrap) enabling the projects to see even small improvements.
- The model should propose improvement frameworks like the ones presented in the CMM-model (Key Process Area descriptions). In this area we have been much inspired by CMM.

5 Literature

[1] The BOOTSTRAP 3.2 process model. www.bootstrap-institute.com, see paragraph about Methodology.

[2] H. W. Jung et al: Findings from Phase 2 of the SPICE trials. Software Process Improvement and Practice. December 2001. p 205-245.

[3] Mark C. Paulk, Bill Curtis, Mary Beth Chrissis, and Charles V. Weber, "Capability Maturity Model for Software, Version 1.1", Software Engineering Institute, CMU/SEI-93-TR-24, DTIC Number ADA263403, February 1993. www.sei.cmu.edu/publications/documents/93.reports/93.tr.024.html

[4] SPICE: The standards, ISO/IEC TR 15504, part 1-9:1998, www.iso.ch/iso/en/CombinedQueryResult.CombinedQueryResult?queryString=TR+15504. The SPICE User Group: More information about development of the standard and the trials. http://seg.iit.nrc.ca/spice/

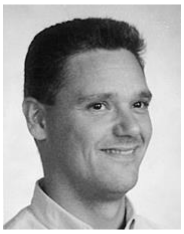
[5] The EFQM Excellence Model. www.efqm.org/new_website/model_awards/model/excellence_model.htm

[6] A.M. Jonassen Hass and Jørn Johansen: Learning from Assessments. In Lars Mathiassen, et.al.: Improving Software Organizations. ISBN 0-201-75820-2.

See also DELTA Software Process Improvement www.delta.dk/services/consulting/sw-process-im/default.htm

Authors CVs

Finn N. Svendsen has 22 years of software experiences, mostly within quality assurance from sectors as industrial products, telecommunication, defence and business management and job functions as project manager, software engineer, certification lead-auditor and at present software quality consultant. He holds a M.Sc. in geophysics, B.Ba in Informatics and M. in Industrial Informatics. He is president of EOQ SW group and has written a number of papers in the field of software quality.
He can be reached at fsvendsen@grundfos.com.



Morten Thy Christensen has worked with embedded and PC software engineering and development for more than 11 years within the industrial sector. For the last 5 years his primary interests has been within Software Quality Management and Software Process Improvement (SPI). He is Project Manager of the Grundfos SPI Programme, Chairman of the SEPG and assessor in the Grundfos self-assessment programme. He holds an M. Sc. in Electronic Engineering.
He can be reached at mtchristensen@grundfos.com.

Kurt Greve has 24 years of software experience, mainly with systems development in various sectors such as industrial automation, defence industry and business computing. Job functions include software process improvement, software- and hardware engineering, system design, project management and development management. At present he is working with software process engineering as a member of the Grundfos SEPG (Software Engineering Process Group), where he is responsible for processes related to software subcontract management and to knowledge management, as well as being a member of the Grundfos self-assessment team.
He can be reached at kgreve@grundfos.com.

Niels Jørgen Strøm Mortensen has worked with embedded software engineering and development for more than 18 years within the industrial sector in such job functions as project manager and software engineer.
For the last 6 years he has also worked with software process improvement as a member of the Grundfos SEPG, as a member of several PIT's of the SPI programme, and as assessor in the Grundfos self-assessment programme.
He holds a B. Sc. in Electronic Engineering with Honours.
He can be reached at njstroemm@grundfos.com.

Experiences with the Combination of Skill Cards and Process Assessments in an Integrated Improvement Framework

Richard Messnarz, ISCN LTD, Janez Hrastnik, IZIT, Slovenia,
Damjan Ekert, ISCN Technical Office, Austria

Abstract

CREDIT was an EU Multimedia project running from 1998 – 2001. It analysed skills assessment processes applied in Europe and also developed a generic platform which can be configured for different skill sets (according to DTI UK skill card standards), supports skills assessment, evidence collection, and advise through an Internet environment.

After the project ended the system has been adapted to be configurable for more general skill sets as well as process assessment models (CMMI, automotive SPICE, etc.).

On a practical case the paper discusses how skills assessment is typically performed and how this relates back to process assessment. This shall create a more holistic view of process improvement where people are actors and part of the improvement process.

Keywords

Skills, Assessment, Platform, Process Assessment

1 What is Skills Assessment

1.1 The Underlying Strategy

Imagine that in a future Europe specific job roles (skills sets), such as automotive engineer, automotive project manager, web software developer, etc. are agreed among the major industry players and universities across European countries.

In this case the job roles

- Will be described according to defined skills definition standards
- Will be assessed according to standard skills assessment processes.

This development, supported by the European Commission in many areas, is the so called “Skill Card” strategy.

Such a common set of skill sets in Europe is needed due to the internationalisation of the businesses (e.g. a German manufacturer with sub-divisions in 170 countries of the world ...) and the free mobility of workers.

However, the practical implementation of skills assessment systems (e.g. the one developed in the Credit project [CREDIT]) showed a possible major fault in its usage by managers. Skill cards were not meant to just assess people, but much more positively to allow people to analyse their current skills level and to set up a learning portfolio to once reach/satisfy a certain job role. So it is completely wrong to use it only as an assessment but rather use it as a positive tool for human resource development supporting the training plans establishment in large firms (required by ISO 9001:2000 [ISO] anyway).

1.2 The Underlying Skills Definition Model

The below figure only gives a simplified picture of the standard skills definition model.

A **Domain**, contains

  Job Roles, which contain

    Units, which contain

      Elements, which contain

        Performance Criteria, which must be proven by

          Evidences.

Figure 1: Basic elements of the skills definition model

Domain: An occupational category, e.g. childcare, first level management or software engineering.

Job Role: A certain profession that covers part of the domain knowledge. E.g. domain = automotive, job role = automotive SW project leader

Unit (UK standards): A list of certain activities that have to be carried out in the workplace. It is the top-level skill in the UK qualification standard hierarchy and each unit consists of a number of elements.

Element (UK standards): Description of one distinct aspect of the work performed by a worker, either a specific task that the worker has to do or a specific way of working. Each element consists of a number of performance criteria.

Performance criterion (UK standards): Description of the minimum level of performance a participant must demonstrate in order to be assessed as competent. A performance criterion may have different relevant contexts.

Evidence: Proof of competence.

1.3 The Underlying Skills Assessment Model

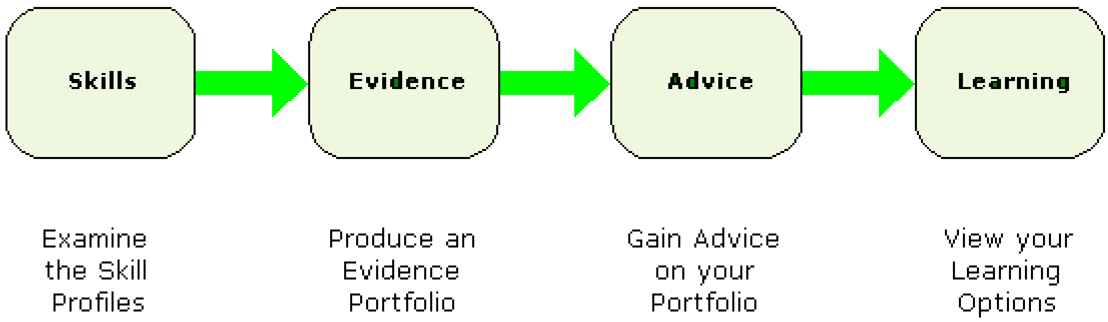


Figure 2: Basic steps of the skills assessment model

Skills - Examine your Skills Profile

- Select a set of skills or competencies, which are required by your profession or job using national standards or your company standards.
- Select a job role you would like to achieve.
- Complete a Self-Assessment of your skills.

Evidence - Produce an Evidence Portfolio

- Collect evidence about your level of skills
- Download a computer programme to guide your assessment which can be used without your internet connection
- Answer some pre-set questions
- Collect a portfolio of evidence that gives details of your relevant education, training, work experiences and other activities that can be used to support your level of skill.
- Send your portfolio via the system to assessors who will assess your evidence and provide written feedback
- Assessors evaluate your competence producing a competence profile

Advice - Gain Advice on your Profile

- Receive written feedback from your assessor
- Highlight areas where you have demonstrated a level of competence
- Receive a learning route to help you develop skills where competence has not been demonstrated
- Decide whether you need to collect new evidence or take further training

Learning - View your Training Options

- Match your training needs with available educational materials and courses
- Register for the recommended training modules
- On completion of your training you can update your portfolio with fresh evidence using the Evidence Planner
- Re submit your portfolio to your Assessor

1.4 Where is it Currently Used

Skills assessment is used within so called APL (Accreditation of Prior Learning) frameworks. This is much supported by countries like UK (promoted by the Department of Trade and Industry), the Netherlands (promoted by national educational institutions and universities), France (with probably Europe's largest open university), and of course the US.

The idea of APL is that people learn a lot on the job, and that old traditional (university exam based) systems only provide certificates if you went to a national college or university, but what you learned on the job is not accredited.

APL would then provide such skills definitions and a network based approach to assess against job roles and get accreditation for specific job roles based on evidences (also evidences from practice count).

Training units of large co-operations (who were partner in CREDIT), such as the Financial Times Management are using such accreditation techniques and skills assessments to upgrade skills of their employees. For instance, the role "first level manager" has been designed and applied.

However, in engineering fields, although ISO 9001:2000 and all other models commonly speak about training plans and human resource management, still such an APL based approach has not been applied so far.

2 How Does This Relate With Process Improvement

This chapter describes the relationship based on a practical example. The data shown are from a real case but have been made anonymous. As the TR15504 assessment model is well known we do not re-discuss it here in this paper. Figure 3 shows part of an outcome of a process assessment using a tailored version of TR 15504. Processes have been measured on the 0-5 capability level scale.

If readers are not familiar with TR15504, for the following text you at least need to understand the levels (shortened but taken from original definitions). If you are familiar with the capability levels, skip the below definitions.

Level 0: Incomplete. There is general failure to attain the purpose of the process. There are little or no easily identifiable work products or outputs of the process.

Level 1: Performed. The purpose of the process is generally achieved. The achievement may not be rigorously planned and tracked.

Level 2: Managed. The process delivers work products according to specified procedures and is planned and tracked. Work products conform to specified standards and requirements. The primary distinction from the Performed Level is that the performance of the process now delivers work products that fulfil expressed quality requirements within defined timescales and resource needs.

Level 3: Established. The process is performed and managed using a defined process based upon good software engineering principles. The primary distinction from the Managed Level is that the process of the Established Level is using a defined process that is capable of achieving its defined process outcomes.

Level 4: Predictable. The defined process is performed consistently in practice within defined control limits, to achieve its defined process goals. Detailed measures of performance are collected and analyzed. This leads to a quantitative understanding of process capability and an improved ability to predict and manage performance.

Level 5: Optimizing. Performance of the process is optimized to meet current and future business needs, and the process achieves repeatability in meeting its defined business goals. Quantitative process effectiveness and efficiency goals (targets) for performance are established, based on the business goals of the organization. Continuous process monitoring against these goals is enabled by obtaining quantitative feedback and improvement is achieved by analysis of the results.

The results illustrated in Figure 3 are based on the P-AVG (average of assessment results from a selected set of projects in the organisation) in comparison with the assessment of the organisation quality management system (ORG). This way to compare profiles has first been proposed by the BOOTSTRAP methodology (TR 15504 compliant methodology).

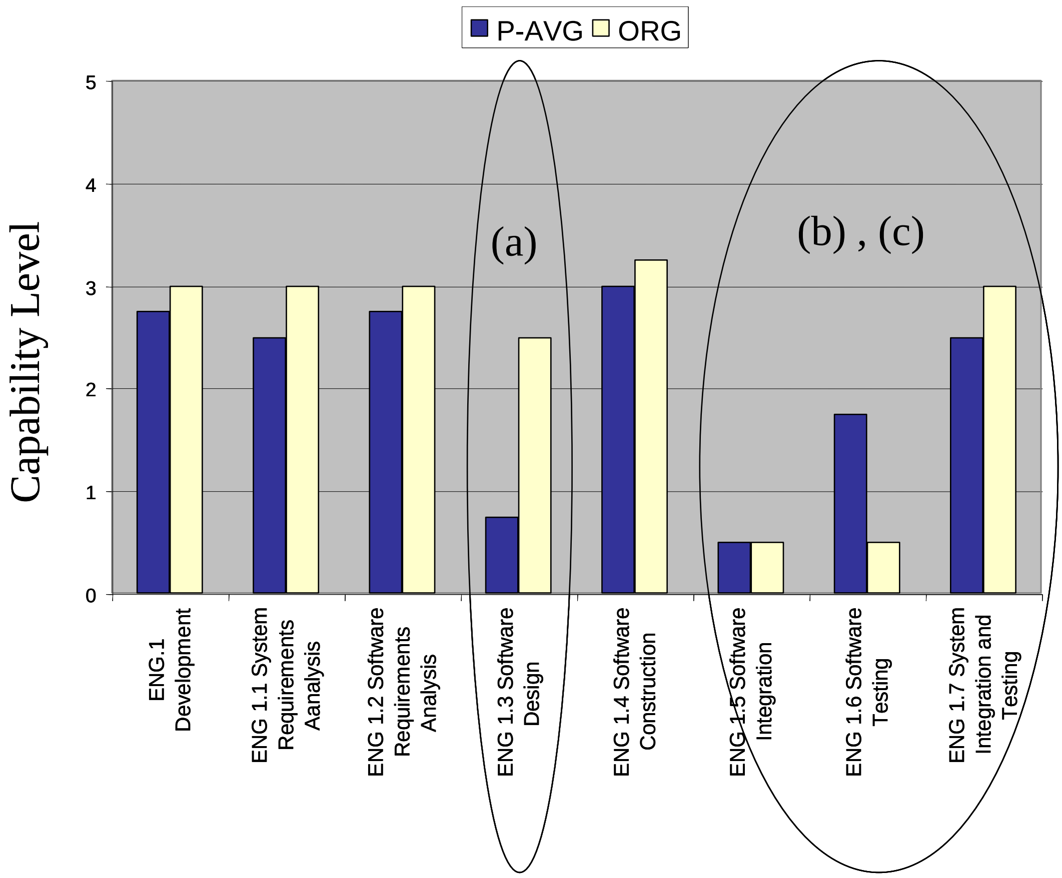


Figure 3: Capability Level Profile

Such an analysis can highlight a number of important inputs for the establishment of improvement plans:

(a) If the capability of a process is higher in the management assessment than in the project average you can conclude that practices and supporting methodologies and tools provided by the management were not properly applied. This can, of course, have many reasons, such as no resources to apply what is provided, missing motivation, missing skills and training, too theoretical models but not practical solutions, etc.

(b) If the capability of a process is higher in the projects than in the management you can conclude that projects have acquired skills without management support driven by the demand put forward by the customers, and that this knowledge has to be assessed for its re-use potential within the organisation.

(c) If the capability of a process is low in both, the projects and the management, then this is a field where no internal know how is available, external knowledge must be acquired.

Based on such a first analysis you might conclude (see Figure 3)

(a) In Software design the organisation provided defined practices, requirements and work product descriptions, management and tracking support, and defined but not tailorable models (they were not tailored – thus level 3 not fully achieved), while projects did not apply them as they claimed that what the organisation proposed could not be tailored to their specific project needs. So the motivation of usage kept low.

(b) In Software Testing the organisation had not invested much in the past but project managers had agreed with their training managers to receive such an education. So the project managers had developed such skills more. However, as long as the organisational support is missing it would not be possible to reach a level higher than 2, to standardise across all projects a common model and view.

(c) In Software Integration both, the management and the projects, are low and here a major field for improvement is highlighted.

The conclusions in (a) – (c) above can, of course, not be drawn without more analysis and information material than just looking at the profiles. Additional factors to be taken into account are –

- Business factors: If an organisation deals with many different business fields which might require different software design knowledge and techniques, then the improvement plan has to set priorities. What comes first, and what will not be supported, has to be decided. In case (a) above, for instance, it might well be the case that the same model cannot be tailored to all business application developments of the organisation.
- Knowledge factors: It has to be decided what knowledge is there and needs to be exploited and multiplied, and what knowledge has to be acquired. In case (b) for instance the experience from the trained project managers and existing practical results from the projects can well be exploited and elaborated as a model agreeable among a number of projects across the organisation.
- Cost factors: in case (c) for instance usually high improvement cost arise because knowledge, practices, experiences, etc. still need to be acquired over a long period of time.

Business managers favour situations (a) and (b) as they see an immediate business impact when they multiply the existing knowledge. So if existing and not so far exploited knowledge can have immediate benefit, would it not make sense to dig further in this gold mine and find out what skills the people have which could be exploited ?

2.1 Relationship Between Processes and Roles

In the organisation process category for process **ORG.2.1 Process establishment process** TR15504 defines practices that deal with the identification of roles, the assignment of people to roles, and the establishment of processes. etc.

Basically, when you establish processes, you need roles which are played by staff members, who produce results according to the work product descriptions, and who execute the practices and management tasks.

Figure 4 illustrates such a process model (taken from a real case, and simplified for this paper) for the Software Testing process.

DD	Design Document
KPL	Known Problem List
RR	Review Report
SFR	Software Failure Report
TC	Test Cases
TD	Test Data
TP	Test Protocol
TPL	Test Plan
URD	User Requirements Document

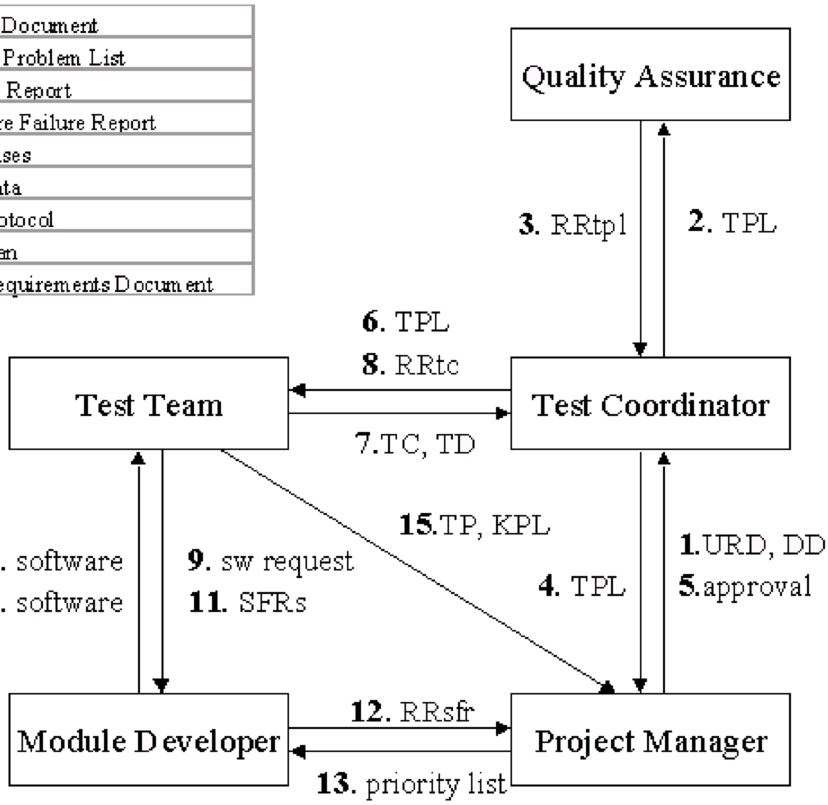


Figure 4: Software Testing Scenario (from a practical case)

Each process like Software Testing therefore falls into specific roles (to perform practices), work products (produced by the roles), and document flows (see above numbering of arrows). It must be noted that starting from the same TR15504 process Software Testing the scenarios might look different depending on the application field (e.g. automotive or multimedia or telecom etc.) and it might differ from firm to firm. However, usually within one corporation such test processes are agreed for the different platforms supported by the corporation.

The above scenario is described below, the numbers correspond to the numbers displayed in Figure 4.

1. The Project Manager provides the Test Coordinator with the User Requirements Document (URD) and the Design Document (DD).
2. Based on the requirements and design document the Test Coordinator creates a Test Plan (TPL) and forwards it to Quality Assurance for review.
3. The Quality Assurance reviews the Test Plan (TPL) and provides the Test Coordinator with refinement notes documented in Review Report (RRtpl).
4. According to the Review Report (RRtpl) the Test Coordinator adopts the Test Plan. The Test Plan receives the status "reviewed". The Test Plan is forwarded to the Project Manager (PM) for approval.
5. The Project Manager approves the Test Plan (TPL) and adopts the Project Plan (PP). The Test Plan receives the status "approved". The approved Test Plan (TPl) is forwarded back to the Test Coordinator.

6. The Test Coordinator provides the Test Team with the Design Document (DD) and approved version of the Test Plan.
7. The Test Team (TT) creates the project specific Test Cases (TC) and forwards the document to the Test Coordinator for review.
8. The Test Coordinator reviews the Test Cases (TC) and provides the Test Team with refinement notes documented in Review Report (RRtc).
9. According to the Review Report (RRtpl) the Test Team adopts Test Cases (TC) and Test Data (TD). The Test Cases receives the status "reviewed". The Test Team requests the latest version of the software from Module Developer.
10. The Module Developer provides the software to the Test Team.
11. According to the Test Plan (TPL) the Test Team performs all planned tests and documents the tests in Test Protocols and software failures in Software Failure Reports (SFRs). The Test Team provides all Software Failure Reports (SFRs) to the Module Developers.
12. The Module Developer reviews the Software Failure Reports (SFRs) and documents the findings and conclusions (description, level, effort, risk) in the Review Reports (RRsfr). The Review Reports (RRsfr) are forwarded to the Project Manager.
13. Taking into account the priorities defined in the User Requirements Document (URD) and findings documented in Review Repots (RRsfr) the Project Manager creates the priority list for failure correction and provides the Module Developer with it.
14. The Module Developer fixes the software failures according to the priority list and provides the latest version of the software to the Test Team.

    The steps from 11. to 14. are repeated until all problems (depending on priorities decided) are corrected and the Project Plan (PP) milestone (Software is Tested) is reached.

15. The Test Team provides the Project Manager with the Test Protocols (TP) and the Known Problem List (KPL).

Ideally there will be many roles with underlying skills sets (requirements) defined either from standards or across organisations (as company standards) against which skills assessments are done and people receive role based learning/training plans.

2.2 Relationship Between Roles and Skills Assessment

As an example, Figure 5 shows the units, and elements of the ASQF certified tester [ASQF]. This model can be inserted into a CREDIT skills assessment environment.

DOMAIN: ASQF - Certified-Tester Foundation Level

UNIT 1: Basics of software testing

UNIT 2: Testing during the lifecycle

UNIT 3: Dynamic testing

UNIT 4: Static testing

UNIT 5: Test Management
- ELEMENT 5.1: Organization
- ELEMENT 5.2: Configuration management
- ELEMENT 5.3: Test expenses, monitoring and controlling

ELEMENT 5.4: Error message management ELEMENT 5.5: Testing standards UNIT6: Support Tools for Tests

Figure 5: Skills Model – Underlying the Certified ASQF Tester at Foundation level

In a typical process assessment (Figure 3) you identify a number of areas where knowledge has to be acquired, such as Software Testing. Skills Assessment offers you a well defined approach to find out -

(a) What skills are available and can be multiplied/exploited.

(b) What skills are missing and need to be acquired for people who will play specific roles, such as Test Co-ordinator.

Figure 6 is taken from the same case which is illustrated from the process side in Figure 3.

In the specific practical case a lot of knowledge existed about managing and correcting errors, and also they archived test cases and had a test team, but

1) They did not know about existing testing standards to compare with.

2) They did not put all test cases and plans under rigorous version control.

3) They did not fulfil all roles (see Figure 4) what a test process requires.

4) They did not collect test metrics and control the efficiency of test performance.

Figure 6 displays a typical skills profile of the responsible manager (organisational management), from which a training plan (in relationship with the ASQF tester, for instance) can be derived.

Figure 6: Skill Profile (DTI UK Standard) for the Unit 5 Test Management in the ASQF Certified Tester Model

The following acronyms have been used in Figures 6 and 7:

Acronym	**Explanation**
CM	Configuration Management (of Test Results and Plans)
TEMaC	Test, Expenses, Monitoring and Controlling
EMM	Error Message Management
TS	Testing Standards
BoST	Basics of Software Testing
Tdtl	Testing during the life-cycle
Dt	Dynamic testing
St	Static testing
TM	Test Management
STfT	Support Tools for Tests

Figure 6 gives more insight into the reasons why Software Testing is weak and looks at the problem from a people/skills perspective. From such a skills profile a typical learning plan can be established to define what project managers need to know to manage Software Testing professionally.

Figure 7 displays the skills hierarchy in the tool set which is currently being configured with the ASQF certified tester model.

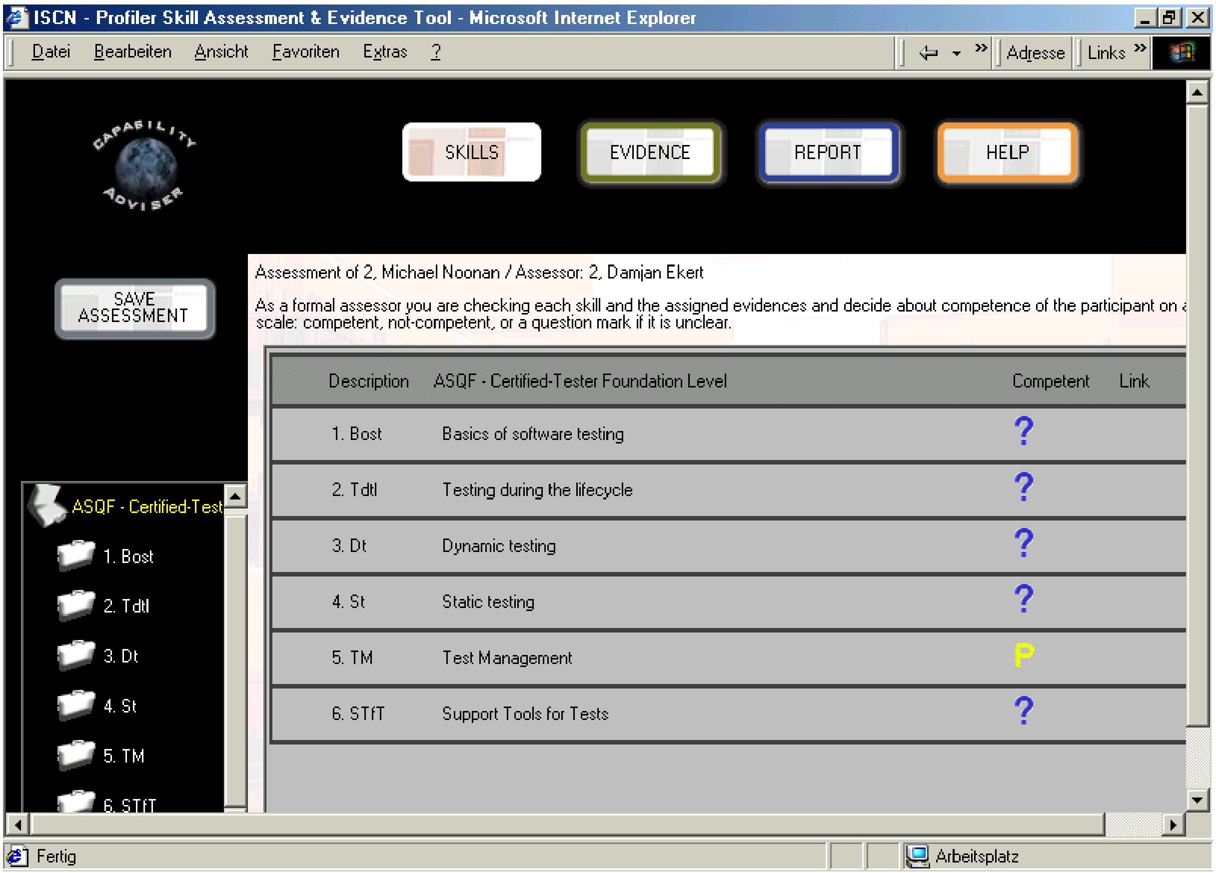


Figure 7: Skills Assessment Tool Configured for the ASQF Certified Tester Model

Please note that in Figure 6 by purpose we set all other units (different from TM) to not evaluated in this paper to not give away information about other units (not related to our arguments).

3 Integration with Process Improvement Planning

As illustrated in Figure 8 assessment reports including action plans might in future also contain skills acquisition plans. Such skills acquisition plans can be derived from skills assessments as described in the previous chapters of this article.

1.	Goals of the Assessment
2.	Assessment Planning
	The Assessment Team
	Selected Projects
	Tailoring - Processes / Levels
	Overall Assessment Plan
	Interview Plans
	Materials/References Provided
3.	SUMMARY FOR THE TOP MANAGEMENT
3.1	Introduction
3.2	Current Status and Capability Profiles
4.	ANALYSIS OF CURRENT STATUS
4.1	Strengths
4.2	Weaknesses
4.3	Improvement Recommendations
5.	Business Goals (to be considered)
6.	Target Profile
7.	Additional Factors/Constraints
8.	ACTION PLAN
8.1.	Proposed Improvement Projects
8.2.	Priorities
8.3.	Risks
8.4.	Action Plan
8.4.1	Skills Acquisition Plan
APPENDIX A	
A.1.	Agenda of Improvement Workshop
A.2.	Outcomes of the Improvement Workshop
APPENDIX B: ASSESSMENT Project 1	
B.1.	Strengths / Weaknesses Feedback
B.2.1	Profile
APPENDIX C: ASSESSMENT Project 2	
Etc.	

Figure 8: Assessment Reports with Action Plans and a Skills Acquisition Plan

Practice shows that the performance of a process assessment is a quite heavy effort. Some organisations might have problems in adding to this effort a skills assessment effort. There are the following two answers to this problem:

1) ISO 9001:2000 clearly defines the requirement of a training plan. Systems which help with the definition of skills, self assessment against skills, and process assessments in combination of skills assessment put forward a well defined structure for managing training plans.

2) In cases of process assessments you would not do skills assessments for roles underlying all processes (selected for the assessment) but just for a tailored set where skills acquisition needs are obvious (e.g. Software Testing in Figure 3).

4 Acknowledgements

We would like to thank the European Commission for supporting the CREDIT project. We would like to thank Dr Jacobeijn Sandberg from the University of Amsterdam for the contributions concerning APL, skills models, and European educational strategies. We would like to thank Ken Street from Telematica for developing the server architecture within the CREDIT project. We would like to thank the Cork Institute of Technology for their support in trialing the system.

5 References

[ASQF] ASQF Certified Tester, Training and Accreditation Model, ASQF, Nuremberg, Germany

[CREDIT] CREDIT Project, Accreditation Model Definition, MM 1032 Project CREDIT, Version 2.0, University of Amsterdam, 15.2.99

[DTI] DTI - Department of Trade and Industry UK, British Standards for Occupational Qualification, National Vocational Qualification Standards and Levels

[FT1] Financial Times Management (1997). Manage people, Unit C15 Project: Dealing with poor performance. Manchester: The Open College.

[FT2] Financial Times Management (1997). Manage people, Unit C12 Project: Working with your team for results. Manchester: The Open College

[ISO] EN ISO 9001 – Quality Management System - requirements, 2000-12-17

[KLARUS] Klarus, R., & Blokhuis, F. (1997). Accreditation of prior learning in the Netherlands. 's Hertogenbosch: CINOP.

[MESS] Messnarz R., Tully C. (eds.), The PICO - Book: Better Software Practice for Business Benefit - Principles and Experience, IEEE Computer Society Press, in publication

[SPICE] SPICE: ISO/IEC TR 15504-2: 1998(E). Information Technology, software process assessment. *Part 2: a reference model for processes and process capability*

6 Author CVs

Dr Richard Messnarz

Dr Richard Messnarz made his MSc at the University of Technology in Graz in Austria. In 1995 he received a PhD from the same university for the work on a QUES - "A Quantitative Quality Evaluation System".
He has participated in a number of European research and industry projects since 1990. He was also the project leader of the Comett expert exchange project ISCN in 1993 which led to the foundation of ISCN itself in 1994.
He is the Executive Director of ISCN. He was the technical co-ordinator in previous process improvement initiatives such as PICO (Process Improvement Combined apprOach), Victory (Virtual Enterprise for Process Improvement), and the co-ordinator of a software engineering group within the pilot project for building a prototype of a virtual university. He is the editor of a book published in 1999 from IEEE "Better Software Practice for Business Benefit - Principles and Experience". He is currently the technical director of a large e-working initiative TEAMWORK, and manages in the initiative MediaISF the technical development for an e-working solution for news agencies collaborating on Eastern European enhancement topics.
He is a chairman and main organiser of the EuroSPI conference series (the former ISCN series).
He is a SPICE and BOOTSTRAP lead assessor.

MSc Janez Hrastnik

MSc Janez Hrastnik made his MSc at the University of Technology in Graz in Austria. He is the founder and director of IZIT in Slovenia which closely cooperates with ISCN LTD in different topics since 1997 such as assessment methodologies, and quality assurance systems.
He has experience as a quality manager from middle sized IT firms, he is a trained ISO 9001 auditor, and he is working for a number of clients in collaboration with Know-Center, a Knowledge Management Competence Centre.

Damjan Ekert BSc

Damjan Ekert is a Master student at the University of Technology in Graz, Austria. He is working with ISCN LTD / ISCN GesmbH Austria since 2001. He is a chief programmer of the Capability Adviser system and has also been integrated into the team developing the NQA e-working platform solutions.

Organisational Culture in Extreme Programming

Peter Wendorff

Abstract

Extreme Programming (XP) is a new software development method that is particularly popular with practitioners. XP claims to be superior to other methods in some situations where time-to-market cycles are compressed and evolving requirements make upfront designs unsuitable. So far scientists have provided little theoretical or empirical evidence of XP's effectiveness. In this paper we show that the concept of organisational culture provides an appropriate conceptual framework for the analysis of XP. This crucial insight provides a completely new perspective on XP and its effectiveness. We conclude that the concept of organisational culture is a fundamental part of the scientific basis for XP. Therefore, many aspects of XP should be reassessed on the backcloth of organisational culture. This new perspective promises huge potential for the clarification of XP, its improvement, research into its effectiveness, and research into its applicability.

Keywords

software engineering, project management, extreme programming, culture, organisational culture

1 Introduction

In colloquial language the term "culture" is often associated with the concept of national culture as studied in anthropology. Why is the concept of national culture useful? It alerts us to the possibility that the customs in a foreign country may differ from the customs in our native country considerably. One result of this awareness may be that we inquire about possible differences prior to a visit. Doing so we can possibly avoid behaviour that might be regarded as offensive there, or we can interpret behaviour that we experience there better. Thus, an appropriate idea of a foreign national culture usually faciliitates communication and interaction in that environment.

Organisations rely on division of labour to accomplish tasks. Usually this involves communication and interaction of individuals, and clearly, the effective and efficient execution of these activities is necessary to ensure high organisational performance. Researchers in the field of management have applied the idea of culture to organisations lately, arguing that this perspective contributed to our understanding of certain phenomena that can be observed in organisations. The popularity of the cultural perspective on organisations has increased since the early 1980s considerably, partly because of claims made by some researchers and consultants that there was a definite link between certain aspects of organisational culture and organisational performance.

Extreme Programming (XP) is a new software development method (SDM) that has become popular in the late 1990s [2]. XP claims to be superior to other methods in some situations that are characterised by vague requirements and rapid change. It belongs to a new class of methods, called "agile" SDMs [4]. These methods share a core of values and principles published as the "Manifesto for Agile Software Development" on the World Wide Web [1]. XP is by far the most widely used agile SDM at the moment, and even the first ever dynabook of the Institute of Electrical and Electronics Engineers (IEEE) has been devoted to it [8].

XP has quickly gained a remarkable degree of acceptance in parts of the software engineering community. Interestingly, it has provoked a vivid and often controversial exchange of opinions, for example, published in the dynabook mentioned above. The observation by Jawed Siddiqi of "both enthusiastic support and equally vigorous criticism" of XP in this dynabook deserves some scrutiny. We think that a culture perspective may help to explain this fundamental disagreement. In section 2 we give a short overview of some aspects of organisational culture. In section 3 we analyse some fundamental elements of XP using a simple framework of organisational culture.

2 Organisational Culture

The concept of organisational culture has attracted much attention from scholars as well as managers recently. Much of this interest has been due to the apparent failure of traditional organisational analysis to explain phenomena in organisations based on objective and formal structural characteristics.

The culture perspective uses established ideas from fields like anthropology, psychology, and sociology. One of its central assumptions is, that organisations cannot be understood comprehensively in terms of their formal characteristics alone, but that there exist influential informal elements in organisations. Informal elements within organisations may include myths, heroes, traditions, mental models, animosities, patterns of behaviour, social perceptions, bias, groupthink, group dynamics, politics, coalitions, friendship, revenge, etc. The culture perspective emphasises the importance of these informal aspects of organisations for the analysis of organisational life. Much of the informality in organisations results from the fact that their members are human beings, and that human behaviour defies a definition in purely formal and rational terms. Therefore, the analysis of processes in organisations should take these informal aspects into account.

2.1 Elements of Organisational Culture

A look at contemporary textbooks on management shows, that there is no single, universally accepted definition of organisational culture. Nevertheless, most definitions refer to some points in the following definition:

Organisational culture refers to

- a common set of beliefs, attitudes, perceptions, assumptions, and values
- that is shared by the majority of an organisation's members
- where it is clearly observable who shares in the culture and who does not
- while it reflects accumulated common learning by organisational members
- who develop it as a response to perceived internal or external requirements
- regarding it as a valid source of appropriate explanations for relevant situations
- for which it suggests or prescribes a range of acceptable behaviour
- that is taught to new members to facilitate their understanding and integration
- and is therefore stable and persistent over long periods of time

As organisational culture is a theoretical artefact, that can only be measured indirectly, it is important to identify elements of an organisation that indicate its culture. A popular layered conceptualisation of organisational culture has been introduced by Schein [11], who differentiates three levels of manifestation. At the first level there are "artefacts", that are most easily discernible by an observer. At the second level there are "values", that are more difficult to observe. The third level are "basic assumptions", that are most difficult to detect and express. We will now briefly explain these three levels (cf. [3, pp. 12]).

Artefacts are the most visible manifestations of organisational culture. Usually they are created by humans in order to solve a problem. Artefacts comprise rules, jargon, stories, symbols, office layouts, and ceremonies. These elements of a culture are visible to an outside observer directly by watching the behaviour of the organisational members.

Values are consciously held reasons for behaviour, but they are not directly observable for an outside observer. Values are strongly related to ethical codes, and they express what ought to be done.

Basic assumptions are the least obvious manifestation of culture. They comprise reasons for behaviour that are not consciously held by organisational members, because they are taken for granted. Basic assumptions are not confrontable or debatable, and examples include the basis on which individuals are respected, whether cooperation or competition is desirable as mode of behaviour, and how decisions are made. These unconsciously held assumptions guide human behaviour, and they are not directly observable for an outside person.

The relationship between the three levels of organisational culture described above can be demonstrated by the following example. We assume two basic assumptions, namely (a) humans are generally lazy and try to avoid effort, or (b) humans are generally motivated and enjoy to do good work. Depending on our basic assumption we could then proceed to the level of values and find that (a) strict control of procedures is desirable, because it leads to productivity, or (b) a motivating workplace that provides opportunity is desirable, because it leads to productivity. Depending on these values, we might then choose an appropriate artefact to serve our value, for example, by (a) using a production line, or (b) forming a semi-autonomous work team.

2.2 Team Culture

In this paper an institutional notion of the term "organisation" is used, and accordingly it denotes something with a physical presence, that is set up formally, in order to achieve certain goals, and is given some degree of autonomy, together with some resources. Having said this, an organisation can be a whole company, or a department, or a project team, etc.

In this paper we will deal with organisational culture in the special context of a software development team. We think that most of the research findings that relate to organisations can be adapted to the special situation of a team with only minor changes. We will use the narrower term "**team culture**" to refer to the specific organisational culture that exists within a particular project team. This means that it is something shared by most of the team's members.

The culture of any organisation is called "**organisational culture**" in this paper. Therefore, organisational culture is used as a broader term of team culture. Finally, if any form of culture is referred to, e.g. including national culture, then the term "**culture**" may be used. Therefore, culture is a broader term of both, organisational culture and team culture.

2.3 Functions of Organisational Culture

The importance of culture for the smooth functioning of organisations has been emphasised by many writers. Organisations provide venues where people with differing backgrounds meet, and clearly this creates a potential for disagreement. In order to act effectively, an organisation has to achieve some degree of consensus and cooperation, for example, through formal regulations. This formal approach often results in organisational designs that rely on extrinsic motivation of organisational members. There is strong evidence that common and intrinsic motivation of members does often increase the effectiveness of organisations. An organisational culture represents a shared basic mindset, and therefore it can motivate action, facilitate agreement, and encourage cooperation. In this way it can lead to intrinsic motivation, thereby reducing the need for extrinsic motivation [3, pp. 89].

2.3.1 Consensus

Contemporary software development of non-trivial systems requires teamwork. In the software industry these teams are often composed of individuals who vastly differ in formal qualification, length of professional experience, and their current professional focus. These diverse backgrounds bear the potential for diagreement and conflict within the team. For such a team to become effective, some consensus must emerge, as is pointed out by Weinberg: "Still, if a team sets to work too quickly - before establishing a real consensus about the structure of the project and the division of work - trouble will manifest itself in one way or another. Social psychologists have verified in other contexts that failure of one or more members to share the group goals affects the group performance - not only through that member's share but through a reduced performance on the part of the others, for they invariably perceive the division within the group or the indifference of one of the members." [13, p. 73]

From the definitions given in section 2.1 it is clear that a team culture can encourage consensus within an otherwise diverse team.

2.3.2 Compatibility

Organisational culture narrows down the range of acceptable behaviour in certain relevant situations. In the ideal case a culture can ensure that the possible options of behaviour are compatible with another, thereby reducing the possibility of conflict. In this way a team's culture can help to reduce the need for formal communication, coordination, and control.

Weinberg illustrates that incompatible behaviour may increase the need for formal communication, coordination, and control: "A member who is competent but who does not get along with the others can be an even more serious problem for the democratic group than an out-and-out incompetent. In an authoritarian group, such a member would not have much contact with others on a working basis anyway, so as long as he gets along adequately with the leader, he presents no particular problem. Indeed, some programmers prefer to work under a strong, centralized leader in order that they do not have to socialize with their fellow workers. But in a democratic team, an antisocial member cuts lines of communications and is a constant impediment to consensus in team meetings." [13, p. 87]

In his conclusion Weinberg p. 5ii stresses the importance of compatible behaviour: "Even so, the number one mistake I've seen in team formation over the years is that of adding members to a team based on some presumed skill, rather than on compatibility of the new members with the rest of the team members. Skill may be hard to come by, but it's always easier to buy or train than compatibility." [13, p. 5.ii]

2.3.3 Motivation

Many organisations rely on extrinsic motivation and formal control to influence the behaviour of their members, only to find that this may lead to job dissatisfaction and alienation. An appropriate organisational culture has the potential to help the adoption of the organisation's goals by its members, leading to more effective and efficient intrinsic motivation instead. This point is demonstrated by DeMarco and Lister: "Teams are catalyzed by a common sense that the work is important and that doing it well is worthwhile. The word *well* in this last sentence is essential: The team assigns itself the task of setting and upholding a standard of prideful workmanship. All team members understand that the quality of the work is important to the organization, but the team adopts a still higher standard to distinguish itself. Without this distinguishing factor, the group is just a group, never a real team." [6, pp. 177]

Referring to the ubiquitous motivational posters and plaques, that are a common sign of the attempt to establish a productive culture by bureaucratic organisations, they remark: "They seem to extol the importance of Quality, Leadership, Creativity, Teamwork, Loyalty, and a host of other organizational virtues. But they do so in such simplistic terms as to send an entirely different message: Management here believes that these virtues can be improved with posters rather than by hard work and managerial talent. Everyone quickly understands that the presence of the posters is a sure sign of the absence of hard work and talent." [6, p. 178]

2.3.4 Performance

Much of the interest in organisational culture has been created by books that link strong organisational cultures with certain characteristics to competitive advantage and superior performance. One of the most successful books of that kind has been the empirical study of successful American companies by Peters and Waterman first published in 1982 (reprinted as [9]). While there is some evidence in favour of the link between a strong culture and superior performance of organisations, the relationship is not necessarily positive [3, p. 91]. Indeed, the stability and persistence of organisational cultures can become a liability in a fast-changing environment, as it can possibly impede adaptation to a changed situation. This has been exemplified by some follow-up studies of [9] that found that only a few years later some of the erstwhile successful companies had fallen from grace and were struggling with changed market conditions [5, pp. 159]. Other problems experienced with inappropriate cultures are groupthink, selective perception, exclusion of minorities, and blocking of desirable learning processes [10].

2.4 Management of Organisational Culture

If certain types of organisational culture can have a beneficial impact on performance in certain situations, the question naturally arises how cultures can be created, stabilised, or changed by management.

As has been pointed out in section 2.1, culture is the result of a long-lasting and collective learning experience by a group of people. Research has uncovered, among others, three main factors that shape the creation of cultures [3, pp. 42]. First, national culture has a strong impact on organisational cultures. Second, the creation of strong organisational cultures is often inextricably bound to the influence of a dominant leadership personality [11]. Third, the nature of the business and its operating environment have a profound impact on the resulting organisational culture.

A strong organisational culture is taught to new members, and that is one major reason why it is stable and persistent over long periods of time. Research suggests that a careful preselection of applicants reduces the risk of a mismatch between individual personality and team culture. Admitted applicants are then normally subjected to socialisation processes, and again research shows that skilful management of these processes can greatly reduce the likelihood of failure [3, pp. 55].

Many managers believe that organisational culture can be used to influence organisational performance in a favourable way. This has stimulated interest in the process of culture change. Research in this area suggests that successful culture change requires skilful management. A number of models have been proposed to analyse and implement culture change. A consistent outcome of empirical research is that strong and active leadership is a critical success factor for cultural change [3, pp. 115], [11].

3 Organisational Culture in Extreme Programming

In this section we will analyse XP using the seminal work [2] by Kent Beck as our sole source of information on XP, because it is still the most authoritative publication on "XP - its roots, philosophy, stories, myths." [2, p. xvi] We treat the account given in that book as the description of a prototypical XP project, and believe that the cultural flavour depicted there adequately reflects the ideas of the most important creators of XP.

3.1 Artefacts

3.1.1 Ceremonies

Probably one of the most evident ceremonies in XP is the arrangement of facilities at the start of a project. Beck puts much emphasis on this, stating: "Taking control of the physical environment sends a powerful message to the team. They are not going to let irrational opposing interests in the organization get in the way of success. Taking control of their physical environment is the first step toward taking control of how they work overall" [2, p. 80]. This is one of the first actions the team members perform as a group, and clearly it brings people together and encourages social interaction.

Another ceremony is devised by Beck when XP is adopted with an existing team and software. He assumes that this will lead to major revisions of the existing software, called "refactorings" in XP, and stresses that clear goals must be stated for these refactorings. These goals are written on cards, and given this physical representation of the task, its completion becomes an evocative symbol of accomplishment: "Set these goals, put them on cards, and display them prominently. When you can say the big refactoring is done (it may take months or even a year of nibbling), have a big party. Ceremoniously burn the card. Eat and drink well" [2, p. 127]. In particular, if the project switched to XP is in trouble, Beck devises a public ceremony to mark the new beginning: "However, if you're going to switch a troubled project to XP, make it a dramatic gesture. Half measures are going to leave everybody in more or less the same state that they were before. Carefully evaluate the current code base. Would you be better off without it? If so, flush it. All of it. Have a big bonfire and burn the old tapes. Take a week off. Start over fresh" [2, p. 129].

3.1.2 Workspace Layout

The XP workspace layout is described by Beck as an open workspace: "We will create an open workspace for our team, with small private spaces around the periphery and a common programming area in the middle" [2, p. 77]. He recommends to put the development machines close to another on tables in the middle of an open-plan office, side by side, and even discourages the use of cubicles. The rationale behind this is that "XP is a communal software development discipline. The team members need to be able to see each other, to hear shouted one-off questions, to 'accidentally' hear conversations to which they have vital contributions" [2, p. 79]. Another requirement is that there must be enough space for two persons to sit in front of one computer.

3.1.3 Stories

In his seminal book on XP Beck frequently uses stories, either fictional or from his past experience, to highlight certain aspects of XP. A typical example of a fictional story is his description of "A Development Episode", where he gives a detailed account of expected social interaction among the members of a XP team [2, pp. 7]. Addressing the reader as his hypothetical teammate he writes: "This chapter is the story of the heartbeat of XP - the development episode. This is where a programmer implements an engineering task (the smallest unit of scheduling) and integrates it with the rest of the system. I look at my stack of task cards. The top one says 'Export Quarter-to-date Withholding.' At this morning's stand-up meeting, I remember you said you had finished the quarter-to-date calculation. I ask if you (my hypothetical teammate) have time to help with the export. 'Sure,' you say. The rule is, if you're asked for help you have to say 'yes.' We have just become pair programming partners. [...]" [2, p. 7]

Another story illustrates supportive relationships in one of the first ever XP projects. Beck recounts: "I asked the C3 team (that first big XP project I mentioned earlier) to tell me stories about their proudest moment on the project. I was hoping to get stories about big refactorings, or being saved by tests, or a happy customer. Instead, I got this: '*The moment I was proudest of was when Eddie moved to a job closer to home, avoiding a two-hour daily commute, so he could spend more time with his family. The team gave him total respect. Nobody said a word against him leaving. Everybody just asked what they could do to help.*' This points to a deeper value, one that lies below the surface of the other four - respect. If members of a team don't care about each other and what they are doing, XP is doomed." [2, pp. 34]

Yet another story commemorated by Beck describes the deliberate creation of a symbol, in the shape of a kitchen timer, to control the length of meetings: "For example, on the Chrysler C3 project, design meetings were going on for hours without resolution. So I bought an ordinary kitchen timer and decreed that no design meeting could be longer than 10 minutes. I don't believe the timer was ever used, but its visible presence reminded everyone to be aware of when a discussion had ceased being useful and had turned into a process for avoiding going and writing some code to get the answer for sure" [2, p. 74].

3.1.4 Rules

Rules play a central part in XP, and Beck explains: "XP is a discipline of software development. It is a discipline because there are certain things that you have to do to be doing XP. You don't get to choose whether or not you will write tests - if you don't, you aren't extreme: end of discussion" [2, p. xviii]. A major building block of XP are the 5 so-called "basic principles" [2, pp. 37] and the 12 so-called "practices" [2, pp. 53]. Both the basic principles and the practices are essentially rules.

3.2 Values

XP explicitly defines a set of four values, namely communication, simplicity, feedback, and courage [2, pp. 29]. Later Beck adds another value, underlying these four values, namely respect [2, p. 35].

3.3 Basic Assumptions

In the concluding chapter of his book Beck states: "All methodologies are based on fear. You try to set up habits that prevent your fears from becoming reality. XP is no different in this respect from any other methodology. The difference is in what fears are embedded in XP. To the degree that XP is my baby, XP reflects my fears" [2, p. 165]. Fears are the result of subconscious cognitive structures that correspond to the notion of basic assumptions mentioned in section 2.1. Beck's remark makes evident that he is aware of basic assumptions as determinants of software engineering methods like XP. In this section we will elaborate two basic assumptions that we think are important, underlying basic assumptions of XP.

3.3.1 Consensus on Simplicity

XP puts much emphasis on simplicity of the resulting software. Simplicity is addressed as one of the practices [2, p. 54], one of the basic principles [2, p. 37], and one of the four values [2, p. 29].

A look at [7] immediately proves that it would be difficult to define some software metric "Simplicity" that could be validated empirically. Concepts like simplicity or complexity are in fact highly subjective, and often one person's simplicity is another person's complexity.

In XP the practice of "collective ownership" allows any pair of programmers to change any piece of code at any time. There is even a duty to "improve" the code, if the opportunity arises. Therefore, the same piece of code might be changed by different programmers, each claiming that the change led to a simpler design. Given the subjective nature of the term "simple design" this could lead to a number of contradicting changes. As long as different people have fundamentally different ideas of "simplicity", this process might go on forever.

Beck does not present a definition of simplicity in [2], and because of the subjective nature of the concept this would be questionable anyway. Nevertheless, because of the great importance of the concept of simplicity in XP, it must be made sure that the quest for simplicity doesn't lead to numerous contradicting changes of code and, possibly, ensuing acrimonious discussions. The only way to achieve that is, that the potentially conflicting notions of simplicity of different team members do converge over time.

One basic assumption of XP seems to be that within an XP project team the potentially different notions of simplicity converge with acceptable effort.

It is useful to reconstruct basic assumptions like this, because it may help to identify sources of fundamental disagreement, namely implicit, incompatible basic assumptions. We want to illustrate this very important point using an example. In this example we assume a scientist who has worked in the field of software measurement for some years, and who takes it for granted that concepts like complexity (or simplicity) are subjective by nature. Actually, his way of thinking would reflect the current state of affairs in software measurement [7]. Being introduced to XP, this scientist would quickly discover the central role of simplicity in XP. Being keenly aware that there is nothing simple about simplicity in software measurement, he would look for an elaborate definition of the concept, and would inevitably be disappointed by the content of [2] on the subject. The likelihood is, that this software engineer would regard XP as unfounded and unscientific and dismiss it.

The important point to note here is that both Kent Beck as well as our hypothetical software engineer are highly qualified software engineering experts. Nevertheless, they may arrive at opposing conclusions, one thinking that a concept like simplicity can be useful, even without an elaborate scientific definition, the other thinking that a concept like simplicity without a scientific definition is subjective, arbitrary, and therefore pretty useless. The fundamental disagreement of the two experts in this example is the result of their fundamentally different, implicit basic assumptions, or in other words, their disagreement results from their differing cultural backgrounds. It is obvious that deeply ingrained assumptions of that kind are barely debatable.

3.3.2 Human Motivation

One of the twelve practices in XP is called "collective ownership" [2, pp. 54].

Usually the owner of a thing takes care of it and keeps it in order. If the owner of a thing is a person, then that person will ensure that the property is not abused or damaged. In cases where there is no obvious human owner of a thing, e.g. in the case of a company car, some people exhibit a tendency to treat that property with less regard than their own. In that situation the owner often appoints a proxy endowed with some authority to act on behalf of the owner. Alternatively, the owner may rely on the responsible behaviour of the users of his property. In the case of company cars, at least, different companies prefer different solutions, some rather rely on a single person in charge, some rely on some form of collective responsibility, others resort to alternative solutions.

In XP "collective ownership" is very important, because it mandates that "anybody who sees an opportunity to add value to any portion of the code is required to so at any time" [2, p. 59]. Other practices, like for example, the so-called "refactoring" [2, p. 58] crucially depend on the practice of collective ownership. XP supports the practice of collective ownership through a value called "courage". The value courage acknowledges that it often takes courage to fix complex design errors, due to the high effort and the possible complications. To lend additional support to collective ownership, XP requires that automated tests must be present for any program feature, to give programmers the opportunity to verify the correctness of any modifications of the software. The assumption here is, that programmers who are afraid of programming errors might compromise the practice of collective ownership to avoid being blamed for programming errors. If the risk of blame is reduced, then they may be more inclined to put collective ownership of the code into practice.

Beck does not specify command structures, rewards, or punishments to ensure that the members of a XP team act according to the practice of collective ownership. Nevertheless, XP critically depends on the acceptance of collective ownership by the team members.

One basic assumption of XP seems to be that in an XP project team intrinsic motivation can be achieved that ensures the implementation of collective ownership.

The reconstruction of this basic assumption again reveals a potential source of fundamental disagreement over XP. This assumption refers to human nature, and surely, this is an area where fundamentally different basic assumptions prevail. One typical example of this are McGregor's Theory X and Theory Y assumptions about human nature, that can potentially turn into self-fulfilling prophecies [12, p. 16].

The important point to note here is that the conclusion about the feasibility of collective ownership of code will critically depend on basic assumptions about human nature. A person, for example, with basic assumptions comparable to McGregor's Theory X, assuming that people basically dislike work, need direction, and avoid responsibility, will probably arrive at the conclusion that collective ownership is unfeasible. A person, on the other hand, with basic assumptions comparable to McGregor's Theory Y, assuming that people like work, are creative, and accept responsibility, will probably arrive at the conclusion that collective ownership is feasible and a promising idea. This fundamental disagreement is the result of irreconcilable basic assumptions, that are barely debatable.

4 Conclusion

Extreme Programming (XP) is a new software development methodology that is eagerly embraced by many practitioners, but scientists have largely failed to address its effectiveness. In this paper we have used the layered conceptualisation of organisational culture developed by Schein to analyse XP. We have found that the culture perspective is an appropriate way to look at XP. This leads to the following results and suggestions:

1. Team culture is a form of organisational culture, a subject that has been addressed by management researchers intensively and successfully. Therefore, we should make use of these results in order to improve XP. One first step is outlined in this paper. XP uses so called "values", "basic principles" and "practices". There is a striking correspondence between these elements of XP and the layered conceptualisation of organisational culture developed by Schein. Therefore, the culture perspective provides a promising conceptual framework for a systematic assessment and consolidation of these essential elements of XP. This points the way to a clearer definition of XP, with a sound scientific background.

2. Research on the impact of strong cultures has identified some likely benefits that probably explain the effectiveness of XP. A strong culture usually raises consensus, ensures compatibility, provides motivation, and can be a source of superior performance. Future research on the effectiveness of XP should pay more attention to these effects.

3. Strong cultures can have negative effects too. These are potential sources of problems, and therefore, managers should pay constant attention to these warning signs. A strong culture can result in groupthink, may promote selective perception, result in exclusion of minorities, and block desirable learning processes. Future research in this area might lead to intervention techniques to deal with these practically important problems effectively.

4. There is a large body of empirical evidence that suggests a strong influence of culture on performance. XP is informed by the idea of a strong culture, and this probably explains why XP has been found very effective in practice in some cases. We have cautioned against oversimplification though, because management research also suggests that the relationship between culture and performance is complex. Clearly, much more research should be devoted to this issue in future. Nevertheless, the culture perspective provides a plausible explanation for the successful use of XP in some cases.

5. Because team culture plays an important role in XP, we can possibly use it to predict XP's failure at an early stage of a software development project. If we fail to establish a team culture conforming to XP's requirements, low performance is likely to follow. This may indicate problems in an XP project long before deadlines slip and budgets overrun. Clearly, research results in this area will be of high practical value.

6. Team culture results from complex social processes. XP defines the role of team coach, but the description of this role given by Kent Beck is rather vague. Nevertheless, from his remarks it can be concluded that he thinks of the team coach as a facilitator of team culture. This idea is not new, it corresponds to the concept of cultural engineer described in some management texts. Therefore, we think that the role of team coach in XP can be clarified and enhanced based on the concept of cultural engineer.

7. Research in the field of organisational culture has consistently indicated the close interrelationship between strong leadership and the emergence of a strong culture. Many scientists believe that the two are inextricably intertwined. This suggests a focus on the subject of leadership in future research on the effectiveness of XP. Empirical research in this area should try to establish how much of the result of a XP project's outcome is due to the bare XP method, and how much of the result must be attributed to the particular leadership personality. The answer to this question is of great practical importance for the conduct of future empirical research in this field, because if the leadership variable turns out to be significant, then it must be monitored in future studies and controlled in future experiments, in order to achieve valid and reliable results.

8. To stabilise a XP project team with an appropriate and successful team culture it will regularly be necessary to control the addition or replacement of team members carefully. Changes to the team can have an adverse effect on the stability of the culture, and thereby lead to reduced performance. Managers should make sure that people added to the XP team have personalities compatible with the culture present in the team.

9. The culture perspective directs attention to the implicit basic assumptions of a software development method. We have suggested two basic assumptions that seemingly underly XP. First, in a XP project team a consensus about the notion of simplicity emerges over time. Second, in a XP project team intrinsic motivation can be achieved that ensures the implementation of collective ownership. It is important to identify basic assumptions, because somebody who disagrees with these basic assumptions will probably find it difficult to put XP into practice effectively. Moreover, we think that part of the fundamental disagreement and often bitter dispute about XP in the software engineering community results from irreconcilable basic assumptions, that are barely confrontable or debatable.

10. A XP project does take place in the context of an organisation. This implies that we have to balance two separate cultures: XP's team culture and the culture of the whole organisation. Naturally, these two don't exist in isolation. Therefore, we must look for ways to avoid a culture clash.

5 Literature

[1] The Agile Alliance: *Manifesto for Agile Software Development.* http://agilemanifesto.org/ (last visited on 16/06/2002), 2001.

[2] Beck, K.: *Extreme Programming Explained: Embrace Change*. Longman Higher Education, 2000.

[3] Brown, A.: *Organisational Culture*. Prentice-Hall, 1998.

[4] Cockburn, A.: *Agile Software Development*. Addison Wesley Longman, 2001.

[5] Dawson, S.: *Analysing Organisations*. MacMillan Press, 1996.

[6] DeMarco, T. and Lister, T.: *Productive Projects and Teams*. Dorset House Publishing, 1999.

[7] Fenton, N. and Pfleeger, S. L.: *Software Metrics*. International Thomson Computer Press, 1996

[8] Institute of Electrical and Electronics Engineers: *Dynabook on Extreme Programming*. http://computer.org/seweb/dynabook/Index.htm (last visited on 16/06/2002), 2000.

[9] Peters, T. and Waterman, R.: *In Search of Excellence: Lessons from America's Best-Run Companies*, Warner Books, 1988.

[10] Rollinson, D. et al.: *Organisational Behaviour and Analysis*. Addison Wesley Longman, 1998.

[11] Schein, E. H.: *Organizational Culture and Leadership*. Jossey-Bass Publishers, 1992.

[12] Schermerhorn, J. R.: *Organisational Behaviour*. John Wiley & Sons, 2000.

[13] Weinberg, G. M.: *The Psychology of Computer Programming*. Dorset House Publishing, 1998.

6 Author CVs

Peter Wendorff

Mr. Wendorff received university degrees in economics and engineering. He has been working in the software industry since 1993 as a consultant. He has cofounded ASSET GmbH, Germany, in 1999. You can reach him at p.wendorff@t-online.de.

AcceleratedCMM for AcceleratedSAP in ERP Implementation for Business Process Re-engineering

Kim Man Lui and Keith C.C. Chan

Abstract

AcceleratedSAP, a methodology for SAP R/3 implementation, is adopted by a diversity of industries around the world. It has proven to be useful and successful. Likewise, the CMM is widespread and it describes the best principles for software organizations so as to increase the software capability of running software projects successfully. At first glance, they are very different and they were developed with totally different objectives. By closer examination, however, AcceleratedSAP and the CMM share some disciplines in common. They both describe what good engineering and management practices are. For the CMM, it is about software process improvment. For AcceleratedSAP, it addresses what and how a set of planned tasks should be done for implementing SAP R/3 and business process reengineering (BPR) in organisation. Both of them are adopted to ensure successes of software projects. In this paper, we discuss how AcceleratedSAP and the CMM can be aligned in the software project hierarchy. By combining the uniqueness of AcceleratedSAP and the CMM, we propose a new model, called AcceleratedCMM that can be adopted for ERP software development so that the advantages one can realize through CMM KPAs can be exploited to improve BPR.

Keywords

AcceleratedCMM, AcceleratedSAP, Business Process Reengineering

CMM, ERP, Software Model

1 Introduction

Future prospects of many companies will be determined by how they overcome challenges associated with globalisation, accompanied by external market competition and the adoption of business process re-engineering (BPR). The concept of BPR embraces not only re-engineering but also best business practices, transformation, optimisation, and continuous improvement. For all these achievements, an integrated enterprise resource planning (ERP) system should be regarded as indispensable. SAP has established itself globally as the dominant supplier of core business systems software. Many multi-national corporations have adopted SAP as their global standard, and an increasing number of smaller enterprises are following suit. An SAP software implementation project is typically complex and costly, affecting the whole organization [1]. Typically, when an SAP project overruns, or its schedule slips, the additional project costs can be unexpectedly high. Thus, the success of an SAP implementation is measured and justified by: (1) project/budget control, and (2) the results/benefits of business process re-engineering. Obviously, SAP is motivated to ensure that every implementation of an SAP product (i.e., SAP R/3) is successful. SAP has devised a methodology called AcceleratedSAP, commonly known as ASAP, which provides a roadmap for cost-effective SAP R/3 implementation [2].

Considering SAP implementations as a subset of more general software projects, we are interested in understanding whether a model of software processes, such as the capability maturity model (CMM), is of any merit or inspiration to the generic ERP implementation. The CMM explicates a high-level disciplined software model that is able to improve continuously (at the CMM level 5) an organization’s software process capability [3]. In a similar fashion, SAP is interested in continually optimising their recommended procedures (ASAP) for efficiently implementing their software.

The CMM was originally intended for large-scale software processes, but in practice its application has been much more widespread. Thus, this model could be adopted in system implementation projects (such as ERP), as effectively and beneficially as in software product development processes. However, relevant case studies are very sparse in the literature. In addition, very little is understood about the bridge between software process improvement (SPI) models (such as CMM) and models/practices for ERP implementation (such as ASAP); clearly, both approaches should not be in any conflict.

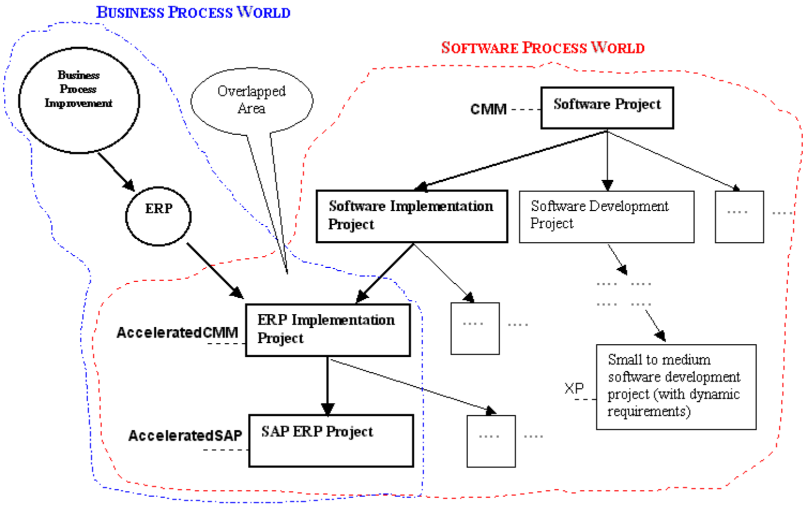


Figure 1: Methodology and Hierarchy of a Software Project

Figure 1 depicts a hierarchy of software projects. At the top level, a general software project combines characteristics from a range of different project types (e.g., implementation or development). Hence at this level, good engineering and management practices, and improvement priorities, are sought to enhance software process capability that can benefit all types of software projects. At lower levels, different types of project become more distinct, and project features/characteristics are defined more precisely. Therefore, any project methodology derived from the upper level may have to be refined to include more specific attributes/features. This however is not the case in contrast to many existing methodologies for different types of projects, which have been developed independently, and therefore, share few common properties being inherited from a high-level model.

As we extend the hierarchy of projects further, the lowest level involves very specific project methodologies that describe all the details of how to execute a certain type of project. Figure 1 shows one example; eXtreme Programming (XP) is a methodology suitable for small software development projects [4]. Consider SAP implementation as part of a hierarchy of projects: a low-level how-to roadmap ASAP is used to implement SAP projects across many different kinds of business, and by many SAP business partners around the world.

Figure 1 illustrates different types of projects in a general tree structure where a lower-level model is derived from a higher-level model (top-down); however, the process could also be reversed (bottom-up) if needed. A hierarchical model of related project methodologies offers some advantages in reusing existing project methodologies in a systematic manner. This hierarchy also uncovers relationships between project methodologies that help determine whether a methodology is applicable in certain contexts. For example, it has been argued that CMM is not applicable for some types of development project because of their differences and uniqueness [5]. Although another paper has reported that the CMM and XP are complementary [4].

It is possible for a project hierarchy to be based on characteristics outside of the field of software. SAP implementations are both software and business re-engineering projects. A project hierarchy should include specific business elements, which affect how software implementation processes are interpreted within certain business contexts/processes.

This paper proposes an ERP software implementation model called AcceleratedCMM that aims to bridge the gap in understanding how ERP implementation projects fit within a hierarchy of software projects. AcceleratedCMM is based on ASAP and the CMM, and it illustrates how key process areas (KPA) from the CMM are organised to gain insights into successful ERP and BPR projects. In academia, AcceleratedCMM suggests a hierarchy of software project models that show the relationships between individual project methodologies.

This paper does not intend to suggest that the CMM is a universal model for software projects. Neither do we advocate ASAP as a universal model for any SAP or BPR implementation. We simply limit our discussion to SAP implementation projects using ASAP .

We begin with a review of ASAP. Comparisons between key practices of ASAP and the CMM are then presented. Next, we propose AcceleratedCMM, which is a framework for ERP implementation projects expressed in terms of the CMM. The last section presents our conclusions and contributions towards software project management and SPI.

1.1 AcceleratedSAP (ASAP)

The latest version of AcceleratedSAP is ValueSAP Rel. 4.6C. As mentioned, SAP implementation projects are costly. To ensure the best use of project resources, ASAP addresses all the details of SAP implementation from the perspective of different management levels: for review by senior management and in working procedures used by line managers. In addition, SAP implementation focuses on business process re-engineering. ASAP must then facilitate how to transform a business from the existing processes, called the "as-is" environment in SAP terminology, to the future processes, called the "to-be" environment.

The ASAP framework divides an implementation project into five sequential phases, each of which indicates a specific purpose toward the achievement of final project goal. The end of a phase marks a milestone in the project life cycle. A phase is composed of several work packages. Each work package describes a set of guidelines. The sequence of work packages indicates when certain activities should start. When all work packages are defined, an SAP implementation project plan can be formulated, which is also used for management review.

A work package contains a number of supporting activities, each of which describes what should be achieved (i.e., goals) and what work product (i.e., deliverables) should be obtained by a certain stage in the project. A work product comprises components that are produced by the different tasks associated with an activity. A task is a "self-contained" element; it includes information on: the use of results from other tasks (i.e., dependency), work preconditions (i.e., prerequisites), procedures, and results. The framework of ASAP is shown in Figure 2, it appears well designed and thorough, although very few studies report the ASAP methodology.

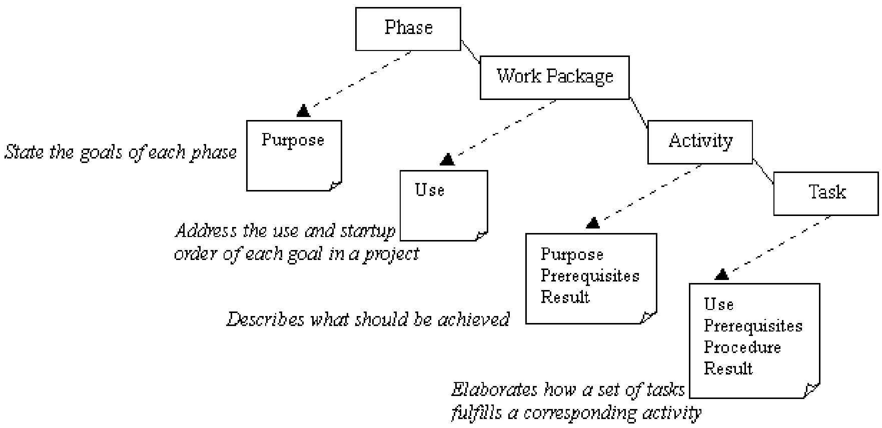


Figure 2: ASAP Structure

ASAP manages a project cycle in 5 phases with 43 work packages, 113 activities and 394 tasks (see Table 1). Unlike the CMM, which is aimed at providing guidelines on a general process of software development [3], ASAP explains not only what to do in general terms (i.e., phase and work package) but also how to do it in very specific terms (i.e., activity and task). Figure 3 illustrates an ASAP tool that organizes the entire implementation roadmap into a hierarchical tree diagram.

Phase		Work Package	Activity	Task
5	*1 Project Preparation* *2 Business Blueprint* *3 Realization* *4 Final Preparation* *5 Go Live and Support*	Phase 1: 6 Phase 2: 8 Phase 3: 19 Phase 4: 7 Phase 5: 3	Phase 1: 17 Phase 2: 21 Phase 3: 54 Phase 4: 14 Phase 5: 7	Phase 1: 64 Phase 2: 84 Phase 3: 195 Phase 4: 32 Phase 5: 19
5		43	113	394

Table 1: ASAP Composition

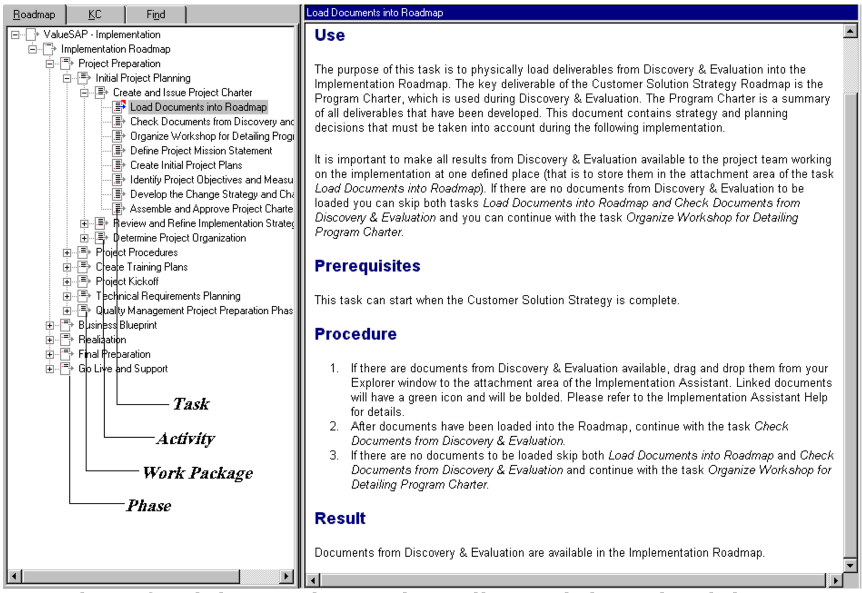


Figure 3: A number of training packages, including web-based training, are available for the ASAP certification examination

1.2 AcceleratedSAP and the CMM

In this section, we look at ASAP and CMM from an executive's perspective. Three areas that are relevant to dealing with various project issues should be examined for a model: management, organization, and techniques [6]. The techniques normally involve in the internal mechanism of a model while management and organization can be roughly understood from the environment. Thus, we attempt to compare the models at a very high level, such as the size of the organization where they can be applied, typical project length, and so on.

	AcceleratedSAP	CMM
Nature	A model for SAP implementation that involves business process re-engineering	A model for SPI
Organization Size	Medium – Large	Small – Large
Project Influence	Organization-wide	Organization-wide
Project Length	0.5 ~ 3.5 years	Ongoing impact. Continuous improvement at level 5
Sponsor	Senior management	Senior management
Applied Industry	All industries implementing ERP	Software community
Type of Process Involvement	Business	Software
Goals	Customer satisfaction	Customer satisfaction
Adoption Popularity	Worldwide	Asia and North America

Table 2: AcceleratedSAP and CMM from a managerial perspective

Table 2 provides a summary of how the two models have been adopted and used in past projects. Clearly, both models have been used in sizeable projects, which may last two or three years and have a profound impact on an organization. The major difference is that ASAP covers business process improvement, whereas CMM is used for software process improvement.

1.3 AcceleratedCMM

In the CMM, a software project is evaluated to determine whether the processes used in the project reach a certain maturity level, which is decomposed (except for level 1 CMM) into key process areas (KPA). KPAs identify a group of activities that are performed collectively to achieve a set of goals that are important for enhancing process capability. However, the CMM does not prescribe when activities associated with KPAs should occur in a project life cycle. KPAs describe the structure of the CMM, just as work packages in ASAP set out the context of that implementation methodology. However, ASAP work packages are also prescribed at specific points in a project life cycle. Based on this structural property, ASAP and the CMM can be represented in a matrix, as shown in Figure 4. As a baseline the ASAP work packages correspond to CMM key process areas, in such a way that they impose the concept of project scheduling on KPAs.

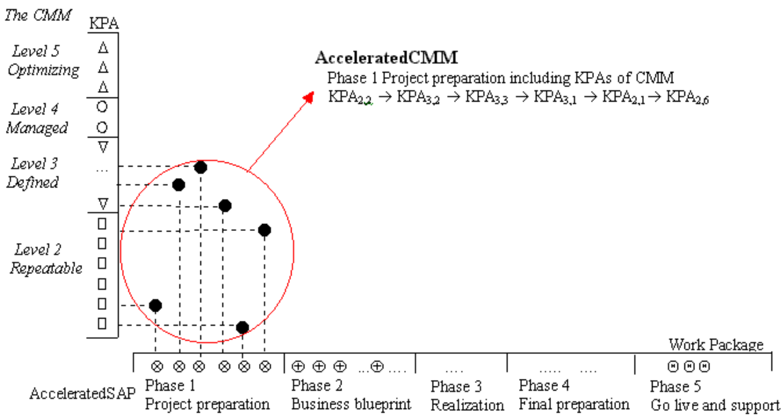


Figure 4: Matrix of AcceleratedSAP and the CMM

Figure 4 illustrates a simple one-to-one mapping only for ASAP phase 1 (project preparation). A complete one-to-one mapping is provided in the Appendix, with the exception that (1) the ASAP work package *Business Process Definition* actually maps to two KPAs and (2) Software Subcontract Management is included in AcceleratedCMM for the management between ERP vendors and their business partners and customers but it is absent from ASAP. Clearly, many work packages are concerned with business processes. Our interest lies in how principles from KPAs can be adapted to develop a well defined and disciplined process, using a methodology that builds business processes in parallel with the software implementation processes. In this case, we have added some new KPAs for certain business processes. For example, Organization Business Process Definition (OBPD) is in principle similar to Organization Process Definition (OPD). In other words, OBPD is a business interpretation of OPD, and to reflect this change, we rename OPD to OBPD.

Having described the foundations of AcceleratedCMM, we now present the details of this methodology. AcceleratedCMM has five phases, each of which contains a set of KPAs. Some fundamental KPAs may occur in several phases. For example, the discipline of software quality management should be used in each phase. As mentioned earlier, in AcceleratedCMM

KPAs are interpreted in the context of business processes, hence the names of some KPAs have been revised. The complete AcceleratedCMM methodology is presented in the Table 3.

Key Process Area	***Purpose***
Phase 1: Project Preparation	
Software Contract Management	Establish an understanding of scopes and responsibilities among all participated parties
Software (S/W) Project Planning	Establish feasible plans for overall project management
Organization Process Definition	Define standards and useable project assets that can be used in the project
Training Program	Plan training activities and execute accordingly
Organization Project Focus *	Establish organization responsibility for project activities and form steering committee to approve and announce management decision
Requirements Management	Establish an understanding between the customer and project team about the customer's requirements
S/W Quality Management	Execute necessary tests and audit project activities
The First Project Milestone	
Phase 2: Business Blueprint	
S/W Project Tracking and Oversight	Track actual progress (such as against budget) so that management can act according to deviation of project's performance
Business Process Change Management *	Define the organization's business processes on the basis of existing processes with the goal of improving quality, increasing productivity and decreasing processes cycle times
S/W Project Planning (Technical)	Establish ERP system technical plans for realization
Training	Plan training activities and execute accordingly
Organization ERP Business Process Definition *	Analyze ERP components as useable assets that can later be applied to business processes (See Phase 3: Integrated Software Management)
Integrated Business Process Product Management *	Integrate ERP business processes with manual operational business activities so that complete, well-defined business processes of an organization are developed in a business blueprint
S/W Quality Management	Execute necessary tests and audit project activities
The Second Project Milestone	
Phase 3: Realization	
S/W Project Tracking and Oversight	Track actual progress (such as against budget) so that management can act according to deviation of project's performance
Organization Risk Management *	Measure risks of business process and system changes and development contingency plans for those changes
Training	Plan training activities and execute accordingly
Integrated Software Management	Integrate business and management activities into ERP system based on the organization's blueprint
S/W Configuration	Establish and maintain the integrity of scenarios, processes, and functions
S/W Product Engineering	Consistently use business blueprint to configure ERP and develop ERP documents such as reports for the ERP production environment
S/W Quality Management	Execute necessary tests and audit project activities
The Third Project Milestone	
Phase 4: Final Preparation	
S/W Project Tracking and Oversight	Track actual progress (such as against budget) so that management can act according to deviation of project's performance
Peer Reviews	Remove defects from business processes of ERP system early and efficiently
Organization Project Focus *	Approve and announce management decision
S/W Quality Management	Execute necessary tests and audit project activities
The Fourth Project Milestone	

Phase 5: Go Live and Support	
S/W Product Engineering	Configure ERP system finally for the ERP live production environment
Quantitative Process Management	Understand capability of the organization's business processes of ERP in quantitative term
Organization Project Focus *	Approve and announce management decision

Table 3: AcceleratedCMM

AcceleratedCMM is a software model for any ERP implementation. The CMM KPAs are clearly distributed across each phase. In addition, the method addresses building business processes based on developing well-defined software processes. AcceleratedCMM shows how CMM can be applied to ERP implementation projects

1.4 Conclusion

Practitioners have been keen on exploring CMM in software product development projects, and also on comparing it with eXtreme Programming [4] and ISO9000 [6, 7]. This paper elaborates on the utility of CMM for ERP implementation projects. In addition, we have shown a logical way in which CMM is aligned with a specific ERP implementation methodology. This paper also provides an informative reference for how CMM can be used to assess and evaluate ERP software projects.

It should be noted that AcceleratedCMM is complete in a way that it covers ERP vendor management by Software Contract Management while AcceleratedSAP does not mention this crucial element contributing toward the success of an ERP project. Besides, AcceleratedSAP does not cover budget control in any planning works such as Initial Project Planning, Project Management Business Blueprint Work and the like. When Software Project Planning and Software Project Tracking and Oversight are adopted for those AcceleratedSAP works, we not only perform the same original tasks of ASAP but also consider the wider principles of KPAs. For instance, ACMM's KPAs suggest monitoring all activities incurring costs including SAP consultant's service tasks and internal resource allocations, which are excluded from AcceleratedSAP. In fact, the purpose of AcceleratedSAP is to implement SAP in a short period of time; therefore, AcceleratedSAP is not complete.

The primary contributions of this paper are to establish the synergy between CMM and ASAP, and then propose a generic model for ERP implementation, called AcceleratedCMM. AcceleratedCMM can improve the software processes of ERP implementation projects since the model was built using the key process areas of CMM. Practically, AcceleratedCMM prescribes a road map, defined by application of KPAs, towards more successful ERP projects. Some KPAs have been appropriately reinterpreted for business process re-engineering, so that AcceleratedCMM can be positioned correctly in the hierarchy of projects illustrated in Figure 1.

1.5 Appendix

	AcceleratedSAP : WORK	AcceleratedSAP: WORK's USE	The CMM: KPA		AcceleratedCMM
Project Preparation	N/A	N/A	S/W Subcontract Management	⇒	S/W Contract Management
	Initial Project Planning	start detailed planning for the project	S/W Project Planning	=	S/W Project Planning
	Project Procedures	establish the principal elements that determine how the project will use and be accomplished	Organization Process Management	=	Organization Process Management
	Create Training Plans	create a project team training plan including business processes, SAP system functionality, and change management	Training Program	=	Training Program

	Project Kickoff	announce formally the initiation of the project to the whole organization	Organization Process Focus	⇒	Organization Project Focus
	Technical Requirements Planning	identify the technical requirements	Requirements Management	=	Requirements Management
	Quality Management Project Preparation Work	provide final verification of all prior project planning and deliverables	S/W Quality Management	=	S/W Quality Management
Business Blueprint	Project Management Business Blueprint Work	establish a proper cycle of project management activities to ensure being on target	S/W Project Tracking and Oversight	=	S/W Project Tracking and Oversight
	Organizational Change Management Blueprint	address the organizational process changes and human resource factors	Process Change Management	=	Business Process Change Management
	Technical Design Planning	define further the technical planning that was started in phase one	S/W Project Planning	=	S/W Project Planning
	Establish Development System Environment	plan and install the development system for the start of implementation.	S/W Project Planning	=	S/W Project Planning
	Design Training Plans	finalize and conduct level two of the project team training	Training Program	=	Training Program
	Organizational Structure Definition	Define the enterprise organizational structure using SAP organizational units	Organization Process Definition	⇒	Organization Business Process Definition
	Business Process Definition	determine the company requirements based on SAP business processes	1) Integrated S/W Management 2) S/W Product Engineering	⇒	Integrated Business Process Product Management
	Quality Management Business Blueprint Work	provide final verification of the status of all deliverables from this phase	S/W Quality Management	=	S/W Quality Management
Realization	Project Management Realization	establish a cycle of project management activities to ensure the implementation project is on schedule. It determines project planning, controlling, and quality assurance activities.	S/W Project Tracking and Oversight	=	S/W Project Tracking and Oversight
	Organizational Change Management Realization	Ensure that Organizational Change Management (OCM) activities take place as planned and are successful in mitigating OCM risks	Process Change Management	⇒	Organization Risk Management
	Create Training Materials	finalize and conduct level three of the project team training.	Training Program	=	Training Program
	Workflow Development	automate a new business process using SAP Business Workflow	Integrated Software Management	=	Integrated Software Management
	Baseline Configuration and Confirmation	configure and confirm settings from the Baseline Scope for scenarios, processes, and functions.	S/W Configuration Management	=	S/W Configuration Management
	Develop System Test Plans	Define technical and performance test plans	S/W Quality Management	=	S/W Quality Management
	Establish Quality Assurance Environment	set up the quality assurance system for integration testing	S/W Quality Management		S/W Quality Management
	Establish Production Environment	Set up the production system for productive	S/W Product Engineering	=	S/W Product Engineering
	Final Configuration and Confirmation	finalize the scope and configure your SAP System for production	S/W Configuration Management	=	S/W Configuration Management
	Prepare and Coordinate ABAP Development	prepare and coordinate ABAP development efforts	S/W Product Engineering	=	S/W Product Engineering
	Develop Conversion Programs	create programs and manual procedures for transferring data from the company legacy systems to the SAP System	S/W Product Engineering	=	S/W Product Engineering
	Develop Application Interface Programs	enhance standard SAP system functions	S/W Product Engineering	=	S/W Product Engineering
	Develop Enhancements	link external application systems with the SAP system	S/W Product Engineering	=	S/W Product Engineering
	Create Reports	identify the internal company reporting needs	S/W Product Engineering	=	S/W Product Engineering
	Create Forms	implement customer-specific forms as required in Business Blue Print	S/W Product Engineering	=	S/W Product Engineering
	Establish User Role and Authorization Concept	set up enterprise-wide user authorizations for system access	S/W Product Engineering	=	S/W Product Engineering
	Establish Data Archiving	to establish data archiving procedures to remove data not required in the R/3 System, but which must be accessible from the R/3 System	S/W Product Engineering	=	S/W Product Engineering
	Final Integration Test	Plan and execute the final integration test	S/W Quality Management	=	S/W Quality Management
	Quality Management Realization Work	provide final verification of the status of all deliverables from this phase	S/W Quality Management	=	S/W Quality Management
Final Preparation	Project Management Final Preparation Work	perform the established cycle of project management activities to keep the implementation project on target. This determines all project planning, controlling and updating activities, and identifies changes between business processes and the organizational structure.	S/W Project Tracking and Oversight	=	S/W Project Tracking and Oversight
	Deliver End User Training	ensure all end users adequately trained before go- live	Training Program	=	Training Program
	System Management	perform the relevant technical activities to prepare for production operation	S/W Quality Management	=	S/W Quality Management

	Conduct System Tests	conduct technical and performance tests to verify that the production environment is ready	S/W Quality Management	=	S/W Quality Management
	Detailed Project Planning	identify issues that impact the initial plan for production support and cutover prepared in Phase Three	Peer Reviews	=	Peer Reviews
	Cutover	obtain final approval from the steering committee for the system to go live	Organization Process Focus	⇒	Organization Project Focus
	Quality Management Final Preparation Work	provide final verification of the status of all deliverables from this phase	S/W Quality Management	=	S/W Quality Management
Go Live & Support	Production Support	move from a pre-production environment to live production operation	S/W Product Engineering	=	S/W Product Engineering
	Ongoing Key Performance Indicator Management	Measure business process performance in quantity	Quantitative Process Management	=	Quantitative Process Management
	Project End	close the project officially	Organization Process Focus	⇒	Organization Project Focus

Abbreviation: S/W is short for Software

2 Literature

[1] MacDonald G, Business Transformation with SAP R/3, The Institution of Electrical Engineers, published by the IEE, Savoy Place, London WC2R OBL, UK, 1996

[2] Hiquet BD, SAP R/3 Implementation Guide: A Manager's Guide to Understanding SAP, Macmillan Technical Publishing, 1998

[3] Paulk MC et al., The Capability Maturity Model: Guidelines for Improving the Software Process, Carnegie Mellon University, SEI, 1995

[4] Paulk MC, Extreme Programming from a CMM Perspective, IEEE Software, Nov/Dec 2001, pp19-26

[5] Johnson DL and Brodman JG, Applying CMM Project Planning Practices to Diverse Environments, IEEE Software

[6] Wang and King, Software Engineering Process: Principles and Application, CRC

[7] Paulk MC, How ISO 9001 Compares with the CMM, IEEE Software, Jan 1995, pp74-83

3 Author CVs

Kim Man Lui

Kim Man Lui received a B. Eng in 1990 from Tamkang University, Taiwan. He was awarded a bursary to study for his MSc in computer science at the University of the Witwatersrand, South Africa. There he received the MSc in 1994. He was invited to be the chairperson for the hardware session of the Asian Conference on Computer Vision 1993 held in Japan.

Mr. Lui has worked in a number of IT positions in the commercial sector from system engineer, analyst programmer, system analyst, project leader and IT manager. He was a Project Manager of an SAP project. Mr. Lui was a certified Sybase Database Administrator in 1996 and also a certified Oracle Database Administrator in 1999. While studying towards his PhD degree at the Hong Kong Polytechnic University, Mr. Lui is currently working as the IT manager for Carlsberg responsible for Pan China IT development and support. Mr. Lui published several international journal and conference papers on pattern recognition, intelligent control, qualitative reasoning and knowledge discovery. He was a co-author of a CMM book published in China. His current interests include software engineering, software process improvement, knowledge management and global software team.

Keith C. C. Chan

Dr. Keith Chan has a B.Math. (Hons.) degree in Computer Science and Statistics and a M.A.Sc. and Ph.D. degree in Systems Design Engineering from the University of Waterloo, Ontario, Canada. He has a number of years of experience in software development and management. Before joining The Hong Kong Polytechnic University, he was with the IBM Canada Laboratory where he was involved in every phase of software development from design to coding, to testing and maintenance. He was a member of the software quality assurance team during the time the laboratory first applied for ISO 9001 certification.

Dr. Chan has been working on different areas in software engineering for many years. He has successfully led the effort to develop a self-assessment tool for ISO 9001 certification and a workflow automation tool to facilitate the adoption of ISO 9001 compliant practices. Dr. Chan has conducted courses and seminars in software quality management, and software project management. He has published in the areas of software processes, software quality management, multi-site software development and cooperative information systems. He has been a consultant to government, semi-government and commercial organizations in Hong Kong, Canada, Singapore and Malaysia. Dr. Chan is a reviewer for IEEE standards in software engineering and is a founding member of the IEEE Standards Association.

Dr. Chan joined the Department of Electrical and Computer Engineering at Ryerson Polytechnic University, Ontario, Canada as an Associate Professor in 1993. Since 1994, he has been with The Hong Kong Polytechnic University where he is currently Associate Professor and Acting Head of the Department of Computing. Since August, 1999, he has been an Adjunct Professor of the Institute of Software, the Chinese Academy of Sciences, Beijing, China. Dr. Chan is consultant to different government and commercial organizations in Hong Kong, China, Singapore, Malaysia and Canada on software quality and software processes. Dr. Chan is a trained ISO 9001 and TickIT Assessor. His research interests are on software engineering and data mining.

Best Practices in E-Commerce: Strategies, Skills, and Processes

Eva FEUER, Sztaki, Hungary, Richard MESSNARZ, ISCN, Ireland, Gabor Nadasi, ISCN, Austria, Nacho SANCHEZ, Gonzalo VELASCO, FGUVA Spain, Eugene O'LEARY, Brian FOLEY, Nuala O'SHEA, Tecnet, Ireland, Mark LYSON, Bruno WÖRAN, Danube, Austria

Abstract

This paper describes the goals, approaches, and first analysis results of an EU Leonardo project developing best practices in e-commerce. The project develops a first skill card for a new profession called e-commerce engineer, and a set of hands-on modular courses which can be applied by both, universities and industry. The initiative involves different market players, such as educational institutions, training organisations, and networks of researchers and users.

The paper will outline the first version of a skill set, the planned structure of skill and course units, and how a learning by doing strategy will offer SMEs in Europe to collaborate with the project consortium in trials.

Keywords

Skill Cards, Skills Standards, E-Commerce, E-Business, Best Practice

1 The E-Commerce Jobs Project

1.1 Background and Objectives

In the Leonardo da Vinci project "Best practice in E-commerce" we study and analyse the required skills and capabilities of software engineers to plan, design, implement and exploit e-commerce applications. The skills required by e-commerce engineers will be described based on skill card definitions and the guide-lines published by the DTI (Department of Trade and Industry) National Vocational Qualification (NVQ) standards [2].

This will allow us to insert the job « e-commerce engineer » in European APL (Accreditation of Prior Learning) accreditation systems [1],[6].

The information society is changing rapidly and nearly all firms (even non-software and non-computer firms) are going to be represented on the Internet and establish commercial services over the net. The customers can get acquainted with a wider choice on the Internet, they can reach the suppliers all over the world. The European firms should enter the electronic market as soon as possible to stay competitive. However, this leads to a new job role and profession required in SMEs, which we could call e-commerce and multimedia engineer.

The project " Best practice in e-commerce " develops

- a skill set for this new profession [3], [4],
- a set of training materials for each skill unit within the skill set for an e-commerce and multimedia engineer and makes those materials available through a server in the CEEC countries and in the European member states.
- An e-commerce demo site to learn from practical experience.

1.2 Why this Work is Needed

There is a continuous social dialog between educational institutions (and commercial training organisations) and the labour market. Education and training must satisfy the needs of the labour market and as the labour market moves towards an electronic information society, e-Commerce is a key educational field for the future.

However, currently the pitfalls are that many organisations go into the field without proper understanding of

- Information design issues
- Security issues
- Commercial issues
- And impact on people and market behaviour.

These aspects are taken into account in the course development and have been considered in the demand analysis which has been performed at the beginning of this year. The results of this demand analysis are discussed in this paper as well.

1.3 Underlying European Educational Strategies and Skill Cards

1.3.1 Strategies Underlying Skill Cards

Imagine that in the future Europeans will have a skill card like a card with a chip which stores your skill profile to fulfil specific professions, job roles, and tasks. Its working like an ID card. This future scenario requires -

- A standard way to describe a skill set for a profession, job, or specific task.
- A standard procedure to assess the skill and to calculate and display skill profiles.

Such a common set of skill sets in Europe is needed due to the free mobility of workers, and e.g. software engineering companies in Germany employ software engineers from Ireland, etc. European countries such as UK, The Netherlands, and France have already well established open universities which support APL (Accreditation of Prior Learning). In APL the skills of students are assessed, already gained skills are recognised, and only for the skill gaps a learning plan is established. The skill assessment bases on defined skill units and a skill profile displaying how much of the skill units are covered.

In a previous project CREDIT (Accreditation of Skills via the Internet) [1] in which some of the project partners were involved such an Internet based skills assessment system has been built. Therefore another possible scenario of the future is that representative educational bodies per country in Europe maintain skill profiles in databases which can be accessed via defined ID codes for people.

1.3.2 Skill Card Structure

For developing the skill set of the e-commerce engineer we base on the skills definition proposed by the DTI (Department of Trade and Industry) in the UK for the NVQ (National Vocational Qualification) standards. These models have been re-used and slightly modified by other countries when they started employing skill cards, and so we also base our work on these models [1], [2].

A skills definition contains the following items (see Figure 1):

Context (UK standards): A category of ranges; it represents some terminology used in a performance criterion that consists of different context, conditions or circumstances. A participant must be able to prove competence in all the different circumstances covered by the context.

Domain: An occupational category, e.g. childcare, first level management or software engineering.

Element (UK standards): Description of one distinct aspect of the work performed by a worker, either a specific task that the worker has to do or a specific way of working. Each element consists of a number of performance criteria.

Evidence: Proof of competence.

Knowledge and understanding category (UK standards): A category of knowledge and understanding descriptions.

Knowledge and understanding description (UK standards): A description of certain knowledge and understanding. To be judged competent in a unit a participant must prove to have and to be able to apply all the knowledge and understanding attached to it.

NVQ (UK based): The National Vocational Qualification standard of England, Wales and N. Ireland.

Performance criterion (UK standards): Description of the minimum level of performance a participant must demonstrate in order to be assessed as competent. A performance criterion may have relevant contexts.

Principle (UK standards): A statement of good intentions; it underpins all competent domain practice.

Range (UK standards): Description of a specific circumstance and condition of a performance criterion statement.

Qualification: The requirements for an individual to enter, or progress within a certain occupation.

Job Role: A certain profession that covers part of the domain knowledge. E.g. domain = E-Commerce, job role = E-Commerce Engineer.

Unit (UK standards): A list of certain activities that have to be carried out in the workplace. It is the top-level skill in the UK qualification standard hierarchy and each unit consists of a number of elements.

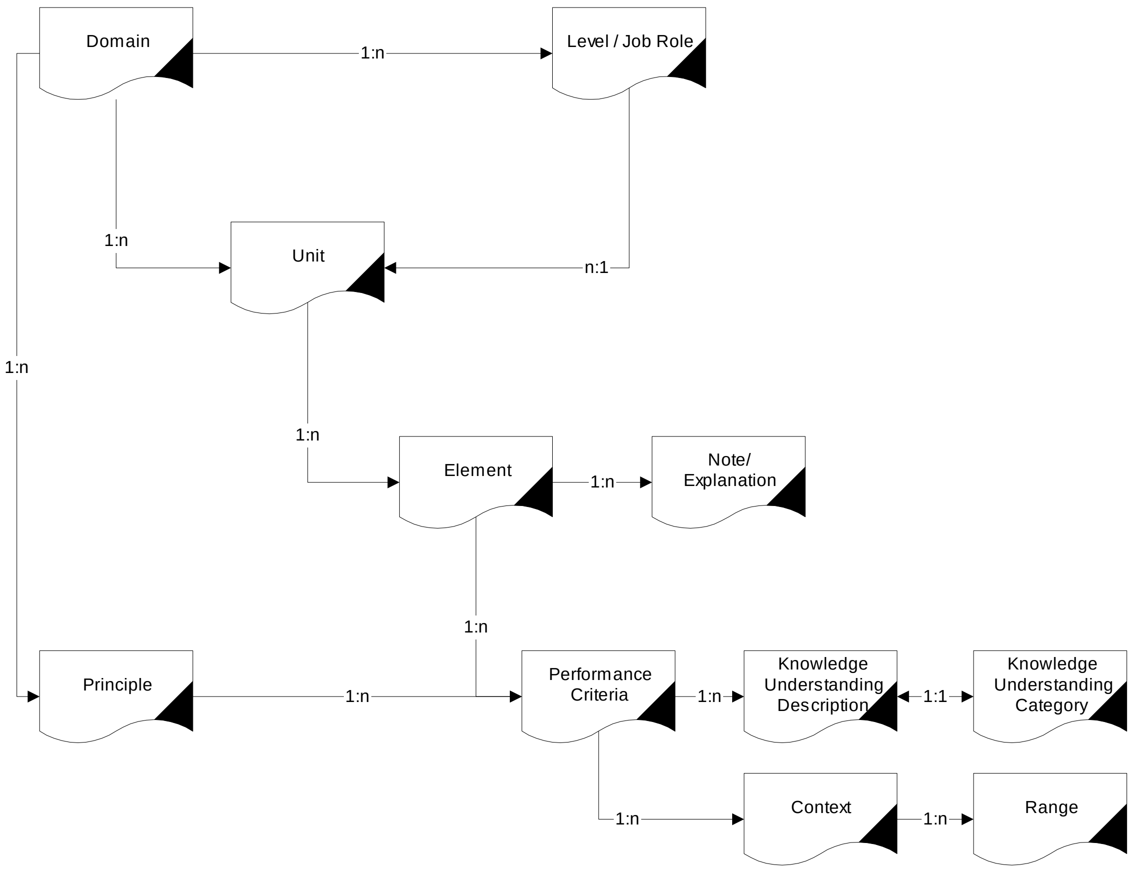


Figure 1 : The Skill Definition Model (1:n = one to many relationship)

2 The Demand Analysis and Results

A demand analysis has been performed by involving chambers of commerce and educational institutions from Austria, Hungary, Ireland, and Spain [5], [8], [9]. The outcomes can be summarised as follows:

In the original proposal we had promised to develop a skill set for an e-commerce and multimedia engineer.

However, the feedback especially from chambers of commerce did not focus just on technical issues but very much on business strategies and business related issues. If the business managers do not get the right understanding, set the right goals, and give financial support to such engineers, e-commerce will not happen as well. And that is a good point as well. So we had to gradually shift away from just a technical viewpoint on an e-commerce engineer to more a multidisciplinary viewpoint addressing business as well as technical topics.

Originally we promised two courses, and two skill sets, such as a beginner and an advanced level of an e-commerce engineer (see previous level definitions in the skills definition model). Based on the demand analysis and feedback we got we had to shift this focus to rather three job roles:

- The e-business strategy manager (high level strategy)
- The e-business manager (business implementations level)
- The e-commerce engineer (technical realisation level)

The introduction strategy of electronic commerce in SMEs impacts on their business and technological processes. Within the enterprise the evolution towards e-commerce is initiated on top management level, is designed and implemented both on management and technical levels. The actors need to gain knowledge about how to do e-business and be aware of the skills necessary to produce sufficient results.

Figure 2 provides an overview of this change in mind, leading to a combination of business and technical topics.

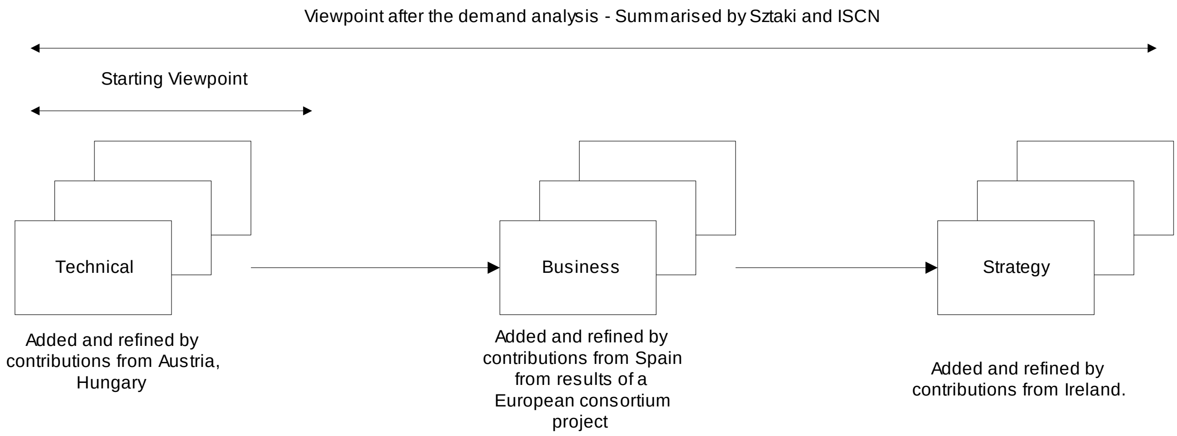


Figure 2 : Change of mind to a combined view of technical and business skills

An e-commerce introduction project [9] must be developed and implemented from a business point of view and not from a technological point of view. Although technology can play a very important role, it is a tool for obtaining results but never an aim itself. Although technology is very important in an e-commerce project, it should bare in mind the company culture and the staff working there as they are the “engines” of the organisation.

Reasons for failure of e-commerce projects were commented by chambers of commerce as follows:

- Because they ignore, totally or in part, the important opportunities offered by ICT. Therefore the management of the enterprises do not support these projects.
- Because in many times e-commerce is confused with having a web page.
- The opportunities offered by the ICT are underestimated.
- There is a lack of professionals that have a general overview of the e-commerce project, from both a management point of view and also a technological point of view.
- Organisations’ processes are not re-oriented when they go digital, in order to have a correct adaptation of the businesses.
- Internet communication actions are not correctly developed.
- It does not exist a methodology for the development of the e-commerce project.
- It is not considered as a priority and SMEs are quite reluctant to changes.

For instance, the SME -eBusiness Survey 2001 undertaken by the Chambers of Commerce of Ireland demonstrated the level of penetration that eBusiness has achieved in Irish businesses. (http://www.chambersireland.ie): 826 people were interviewed in the survey, all from businesses with 250 or fewer employees. The respondents were senior staff in the companies.

The survey revealed that while there was a high level of Internet access among Irish SMEs (over 81%), use of the Internet tended to be in large part limited to marketing and sourcing material (82%) and emailing customers (73%) and there was a low level of transference of general business processes to the Internet - with 45% of companies having a website. The survey also revealed that Irish business felt that not enough was being done to inform them about e-Business or in providing practical help for companies getting involved in e-Business.

The perceived main obstacles to e-Business were lack of technical skills, security issues, and the cost of hardware/computer equipment. The cost of specialist technical skills in computer/e-Business was seen as the greatest obstacle, with 40% of respondents saying that this was a major problem. While a third of the businesses surveyed perceived lack of security as a major problem. In the Enterprise Ireland survey of its client base similar obstacles were highlighted - organizational culture, cost and training existing staff were seen as the greatest barriers.

3 The Modular Course Architecture

The demand analysis also identified first skill units and skill elements (re units and elements compare with the skill definitions in Figure 1) which shall be covered by each of the three job roles:

1. E-Commerce Engineer Technical Course
2. E-Business Manager Course
3. E-Business Strategy Course

The identified units and their elements for the two of the three job roles are described in Figures 3, and 4.

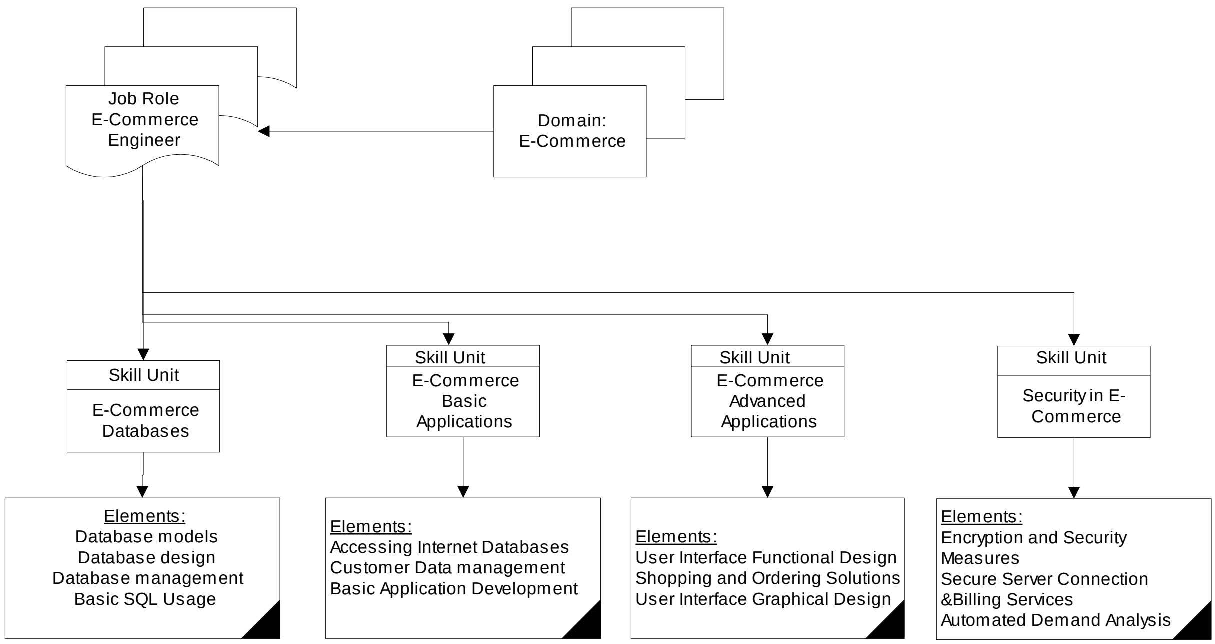


Figure 3 : Skills Definition of the Job Role E-Commerce Engineer

As you can see in Figures 3 and 4 -

- The technical course addresses already experienced Internet developers who want to extend their skills to learn about practical e-commerce solutions. Ready-to-use existing software solutions and scripts will be used in the training sessions which can be adapted and extended by the students to their own company specific solutions.
- The business manager course addresses section and division managers of large firms or SME owners who want to learn about practical solutions, consequences, impacts, and potentials of e-commerce solutions. This will involve Business to Business, Business to Customers, Customers to Customers, etc. applications.
- The e-business strategy manager course will use two of the modules of the e-business manager course (to get an overview about potential applications) and focuses towards e-strategy analysis tasks, such as business planning, marketing planning, financial impact analysis, technological impact analysis, etc.

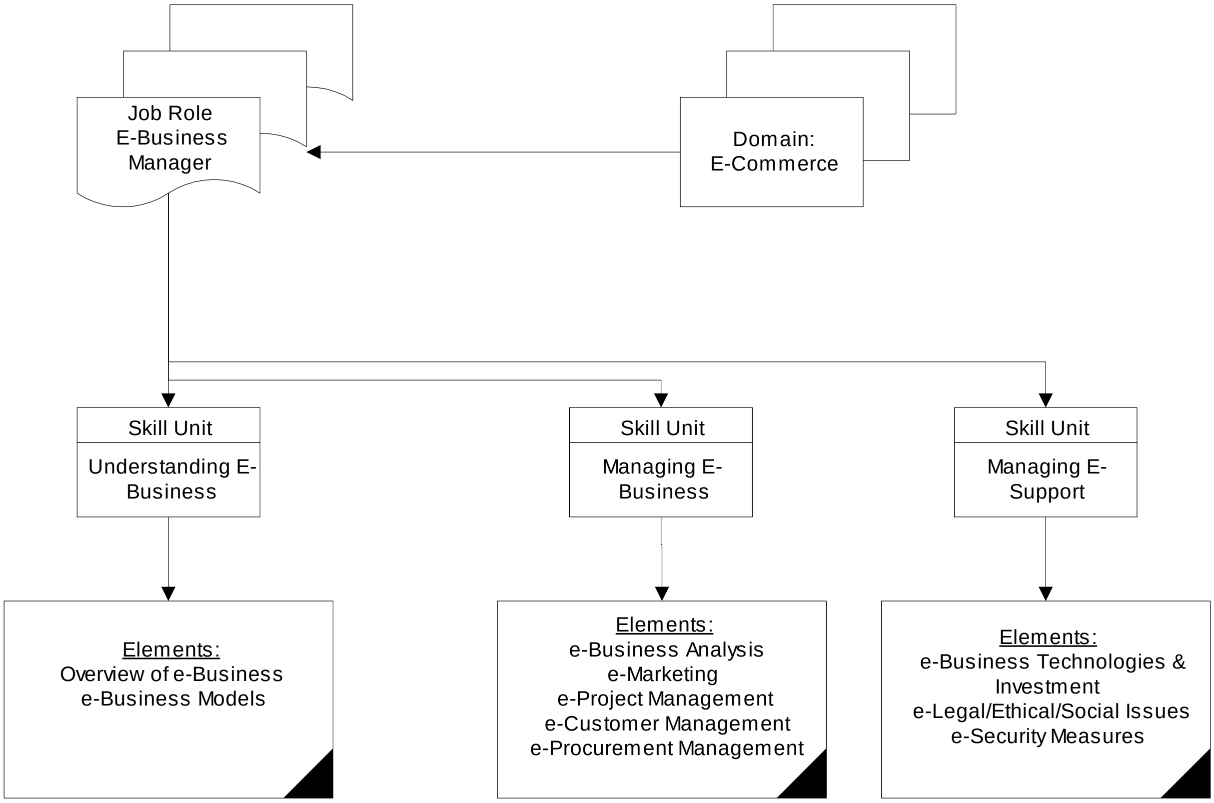


Figure 4 : Skills Definition of the Job Role E-Business Manager

The skill set definition also defines the course units and their course elements. In the development of a course programme each skill unit will become a course unit, and each skills element will become an element of the course unit. Thus a picture as illustrated in Figure 3 already describes the modular course architecture.

Please note that the results described in Figures 3 and 4 are preliminary and will be further refined in 2002.

4 Outlook and Potential Users

The results from the surveys in Ireland suggest that there would be a strong demand from Irish SMEs for a course that would inform them of the relevance of e-business to their organisation and to aid them in planning adequately for implementing an e-business strategy. The course proposed in the E-Commerce Jobs project aims to address this need.

It is important to highlight also that the course required by Irish SMEs is not a purely technical course aimed at IT staff but rather a course with a mix of business and technical aspects. The SME e-business survey from 2000 highlighted the fact that SMEs did not view e-business as being mainly the concern of their IT staff - 70% of respondents disagreed strongly with the premise that "e-business in our company is mainly for the IT people". In a similar light 45% of the Enterprise Ireland respondents claimed that the CEO was responsible for driving the company's e-business strategy.

Our partner TecNet and its member Institutes of Technology are ideally placed to deliver these courses. Geographically the institutes are located in thirteen locations across Ireland. Lecturers in the institutes are skilled in the delivery of such courses and the intention will be to incorporate the courses designed in this project for use as modules in the Institutes' diploma and degree programmes.

In Austria since a few years, there is an increasing tendency of establishing “Fachhochschulen” (polytechnics) in Austria, which do represent, in a certain way, a competition to the existing universities. The strategy of the Austrian project partner, DANUBE, is to collaborate in a closer way with these “Fachhochschulen” in order to gain efficient and quick results. To organize the e-com course within classical universities means a large and complicated organisational process which does not correspond to the fast changing needs in business and technology. The course should therefore be offered as a postgraduate course or a semester course within a Fachhochschule. The process of implementing the course would be much faster. Even so, close relations of the Fachhochschulen with the business world is an important fact.

Besides the polytechnic involvement, the course could be organized in collaboration with private or public training organisations or employment related organisations as for example:

- The Austrian Employment Service (AMS),
- Training organisations of the chamber of labour (BFI). This Institution is involved in similar projects and course development. In collaboration with BFI and DANUBE, an e-commerce platform has been created in order to strengthen the knowledge of EU-based e-commerce training courses.
- Training Institution of the Chamber of commerce (WIFI): This organisation might have a strong interest in running the e-com jobs training course or parts of it either as short courses or as a semester course in it’s “Fachhochschule”.
- The Austrian Computer Society (OCG) is a very competent potential partner. Experienced in computer related course development on a national level (as for example the implementation of the ECDL – European Computer Driving licence- in Austria) this organisation could be the platform for running the E-Commerce and Job Role Qualifications course.

In Spain concerning SMEs, after carrying out a first survey with the Chamber of Commerce of Valladolid, E-commerce providers solutions, ICT engineers and e-business organisations, we can say that the organisations have many difficulties to improve their results using the Internet due to the following reasons:

- Total or partial ignorance of the many existing opportunities that offer the new technologies and particularly the Internet, which implies little support by the management.
- The many existing possibilities that offer the Internet are underestimated.
- There is a lack of staff with a global vision of the project, from both a business point of view and a technical one.
- The different processes of the firm are not reorganised in order to have a proper adaptation of the business.
- Incorrect development of the communication actions in the Internet.
- There is a lack of methodology in the implementation of the project.

The most important one seems to be the lack of training. After analysing the current situation, we considered the following necessary actions for Spain.

- Training in the Internet business area: it is not so necessary training focused on technical aspects but more on how to use the Internet in order to do businesses.

- Think about the Internet as a way to improve the businesses results and that must be a responsibility of the management of the firm.
- Need of internal and external experts in this area.
- Think about the businesses in the Internet in a strategic way.
- To be ready to reconsider the business model and the processes of the firm.

In short, the E-commerce Job project appears as a solution, not only for the existing and potential businesses but also for the future workers of those businesses. The University is supposed to be the most important source of future business managers in Spain and therefore should adapt the training offered to them to the market trends and not wait until is too late. For this reason FGUVA considers the implementation of the pilot course as a great opportunity to strength the cooperation between the university and the business world and prepare Spanish students and graduates for the e-business world.

ISCN as the co-ordinator of the EuroSPI initiative addresses some 30000 IT managers across Europe and through a major annual conference can promote this knowledge and the courses to leading European industries. In-house courses will be held.

Finally the co-ordinator of the project MTA SZTAKI, the Computer and Automation Research Institute of the Hungarian Academy of Sciences, plays a lead role in preparing the Hungarian industry for EU integration. The results of this project will be entered into these services and will further European integration.

5 Literature

[1] CREDIT Project, Accreditation Model Definition, MM 1032 Project CREDIT, Version 2.0, University of Amsterdam, 15.2.99

[2] DTI - Department of Trade and Industry UK, British Standards for Occupational Qualification, National Vocational Qualification Standards and Levels

[3] Financial Times Management (1997). Manage people, Unit C15 Project: Dealing with poor performance. Manchester: The Open College.

[4] Financial Times Management (1997). Manage people, Unit C12 Project: Working with your team for results. Manchester: The Open College

[5] Information Society Commission, Benchmarking Ireland in the Information Society, Irish Government, 1997

[6] Klarus, R., & Blokhuis, F. (1997). Accreditation of prior learning in the Netherlands.

's Hertogenbosch: CINOP.

[7] 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

[8] M. Biro, R. Messnarz, Key Success Factors for Business Based Improvement, in Software Quality Professional, Volume Two, Issue two, American Society for Quality, March 2000

[9] Messnarz R., Biro M., Velasco G., O'Suilleabhain G., The Victory Project - A Virtual Enterprise for SPI, in Proceedings of the EuroSPI 2000 Conference, Copenhagen Business School, 7-9 November 2000

6 Author CVs

Eva Feuer

Éva FEUER is Head of Quality Management, senior research associate at the Computer and Automation Research Institute. She has more than 20 years of IT and project management experience.
M.Sc. in mathematics and computer science (Jozsef Attila University, Hungary). She is certified ISO 9000 TÜV CERT auditor and BOOTSTRAP software process assessor. She managed Hungarian participation in European multinational projects and organisations (European Software Institute, Bootstrap Institute, Idealist-East Leonardo projects, Pass ESSI Pie project, Vasie ESSI project). She is head of the Information Society Technologies Liaison Office for the European Union's 5th Framework Programme in Hungary. Her main R&D fields are: information systems, software development, quality management, software process assessment, process improvement and software product quality audit. She has publications in international scientific journals and conference proceedings in the field of software quality management.

Dr Richard Messnarz

Dr Richard Messnarz made his MSc at the University of Technology in Graz in Austria. In 1995 he received a PhD from the same university for the work on a QUES - "A Quantitative Quality Evaluation System".
He has participated in a number of European research and industry projects since 1990. He was also the project leader of the Comett expert exchange project ISCN in 1993 which led to the foundation of ISCN itself in 1994.
He is the Executive Director of ISCN. He was the technical co-ordinator in previous process improvement initiatives such as PICO (Process Improvement Combined apprOach), Victory (Virtual Enterprise for Process Improvement), and the co-ordinator of a software engineering group within the pilot project for building a prototype of a virtual university. He is the editor of a book published in 1999 from IEEE "Better Software Practice for Business Benefit - Principles and Experience". He is currently the technical director of a large e-working initiative TEAMWORK, and manages in the initiative MediaISF the technical development for an e-working solution for news agencies collaborating on Eastern European enhancement topics.
He is a chairman and main organiser of the EuroSPI conference series (the former ISCN series).
He is a SPICE and BOOTSTRAP lead assessor.

Gonzalo Velasco del Barrio

Mr Velasco made a master degree in International Trade at Valladolid Chamber of Commerce (Spain) and graduated 1997. He also made a honours degree for bachelor of arts in business administration at the university of Coventry (United Kingdom).
Since 1997 he worked in Spain on the development and management of EU, national and regional projects at FGUVA (General foundation to further the integration of Spanish regions into Europe) training and innovation departments.
He is currently the person in charge of the development and the management of two networks: Innovation Relay Centre Gallaecia (IRC) and the Envirdis Network dealing about innovation issues both of them are framed in the Innovation programme. He has experience in Leonardo Projects and other transnational projects.

Eugene O'Leary

TecNet is a strategic network of thirteen Irish Institutes of Technology (technical university equivalents). The network was established by the Council of Directors of the Institutes of Technology and is jointly funded by Enterprise Ireland (Ireland's enterprise development agency).
Eugene O'Leary is chief executive of TecNet. He has worked in third level education as well as in SME's and has expertise in information technology and its application to business development. He is an experienced project manager having worked with over twenty European Projects including, ADAPT, Integra, Leonardo, Socrates, Euroform and others. He is currently co-ordinating an IST FP5 project. He also sits on a number of Consultative Committees in the Irish Department of Education.

Bruno Wöran

Mr Wöran made a master degree in economics and graduated from the University of Economics in Vienna in 1991. He also made a master of business administration degree at the University of Nebraska, Oklahoma, and graduated there in 1993. In 1993 he then joined Austrian strategic EU institutions to further the Austrian integration into Europe. Since 1996 he is the managing director of Danube. DANUBE is the regional service and information centre for the European vocational training Programme LEONARDO DA VINCI and the 5th Framework Programme for research and technological development. At the same time DANUBE is consortium partner in the INNOVATION Relay Centre Austria, a network of technology transfer and validation centres throughout Europe. DANUBE is the co-ordinator of several pool contracts for university-enterprise transnational mobilities of students and graduates for 4 universities and 8 polytechnics and partner institutions in former Eastern Europe.

The eServices Capability Model (e^{scm}) For Excellence in IT-enabled Outsourcing Services

Rajesh Gupta
Bennet Suresh Kumar
Satyam Computer Services Limited, India

Abstract

Outsourcing has long been available - since the mid-twentieth century, to organizations as a strategy for reducing cost, improving quality or focusing on core competencies. What is new in outsourcing, however, is the increasing impact of Internet revolution on it and emergence of a new breed of outsourcing services to take advantage of increasing bandwidth in telecom and data network. The new types of services that are being increasingly outsourced these days are called IT enabled outsourcing services (IT-EOS). However, there are several challenges for a service provider of IT-EOS, like managing and meeting client expectations is a major challenge in IT-EOS, where the nature of the services themselves, and the rapid changes in technology and tools, introduce an additional level of complexity. The paper begins with the emerging trends in IT enabled outsourcing services. Then, it delves into the critical issues for successful delivery of IT-EOS pertaining to the various phases in a typical IT-EOS process.

Carnegie Mellon University – which has produced several successful capability maturity models in past, has come up with a new model (November 2001) to address these critical issues. The model is called the eServices Capability Model (e^{scm}). The next part of the paper presents the framework of the model and goes on to provide a practical way to implement the model. The paper explains the approach for implementing the model by taking up a case of an organization that provides data center services and located in Asia Pacific region: what made them to chose the model, what are they doing to adopt the model and expected benefits out of its implementation.

Keywords

e Services Capability Model, ESCM, IT enabled services, IT enabled outsourcing services

1 Emergence of IT-enabled outsourcing services

Outsourcing has long been used, since the mid-twentieth century, by organizations as a strategy for reducing cost, improving quality or focusing on core competencies. Initially outsourcing was used primarily for the manufacturing of industrial components, as well as for some non-critical services such as facilities management. Outsourcing in information technology (IT) commenced in the 1970s when organizations started to outsource parts of their IT operations to service providers, to dramatically reduce costs. Thereafter, the 1980's and 1990's witnessed the conclusion of some landmark outsourcing agreements, which involved the shifting of entire IT operations to external service providers.

The reasons for outsourcing have been both strategic as well as tactical. Strategic factors, that drive an organization to outsource, are those that have a direct and long term impact on a company's overall marketplace positioning. They include:

- Corporate focus on core business functions;
- Increasing regulatory and compliance requirements due to organizations operating across geographical boundaries;
- Utilizing world-class capabilities;
- Mitigating risks through sharing;
- Increasing cost pressures.

Tactical reasons are for short-term gains and include reasons like infusing cash into business operations and reducing and controlling operating costs, reducing the risk of entering new markets, leapfrogging competitors.

The late eighties and nineties witnessed the emergence of outsourcing of IT-intensive business processes. These business processes included information services, human resource services, financial and payment services, and other services that used IT heavily for their design, deployment and/or delivery. Later, other services were also included in this category like engineering services, geographical information systems, multi-media content development, and transcription services. The outsourcing of such services is called IT Enabled Outsourcing Services (IT-EOS).

IT-enabled outsourcing services include outsourced IT-intensive business processes, projects and tasks that use Information Technology as an enabler for designing services, coordinating service deployment, and delivering services [3]. Some important IT-enabled outsourcing service segments are:

- Data capture, integration and analysis services (comprising data processing, data warehousing and data mining);
- Engineering services (including engineering design, architectural design, product design, geographical information systems);
- Human resource services (including back office operations and strategic HR function outsourcing);
- Information Systems outsourcing (including hardware/software maintenance, application development, network management);
- Multimedia and animation services (including content development, content management, websites, animation films);
- Remote customer interaction services (comprising call center outsourcing, fulfillment center outsourcing, e-support centers); and
- Transcription services (including medical and legal transcription services)

There are primarily two reasons, which fuelled the growth of IT-EOS. These are:

- The rapid development in technology has now made it possible to transmit data securely and rapidly using telecom or data networks. In other words, it is now possible to provide service to a client or an end-user without involving any (or only minimal) physical delivery of work products from a service provider.
- The cost of usage of the Internet and global telecommunication infrastructure has been considerably reduced.

Since geographical distances do not matter, organizations have flexibility to choose their service providers from any part of the world. Organizations are now outsourcing their business processes, projects or tasks and thus establishing 'extended' business partnerships with service providers, and are transforming themselves into 'extended enterprises'.

The economic impact of this technology-enabled revolution is estimated to be tremendous with estimates of the market for such services ranging from US$ 500 billion to 700 billion by 2004 [4,5]. According to McKinsey & Co.[6] the volume of IT-EOS is expected to grow fifteen-fold by 2008, and within the next 20 years it is expected to cross over a trillion dollars Further, 90% of companies in a recent survey [7] stated that they have plans to increase outsourcing spending this year. Spending growth for outsourcing arrangements are expected to grow considerably over next several years. The growth rates have been projected [7] as: 15% CAGR in US and 50% CAGR in Asia,

The market is growing. However, examples of failures in outsourcing relationships abound. Failures typically happen throughout the outsourcing process, i.e. during requirement specification, contract execution or service transition.

- Through 2005, the rate of technology and industry change will cause 40% of outsourcing service providers to fail [8]
- 25% of organizations that have outsourced report relationship failures within two year time period [4]
- More than half (53%) of all outsourcing customers report having re-negotiated a contract, and in nearly one-quarter of these re-negotiations the original service provider lost the account. The primary reason for failures is lack of defined service objectives and inadequate measurement of service related activities used to quantify and report progress [5]

2 Critical issues in IT-EOS

There are numerous issues that are critical for the service providers to meet its client's requirements. These can be grouped into five groups, as under:

i. Management: Issues related with setting direction, providing resources, taking appropriate corrective and preventive actions to meet client needs.

ii. Operations: Issues related with formulation of contracts with clients, designing services to meet client requirements and delivering the services to clients and end-users; meeting and managing SLAs.

iii. People: Issues related with personnel acquisition, training and development

iv. Technology: Issues related to management of technology and that includes technology acquisition, deployment and integration with that of clients

v. 'Knowledge: Issues related with capturing learning and improving the quality of service

The five different groups, as above, relate to the five organisational elements of the e^{scm} framework (described in section 3.0).

2.1 Critical issues for Management

These are:

i. Defining and communicating objectives at all levels in the organisation

ii. Creating a client focus and setting performance expectations

iii. Managing client relationships

iv. Providing adequate resources for performing service-related and support activities

v. Proactively managing risks

vi. Defining profitability expectations

vii. Defining clearly roles, responsibilities and authorities for personnel

viii. Verifying process adherence and taking appropriate corrective and preventive actions

ix. Identifying areas of cultural difference with clients, both in terms of organisational and societal culture, and taking suitable actions to achieve a cultural fit with clients

2.2 Critical issues for Operations

The critical issues from operations perspective can be explained with reference to the typical phases in an IT-EOS relationship i.e. pre-contract, contract execution and post-contract phases.

2.2.1 Critical issues during the pre-contract phase are:

i. Understanding explicit and implicit client requirements

ii. Ensuring that the service provider can fulfil these requirements, prior to making commitments to clients

iii. Understanding the roles and responsibilities of the client and the service provider

iv. Providing accurate and comprehensive responses to client inquiries and requests

v. Negotiating and establishing contracts that clearly define the outsourcing relationship with clients

2.2.2 Critical issues during the contract execution phase are:

i. Translating client requirements into service deliverables of desired quality

ii. Reviewing and controlling the service design, deployment and delivery activities in order to adhere to desired service levels

iii. Effectively recovering the service in case of failures

iv. Managing client's assets

v. Managing sub-contractor relationships to ensure adherence to service levels

2.2.3 Critical issues during the post-contract phase are:

i. Ensuring a positive client experience

ii. Analysing reasons for occurrence of a termination

iii. Ensuring against re-occurrence of termination for cause

iv. Learning from client experiences

2.3 Critical issues for People

These are:

i. Identifying and developing required personnel competencies

ii. Developing and maintaining a work environment that improves employee satisfaction, motivation and retention

iii. Managing and rewarding personnel performance

iv. Encouraging personnel to innovate

2.4 Critical issues for Technology:

These are:

i. Managing technology capacity and capability

ii. Maintaining technology availability, reliability and accessibility

iii. Ensuring the security of the technology infrastructure and information

iv. Managing rapid technological advances

2.5 Critical issues in Knowledge Management

These are:

i. Learning from past experiences.

ii. Improving the performance of organisational processes

3 The e-Services Capability Model (e^{scm}) : An Introduction

After having realised that no existing quality or capability models address these critical issues with which today's IT service- providers are faced with, IT-Services Qualification Centre (IT^{sqc}) (http://itsqc.srv.cs.cmu.edu) of Carnegie Mellon University (CMU), Pittsburgh (U.S.A) established, in early 2000, a consortium of various industry leaders. Noteworthy amongst them are Satyam Computer Services Limited (www.satyam.com) and Accenture (www.accenture.com).

The IT^{sqc} has been established with an aim to develop capability models for the networked economy. The e-Services Capability Model (e^{scm}) is one such capability model developed specifically to provide guidance to those organisations that are providing IT-enabled outsourcing services. The e^{scm} was released last year during November 2001 and is now freely available for downloading at http://itsqc.srv.cs.cmu.edu/escm

The e***scm*** was developed jointly by a team from Satyam and from IT^{sqc}, and has on its Technical Advisory Board world-renowned quality experts, including the authors of the Software -CMM and the People-CMM.

3.1 e^{scm} Framework

The e***scm*** provides guidance by way of a set of practices that an organisation needs to implement. The e***scm*** framework contains one hundred practices for IT-EOS providers that address the critical capabilities for IT-enabled outsourcing services. All the practices are organised into three outsourcing phases and five organisational elements. Each practice is associated with a capability level.

Phases

e***scm*** provides guidance for the entire outsourcing life cycle, which has three different phases: (i) Pre-Contract phase (ii) Contract Execution Phase, and (iii) Post contract Phase:

- The **Pre-Contract Phase** covers the time and activities until the finalization of the contract with clients and has an overall objective of understanding client requirements and assessing capabilities to meet the requirements, in order to form outsourcing relationships. There are two stages in the Pre-Contract Phase:
    - Requirements Management covers the exchange of information and requirement elicitation, with emphasis on formation and management of the relationship; and,
    - Contract Formulation covers negotiation and contract finalization

- The **Contract-Execution Phase** has an overall objective of translating client requirements into deliverables of desired quality through service design, deployment, delivery, and enhancement. There are two stages in the Contract-Execution Phase:
    - Service Design and Deployment covers the design of service based on specified requirements and transition and/or deployment of designed service;
    - Service Delivery and Enhancement covers the delivery of the service and improvements made based on feedback and analysis of performance

- The **Post-Contract Phase** includes the time and activities that occur after the contract gets completed or terminated. This phase has an overall objective of learning from the contractual experience and ensuring a positive client experience, even in instances where an engagement is terminated because of irreconcilable issues.

The practices, that are applicable in more than one phase ,are classified as **Overall practices**.

Organisational Elements

During all the three phases, a successful outsourcing requires the co-ordinated functioning of various organisation elements. The e***scm*** categorises five Organisational Elements that are critical to successful outsourcing:

- Organisational Management,
- People,
- Business Operations,
- Technology, and
- Knowledge Management.

The practices in e***scm*** address the role of these Organisational Elements across the phases of the outsourcing process.

- **Organisational Management**: responsible for setting direction, providing resources, verifying activities and taking appropriate corrective and preventive actions, to meet or exceed clients requirements and for continual improvement.
- **People** element provides guidance on personnel acquisition, training and development, which is critical to success since personnel play a key role in the design and delivery of IT-EOS.
- **Business Operations:** deals with formulation of contracts with clients, designing services to meet client requirements and delivering the services to clients and end-users. It contains all those practices that are required for meeting SLAs agreed upon with clients
- **Technology** element provides guidance on issues like technology acquisition, deployment and management, since IT-EOS heavily depends on technology infrastructure for service design, deployment and delivery.
- **Knowledge Management** deals with the capturing and using the learning from client engagements as well as processes used there in. It includes knowledge about process and service performance, and of experiences with clients and end-users.

Capability Levels

e***scm*** describes an improvement path for a service provider to progress along from a minimal level of having the capability to deliver a service that meets client requirements, up to the highest level of enhancing value through continuous innovation. There are five levels of capability that define this path and these are:

- Level 1 - Initial;
- Level 2 - Performing to meet client requirements;
- Level 3 - Controlling through measurement;
- Level 4 - Enhancing through innovation; and,
- Level 5 - Sustaining excellence.

At Level 1, service providers lack sound management practices. They operate without formalised systems and procedures leading to frequent crises, exceeded budgets and missed schedules. Thus, service providers at the Initial level are often unable to effectively address their clients' requirements.

At Level 2, a service provider has formalised procedures for capturing requirements and delivering the service per commitments made to clients.

At Level 3, a service provider is able to continuously learn from experience, and measure and control its activities.

A service provider at Level 4 is able to proactively respond to changes in the external or internal business environment, in addition to enhancing the capabilities gained at Level 2 and Level 3.

Level 5 implies continuous sustenance of excellence in the organisation, which is demonstrated by its ability to enhance value to stakeholders and sustain capabilities at Level 4 for at least two years.

The relationship amongst the three different phases, five organisational elements, and five capability levels in the e^{scm} framework are shown in figure 1.0 [3].

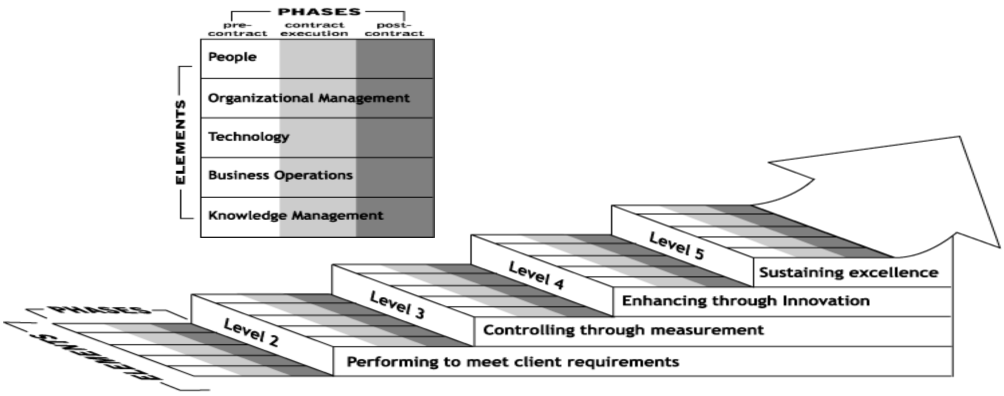


Figure 1: shows relationship amongst the three different phases, five organisational elements, and five capability levels in the escm framework

4 Using e^{SCM} to improve capabilities

Here, we briefly describe the steps taken by an organisation to adapt to the model. The organisation referred here is a large data centre unit, operating out of the Asia Pacific region. It has an Information Services Centre (ISC) that provides IT enabled services to its clients like:

- Main Frame operation management services
- Main Frame hosting service
- Server operation management service
- Server hosting service
- Disaster recovery service

The organisation is accredited under ISO 9001 (1994) under Tick IT scheme. Further, the organisation has adopted six- sigma methodology to improve some of its key performances.

Quality is key to the success of their business, but expectations from their customers for quality seemed insatiable. A few years ago, data processing with 90% quality was good, now they do not seem to be content even with 99.9% quality.

Further, despite the heroic efforts of their competent and committed workforce, things in their shop do "go bump in the night." If an application error results in even a few minutes of downtime for the system, thousands of transactions may be rejected. A bump in the night can be incredibly costly not only for them but also for their clients as well as their end- users. One tiny mistake in a handoff or one small misunderstanding can cost millions of dollars in penalties and a deep dent in their good will.

As the size of their business grew during the last a few years, so did the complexity of dealing with different clients, each located at different corners of the globe – each with their unique set of business imperatives and cultural factors. And consequently, the risks of failures also increased. These risks could be due to several reasons like breach of security in their technology network, not being able to bridge the cultural gap with the clients, not being able to manage SLAs effectively. Needless to add, these risks are additional to the traditional risks in an outsourcing services of not meeting quality parameters or not meeting budget or schedule.

Of late, the organisation did experience some of these problems increasingly. The reasons for this could be grouped in two categories as:

- **Issues with process implementation:** Organisation had a set of processes but for some reasons, they were not implemented the way it should have been. The approach to a client's request typically would be "let's do it this time somehow, we shall follow the process next time onwards". And their internal auditors neatly understated it, "Compliance with the outsourcing processes is a problem."

- **Process inadequacy**: The organisation had not realised the importance for a host of processes that are critical for the success of an IT-EOS. Consequently, they were never considered worthy of being defined. For example, the process of collecting, storing and using client related information for doing a job was not considered important enough for documentation. Similarly, it was never realised that the inability to bridge the cultural gap with their clients was such a serious limitation that might wreck the organisation to lose the client forever.

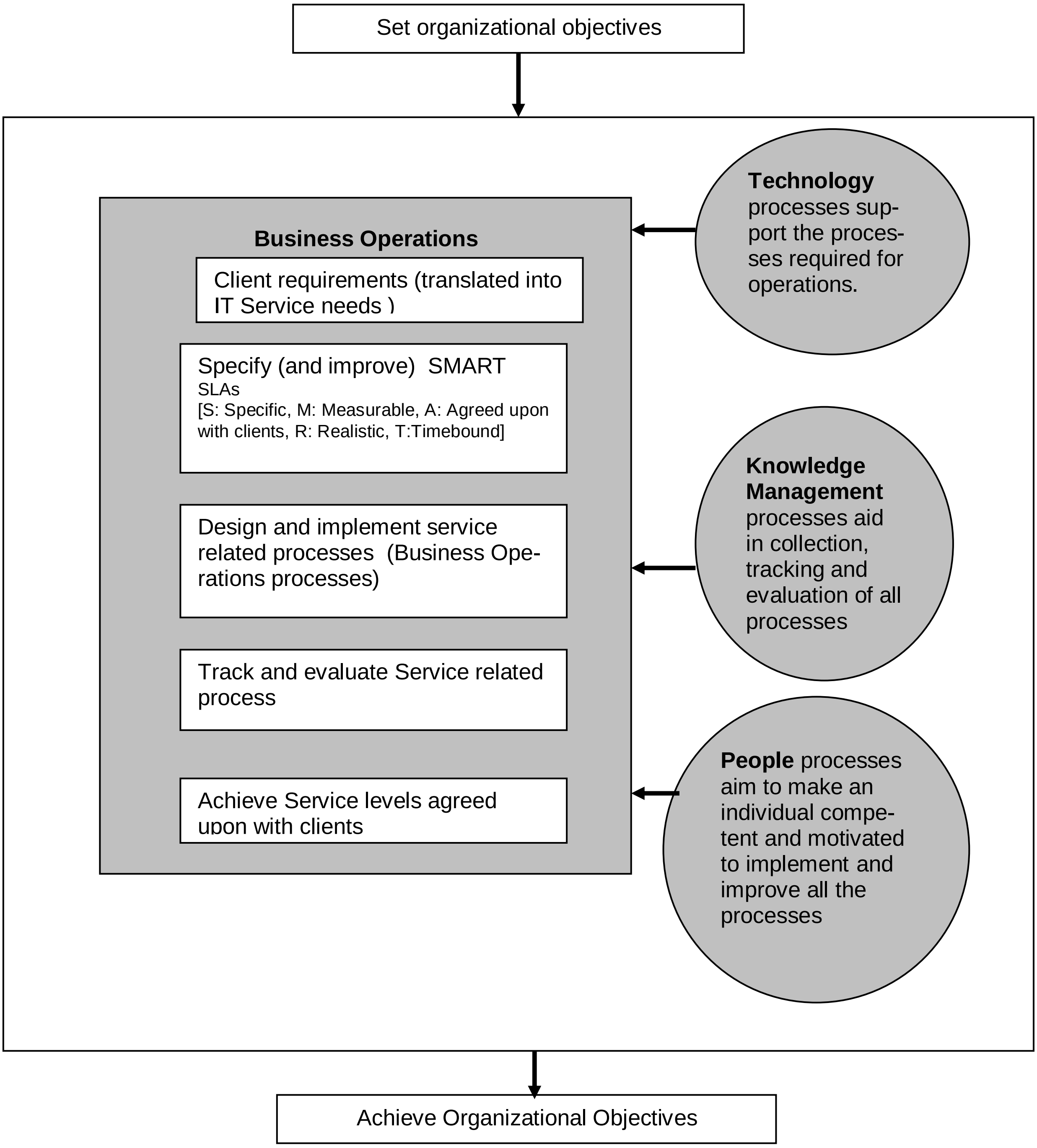
While problems with process implementation could be dealt by strengthening the 'Quality Assurance Group' and taking other such measures, it was the second set of problems (i.e. Process inadequacy) where ESCM is expected to contribute to improve their capability and competitiveness.

The organisation has adopted the model to address its business goals and therefore, has focused all its processes to meet or exceed its business objectives. In order to address the process inadequacy gap, the organisation seeks to implement the model by implementing service related and support processes. Service related processes map to the Business Operations element of the model and are aimed at meeting or exceeding SLA's agreed upon with clients. To enable the organisation to leverage the benefits of these Service related processes, a set of support processes are required. These support processes map to the rest of the Organisational Elements of the model (viz. Technology, Knowledge Management, People and Organisational Management).

These support processes aid implementation and improvement of service related processes so as to enable the organisation to meet or exceed service levels agreed upon by the client. Technology processes are defined so that it can support the processes required for operations. Knowledge Management processes aid in collection, tracking and evaluation of all processes. Organisation has defined People processes that will enable an individual to implement and improve all the processes. Lastly, Organisational Management processes are defined so as to enable the leadership of the organisation to show visible commitment, reviewing the progress and providing resources for meeting or exceeding clients needs and expectations, and bring continual improvement in all its processes. The figure 2.0 describes the relationship of various processes in the organisation.

The figure 2.0 describes the relationship of various processes in the organisation.

Organizational Management processes enable the leadership of the organization to establish organizational objectives, review the progress and provide resources for meeting or exceeding clients needs and expectations, and bring continual improvement in all its processes.



Moreover, the organisation is taking additional step towards:

- Establishing a Knowledge Management system for regular collection, retrieval and usage of information in the timely manner.
- Setting up a metrics program to identify, collect and analyse relevant measures for tracking, evaluation and improvement
- Ensuring security of technology infrastructure and information stored

A few of the benefits that the organisation, consequent to implementation of the model, is looking forward to, include:

- Increased customer satisfaction by way of meeting or exceeding SLA s agreed upon with them
- Increase in productivity
- Substantial reduction in rework, loss of opportunities
- Improved employee morale
- Increased investor confidence
- Reduced time to market of new services

5 Summary

Though rapid boom in the IT-enabled outsourcing services are expected in coming days but chances of failing in such outsourcing relationships is equally probable.

Carnegie Mellon University has recently introduced a capability model for the use of Service Providers of IT enabled outsourcing services. The escm is a capability model that promotes excellence in IT-enabled outsourcing services, through an emphasis on continuous enhancement of services through innovation. It defines a road map for organizations to progress along a path to this end, by first enabling them to develop capabilities to perform processes to meet client requirements, measure and manage these processes, and then use innovation, benchmarking and knowledge as a means of effecting proactive enhancement of services.

It is important that one needs to interpret the model in the context of the business of the organization and then define processes to address issues that are critical for the success of the outsourcing relationships. As is true for any improvement effort, here also the visible commitment from the top management of the organization is imperative for spelling success of the organisation.

6 Literature

- TechTarget.com (2001). http://whatis.techtarget.com/
- Drucker, P. (March 29, 1995) The Network Society, Wall Street Journal, pg. A12.
- The Capability Model for IT-enabled outsourcing service providers – Carnegie Mellon University, School of Computer Science, Vol. 1 CMU-CS-01-163 – Hyder E., Kumar B., Mahendra V., Siegel J., Gupta R., Mahaboob H., Subramanian P. http://itsqc.srv.cs.cmu.edu/escm
- Ozanne, M.R. (February 29, 2000). Barometer of Global Outsourcing - The Millennium Outlook, Sponsored by Dun & Bradstreet.
- Dataquest's 1999-2004 Market Forecast for Business Process Outsourcing (BPO). http://www4.gartner.com/
- McKinsey and Company. (1999). Highlights of the NASSCOM – McKinsey Study Report 1999 – Study on IT-enabled Service Sector, sponsored by the National Association for Software and Services Companies, India.
- Corbett & Associates (2002), IT – enabled Services as Development Drivers in Low- Income Countries http://www.is.cityu.edu.hk/research/ejisdc/vol9/v9r4.pdf
- Gartner (2002), IT Services and Sourcing: Business Imperatives In The Face Of Commercial Uncertainty http://www4.gartner.com/2_events/conferences/asm3i/asm3i.jsp

7 Author CVs

Rajesh Gupta

Mr. Rajesh Gupta, MBA, PMP is a Process Consultant (Quality) at Satyam Computer Services Limited, an Indian based SW-CMM Level 5 organization. He holds B. Tech degree from Indian Institute Of Technology (IIT) Kanpur and MBA degree from Xavier Institute of Management (XIM) Bhubaneswar where he won Gold Medal for securing the top rank in the batch. He has over twelve years of experience in the field of Project Management and Process Management. He has presented numerous workshops and training sessions for many clients frm a wide variety of industries.He is a certified Project Management Professional (PMP) by Project Management Institute (PMI) USA, and has earned the Certified Quality Manager (CQM) designation from American Society of Quality (ASQ) USA.

He is co-author of the new model on IT enabled Outsourcing Services, called e Services Capability Model (e^{SCM}) http://itsqc.srv.cs.cmu.edu/escm/index.html), developed by Carnegie Mellon University, USA. He is a Lead Auditor ISO 9001 and is due to be authorised by CMU as Lead Evaluator e^{SCM} . Presently, he, as a visiting scholar to CMU, is working on the development of other Appraisal methods for e^{SCM}.

Besides, he has already published three papers on quality related topics in national publications.

Bennet Suresh Kumar

Mr. Bennet Suresh Kumar is currently with the Outsourcing Process Consulting Practice of Satyam Computer Services Limited, one of India's leading IT organizations assessed at SEI CMM Level 5. As a Process Consultant, he translates his experience in working with business processes in many leading organizations, into designing and deploying effective processes for clients.

He is a co-author of the escm and was involved in the escm initiative at Carnegie Mellon University, from initial model conceptualization to final publication. He will be one of the first few Qualified Lead Evaluators' of the eServices Capability Model, globally.

Measuring e-business effectiveness: Adaptation of process driven quality approach in web-publishing

Csilla Baksa, Janos Ivanyos, Zsolt Ziaja, Memolux, Hungary, Richard Messnarz, ISCN, Ireland

Abstract

Based on the experiment of the MEDIA-ISF (Media – Information sans Frontieres) best practice action [9] under the 5th framework programme of the European Union, the authors summarise the results of how the process driven quality model and tool are supporting the measurement and evaluation of the adapted web-publishing business model. This media business adaptation contributed towards a new and innovative model for e-working, e-commerce and dissemination in Europe building a bridge between relevant information on best practice gathered in European regions and Europe's media (news agencies, journalists, newspapers, portals, and so forth).

Measurement and evaluation of running e-business models have become more important since the reappraisal of the business success measures of electronic commerce and e-work processes. In general the former differences of the financial, IT and quality evaluation models of business processes have been disappearing, the audit standards converge to each other. By this way the IT – or we can even use the traditional term of “Software Process Improvement” – methodology can be practically adapted in non-software business environment as well, especially when it uses extended Internet-based technology for the commercial and teamwork cooperation.

The described measurement and evaluation experiments are based on the usage of already existing blocks of former European technology, methodology and business results (such as NQA Integrated Teamwork and Network based Quality Assurance model and tool [7], [8], EFQM Excellence Model [2], [3], [4], BESTREGIT innovation transfer experiments [1], and MEDIA-ISF electronic content distribution model). The dissemination of these experiments contributes to the further extension and European wide adaptation of the Software Process Improvement results in other user domains.

Keywords

Best Practice, E-Work, MEDIA-ISF, Measurement

1 The Underlying Service Model

The MEDIA-ISF model [9] was established on the following 3 level architecture of information service delivery:

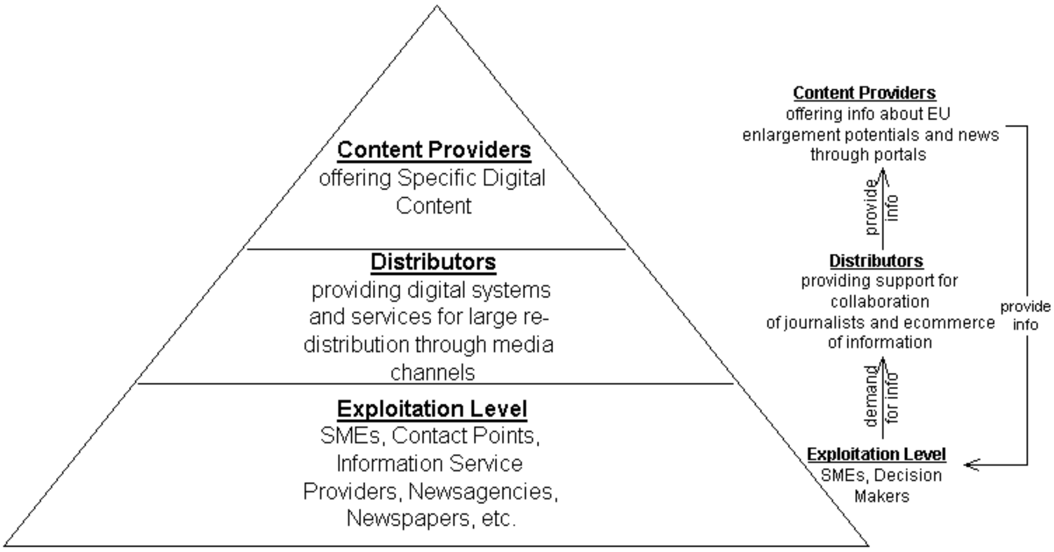


Figure 1: The Underlying Information Service Architecture

Content providers in the MEDIA-ISF trials are different news agencies (Bruxinfo, Euractiv, EON, etc.) who collaborate on topics for the European integration process, as well as representatives from the EU Innovation Relay Centres network. On the technical side their work is supported by a team-work environment over the Internet which follows defined quality processes for media publishing and it enables the different content providers to collaborate on subject dossiers in a controlled way.

Distributors are service providers who offer the developed content to registered users. Content providers like a news agency can be also such distributors, as well as each content provider might have a group of national contact points who offer the content to regional users (SME service contact points). On the technical side the distribution is supported by a portal (Media Bridge) which receives submissions from the team-work server and distributes information automatically further (by key word selection) to interested content groups.

Both technical support systems, the teamwork server and the Media Bridge, are integrated via a key word system, which can be configured per media publishing project.

On the **exploitation level** SMEs (or just European citizens) connect to the distributors, register, and receive the high quality information (subject dossiers) which was elaborated in a defined virtual teamwork process and the publication of them was supported by the Media Bridge.

A major advantage of the MEDIA-ISF virtual organisation model is the “catalyst“ function provided by the interconnection of different digital content groups. While in the past the same subject (e.g. EU position on IT investment in Eastern Europe) had to be developed by each news agency themselves, they can now share knowledge and come up with a joint subject dossier at higher quality of information.

We have to highlight, that MEDIA-ISF technical architecture is not bound to just the case of news agencies and EU content providers. It could be used also for any other content areas in the EU where different content groups want to share knowledge and collaborate on topics, which should be published in news servers of Europe. This way MEDIA-ISF is a new system approach of a future European dissemination platform.

In this paper we will also discuss the measurement topics of the web-publishing adaptation and refer to the model in general or to the Virtual Organisation Management issues only when these issues give wider context of usability.

However, as outlined above, we assume that the most important factor for sustainability of the model is based on the connection of many different content groups to work in many different EU domains in the future.

2 Defined Service Processes

As stated in the project agenda the general approach of the EFQM model [2], [3], [4] was applied. So areas such as leadership, people, policy, resources, processes, business goals were analysed and teamwork processes relevant for this MEDIA-ISF application platform have been defined. Figure 2 illustrates the main identified processes for which team-work scenarios have been analysed and documented.

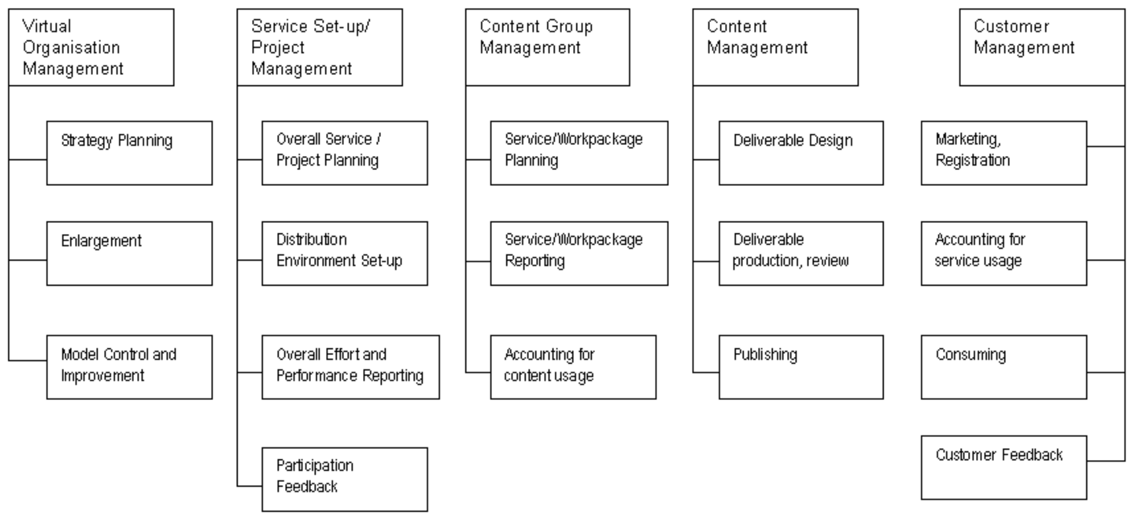


Figure 2: The MEDIA-ISF Process Map

The team-work processes have been analysed applying the BESTREGIT methodology [1], [5] what came to the role based and result-driven work-flow scenarios. The BESTREGIT methodology delivers models which can be integrated into the team-work platform system NQA [6], [7], [8] by configuration (no change of code and just by configuration of user interface templates).

Figure 3 shows a typical team-work scenario for e.g. Deliverable Publishing in the Content Management process area.

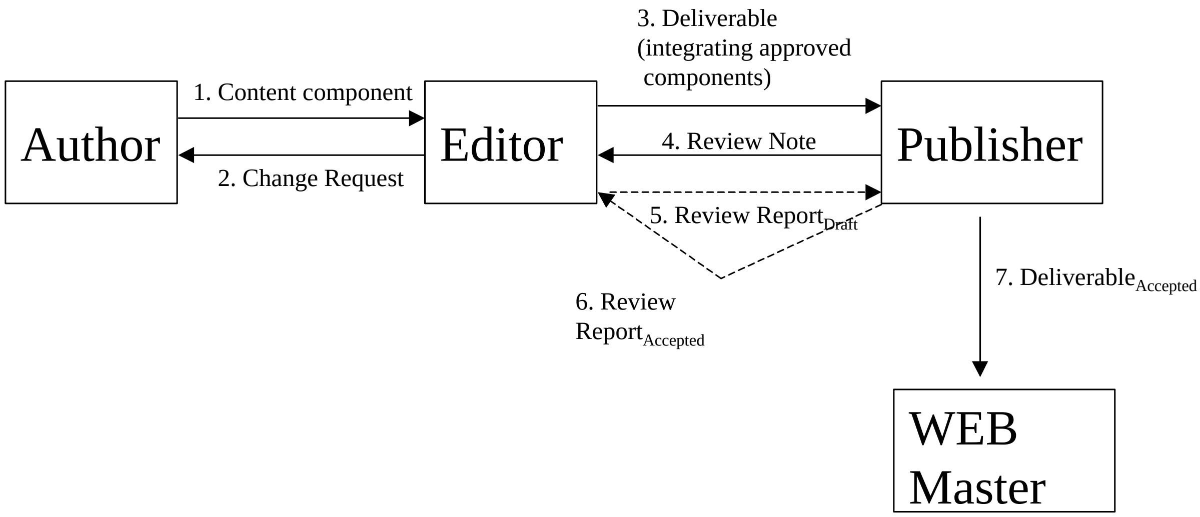


Figure 3: The Deliverable Publishing Team-work Scenario

In such a scenario

- People get accounts on the system and are assigned to roles like "Author" in a specific media publishing project;
- For each result templates are generated by the system and used by the roles to produce the results;
- The team-work system supports the above highlighted submission flows, as well as control measures and metrics.

Again it must be noted that the underlying technology is configurable to any set of work scenarios and collaborative authoring project types, so that the same platform can be adapted to many other content domains in Europe. Its strengths bases on the fact that it does not only give you support to share knowledge or files etc. but to configure defined work processes which follow a role based team-work philosophy.

3 The Underlying Technology

The NQA approach bases on three principles, which have been discussed and published at previous ISCN conferences (http://www.iscn.ie/conferences) and in the book published by IEEE [MESS99b].

The three principles

- Role and information flow based team work process management
- Development by configuration
- Re-Use pool concept

are discussed below.

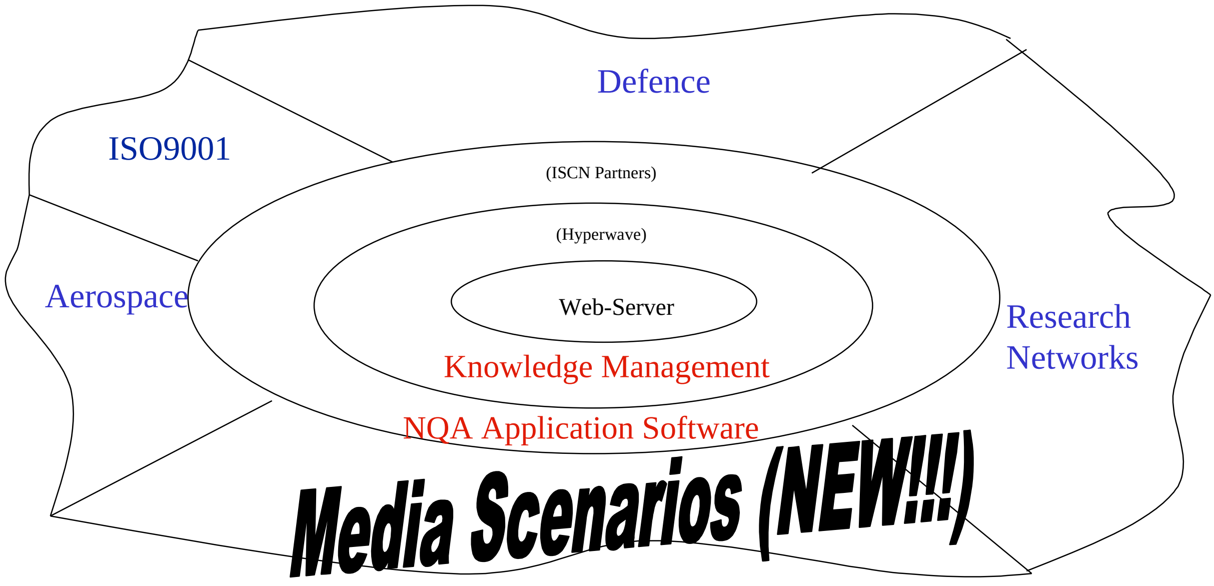


Figure 4: The NQA/Teamwork Platform System

A major feature to make such a virtual approach applicable for different environments is that such a system must be kept completely configurable. The menu, the data, the functions, the document/information flows can be configured for different user scenarios and this high configurability is the major feature of an NQA virtual office.

NQA [7], [8] configures

- Project types
- Team-work scenarios (each project type contains a number of working scenarios)
- Document and report templates and submission flows per project type
- Roles which are played by people who get accounts
- Assignment of people to teams who are assigned to projects

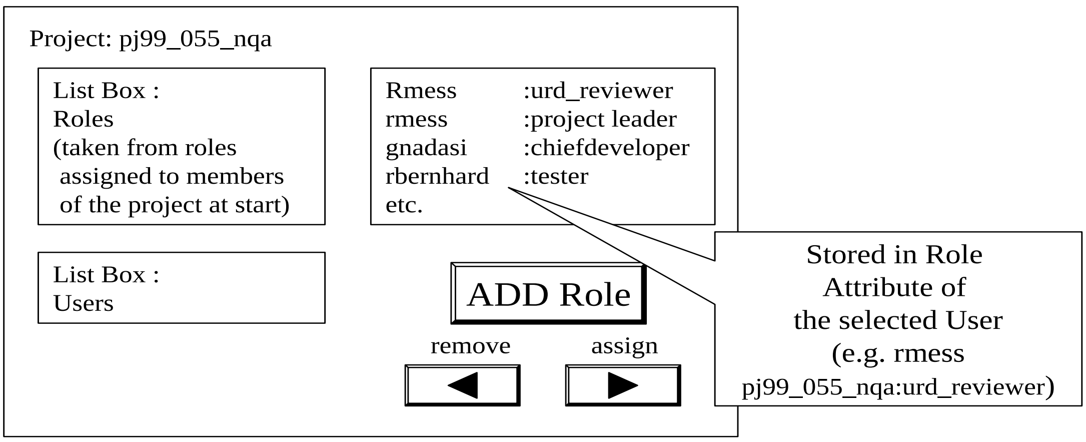


Figure 5: Configuration of Roles and Users

Once this is done an administrator can create projects of that type and support team-work on the Internet. There is a large re-use factor as e.g. a configuration for ECSS (new European Space Agency Standard) has been done and can, for instance, be re-used by all these organisations. The same way is open for the developed MEDIA-ISF scenarios and project types.

People are assigned to project groups as well as to roles defined in a particular project, and distribution lists are automatically generated so that documents and reports can be automatically submitted to the appropriate team members who are responsible for specific role(s) in the project group. Tasks are created for these roles and the completion can be traced.

3.1 The Media Bridge Interface

The below architecture was implemented within the NQA tool for supporting web-publishing:

- Using the NQA system journalists collaborate in a team to share content components through a server and integrate deliverables (articles, newsletters, etc.) to be published. Once a content component is to be published the PUBLISH function is called and info about the content components is submitted to a mailing list of web editors at news agencies.
- PUBLISH also copies the specific content item from the private media publish project area to a public collection which allows that all news agencies receiving the information can directly link the corresponding content from their news servers.
- Also each PUBLISH information submitted from NQA contains a set of key words (configurable per media publish project) so that a Media Bridge (portal function) selects those content providers automatically who should be notified of the availability of that content.

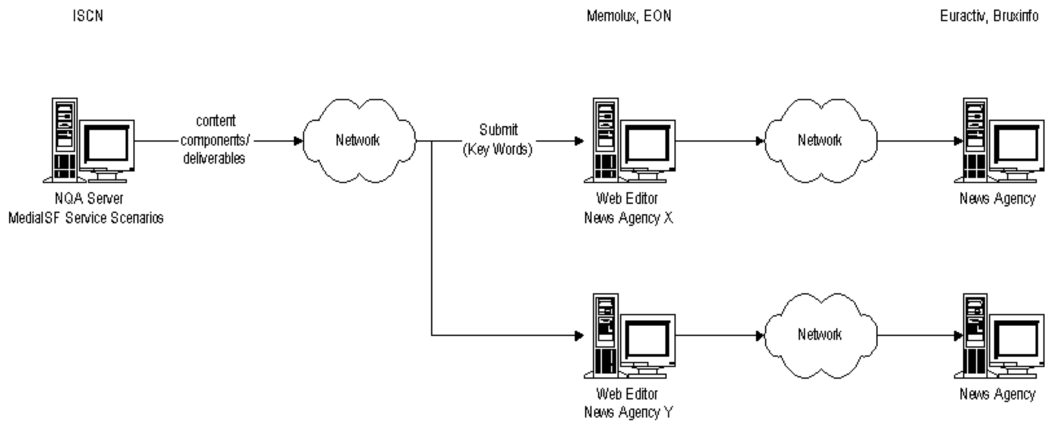


Figure 6: Interface between the NQA System and the News Agencies

4 EFQM Implementation and Measurement

The MEDIA-ISF model used the thematic approach of the EFQM Excellence Model [EFQM_Model] and its self-assessment methods [4] focusing on the specialities of the multi-regional, multi-lingual co-operation of potentially many thousand SME organisations through the information network service. The following key conceptual elements of the EFQM Excellence Model were implemented in the case of the media application:

Result orientation

- To establish new media-based, electronic information network service presented by the "Media on

EU Enlargement" content of the virtual organisation, switching from the present media oriented approach to a more subject oriented approach, and opening the way for multilingual exploitation as well

- To make the forum where SME businesses can capture the needs for their services, can find their stakeholders, partners, human resources, customers, suppliers to meet the needs through the information network services of the digital contents provided by the virtual organisation, using adequate search engine opening the way for subject oriented systemisation and retrieval of the segments of each objects
- To build up the virtual organisation which opens new business opportunities for the participants being involved in Europe wide co-operations via controlled and visible way provided by the "Virtual Organisation" digital content distribution
- To set up the processes supported by the relevant technology to balance the participants' expectations and interests at each level of the virtual organisation

Customer focus

- To demonstrate the business drivers of the virtual organisation, which ensures a unified high quality level for the clients of the digital content related information services
- To assess the requests and benefits of the SME business circles for the planned multimedia-based information services and use the evaluated results to improve the e-work and e-business models of the digital contents
- To make for the customers easy and fast to find services, which help them to activate their enables by the innovation and learning support of the "Exploitation" level of the digital contents provided by the virtual organisation

Leadership and Constancy of Purpose

To make conceptual refinements based on the EFQM Excellence Model for the virtual organisation, which ensures that the stated quality goals and the consistent proceeding be on high quality level according to the goals that meet the participants' expectations and interests as shared values and ethical behaviour

Management by Processes and Facts

- To provide a solution for e-work and e-business management of a European wide multilingual service network
- To initiate gathering best practice processes on the base of the self-assessments regarding e-work and e-business processes of the virtual organisation
- To make full understanding and usage of the information service processes by all of the stakeholders of the 3-level information services of the virtual organisation

People Development & Involvement

- To empower people to act and openly share their knowledge and experience through the different levels of the organisational structure
- To help people to get the most benefit from furthering organisational objectives

Continuous Learning, Innovation and Improvement

- To provide benchmark database initials for the participants
- To provide the possibility of easy adaptation and re-usability of the system supporting the information service processes
- To form the learning organisation culture integrated into the information service processes for the continuous improvement

Partnership Development

- To enhance working possibilities through offering the opportunity for small and medium business oriented enterprises and individuals (from starters or re-starters, part-time workers or scientists, self-employed to unemployed people, etc.) to subscribe for the virtual organisation, and after an educational preparatory period become active players in the e-work and e-business models provided by the services of the implemented digital contents
- To increase efficiency of the system supporting virtual organisation, making it flexible enough to offer the possibility of broad diversification in the services, ensuring that without additional investment, based on the same technology, the new participants can have the potential ability to answer easily to a much wider range of demands in the increasing virtual market
- To demonstrate the business sustainability of the virtual organisation, which ensures competitive businesses for information service providers
- To demonstrate the co-operation of the participants within the virtual organisation based on the concepts of the European model of excellence, which ensures both a unified high quality level and competitive businesses
- To use the 3-level organisational structure providing e-work solution to co-operate in the development of e-business models on the basis of shared knowledge

Public Responsibility

- To make possible to narrow the information gap between the centralised European newsrooms and the local service providers by using the foreseen multimedia based personalised on-line service, thus cutting down the time and cost of collecting and distributing information
- To increase the European competitiveness against other parts of the world, tracking special aspects of new types of organisational and employment (e-work) structures utilising quick information flow and virtual network through Europe
- To increase the participation of Western and Eastern European SMEs in cross-border e-business by offering adaptable technology, organisational methods and up-to-date legislation information of retrieving and using information network services

Based on these conceptual elements the following measurement plan has been established:

Process	Key Goal Indicators	Key Performance Indicators	Critical success factors	Measurement method
Virtual Organisation Management				
Strategy Planning	Model Design	Scenarios, Quality System	Used scenarios, Change Management	Number of documents and reports
	Evaluation	Model Scope	Continuous improvement	Assessment results
	Strategy Communication	Public strategy	Clear mission	Publications
Enlargement	Demonstration	Dissemination	Effective dissemination	Publications, demonstration materials
	Training	Audience	Audience from different user domains	Audience on training events

Process	**Key Goal Indicators**	**Key Performance Indicators**	**Critical success factors**	**Measurement method**
	Contract	Consortium agreements	Consortium growth	Number of consortium partners
	Awarding / Development projects	Applications / Proposals	Successful applications	Applications, Success rate
Model Control and Improvement	Non-conformance Management	Corrective and preventive actions	Customer complaints	Handled Customer complaints
	Measurement Control	Process map for Measurement	Automation	Coverage of process map
	Model Self Assessment	EFQM principles	Conformance with EFQM requirements	Assessment reports
	Audit	Sustainability	Achievement of business objectives	Audit reports
	Model Improvement	Model maintenance and review	Flexibility	Reaction time, Model submissions
Service Set-up / Project Management				
Overall Service / Project Planning	Overall Project Planning	Project scenarios, Participants, Project deliverables	Scenarios, assignments	Applied scenarios, scenario assignments
	Overall Service Management	Overall Service Plan, Service Scenarios, Participants, Service modules	Business plan, scenarios, assignments	Business analysis, applied scenarios, scenario assignments
Distribution Environment Set-up	MEDIA-ISF Exploitation Model Mapping and Implementation	Implementation scope	Configuration, bridges	Implemented content groups/ projects/ services, bridge contacts
Overall Effort and Performance Reporting	Overall effort reporting	Effort plan, change management	Effort table	Effort table usage

Process	**Key Goal Indicators**	**Key Performance Indicators**	**Critical success factors**	**Measurement method**
	Performance Reporting	Content-specific key performance indicators	Business metrics	Content-specific metrics
Participation Feedback	Participation Issue Management	Change management	Handling of partner needs	Handled partner needs
Content Group Management				
Service / Workpackage Planning	Service Scheduling	Deadlines, participation	Timeliness, completeness	Delays, refusal
	Workpackage Planning	Deadlines, participation	Timeliness, completeness	Delays, refusal
Service / Workpackage Reporting	Service / Workpackage Performance Reporting	Content-specific key performance indicators	Business metrics	Content-specific metrics
Accounting for content usage	Profitable content production	Share on Service revenues	Content group accounting	Rate of Content providers share on service revenues
Content Management				
Deliverable Design	Deliverable requirement definition	Completeness, correctness	Applicability	Review notes
Deliverable production, review	High quality product	Review scenario	Accepted deliverables	Review notes, acceptation rate
Publishing	Wide range content distribution	Publishing scenario	Distribution growth	published, rejected/unpublished contents
Customer Management				
Marketing, Registration	Demonstration	On-line and traditional offers	Availability through information service providers	Number of interconnected information portals and available dissemination materials
	Training	Audience	Participation of professionals	Online and traditional training courses, Participation rating

Process	**Key Goal Indicators**	**Key Performance Indicators**	**Critical success factors**	**Measurement method**
	Registration	Registrations	Growth of active user groups	Registration figures of participants
	Awarding	Applications	Visibility for other user groups	Applicants, winners, Success rates given by other user groups
Accounting for service usage	Profitable service	Revenue from information service	Profitability	Registration fee, growth rate
Consuming	Value-added service consumption	Commercial channel usage	Content usage	Downloads
Customer Feedback	Customer Issue Management	Change management	Fulfilment of customer requirements	Handled customer needs

Table 1: The Process Map for Measurement

4.1 Some Measurement Results

4.1.1 EFQM Self Assessment

The EFQM principles were used to set up the assessment criteria. After 4 months of the media validation phase, the following self-assessment results were achieved for the running web-publishing activities:

	FUNDAMENTAL CONCEPT	START UP	ON THE WAY	MATURE
1	Results Orientation	All relevant stakeholders are identified	Stakeholder needs are assessed in a structured way ✓	Transparent mechanisms exist to balance stakeholder expectations
2	Customer Focus	Customer satisfaction assessed	Goals and targets are linked to customer needs and expectations. Loyalty issues are researched ✓	Business drivers of customer satisfaction needs and loyalty issues are understood, measured and actioned
3	Leadership and Constancy of Purpose	Vision and Mission are defined	Policy, People and Processes are aligned. A leadership 'model' exists ✓	Shared Values and ethical role models exist at all organisational levels
4	Management by Processes and Facts	Processes to achieve desired results are defined	Comparative data and information is used to set challenging goals ✓	Process capability is fully understood and used to drive performance improvements
5	People Development & Involvement	People accept ownership and responsibility to solve problems	People are innovative and creative in furthering organisational objectives ✓	People are empowered to act and openly share knowledge and experience
6	Continuous Learning, Innovation and Improvement	Improvement opportunities are identified and acted on	Continuous improvement is an accepted objective for every individual ✓	Successful innovation and improvement is widespread and integrated
7	Partnership Development	A process exists for selecting and managing suppliers	Supplier improvement and achievements are recognised and key external partners have been identified	The organisation and its key partners are interdependent. Plans and policies are co-developed on the basis of shared knowledge ✓
8	Public Responsibility	Legal and regulatory requirements are understood and met	There is active involvement in 'society' ✓	Societal expectations are measured and actioned

Figure 7: EFQM Self Assessment Results

4.1.2 Business Achievement Measurement

Progress and success metrics related to the business objectives of the web-publishing adaptation of the MEDIA-ISF model have compared with the baseline (at start), short term (after 6 months) and expected (from the second year) status:

	Baseline	**Short term**	**Expectation**
Increasing customised volume and topic of information available for network participants	Existing separate portal services of EurActiv, EON, Bestregit, IRCA, etc.	New MEDIA-ISF Portals linked to the extended EurActiv, EON, BruxInFo, APS portals	Dozens of inter-connected portals
Increasing number of registered end-users	N/A	Up to 2000	Over 4000
Increasing volume of activity types available for registered media SMEs	Searching	Searching and Teamworking	Commercialising
Increasing participation in the learning process-es	EJC Trainings, Bestregit	Additional Training Activities and new modules for exist-ing courses	Additional self-learning by the MEDIA-ISF Model
Increasing number of active end-users provid-ing service by MEDIA-ISF	N/A	More than 10	Up to 200
Increasing user satis-faction	Implementing rating mechanism	Presenting the metrics	Follow-up the measured satisfac-tion data
Increasing productivity	Cutting the aver-age cost/web article	Cutting the average cost/customised web article	Cutting the average cost/commercialise d content page
Increasing volume of new partner-ship/cooperation in the value chain	Consortium for MEDIA-ISF pro-ject	MEDIA-ISF Best Practice Partner-ship (2 content groups)	5-10 new content groups within 2 years
Increasing volume of interconnections of dif-ferent business groups based on multi-linguality and multi-regionalism	MEDIA-ISF pro-ject	Interconnection of the media and in-novation business groups of UK, IRL, B, NL, A, SIT, H	Extension to En-largement countries and other Europe-an regions
Higher Quality Control on virtual collaboration	Non standard methods	ISO9001:2000 conform workflows	Excellence Model
Increasing volume of technology transfer	Bestregit	NQA technology implemented for MEDIA-ISF	Information Service Distribution for NQA and other new tools
Process Improvement	Bestregit	Media authoring & distributing	New content relat-ed processes
Increasing role in policy and strategy making of the MEDIA-ISF organi-sation	MEDIA-ISF pro-posal	MEDIA-ISF Exploi-tation Plan	Extension towards other Exploitation Models

Table 2: MEDIA-ISF Business Objective Measurements [MISF2001]

5 Outlook

The overall service architecture and the technical concept discussed in the previous chapters of this paper can be adapted "by configuration" to many different content groups. An adaptation would just require

- the configuration of new collaborative scenarios on NQA,
- the adaptation of parameters to be given to the call interface of the transaction processing system and the European portals

6 Literature

[1] Bestregit Improvement Guideline , http://www.victory-group.net, R. Messnarz. C. Stöckler, G. Velasco, G. O'Suilleabhain, Results of the EU Leonardo da Vinci Project BESTREGIT, July 2000

[2] EIGHT ESSENTIALS OF EXCELLENCE - The Fundamental Concepts and their Benefits, http://www.efqm.org

[3] THE EFQM EXCELLENCE MODEL – Companies version, ISBN 90-5236-082-0

[4] ASSESSING FOR EXCELLENCE – A practical guide for self-assessment, ISBN 90-5236-093-6

[5] 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

[6] Messnarz R., Tully C. (eds.), The PICO - Book: Better Software Practice for Business Benefit - Principles and Experience, IEEE Computer Society Press, in publication

[7] 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

[8] Messnarz R., Nadasi G., Bernhard R., O'Leary E., Foley B., Experience with Teamwork in Distributed Work Environments, in: Proceedings of E-Work and E-Commerce 2001, eds. Brian Stanford Smith, Enrica Chiozza, IOS Press, Amsterdam, Berlin, Oxford, Tokyo, Washington, June 2001

[9] György Fóris, János Ivanyos, Richard Messnarz: MEDIA-ISF Information sans Frontieres, in: Proceedings of the European Software Process Improvement Conference, 10-12 October 2001, Limerick, Ireland

7 Author CVs

Csilla Baksa

Ms. Csilla Baksa was graduated as mathematics-physics teacher in 1984, and programmer mathematician in 1987. She participated in Computer Information Systems Auditor training in 1998 and got IQA IRCA Lead Assessor (Quality Auditor and Certifier) certification in 1999.
After working as system designer and analyst for several years she joined to MemoLuX in 1996. She is responsible for quality assurance, process improvement and project management assistance of MemoLuX projects, including Whirlpool, EDS, PASS, MEDIA-ISF etc.

Janos Ivanyos

Mr. Janos Ivanyos is one of the founders of company Memolux and the managing director responsible for Information and Communication Technology support and development since 1989. He was graduated as an economist at University of Economics, Budapest in 1984. He was working for the Computer Center of the National Planning Office for 4 years and then established his first computer service company. He managed several IT projects including IT outsourcing services to Unilever Hungary, Heating Works of Budapest, National Employment Office, Ministry of Finance, etc. He was the Project Manager and Contractor for the PASS Esprit project during the period of 1997-1999. He is the Technical Coordinator of the running MEDIA-ISF (IST-2000-29651) project.

Zsolt Ziaja

Mr. Zsolt Ziaja was graduated as software engineer at the Technical University, Budapest in 1994. He started to work for the University as lecturer, and joined to Memolux organisation 7 years ago. His practice includes Unix-based development on heterogeneous platforms and he led several software development projects. He was the project manager of the baseline of PASS Process Improvement Experiment Esprit project. He is a working group manager in the running MEDIA-ISF (IST-2000-29651) project.

Dr Richard Messnarz

Dr Richard Messnarz made his MSc at the University of Technology in Graz in Austria. In 1995 he received a PhD from the same university for the work on a QUES - "A Quantitative Quality Evaluation System".
He has participated in a number of European research and industry projects since 1990. He was also the project leader of the Comett expert exchange project ISCN in 1993, which led to the foundation of ISCN itself in 1994.
He is the Executive Director of ISCN. He was the technical co-ordinator in previous process improvement initiatives such as PICO (Process Improvement Combined apprOach), Victory (Virtual Enterprise for Process Improvement), and the co-ordinator of a software engineering group within the pilot project for building a prototype of a virtual university. He is the editor of a book published in 1999 from IEEE "Better Software Practice for Business Benefit - Principles and Experience". He is currently the technical director of a large e-working initiative TEAMWORK, and manages in the initiative MEDIA-ISF the technical development for an e-working solution for news agencies collaborating on Eastern European enhancement topics.
He is a chairman and main organiser of the EuroSPI conference series (the former ISCN series). He is a SPICE and BOOTSTRAP lead assessor.

Software Process Assessment in a Component Environment

John Torgersson
University College of Borås, Sweden
School of Business Administration and Informatics
Tel: +46 (0)703 - 92 41 84
e-mail: john.torgersson@hb.se
Address: Allégatan 1 SE– 501 90 Borås

Alec Dorling
University College of Borås, Sweden
School of Business Administration and Informatics
Tel: +46 (0)33 - 17 46 73
e-mail: alec.dorling@impronova.com
Address: Allégatan 1 SE– 501 90 Borås

Abstract

The current international standard for software process assessment, ISO/IEC TR 15504, has proven to be suitable for general software development approaches. However, there is a lack of coverage of Component-Based Development CBD specific issues when it comes, both to the content within the processes and the terminology used. Thus, there is a need for new or changed processes as the basis for assessing organisations using a component based approach for software development. To meet this need a Process Reference Model (PRM) that captures these changes has been developed by the OOSPICE project. Changed processes also trigger the need for new software process assessment methodologies to be able to determine the capability of CBD organisations and by that, be able to improve the quality of software components. Such a methodology is also under development by the OOSPICE project.

Keywords

OOSPICE, Software Process Assessment, Component, Software Process Improvement and Capability dEtermination, Component-Based Software Development, Component-Based Software Engineering, SPICE.

1 Introduction

An accepted principle in software engineering is that process quality determines product quality (Humphrey, 1989)(Cignoni, 2000). Process control is essential for predictability and repeatability in all kinds of software development (Wallnau, Hissam, Seacord, 2002). The assessment methodology ISO/IEC TR 15504, the international standard for software process assessment, is now well reputed and used in a variety of markets and different kinds of organisations. The large number of user trials has proven that the methodology is highly suitable for software process assessment and improvement. Existing software processes, however, are rooted in general software development approaches and do not suit component-based development (CBD) (Wallnau et. al 2002). Although the CBD approach is founded upon a solid background of experience from the field of software development it is still immature and in need of processes based on best practices (Heinman,Councill, 2001). Furthermore, assessors using the methodology in organisations that are using a component-based approach when developing their software have experienced that these organisations find it hard to relate to the processes and terminology in the standard. Therefore, new processes need to be defined that are suitable for the organisations in the component-based software development market.

Thus, there is a gap that needs to be bridged by combining inputs from experts from two important software engineering disciplines: namely software process assessment and improvement and Component-Based Software Development and develop a new process assessment methodology that is suitable for organisations working in a component-based environment.

1.1 Introduction to OOSPICE

OOSPICE[1] is a project to extend the current ISO/IEC TR 15504 process reference model to cover Component-Based Development. CBD is generally acknowledged to be key for improving time-to-market, productivity, quality and re-use in software development. OOSPICE focuses on the processes, technology and quality concerns in component-based software development. Based on the principles of empirical software engineering, the OOSPICE project combines, in an innovative way, three major concepts:

- component-based software development
- software process assessment
- software process improvement.

Currently, suitable processes for CBD are still emerging and their general applicability is being debated. Developers of component-based systems find it hard to determine and assess the appropriateness of available components. OOSPICE aims to address these problems and provides processes, methods, tools and standards to create a technological basis needed for effective CBD and component provision.

The main objectives of OOSPICE are to develop:

- a unified CBD process metamodel,
- a CBD assessment methodology and assessment software tool delivering component-provider capability profiles,
- a CBD methodology and software tool,

[1] The ‘OO’ in ‘OOSPICE’ relates back to the origins of the project where the original focus was object-oriented development. While developing the project, it was concluded that component-based software development needed to be addressed more than object-oriented development. So, in ‘OOSPICE’, for ‘OO’ read ‘CBD’, remembering not to confuse components with objects.

- and a Process Reference Model to extend the ISO/IEC 15504 process assessment standard.

1.2 Scope

OOSPICE has developed a software process assessment methodology for component-based software development. This paper will describe this methodology and focus on a subset of the processes within the Process Reference Model (PRM) in order to motivate the need for the process as well as highlight their CBD-specific issues. The international standard for process assessment, ISO/IEC TR 15504, contains a PRM for traditional software development. Lack of CBD coverage in this model will be pointed out to provide information about what content and which processes need to be incorporated in a CBD PRM.

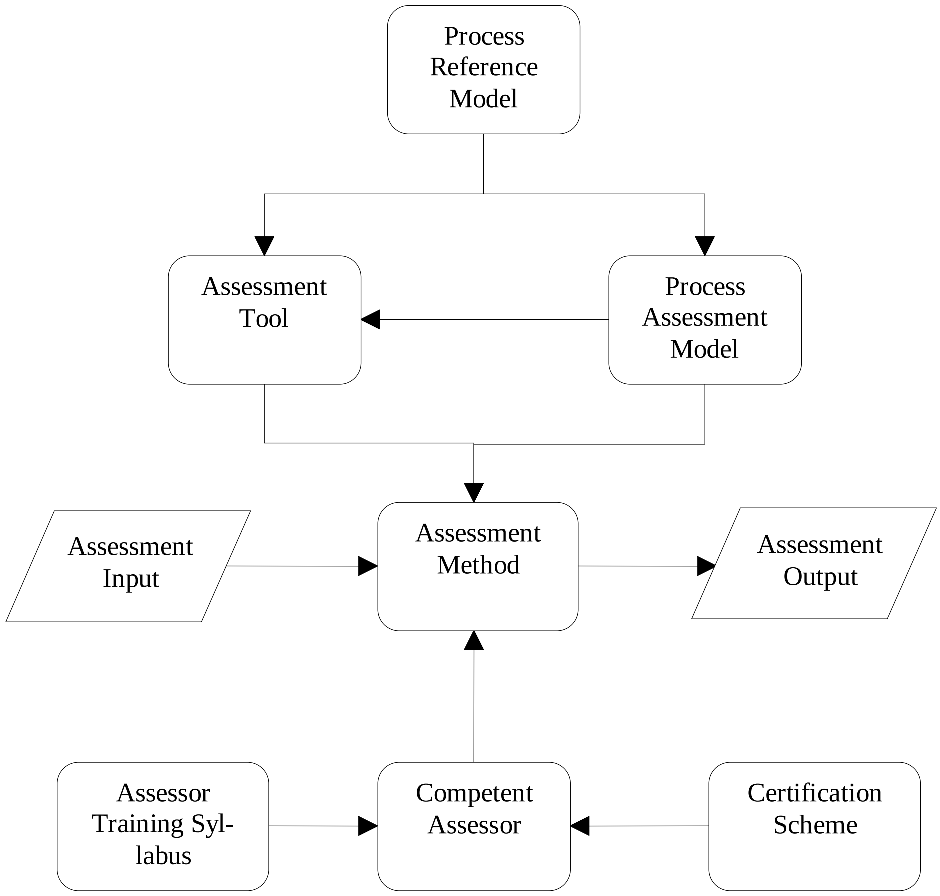
2 CBD Assessment Methodology

The OOSPICE process assessment methodology is a framework for assessing software processes in a Component-Based environment. The methodology aims to be used in all kinds of organisations and market sectors that have adopted a component-based approach to software development. The methodology consists of:

- a Process Reference Model (PRM) containing processes for Component-Based software development (CBD),
- a Process Assessment Model (PAM) that incorporates the PRM and extends with detailed indicators as a basis for assessing processes
- an assessment method that defines how an assessment shall be performed within the assessment framework,
- an assessment tool that is used to aid collection of assessment data, rating and presentation of the result,

The diagram below shows a graphical overview of the assessment methodology.

Figure 1 OOSPICE Assessment Methodology



An assessment is lead by a Competent Assessor who has the required knowledge, skills and experience. The OOSPICE project has defined the basis for the operation of a certification scheme for assessors together with an assessor training syllabus for training courses. To be able to perform an assessment an assessment method and process assessment model is needed that are conformant to the requirements of the international standards ISO/IEC 15504.

2.1 The Assessment Method

The current international standard for software process assessment, ISO/IEC 15504, contains a set of requirements for the performance of an assessment. The CBD assessment method is developed to be conformant to the requirements in the standard.

The objective with the method is to provide as much help and guidance as possible for the assessment team in order to facilitate the assessment process and improve the quality and reliability of the assessment result. Therefore the method emphasises the use of templates. To ensure the output from different assessments are comparable and to decrease the likelihood of leaving out important information when reporting back the result to the assessed organisation.

2.1.1 Structure of the method

Three phases constitute the method namely, the preparation-, assessment and reporting phase. Each phase has a purpose and a set of activities. The purpose is a description of the overall objectives of the phase and gives an indication about what to accomplish within the phase. Each phase is further divided into activities. Each activity has a general description and a number of tasks. The description gives an overview of what is to be done within the activity. The tasks are a further elaboration about how to meet the requirements of the activity. The roles needed to carry out the activity are provided following the tasks as well as the output(s) produced. The outputs are in form of templates, which are documents that help the assessors in collecting the data and presenting the result.

In the method a set of guidelines is also provided and is incorporated as an extra source of information about how to use the assessment method and perform the tasks. The guidelines serve as additional information about how to conduct the different activities.

The phases within the method are phases that an assessment team will go through during any assessment.

Preparation Phase

During the preparation phase the organisation to be assessed and the assessment team are prepared for the upcoming assessment phase. The major part in this phase is the planning and scheduling of the assessment. The assessment plan is created containing information about assessment scope, roles and responsibilities and the key activities to be performed during the assessment.

Assessment Phase

Most of the activities during the assessment phase are performed on site in interaction with the assessed organisation. In this phase the assessment team and the participants from the assessed organisation are briefed on the purpose with the assessment as well as the methodology to be used. After that, data is collected sufficient to meet the assessment purpose and scope and to rate each process attribute up to and including the highest capability level defined in the assessment scope. The rating is based on objective evidence gathered from the collection and validation of data.

Reporting Phase

When the data is collected and the different processes are rated, the result is reported back to the assessed organisation. The report will highlight the process profiles, key results, perceived strengths and weaknesses, identified risk factors and potential improvement actions.

2.2 The Process Reference- and Assessment Model

A Process Reference Model (PRM) has been developed for use in assessment of CBD-processes. The processes are described according to the requirements of the new emerging framework in ISO/IEC 15504. Each Process is defined with a process name or identifier, a process purpose and a set of outcomes.

Organisations will be assessed against a Process Assessment Model elaborating on the PRM that must be able to relate the process information to their CBD work. The assessment model provides guidance when assessing process performance within organisations that have adapted to the CBD approach. The objective is to be able to assess all organisations, not only those that have implemented CBD to a full extent, but also those organisations that are planning to adapt CBD in order to find out in what direction the organisation should move to get closer to full implementation of CBD and how to improve their process performance, which is the key to improving the software quality. Thus the assessment model can be used as an internal tool with the purpose of improving the organisation's processes as well as a tool to evaluate a component supplier.

With the exception of a few differences in naming and in the level of detail, the CBD assessment model is similar in the structure to the exemplar model ISO/IEC TR 15504-5. However, when it comes to the content there is a significant difference. This is a result of the fact that the assessment model is developed to suit CBD.

To validate the assessment model user trials will be conducted by assessing companies of different size, in different markets in countries across Europe. The experienced made from these user trials will then be used to subsequently update and refine the model.

The processes within the PRM are grouped into different process categories. Taken from the existing standard are the Customer-Supplier-, Support-, Management- and Organisational process categories. What has been replaced is the Engineering process category, due to the fact that this is where the major differences are found compared to the processes for general software development. The Engineering process category is replaced by the categories Modelling, Component Assembly and Component Provisioning. The processes within these categories are presented in the table below.

Table 1 Engineering process category

Modelling	**Application Assembly**	**Component Provisioning**
• Domain Engineering • Business Modelling • Requirements Engineering • Behaviour Specification Architecture • Provisioning Strategy • User Interface Specification	• Application Internal Design • Component Assembling • Application Testing • Application Delivery • Component Selection	• Component Internal Design • Component Testing • Component Delivery • Legacy Mining

3 Component-Based Development

It is our conviction that the component market will expand and the use and reuse of software components will increase significant.

"/.../ the benefits of CBD are real and are being demonstrated on real projects of significant size. CBD may be the most important paradigm shift in software development in decades /.../." (Clements, 2001)

However, the component approach is not, by it self, the solution to all problems. Just as before emphasis must be placed on the quality of the processes in order to deliver products that are able to meat the customers' expectations.

In CBD development, different actors experience different lifecycles. A component supplier experience a lifecycle where a component is to be developed often according to contracted requirements while a component consumer is focused on selecting the right components and assemble these into a larger application. (Christiansson,Jakobsson, 2001)

The same view is taken in the OOSPICE project and hence the categorisation of the processes that can be seen in table 1. In this section two of the processes in the Component Provisioning process category will be discussed as an example of what reasoning that has led to the emergence of the CBD assessment model.

4 Component Provisioning

In order to know what to look for and what to assess when a component is developed the specific processes and tasks need to be defined based on CBD best practices. Below two processes are described along with their CBD specific issues with relationship to the current ISO/IEC TR 15504.

The processes are taken from the component provisioning process category which aims to develop components providing commonly used business and/or data services across different systems or for use by third parties. (Allen, Frost, 1998).

4.1 Legacy Mining

Most processes defined for CBD make the implication that the component is developed from scratch. However, organisations are increasingly looking for effective ways of reuse their investments in existing packages, databases and legacy systems within the context of component technology. Unfortunately most processes offer little or no guidance in this area. (Allen, Frost, 1998). The Component Provisioning track must therefore take the existing software into account when defining the processes within that track, rather than ignoring it as a liability. In the OOSPICE assessment model information is provided about what to look for when assessing a process that aims to reuse and wraps up the legacy systems into components.

There are a number of tasks, which are called base practices in the OOSPICE assessment model, that have been found necessary for the completion of the Legacy Mining process. Thus, the assessment model needs to cover these by, as stated above, providing information needed to assess the completion of the base practices.

So what do we need to look for? The existing software assets must be analysed in order to identify the services provided by it. This provides a basis for identification of potential assets that can be wrapped into components. We also need to analyse how components are extract from the legacy systems and how the architecture is adjusted to the changes.

4.2 Component Internal Design

The Internal design of a component can be classes, interfaces of consumed components and functions or sub routines that are used. It is important to ensure that the internal constructs are adequately specified. Further, for each operation of each (programmable) interface of the component being developed, it is decided how the required behavior will be realised, in terms of operation invocations. Where the software being developed requires persistent data to be stored and retrieved a design for the data structure and data storage that provide the persistency is required. This leads to a transformation of the component specification from the specification environment to the implementation environment. (Lefever, 2001)

Based on the above the conclusion has been drawn that when assessing the Component Internal Design process one needs to ensure that the internal constructs are adequately specified. That is, the details in the internal construct should be defined together with the determination which internal and public operations operation should invoke, and the sequence of operation invocation. If the software being developed requires persistent data to be stored and retrieved the data store needs to be design and finally assess the transformation of the component specification from the specification environment to the implementation environment.

The OOSPICE assessment model therefore contains information about what to look for when assessing the Component Internal Design process both when it comes to what is actually performed in the form of base practices and how the inputs are transformed into outputs in form of work products.

When setting this in relation to the ISO/IEC TR 15504 the first thing that is obvious is that the terminology is different. This is an important issue when going into the CBD market since experiences have shown that companies that are about to start a software process improvement project are more likely to adopt a methodology that they can relate to both in content and in terminology. If we choose to look into the technical details of the standard the gap between CBD the standard become even larger. The major difference is the separation of the component specification and component implementation where the specification of the internal construct in a CBD environment is based on a component specification, whereas the software architecture design in the standard is based on the requirements. This separation makes it possible to select different provisioning strategies for different components belonging to the same application based on the component specification.

5 Conclusion

The way assessment is conducted when assessing a process or an organisational unit in a CBD-context doesn't differ from the way it is done when assessing general software development. What differs is the content and information to be found in the assessment model. When making a comparison between the standard and the way components are developed it is easy to understand that the component developers are less likely adopt the current standard when performing assessments. Instead a new methodology is needed that takes into account the CBD-specific issues that exists in a CBD-environment.

The OOSPICE project is bridging this gap by providing complete methodology for conducting assessments in a CBD environment. The methodology contains a new Process Reference Model and an associated Process Assessment Model that will be conformant with the new framework for process assessment in ISO/IEC 15504.

6 References

Allen, P. Frost, S. (1998) Component-Based Develoment for Enterprise Systems. Cambridge University. UK.

Christiansson, B. Jakobsson, L. (2001) Component-Based Software Development Life Cycles. [online] http://www.idt.mdh.se/ [read 2002-05-10]

Heinman, G.T., Councill, W.T. (2001) Component-Based Software Engineering – Putting the Pieces Together. Addison-Wesley.

Clements, P. (2001) From Subroutines to Subsystems: Component-Based Software Development. Published in: Component-Based Software Engineering – Putting the Pieces Together. Addison-Wesley.

Humphrey, W.S., (1989) Managing the Software Process, Addison-Wesley Reading Mass.

ISO/IEC TR 15504: 1998 - Software process assessment.ISO 1998

Lefever, Bruno. Computer Associates (2001). Input to OOSPICE: Industrial best practices.

Wallnau, K.C., Hissam, S.A., Seacord, R.C. (2002). Building Systems from Commercial Components. Addison-Wesley.

ISO/IEC 15504:200X Process Assessment (to be published in 2003)

7 Author CVs

Alec Dorling

Alec Dorling is the international project leader of SPICE (Software Process Improvement and Capability dEtermination) project and convener of the ISO group developing the ISO/IEC 15504 standard. He is an internationally recognized expert in the field of Software Quality Management and Process Improvement. He has previously held key posts the Centre for Software Engineering in Sweden, the European Software Institute in Spain and the National Computing Centre in the UK.

John Torgersson

John Torgersson is currently involved in the OOSPICE project where he is responsible for the development of an assessment methodology suitable for component-based software development. He is also working as a teacher in informatics and business administration at the University of Borås in Sweden.

8 Acknowledgement

The OOSPICE Project is developed as part of the EU-funded Shared-Cost RTD Project (IST-1999-29073).

The project partners are:

- Institute of Systems Sciences, Department of Systems Engineering and Automation, Kepler University Linz
- Huber Computer Datenverarbeitung GmbH
- Volvo Information Technology
- Griffith University
- Dataservice Informatik Ges.m.b.H.
- Centre for Object Technology and Application Research
- Computer Associates
- University of Borås

For more information visit: http://www.oospice.com

SP³E: a Method for Software Project Performance Evaluation

Vincenzo Ambriola & Giovanni A. Cignoni
Dipartimento di Informatica, Università di Pisa

Abstract

There are several initiatives for quality control in the supply of software and related services. In general, the most known and applied are the result of initiatives on the part of the customer: well defined acquisition processes, supplier selection based on certification or maturity assessment, contract monitoring. All these initiatives are aimed either to give confidence about the supplier capabilities or to directly control the development activities during their enactment.

On the side of the supplier, efficiency in the development of the software product is the most important issue, as it is the mean to be profitable. Evaluating the performance of a project has for the supplier the same importance that controlling the quality of the product has for the customer.

This paper presents a method for evaluating the performance of a software development project. The method implements a new approach that gives the supplier the ability to gather data about the performance of its projects while complying with the control initiatives of the customer.

Keywords

software acquisition, software process assessment, contract monitoring, project performance

1 Introduction

Without a well defined, enacted, and controlled development process it is almost impossible to *efficiently* deliver software quality. As product quality is the key factor for customer satisfaction and efficiency is the mean to be profitable, suppliers have strong and motivated arguments to control both quality and performance of their development projects.

On the customers side, the main concern is to receive in due time the required software system. Customers are not strictly involved in the economical aspects of the supplier's process: as far as the development process of the supplier is not related to product quality, the customer is not interested in its details, neither in the causes that ended in a delay or in low product quality. Customers want to play an active role in quality control, but their scope is limited to gaining confidence about the capabilities and the controllability of the supplier process during the enactment of a contract.

This commitment on the part of customers has been the motivation of several initiatives. Some of them are aimed to prevent risks, either by supporting the customer in the choice of the "best" supplier or by allowing the customer to manage a mature acquisition process. *Software Acquisition* (SA) methodologies directly aim to provide guidelines and tools to organize and control the customer acquisition process. The success of *Software Process Assessment* (SPA) methodologies is largely due to the "certification" they provide about the capabilities of the assessed software company. Customers can use this certification in the selection phase, both for identifying the "best" supplier and for verifying the controllability of its development process.

There are other initiatives that support the customer during the project enactment for strictly and directly controlling the supplier activities. An example of this kind of initiatives is *Contract Monitoring* (CM), as defined by acquisition methodologies or, as a particular case, by the rules that, by law, are applied to Information Technology (IT) contracts of Italian public bodies. Even in this case the initiative is on the part of the customer.

In order to comply with the control initiatives of their customers, suppliers must dedicate some noticeable effort to demonstrate their capabilities with SPA and to support the requirements of the CM control activities.

This paper presents a method for *Software Project Performance Assessment* (SPPA). SPPA is a new approach in the control of software projects. Presented as an initiative of the supplier, SPPA aims at capitalizing the effort spent by the supplier to comply with the control initiatives of the customer. The idea of SPPA is to build up a set of activities, performed by the supplier, whose results:

- can be presented to the customer to fulfill its need of confidence in the capabilities of the supplier and in the correct management and progress of a specific project;
- can be used by the supplier to assess its performance during the project, providing means to evaluate the efficiency of its development process.

Section 2 of the paper presents the general context of the current control initiatives, briefly describing those from the part of the customer. Section 3 introduces the general goals of SPPA. Section 4 describes the details of *SP^{3}E*, our proposed method for SPPA. Section 5 presents some examples from the first experimentations with the SP^{3}E method. Conclusions and future work are in Section 5.

2 Control initiatives on the part of the customer

In this section we give an overview of three control initiatives that are on the part of the customer: software acquisition process, software process assessment, and contract monitoring.

2.1 Software acquisition process

One of the most critical activities performed by large organizations is the acquisition of goods and services. The acquisition of software products (SA in short) is one of the many instances of this process and it is now considered as part of the software process. The *ISO/IEC 12207* [1] standard includes SA as the initial *Primary* process, the one that starts the software life-cycle. In the same way, the software process model defined in *SPICE* [2], and now part of the *ISO/IEC TR 15504* [3] standard proposal, identifies SA as the first process listed in the *Customer-Supplier* category.

The *Capability Maturity Model* (CMM) [4] principles were applied to the SA process itself. The result is SA-CMM [5], an assessment methodology that can be used to evaluate or improve the SA process of an organization, or as a set of guidelines that can help in the definition of a mature SA process.

In Europe, since the mid-eighties the SA process was considered of crucial importance. The European Commission sponsored the *Public Procurement Group* (PPG) to define the technical rules for the acquisition of IT product and services. The most important project carried out by PPG is *Euromethod* [6], aimed to investigate on the relationship between customers and suppliers, to define a set of common concepts, and to find an agreement on a common terminology.

2.2 Software process assessment

Process assessment is the evaluation, performed according to a well-defined framework, of the development process of a software company. The term *Software Process Assessment and Improvement* puts together the goals that drive evaluation: definition of the process quality level, identification of the improvement strategies.

The most widely known SPA model, the already cited CMM, was developed as a tool for classification of suppliers. In the CMM perspective, the evolution of the software process is a path of five steps that identifies the maturity of a software company and the actions needed to move from one level to the higher one. In this sense, not only CMM is a classification tool but also a concrete methodology for process improvement.

The ISO 9000 series of standards is a commonly accepted reference that, in some market areas, has been elevated to the rank of mandatory requirement for having access to bidding procedures. Although ISO 9001 [7] is a general series of standards that address the quality system instead of the software development process, it shares many aspects with SPA [8]. For instance, the *Bootstrap* project [9] merged in a single SPA methodology the CMM approach and the ISO 9001 requirements for quality system management.

The *SPICE* (Software Process Improvement and Capability dEtermination) project aimed to establish a framework in which different assessment techniques can express process capabilities with respect to a common scale and to common assessment criteria. With ISO/IEC as a partner in SPICE, the development of project results is currently published as a ISO/IEC TR 15504.

2.3 Contract monitoring

Monitoring (CM in short) is a quality control activity performed during the enactment of a contract. As part of the SA process, CM is defined in several contexts. ISO/IEC TR 15504 defines “CUS 1.3 Supplier monitoring” as a component of the “CUS 1 Acquisition” process. In ISO/IEC 12207 monitoring is an activity of the SA, listed as a primary process. Monitoring the supplier during the contract is an activity defined by Euromethod too. ISO 9001 standard requires the Quality Plans, as documents that support the monitoring activities of the customer.

In Italy, SA procedures to be followed by public bodies are ruled by specific laws that prescribe CM and describe this activity in details [10, 11]. The *Authority for Information Technology in the Public Bodies* (AIPA, using its original acronym for “Autorità per l’Informatica nella Pubblica Amministrazione” [12]) defined CM with respect to both methodological and normative issues.

According to the AIPA definition, the aim of CM is to control specific facets of the contract enactment, which, though bound to the quality of the final product, are not directly related to the phases of the development process. Four “specialized” facets of CM have been identified: supplier process monitoring, project management monitoring, product quality monitoring, investment benefit monitoring.

Acceptance test is the traditional control that the customer applies to the product when it is ready to be delivered. The acceptance test, however, cannot face all the risks the customer is exposed to: it cannot face delays in the delivery, and, even worse, when it discovers defects in the product quality, acceptance test introduces further delays for their removal. The cost of these delays, even in terms of lost revenues, can overcome the whole cost of the software system. In this perspective, CM represents a customer initiative that neither replaces nor overlaps with acceptance test, but aims to directly control the supplier during the development of a required product or the provision of a service. CM represents the will of the customer to actively participate to the development process of a required and crucial product [13].

3 Software Project Performance Assessment

The importance of fully controlling the acquisition of software products and services is becoming more and more evident to customers. Clear signs of this trend are the initiatives aimed at defining, controlling, and monitoring the acquisition process. In particular, contract monitoring is an even stronger element that gives evidence of an active presence “inside” the supplier development process.

According to the 12207 standard, the software acquisition process is a starting point of the whole lifecycle. The introduction of CM as a relevant component of SA makes obsolete this position: SA is no more that runs before but a process that runs in parallel with respect to the development process. For the manager of a software development process, SA has become the reference counterpart.

The clever supplier gives paramount importance to the wishes of its customers. Following this simple and widely accepted paradigm, software suppliers must adapt to and fulfill the customer requests. In our particular context, SPA procedures are a convenient strategy that suppliers can follow to give evidence of their reached maturity level and of their capability of satisfying the commitment prescribed by contractual bindings. In order to give evidence of the capabilities of properly conducting projects and of fulfilling the deadlines contractually stated, suppliers must also accept the strict control regime imposed by CM.

Customer maturity in general requires higher costs for the suppliers. These costs are, however, balanced by the implicit advantages that the supplier can obtain from these additional constraints. The primary interest for the supplier is to improve the process efficiency, then efficiency can be exploited to obtain higher marginal financial benefits, and/or to release precious resources.

Software project performance assessment is an initiative that gives to the supplier the ability to observe and evaluate the performance of a project and, hence, the efficiency effectively exhibited during their enactment.

4 The SP^{3}E method

SP^{3}E, which stands for Software Process and Project Performance Evaluation, is a method for the performance assessment of software projects. The SP^{3}E assessment technique is partially derived from the rapid assessment methodology developed in the ESPRIT/TOPS project [14].
In SP^{3}E performance is defined in terms of efficiency: the SP^{3}E method measures it with respect to costs and product dimensions. Other parameters can be considered part of a performance definition, for instance the quality of the product. But, in a supply context, we can assume that the product will eventually pass the acceptance test. In this case, since the quality goal is accomplished it is more important to evaluate its costs, for instance because the first acceptance test has failed and the supplier must put extra effort in debugging the software. In other words, bad product quality is the cause of poor overall performance and not another kind of poor performance.

4.1 SP^{3}E activities

From a practical point of view, the application of the SP^{3}E method on a given software project consists in three distinct activities: *process assessment*, *project monitoring*, and *performance evaluation*.

Process Assessment. This activity is performed at the very beginning of the software project. Its general goals are to acquire a detailed knowledge of the development process and to assure its controllability. To evaluate performance we need to keep track of schedule, product dimension and costs. Technically speaking, this means that management activities and some development activities have to be carried out at level 4. It is important to note that, during the process assessment, it is possible to discover some mandatory improvements. The assessment result is the certification that level 4 is in place, or the identification of the needed changes to bring the process at level 4. The practical result of the assessment is the “map” of the process adopted in the project. This map will be used as a guide to monitor the project activities. The assessment performed is compliant to ISO/IEC TR 15504. Maturity evaluation and general improvement suggestions may be welcome “side results” of the assessment.

Project Monitoring. This activity is performed during project enactment. Its general goals are to keep track of the changes to the process and to measure, for each activity, the effort and the dimension of the developed products. The need of a monitoring activity is to assure that all changes to the process are tracked. Often, the process is defined and managed, but changes are not documented, especially those deriving from unexpected problems. In a performance assessment this means the presence of hidden effort. In other cases, there are changes that do not produce a performance loss, even if they have a negative impact on the project. For instance, an activity can be delayed because of resource shortage, as far as penalties are not applied, costs and revenues do not change. As an additional result, data from the project monitoring activity can be presented to the customer to provide evidence of the good management of the project.

Performance Evaluation. This activity is generally performed at the end of the project. It consists of the analysis of the data collected during monitoring to evaluate the overall performance of the project. Punctual performance can be evaluated also during the enactment of the project, but we have to wait until the very end of the project to have a stable configuration of data. The data can be analyzed and compared with reference models.

SP^{3}E activities should be carried out by professionals that are independent from the supplier company. Apart of the performance evaluation, this can also help if results are presented to the customer, for instance in the context of CM. SP^{3}E activities are supported by forms that help in collecting data about the project:

- *project form*, collects data about the project, the people involved, the available documentation, and the product to be delivered;
- *work-package form*; collects data about a single work-package: description, schedule, effort estimated and actual; the first configuration of the set of work-package forms is generated during the process assessment and is constantly updated during the project monitoring activity; process attributes of the single work-package are verified during the process assessment to assure the controllability of the work package;
- *component form*; collects data about a single software component: description, related work-packages, dimension; the set of component forms is generated and constantly updated during the project monitoring activity.

The current version of templates for SP^{3}E forms and the guidelines for their compilation are available at [15]. As sketched in fig. 1, SP^{3}E activities are organized in a way that fruitfully exploits activities that the supplier must anyhow perform in order to satisfy the new requests of the customer, in particular the CM activities performed inside the SA process.

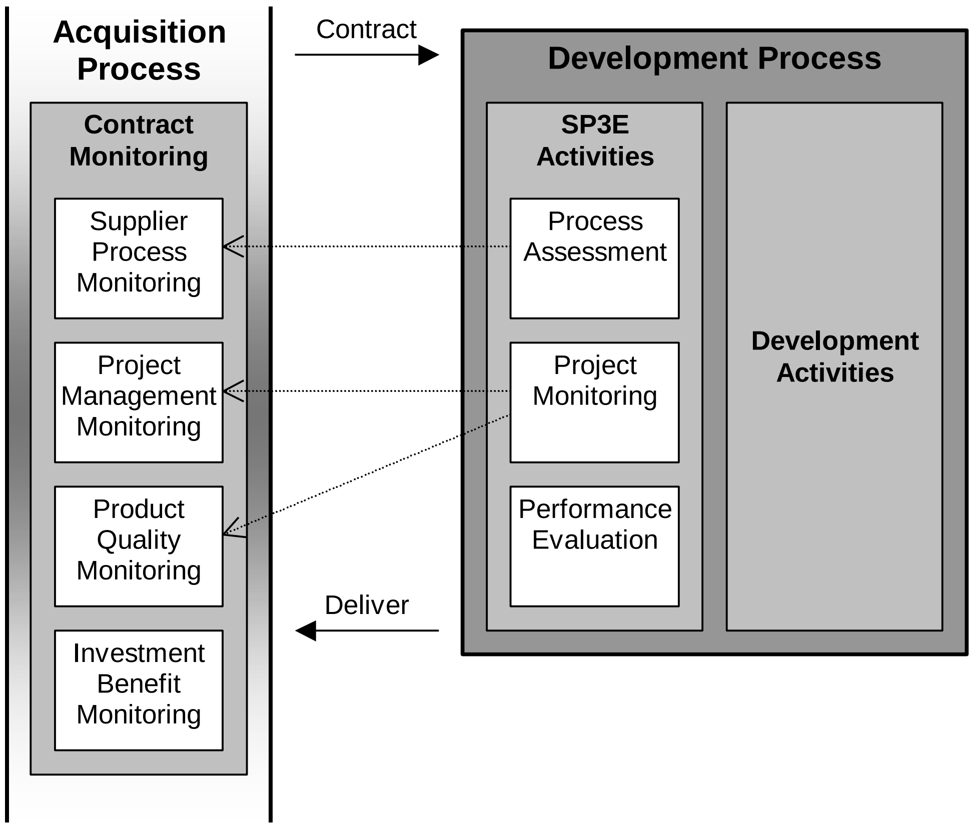


Fig. 1. Relations between SP^{3}E activities and customer control activities

4.2 SP3E results

The result of SP^{3}E activities can be exploited to derive two kinds of performance evaluation of the assessed project: *quantitative* performance evaluation, as a direct measure of the project productivity, and *qualitative* performance evaluation, as a set of considerations that can help the project management to identify improvements suggestion for their development process.

Quantitative Performance Evaluation. In SP^{3}E we focus on efficiency as the most important facet of project performance. Efficiency can be measured as a ratio between product size and effort or between product size and costs. In SP^{3}E product size is measured using *Object Points* as defined in the proposal of the CoCoMo 2 model [16]. Effort is measured in man/month, as usual. The size/effort measure, however, is a rough approximation of the efficiency because there are substantial differences among man/month costs of differently skilled personnel. For this reason SP^{3}E measures the industrial cost of each activities and produces the size/cost ratio as a measure of the efficiency.

Qualitative Performance Evaluation. Apart of performance measures, SP^{3}E provides several "maps" that can be useful in the study of the project performance. Basically, SP^{3}E maps are coloured Gantt charts. We use Gantt charts as basic representations of project activities. Then we use different colours to highlight activities that differ with respect to a characteristic that influences performance. For each characteristic SP^{3}E provides a map, the different colours describe how the activities are differently positioned with respect to that characteristic. At the moment we have identified three characteristics: *effort cost*, to highlight cheap and expensive activities, *size/time*, to highlight the activities with respect of time to deliver, *tools dependence*, to highlight the activities with respect to the different development tools used for their enactment.

Quantitative performance evaluation is proposed mainly as a benchmark tool. In its first approximation, size/effort, the performance result can be compared with the "standard" value that can be obtained applying the CoCoMo 2 model to the project. The size/cost result can be compared with data from other projects of the same company.

Qualitative performance maps are proposed as an investigation tool. A typical way to study the maps is to compare them and to search correlations among characteristics. For instance, it is possible to "see" if the introduction of a development tool was useful with respect to time to deliver or to cost reduction.

5 Experimentation with the method

The SP^{3}E method is yet in development. Several trials were conducted to experiment with the method but, currently, no complete application of the method in the industry has been done. The main motivation is that, in the trials we did, the initial assessment showed too poor maturity levels. If the process is not improved, it is impossible to keep track of activities and costs to evaluate the performance. As it is too expensive to change the process, the experiment terminated after the first phase. Even if this is quite frustrating for us, it is indeed a result for the company. Performance evaluation seems to be a good lever for convincing companies to perform an initial process assessment. The proof of evidence that they are not able to have a reliable assessment of their performance is a strong motivation to start an improvement path.

Having failed our first attempts to experiment the method in an industrial setting we directed our efforts to experiment the method with students involved in software engineering projects. In the course of *Tecniche di Programmazione* (Programming Techniques, TP), held at the University of Florence, students are grouped in development teams that act like small software companies. The goal of the course, that in the curriculum follows a classic Software Engineering course, is to apply in a practical project both development techniques (analysis and design using UML, testing) and management techniques (cost estimation, project planning and control).

In the simulation proposed by the course there is a customer that asks for a software application. Teams must perform requirement analysis, present a formal offer that includes a price, and deliver the application in the given time. An important constraint imposed to teams is to stay in the cost estimation from which they derived the offer price. This year there were six teams, grouped in three consortiums. Teams were required to define their process and to consistently plan and control it. To assure that cost are estimated, activity planned and controlled, delays tracked and cost accounted, teams were subjected to three formal inspections, whose results influenced the final score for the course. Before each inspection, teams were required to consolidate activities and costs and, if necessary, to update their project plans.

In the project, according to the performed activities and their declared skills, students can play several roles and each role has a different cost. For instance, a student who already had followed a specialized course in Java was considered a senior Java programmer, while a student who had only followed a general programming course was considered a junior programmer. The same rules applies to other skills participating to the project (Java, SQL and UML) and to activities that involve more responsibility (such as analysis, design, and management). The student teams had to manage these issues as if they were in a true company. Their plans were required to keep track of all the changes about duration effort and responsibility (as in a "mature" company).

From the performance assessment point of view, the student teams, while not a real software company, acted perfectly because their process was well defined and data were consistently kept. In this perspective their projects were subjected to a strict monitoring activity and, hence, well suitable for experimenting the SP^{3}E method. In the following we show an example of an SP^{3}E map drawn using the data of two student teams and we use these maps to discuss their performance.

Figure 2 shows two effort cost maps. They present the high level activities of two teams (analysis and offer proposal, design and coding, integration and verification) and indicate how the average cost of activities changed during the project. The upper bar represents initial planning, the lower bar represents actual results. The length of the bars is proportional to effort (not time of deliver but how many man/hours were totally consumed). The number inside the bars is the average cost per hour (the same information is given visually by the bar colour: light means cheap, dark expensive).

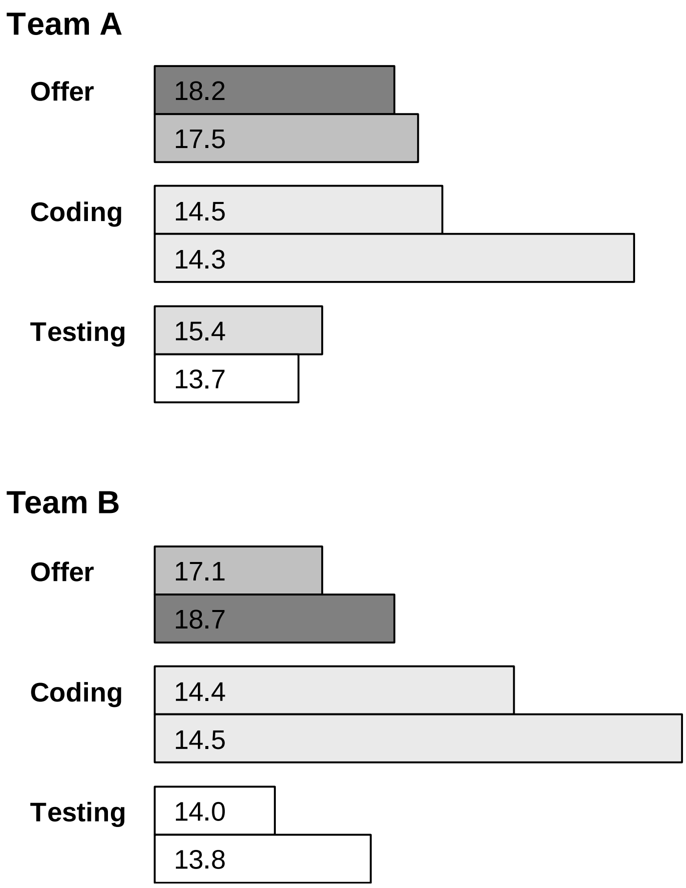


Fig. 2. Effort cost maps for the performance of two student teams.

Using the maps of Figure 2 it is possible to evaluate the performance of the two teams with respect to role management. Both teams faced an increased effort with respect to initial planning. Team A was very efficient in managing the extra required effort without too much increasing the overall costs. In fact, the average cost per hour decreased. Going into more details (using the tracked management data) we notice that, for instance, in the offer phase, when they realized that the number of meetings with the customer was rapidly increasing they decided to use lower skilled people for collecting information and for writing the initial version of the analysis documents, saving the time of skilled and more expensive people for the last revision of the documents. A similar strategy was used for testing, where, to stay in the budget, they made the most relevant cut deciding for a naïve verification rather than designed test suites when they were more confident that a component was well coded. Even if the purpose of the method is not to compare, but rather to investigate and discuss the efficiency of a project, in this example it is clear that Team A exhibits a more efficient resource management.

6 Conclusions and future work

In this paper we identify a new activity in the context of quality control of software project: SPPA. This activity is organized to fruitfully exploit the effort that the supplier must anyhow perform in order to satisfy other control initiatives requested by the customer: SPA and CM. We also propose a method, SP^{3}E, for performing SPPA. The method is briefly presented in general and example of its use is discussed as a mean to evaluate the performance of two development teams with respect to role management.

Our method is focused on efficiency, as the main indicator of performance. However, efficiency is not the only performance indicator that we deem important for SPPA. Other indicators, when combined with efficiency, can give a more precise and complete picture of the performance of a software development project. For instance, we have in mind to use the competence level of the personnel associated to software projects and the capability of compressing activities along the time dimension, both by using at best the available technological tools and by increasing the automation level of the development process. We are fully aware that the research is still at an initial stage but we actively plan to validate the SP^3E method by means of experimentation on real industrial projects.

The definition of a method for evaluating software project performance was partially supported by the *Estate* project [15], funded in the context of the *Lyee International Collaborative Research Project*. Lyee [17] is a software development methodology based on a specific analysis method and a tool for automatic code generation. In the Estate project the SP^3E method will be used to assess the performance of the projects carried out using Lyee.

7 Acknowledgements

The affiliation of G.A. Cignoni is with the University of Pisa, because the research about SPPA was developed there, but the reported data came from a course held by him at the University of Florence. We want here to tanks the students of the course of Tecniche di Programmazione.

8 Literature

Some of the listed references cite Italian laws: [10] is the Italian law where monitoring was defined for the first time; [11] is the first edition of the official guidelines for monitoring.

[1] International Organisation for Standardisation, “ISO/IEC 12207:1995 Information Technology – Software lifecycle processes”, 1995.

[2] SPICE Document Suite, available at http://www-sqi.cit.gu.edu.au/spice, accessed March 2002.

[3] International Organisation for Standardisation, “ISO/IEC TR 15504:1998 Information technology – Software process assessment -– Parts 1-9”, 1998.

[4] W.S. Humphrey, *Managing the software process*, Addison-Wesley, 1989.

[5] J. Cooper, M. Fisher, S.W. Sherer (Eds.), “Software Acquisition Capability Maturity Model”, Technical Report CMU/SEI-99-TR-002, 1999.

[6] Euromethod Version 1, available at http://www.esi.es/Euromethod/, accessed March 2002.

[7] International Organisation for Standardisation, “ISO 9001:2000 – Quality management systems – Requirements”, 2000.

[8] M.C. Paulk, “How ISO 9001 compares with the CMM”, *IEEE Software*, vol. 12, no. 1, January 1995.

[9] P. Kuvaja et al., *Software process assessment and improvement: the Bootstrap approach*, Balkwell Business Publishers, 1994.

[10] Decreto legislativo n. 39 del 12 febbraio 1993, *Norme in materia di sistemi informativi automatizzati delle Amministrazioni Pubbliche, a norma dell’art. 2, comma 1, lettera m, della legge n. 421 del 23 ottobre 1992*.

[11] Circolare n. 5/AIPA del 5 agosto 1994, *Art. 13, comma 2 del decreto legislativo n. 39/1993: monitoraggio dei contratti di grande rilievo relativi a progettazione, realizzazione, manutenzione, gestione e conduzione operativa dei sistemi informativi automatizzati: criteri e modalità*.

[12] AIPA Web site, http://www.aipa.it/, accessed March 2002.

[13] V. Ambriola, G.A. Cignoni, “Monitoring the software process: a solution from the customer point of view”, *Proc. of the Int. Conf. on Software Quality Engineering (SQE ’97)*, Udine, Italy, May 1997.

[14] G. Bucci, M. Campanai, G.A. Cignoni, “Rapid Assessment to Solicit Process Improvement in Small and Medium Organizations”, *Software Quality Professional*, vol. 4 n. 1, December 2001.

[15] Estate Project Web site, http://www.di.unipi.it/estate/, accessed March 2002.

[16] B.W. Boehm et al., “Software Cost Estimation with Cocomo II”, Prentice Hall, 2000.

[17] Lyee Web site, http://www.lyee.co.jp/, accessed March 2002.

9 Author CVs

Vincenzo Ambriola

Vincenzo Ambriola is associate professor of Software Engineering at the Department of Computer Science of the University of Pisa. Author of more than fifty scientific publications on international journals and conferences, he is currently in the editorial board of Software Process Improvement and Practice. Vincenzo Ambriola has been member of the program committee of more than twenty international conferences on Software Engineering (European Conference on Software Engineering, Software Engineering and Knowledge Engineering, Software Maintenance and Reengineering, among the others). He has been principal investigator of numerous research projects sponsored by the Italian industry as well as the National Council of research. His main research interests are in Software Engineering and, in particular, in the area of Requirements Engineering and Software Design.

Giovanni A Cignoni

Giovanni A. Cignoni received in 1992 the Laurea degree in Computer Science from the University of Pisa. From 1993 he is a freelance consultant in Information Technology, with particular expertise in the fields of software engineering, software quality, and software process management. As consultant of the Dept. of Computer Science and of the Dept. of Mathematics of the University of Pisa, he participated to Italian and European research projects. His main research interests are in Software Engineering and in particular in the area of management and control of the software development process. As contract professor he held courses about software engineering at the Universities of Pisa, Florence and Padova. He is author of international publications and co-author of the books “Laboratorio di programmazione” (in Italian, “Programming workbench”) and “Il test e la qualità del software” (in Italian, “Test and software quality”).

The Benefit of BOOTSTRAP Assessments

Dipl.-Kfm Peter Bölter, Director BOOTSTRAP Institute

Abstract

The BOOTSTRAP Institute (BI) is operating an independent institute to develop and maintain SPI methods, license it to users and service providers and run a benchmarking database since almost 20 years. The BI was involved in the SPICE project and was representing the DIN in the final discussion. The database includes more than 500 assessment results compliant to ISO 15504 (SPICE). Those results contain maturity metrics and information about investment and archived benefits. SQS is European market leader in consultancy of IT quality management and testing. Since 1995 SQS is full member of BI and has provided SPI services based on BOOTSTRAP and on other methods (CMM, CMMI, SPICE). Today BI and SQS run a project to analyse these data, compare it to existing publications and prove the benefit of SPI.

The BI presents these results to participating experts and discuss the figures. As those information can usually only be accessed by members of BI, this will be one of very few occasions for public audience.

Keywords

SPI, Assessment, Benchmark, Experience Report, ROI, Cost-Benefit-Analysis; Training, Certification of Professionals

Summary

Looking on 20 years of experience, many improvement projects were performed, because involved people thought the improvement was needed. But today's economic situation doesn't allow investments anymore if the return is not estimated and profitable. This change in the market requirement contains a new challenge for SPI methods.

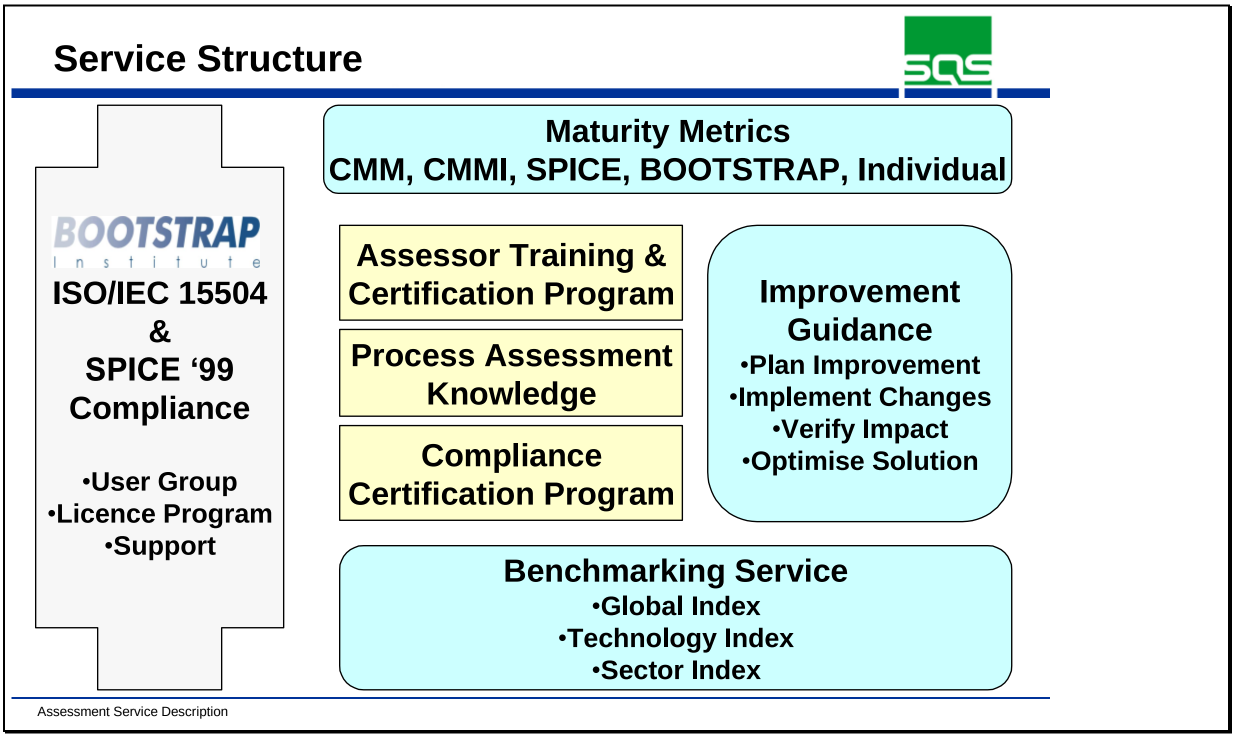
Therefore planning an assessment and improvement process must include expected benefit. Tangible and intangible need to be specified and controlled afterwards..

Summarising the experience from BOOTSTRAP results, the mean business value of an assessment and improvement process is about 6/1

These figures are verified against studies and available publications from other similar approaches.

Introduction to BOOTSTRAP

BOOTSTRAP covers assessment, improvement and benchmarking services of the IT sector including hardware, software and service.



BOOTSTRAP is an ISO 15504 compliant assessment and improvement method providing:

- BOOTSTRAP Model including
  - Process Dimension (ISO 15504 compliant)
  - Capability Dimension (ISO 15504 compliant)
  - Technology Dimension (Unique)

- Assessment Process (ISO 15504 compliant)
- Improvement Process
- Benchmarking
- Assessor training and accreditation (ISO 15504 and SPICE 4.0 compliant)
- Lead Assessor training and accreditation (ISO 15504 and SPICE 4.0 compliant)
- Trainer accreditation (ISO 15504 and SPICE 4.0 compliant)
- Events pp.
- Maintenance
- Actual available version: BOOTSTRAP 3.2 (release 4.0 planned 11/2002)

To optimise assessment costs due to available budget and assessment needs, BOOTSTRAP assessments can be carried out in the following manners

- Impact
  - Guided Assessments: Guided Assessments follow the whole conduct of code, controlled by a BOOTSTRAP lead assessor, conducted by an assessor team. Guided Assessments should be used to determine the capability baseline of great or medium organisations.
  - Workshop Assessments (Tapistry): Workshop Assessments use the Tapistry approach to combine one day Workshop with self assessments. Tapistry is useful for small or medium organisations or small SPU in large organisations. It was successfully applied in large organisation with low maturity as well. Th workshop needs an experienced trainer.
  - Self Assessments: Self Assessments are performed without an assessor or trainer, using the tool BootCheck and the Self Assessment Guide.
- Tailoring
  - Full Assessment: In a full assessment all processes are assessed. Full Assessments are used to define a capability baseline in big organisations.
  - Focussed Assessment: Focussed Assessments use only a subset of the BOOTSTRAP Model. Subset may be defined due to the business or technical needs of an organisation or an SPU within a great organisation.
  - Specialised Assessments: Specialised Assessments use the toolbox architecture of ISO 15504 and BOOTSTRAP to design standard compliant assessments focussing on special issues as test, configuration management or acquisition.

Assessment Costs and Benefit

To make sure that business goals of an assessment and improvement process are reached, cost benefit analysis has to be performed. Analysis is done in two phases - one performed before the assessment and one after the assessment.

1.1 Cost planning and analysis

Before the assessment is contracted, costs have to be estimated. Estimation depends on what is to be done and what is to be delivered.

The figure below shows the cost for a guided full assessment in a big organisation:

resource table							
BOOTSTRAP Assessment				*effort per No./Assessor [h]*		*sum [h]*	
		No.	*No. of assessors*	*from*	*to*	*from*	*to*
1	Defining Assessment goals and detailed assessment planing	1	1	3,00	8,00	3 [h]	8 [h]
2	Kick-off-Workshop	1	1	3,00	6,00	3 [h]	6 [h]
3	documentation analysis QM-System (SPU/Projects)	1	1	8,00	16,00	8 [h]	16 [h]
4	Conducting the assessmnet interview s					0 [h]	0 [h]
4.1	SPU	1	2	12,00	14,00	24 [h]	28 [h]
4.2	Project A	1	2	12,00	16,00	24 [h]	32 [h]
4.3	Project B	1	2	8,00	10,00	16 [h]	20 [h]
4.4	Project C	1	2	8,00	10,00	16 [h]	20 [h]
4.5	Project D	1	2	6,00	8,00	12 [h]	16 [h]
4.6	Project E	1	2	6,00	8,00	12 [h]	16 [h]
4.7	Project F	1	2	4,00	6,00	8 [h]	12 [h]
5	preparing assessment reports (seperate for each project)	7	1	8,00	12,00	56 [h]	84 [h]
6	Benchmarking against BOOTSTRAP database	1	1	4,00	8,00	4 [h]	8 [h]
7	preparing assessment results presentation (seperate for each	7	1	8,00	12,00	56 [h]	84 [h]
8	presentation assessment results and improvement suggestions	1	2	2,00	4,00	4 [h]	8 [h]
9	preparing final assessment report	1	1	16,00	24,00	16 [h]	24 [h]
10	FeedBack session for internal assessors	1	1	1,00	2,00	1 [h]	2 [h]
11	preparation and conducting improvement w orkshop	1	1	16,00	24,00	16 [h]	24 [h]
12	final presentation	1	2	2,00	4,00	4 [h]	8 [h]
sum (hours)						**283 [h]**	**416 [h]**
sum (days)						**35 [PT]**	**52 [PT]**

Figure 1: Cost calculation for a Guided Full Assessment. SPU processes are assessed detailed at SPU level and global at project level, project processes are assessed detailed at two projects and global in another project.

A typical effort calculation of a Guided Full assessment as shown, calculates 35 PD of effort of the assessors. Internal technical and management stuff is not included, but is needed for the following:

- Participation in meetings
- Participation in interview sessions
- Providing documents and additional information

Effort and costs can be reduced through

- Reducing the number of assessed projects
- Reducing intensity of assessment from a detailed to a global approach or focussing on processes

When the assessment report is delivered and accepted, improvement planning takes place. Improvements are prioritised in a SWOT workshop and action planning and cost effort estimation is to be carried out.

Calculation for each improvement action is done by the same scheme:

Budgeting - 2

Resources	Costs
1. Staff	
2. Consulting services	
3. Training	
4. Travelling	
5. Software	
6. Hardware	
7. Other Costs for the Project	
Total	

Process Improvement Slide 37

1.2 Benefit

Standard ROI analysis groups benefit in

- Tangible benefit
- Intangible benefit

1.3 Tangible Benefit

Speaking over tangible benefits, on the one hand it easy to find reports and case study which prove the existence of high benefit values, on the other hand many managers require to know the benefits without having the data to calculate it.

1.3.1 Case studies and reports for tangible benefit

Benefit analysis of Software process assessment and improvement started in the mid of the 1990 th several case studies were developed

[Herbsleb, Carleton, Rozum, Siegel, Zubrow Benefits of CMM-Based Software Process Improvement: Initial Results]

Their results were

- It is possible to proof net tangible benefits
- It is difficult to proof cause measurement data must made be available.

So most of the case studies use predefined metrics to produce reasonable figures.

The overall result is

Category	Range	Median
Total yearly cost of SPI activities	\$49,000 - \$1,202,000	\$245.000
Years engaged in SPI	1-9	3,5
Cost of SPI per software engineer	\$490-\$2004	\$1375
Productivity gain per year	9%-67%	35%
Early detection gain per year (defects discovered pre-test)	6% - 25%	19%
Yearly reduction time to market	15%-23%	19%
Yearly reduction in post release defect reports	10%-94%	39%
Business value of investment in SPI (value returned on each \$ invested	4.0-8.8	5.0

This figure was very common to the software process assessment and improvement community, when in May 2002 US National Institute of Standards and Technology Program Office Strategic Planning and economic analysis group issued a planning report 02-03

"The economic Impacts of inadequate Infrastructure for software testing"

The study investigated the cost impacts of due to an inadequate testing infrastructure in the transportation manufacturing sector and the financial sector.

Its results were

- The costs of inadequate software testing infrastructure in the sectors raise up to **<u>$ 5.182,7 Mio</u>**
- The potential cost reductions from feasible infrastructure improvements raise up to **<u>$ 2.001,8 Mio</u>**

There are no signs that testing in Europe is better than in the USA. So the potential cost reduction from feasible infrastructure improvements is as large as in the USA means **<u>$2000 Mio</u>**.

Inadequate testing infrastructure is a consequence of inadequate test processes. Areas of improvement must be defined and prioritised to make sure that investments cause the highest value of benefits. To meet this need, a clear idea of the actual status of software test processes must be stated.

BOOTSTAP allows to perform a specialised assessment for test processes, creating a first step towards to save $2000 Mio in Europe.

1.3.2 Problems of tangible benefit analysis

Coming down from the whole picture to the concrete enterprise it is difficult to calculate the concrete benefit for a concrete enterprise. Why? If you want to calculate a benefit you must have a picture of the current status. That means to have metrics which show the current status of costs and income. Metrics are the main topic of BOOTSTRAP Level 4. What does that mean?

Capability Levels of BOOTSTRAP

Level	Characteristics	Most important difference to preceding level	Result
5 Optimising	• changes to the definition, management and performance of the process are controlled to achieve the business goals • changes to the process are identified and implemented to ensure continuous improvement in fulfilment of the business goals	The predictable process optimises its performance to meet current and future needs and achieves repeatability	Productivity &Quality
4 Predictable	• process is supported by defined goals and measures which are used to ensure the achievement of overall process goals • process is controlled through the collection and analysis of measures to control and correct the performance of the process to reliably achieve the defined process goals	The established process is performed consistently within defined control limits to achieve its goals	
3 Established	• process uses a process definition based upon the standard process in order to contribute to the defined organisational business goals • process effectively utilises skilled human resources and infrastructure to contribute to the defined organisational business goals	The managed process performs using a defined process based upon good software engineering principles	

2 Managed	• management of the execution of the process to produce work products within stated time and resource requirements
	• work products
	  • are documented and controlled
	  • meet their functional and non-functional requirements
	  • are in line with the product quality goals
		The performed process delivers work products of acceptable quality within defined time scales and resource needs	

1 Performed	• utilisation of a set of practices that are initiated and followed • identified input work product • identifiable output work products • satisfies the purpose of the process	The implemented process achieves its defined purpose	Risk

BOOTSTRAP The method Slide 8

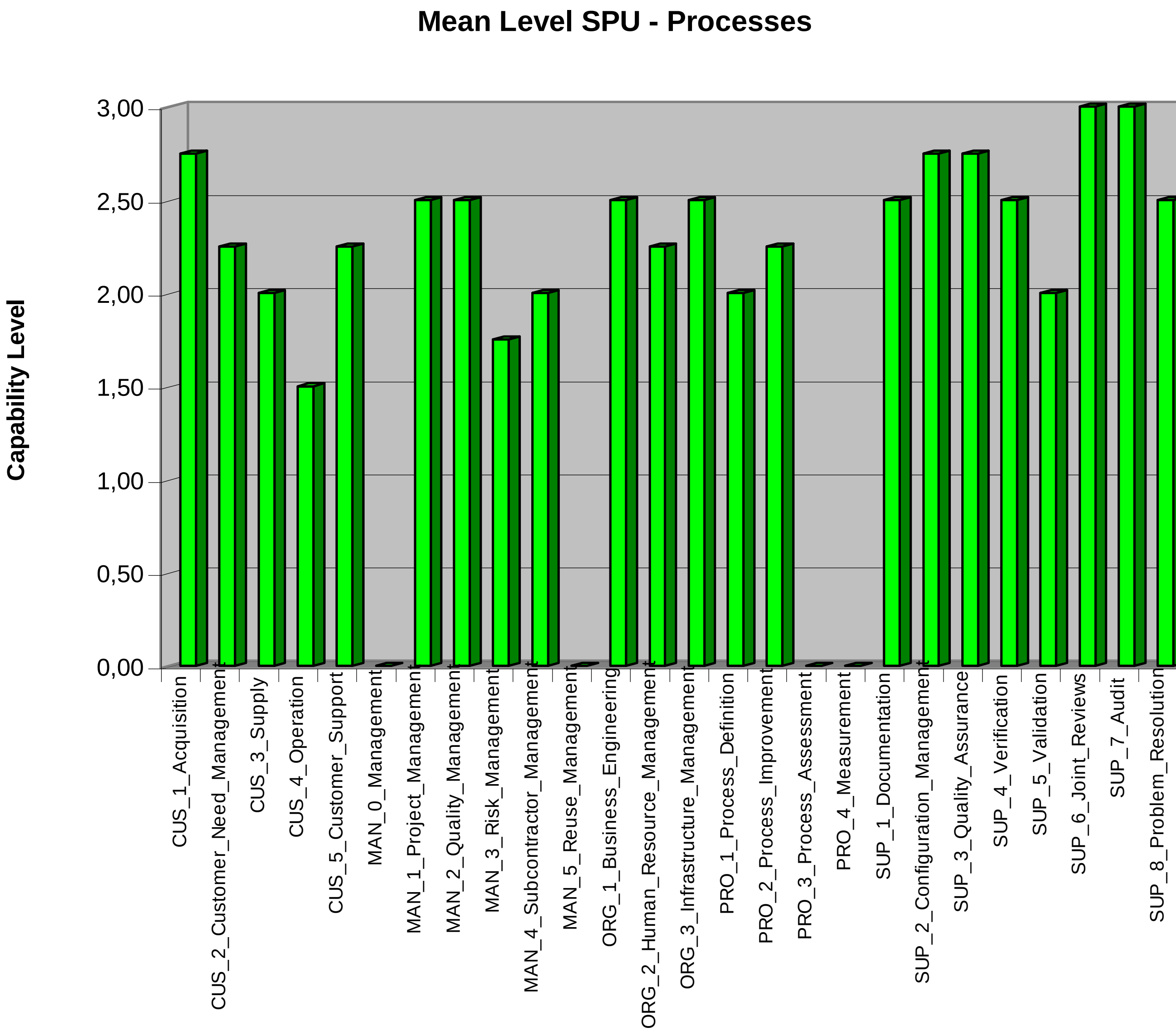
BOOTSTRAP Institute

A level 4 capability for a process states that this process is performed, managed, established and quantitatively controlled.

If you want to calculate tangible benefits, you have just to be where you want to go to.

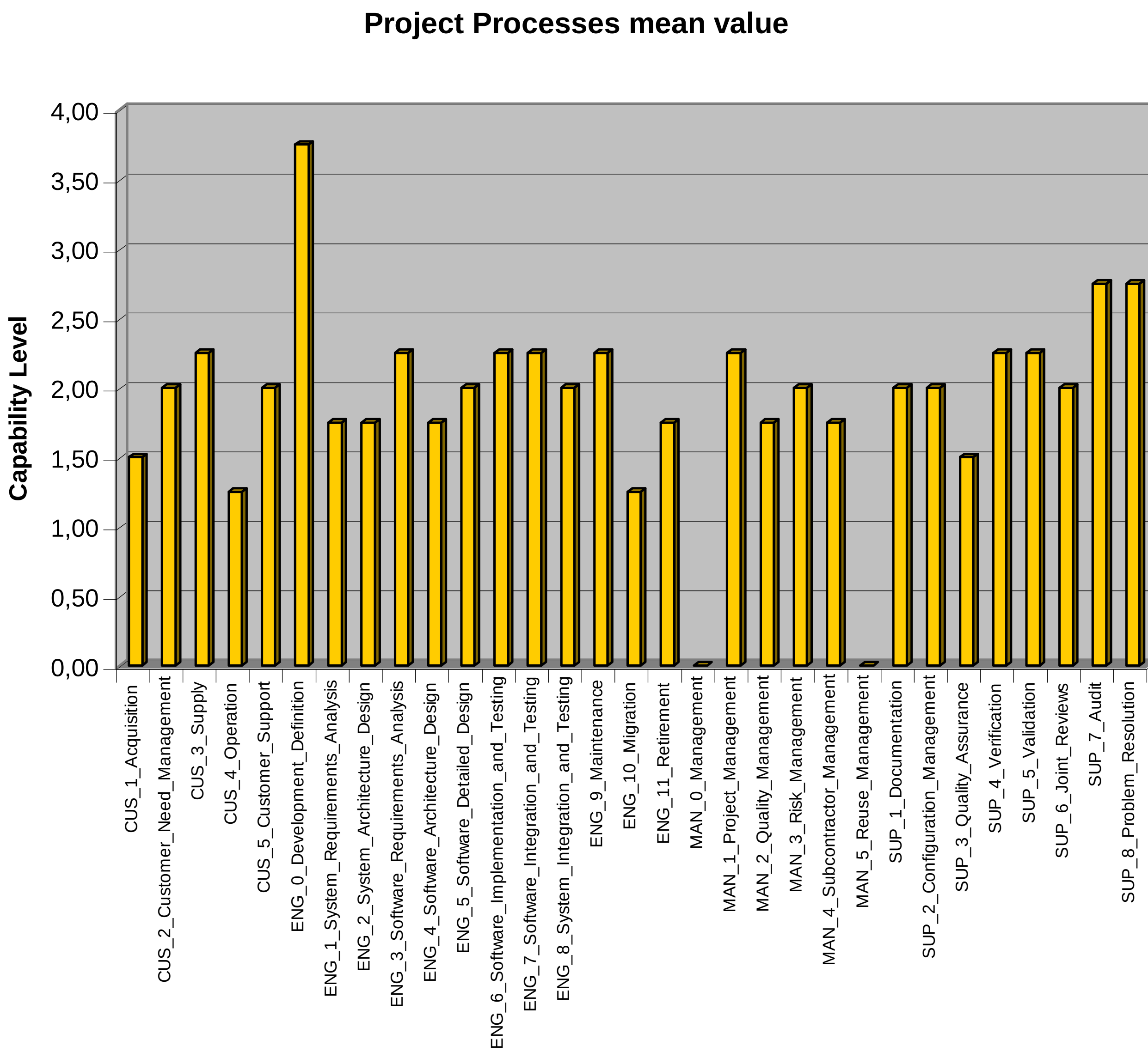
1.3.3 Current status of measurement and benchmarking in the IT industry

As we have seen, tangible benefit analysis needs a process capability level over 3,5. But is this where the industry stands?



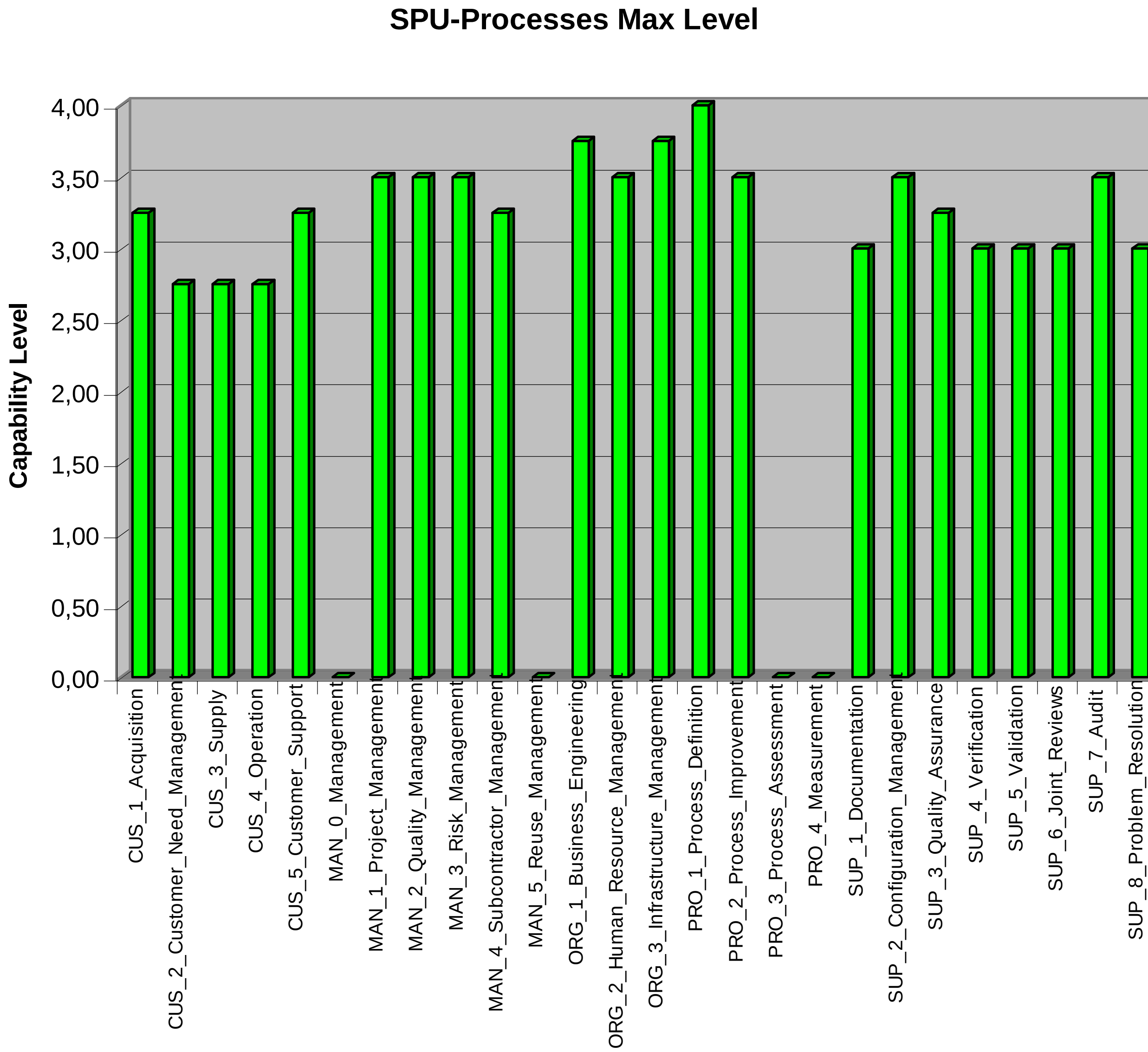
Let’s first have a look at the European Software Industry as a whole picture

The mean value of all SPU processes is 2, i. e. most of the assessed organisations have no standards and no measurements for their key SPU processes.



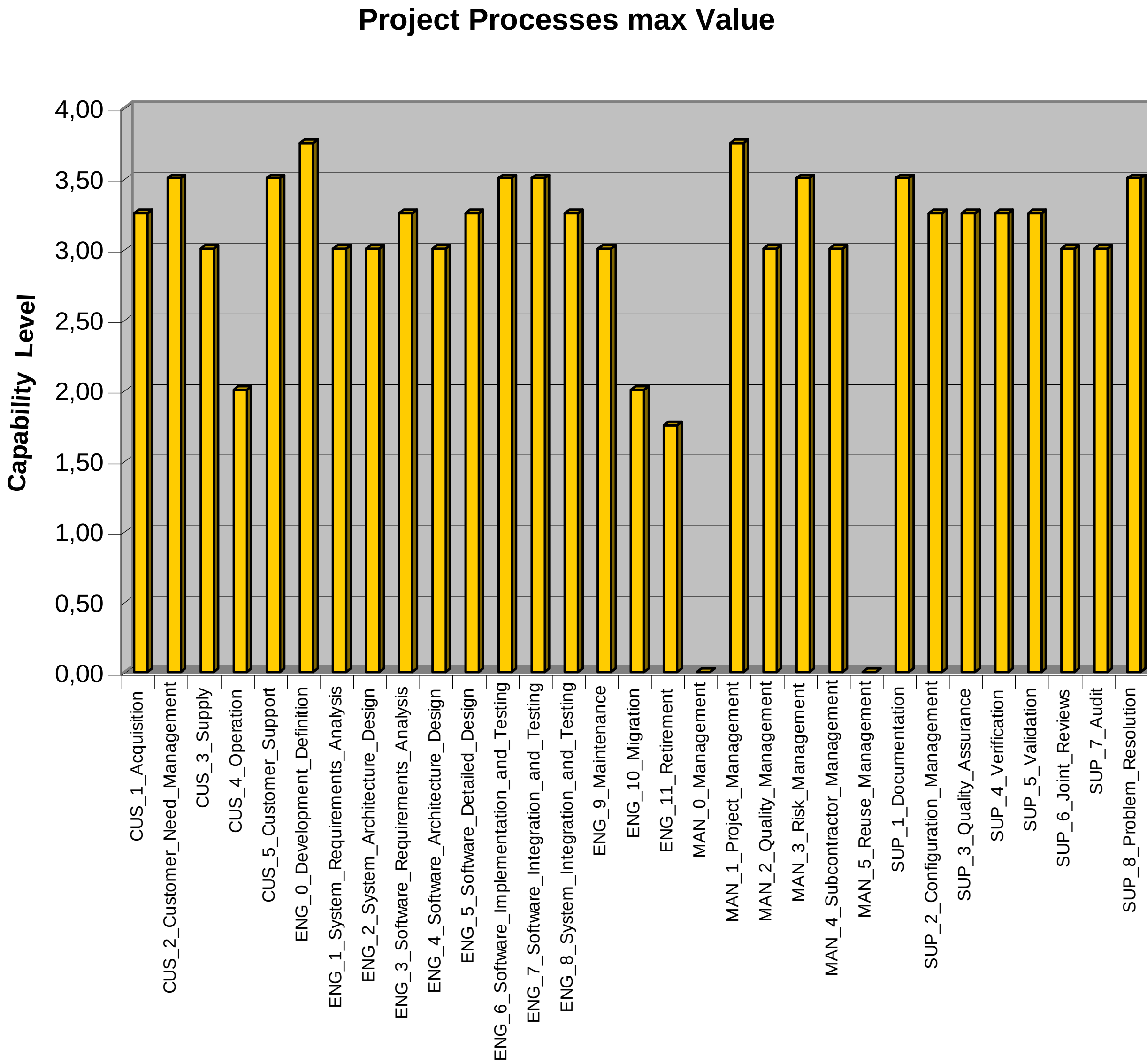
The result for project processes is even worse (1,9), important operative issues are not measured in most of the assessed organisations.

Result most of the Software Producing Units (SPU) do not have the capability to measure their business. So most of the Software Producing Units cannot calculate tangible benefits.



But where are the best?

The best in the market are going straight forward to Level 4 in central SPU processes reaching an average of 2,78 and a process top level of 4 which means that there are standard process definition, measurement and quantitative process control.



And also in project processes the best SPU in the market are moving towards to hit level 4 with an average of 2,93.

What does this mean: The best in the market can calculate tangible benefits so moving forward to ROI decision making and be able to plan and perform assessments and improvements fully according to their business needs.

What about the rest? The rest can proof in intangible benefits.

1.4 Intangible Benefit

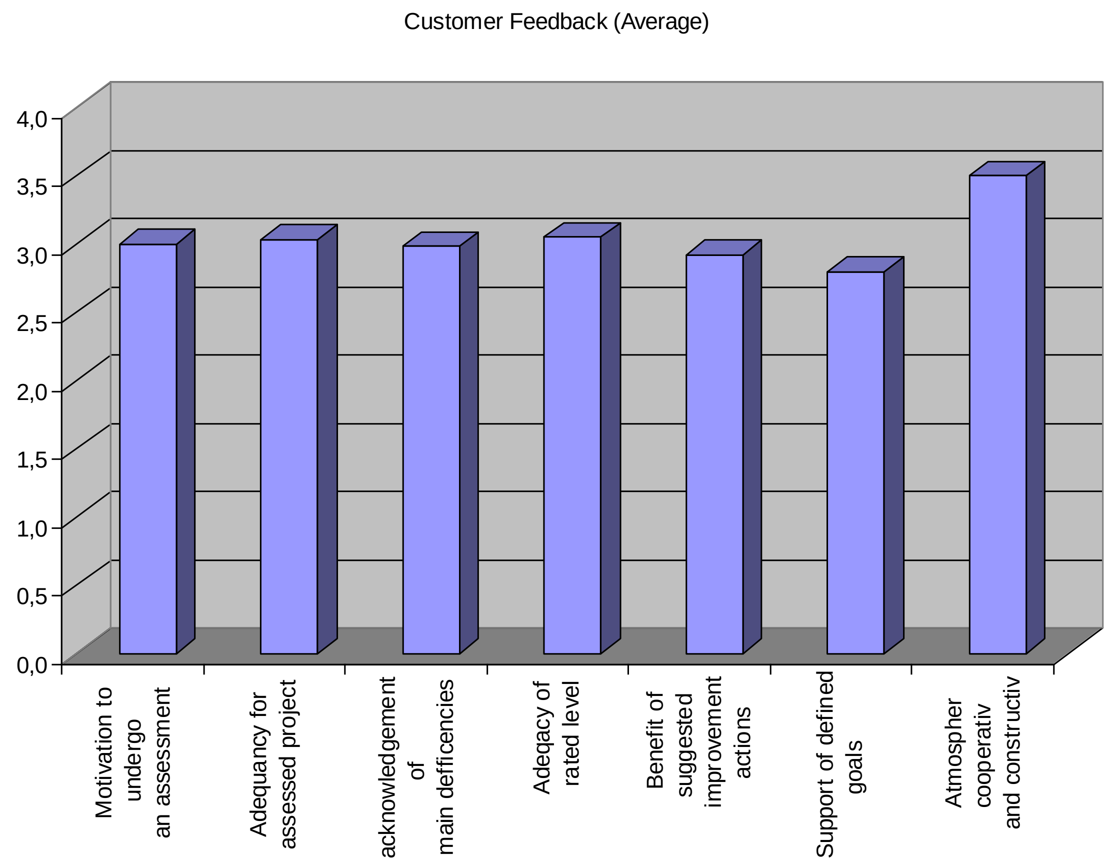
If you want to have an idea of the potential benefit of an investment, and have no chance to calculate the ROI via tangible benefits and be not sure that the published case studies meet your business, you must proof it using intangible benefits.

Typical intangible benefits are:

- Brand Advantage
- Strategic Advantage
- Competitive Advantage
- Intellectual Capital
- Organisational Advantage
- Risk Avoidance

[ALINEAN ROI METHODOLOGY S. 2]

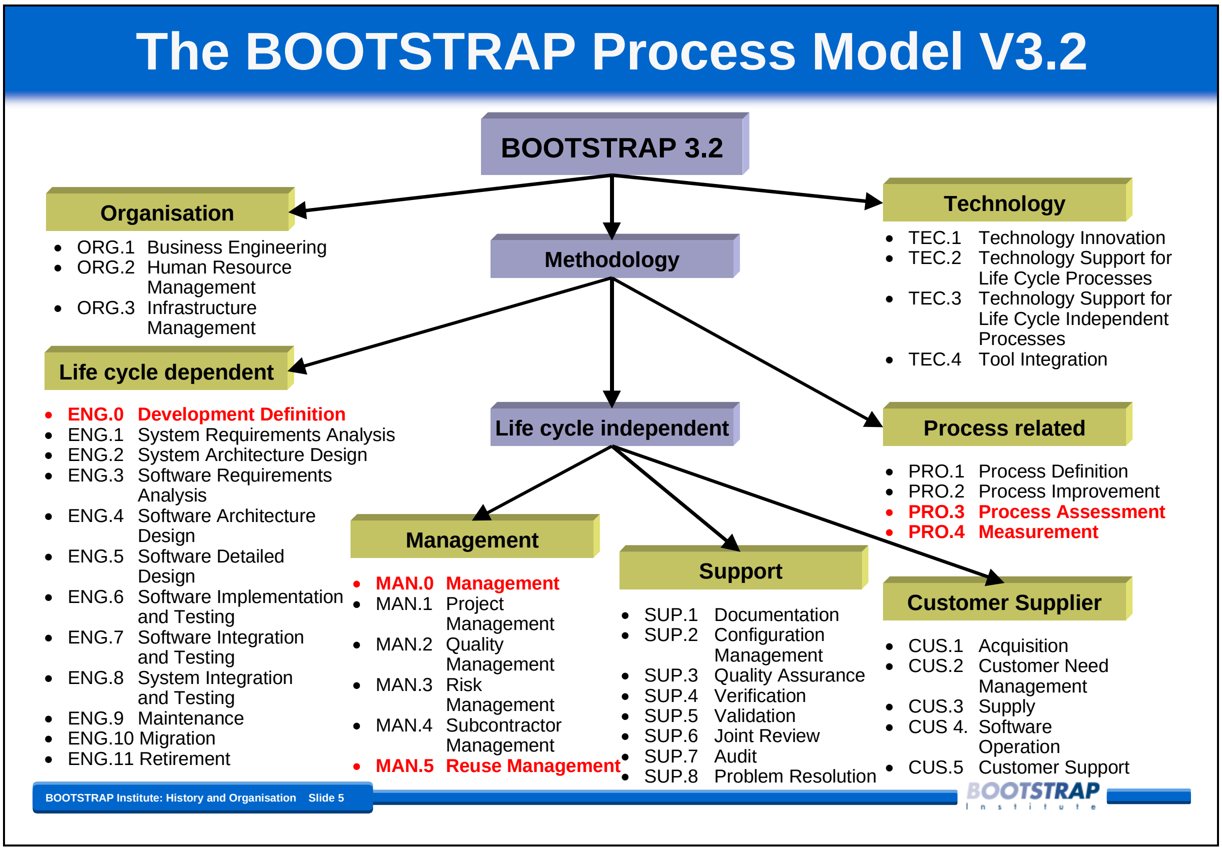
How to proof that these intangible benefits are reached? The BOOTSTRAP Institute uses a detailed feedback form to evaluate customer satisfaction. Parts of this feedback address some of the intangible benefits above:



Customers are satisfied with the benefits of the suggested improvement actions, the acknowledgement of main deficiencies and the support of their stated goals. Knowing that the customer of an assessment and improvement project can trust that the assessment process and the results of the analysis support his goal to reach the planned intangible benefits, it remains necessary to make sure that the model behind the assessment method meets the expected intangible benefits.

1.4.1 Support of intangible benefits during the assessment process

Most of the above mentioned intangible benefits are supported by some central process categories of the BOOTSTRAP process model.



Using a primary impact analysis, intangible benefits can be mapped against the processes of the BOOTSTRAP Model:

	Process	Brand Advantage	Strategic Advantage	Competitive advantage	Intellectual Capital	Organisational advantage	Risk Avoidance	Impact	LEVEL
		Goals & Needs							
	Weight	15	15	15	15	15	25		
1	CUS.1 Acquisition							0,0	
2	CUS.2 Customer Need Management							0,0	
3	CUS.3 Supply							0,0	
4	CUS.4 Software Operation							0,0	
5	CUS.5 Customer Support							0,0	
6	ENG.0 Development Definition							0,0	
7	ENG.1 System Requirements Analysis							0,0	
8	ENG.2 System Architecture Design							0,0	
9	ENG.3 Software Requirements Analysis							0,0	
10	ENG.4 Software Architecture Design							0,0	
11	ENG.5 Software Detailed Design							0,0	
12	ENG.6 Software Implement. and Testing							0,0	
13	ENG.7 Software Integration and Testing							0,0	
14	ENG.8 System Integration and Testing							0,0	
15	ENG.9 Maintenance							0,0	
16	ENG.10 Migration							0,0	
17	ENG.11 Retirement							0,0	
18	MAN.0 Management							0,0	
19	MAN.1 Project Management							0,0	
20	MAN.2 Quality Management							0,0	
21	MAN.3 Risk Management							0,0	
22	MAN.4 Subcontractor Management							0,0	
23	MAN.5 Reuse Management							0,0	
24	ORG.1 Business Engineering							0,0	
25	ORG.2 Human Resource Management							0,0	
26	ORG.3 Infrastructure Management							0,0	
27	PRO.1 Process Definition							0,0	
28	PRO.2 Process Improvement							0,0	
29	PRO.3 Process Assessment							0,0	
30	PRO.4 Measurement							0,0	
31	SUP.1 Documentation							0,0	
32	SUP.2 Configuration Management							0,0	
33	SUP.3 Quality Assurance							0,0	
34	SUP.4 Verification							0,0	
35	SUP.5 Validation							0,0	
36	SUP.6 Joint Reviews							0,0	
37	SUP.7 Audit							0,0	
38	SUP.8 Problem Resolution							0,0	

Weighting can be done in three steps

- Detail the focussed intangible benefits to an operational level
- Define the impact of each process to each intangible benefit
- Define the priority of the intangible benefits

As result you will receive an overall impact of each process to the stated intangible benefits.

As a result customer receives a valuable information over the potential support for his business goals by using the BOOTSTRAP Assessment and a first idea if customer should undertake a full assessment or a focused assessment covering the processes with the highest impact.

1.4.2 Benefits from the assessment process

Looking on the assessment and improvement process as a whole PDCA picture first benefits are earned during the assessment process or as its results.

To understand how a BOOTSTRAP Assessment supports the earning of benefit it is necessary to understand some principles of the assessment process (which is fully ISO 15504 compliant)

- Improvement Orientation: BOOTSTRAP Assessments are performed to identify areas of improvement. That makes a real difference to assessments or audits which are designed for marketing purposes or personal rating.

- Team Credibility: BOOTSTRAP Assessments are performed by well-trained and experienced assessors. The experience and the personal charisma of the assessors is very important, because they are able to discuss best practice improvement proposals in different environment.
- Confidentiality: There are several levels of confidential agreements ensuring an open space atmosphere during the interviews which helps the assesses to speak very open even over weaknesses.
- Listen-Interviewing skills of the assessors: Assessors do not discuss the behave of the interviewees, ask open questions, and listen carefully. This makes sure that all available information is provided by the interviewees and recorded by the assessors.
- Clear roles in the interview: Only one assessor asks questions, no crossfire interviews. That reduces stress of the interviewees and ensures their support.

What are the consequences of such an assessment process and therefore the direct Benefits of a BOOTSTRAP assessment?

- Immediate Improvement: It is difficult to measure a process without affecting him. Mostly assesses start thinking about changing their behave during the interviews. Speaking over the implementation of the BOOTSTRAP processes and having the possibility to speak very open and have someone who listens they often realise that they can change and improve their way to work without any formal improvement project.
- Awareness about the current status. The assessment report contains detailed findings, strengths and weaknesses.
- Improvement Recommendations. The assessment report contains recommendations to each weakness improvement based on industrial best practises. Even if no formal improvement projects are launched, management can use this in areas with management attention as direct basis for decision making.

The next figure shows the typical outcome of BOOTSTRAP assessments related to improvement recommendations:

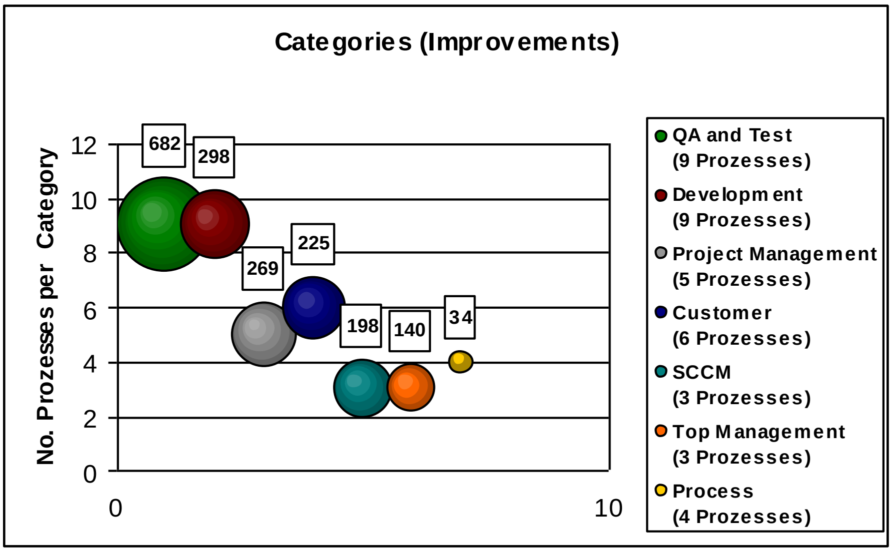
Weaknesses statistics

- **Assessments (2001) 27**
- **Improvement recommendations (in total) 1686**
  - **Maximum 228,00**
  - **Average over all assessments 44,37**
  - **Average per assessment 62,44**
  - **Average per process and assessment 1,64**

Bootstrap Overview

Having a more detailed view, we see the improvements ordered by process categories

Improvement Areas



A great number of improvements is related to the QA and test area. Which has as above mentioned a volume of tangible benefits of **EUR 2.000 Mio.** In detail a BOOTSTRAP Assessment helps to find out where this money can be found.

1.4.3 Benefit from the improvement process

Typical SPI benefits are

- Pride
- Esprit de corps
- Quality of work life
- Company Image
- Emergence of a coherent culture

[Herbsleb, Carleton, Rozum, Siegel, Zubrow Benefits of CMM-Based Software Process Improvement: Initial Results, 26]

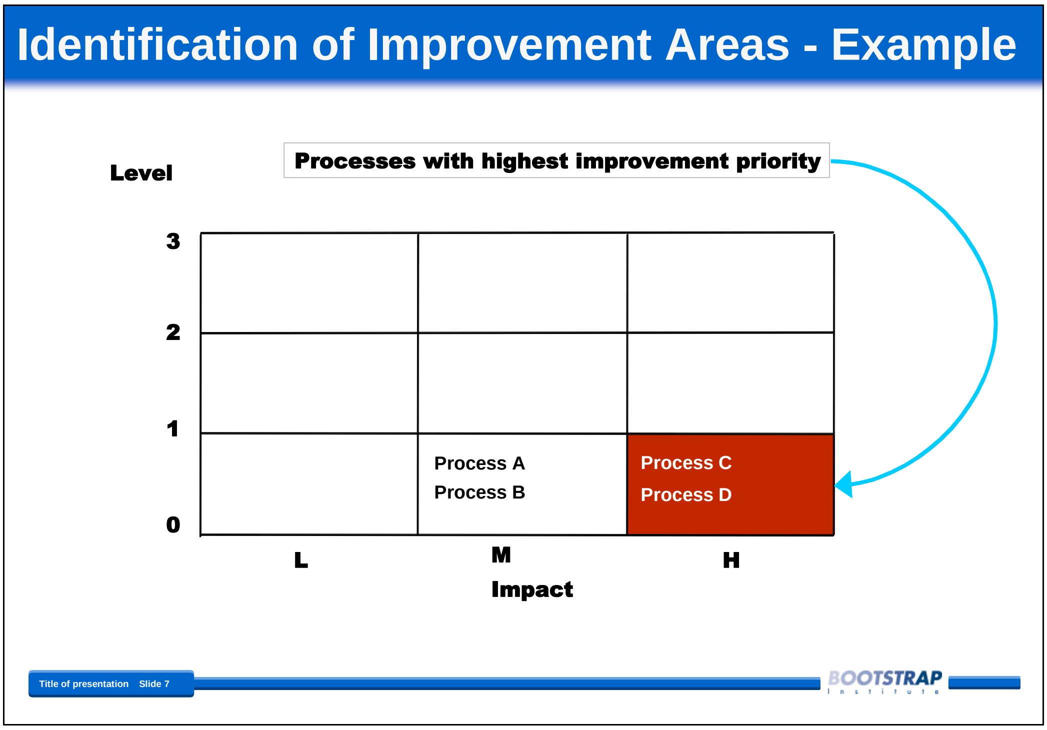
Discussing the intangible benefits from the improvement process it should be clear that improvement opportunities must be clearly addressed and prioritised by business needs.

Having no opportunity to measure, it is clear that there must be qualitative criteria for prioritising improvement actions to develop a practical roadmap for improvement.

There are two major criteria for prioritising improvements:

- The impact of the process subject to improvement to the business goals and needs (one does not invest in improving a process which has no impact to your business)
- The capability level of the process (one normally improve weak processes first)

Having done the assessment and the impact analysis (see above) you now can easily prioritise the processes subject to improvement:



Right now it is very clear which processes have to be improved first to come the widest step forward to earn the intended intangible benefits for each Euro invested.

1.5 A bridge from intangible benefits to tangible benefits

As shown, the number of improvement recommendations can be measured. Normally, there is a substantial number of recommendations in each assessment report affecting several processes from the BOOTSTRAP process model.

Even if it is not really possible to calculate it might be possible to estimate.

One of the first available case studies figures out that the average investment for each improvement is about 25000 $ and the Benefit is about 350000 $ coming to a 6/1 business value

[Herbsleb, Carleton, Rozum, Siegel, Zubrow Benefits of CMM-Based Software Process Improvement: Initial Results 38]

Finding that after each guided full assessment there are 1,6 improvement recommendations per each process and guessing that improvement actions are normally focussed on 1 to 4 processes at a time coming to an average of 3 processes state out that after each assessment 5 improvement projects are planned and performed.

Looking to this we can estimate the benefit figure for each guided full assessment:

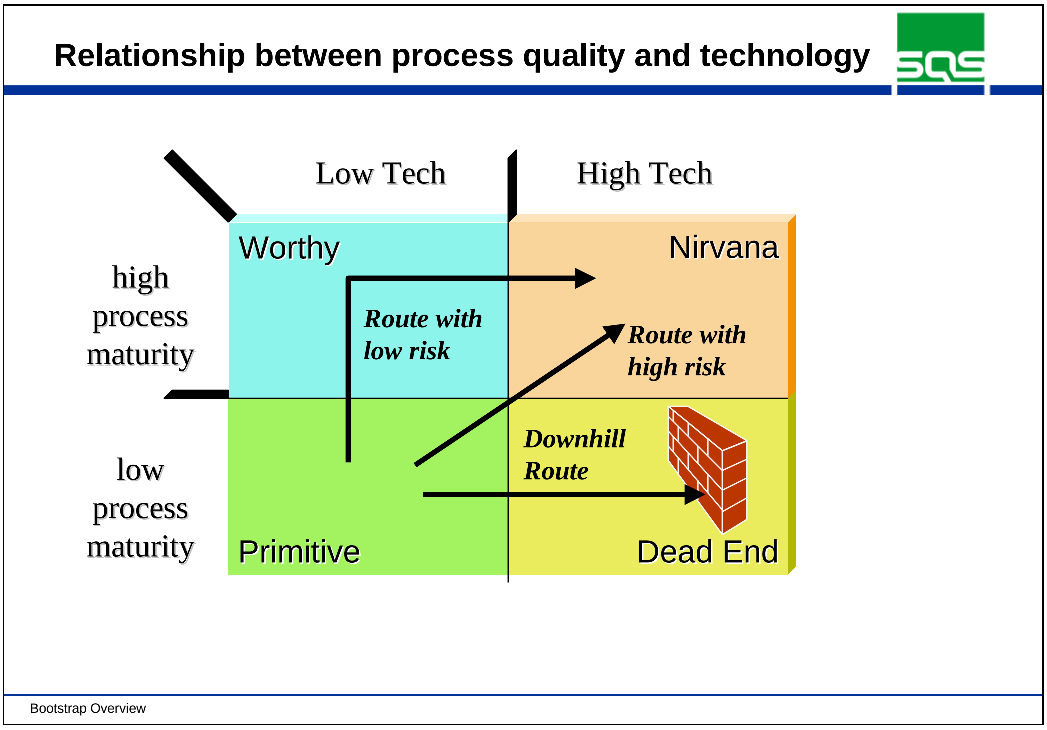
ITEM	Costs	Benefit
Improvement	125000 EUR	1,75 Mio EUR
Assessment	from 100000 up to 150000 EUR	
Total	**225000 up to 275000 EUR** **Average 250000 EUR**	**1,5 Mio EUR**

That does not mean that it is possible to recalculate the benefits of each assessment, because it should be remembered that most SPU have not the necessary data to define a business baseline and to calculate the benefits.

1.6 Risk Management

BOOTSTRAP assessments help to reduce risks in several ways:

- Providing an assessment focussed on risks to give a direct support to risk management, so helping to analyse, prioritise and mitigate risks
- Providing a clear status for each assessed process, risks become more transparent
- Providing a clear roadmap to software process improvement BOOTSTRAP assessments help to mitigate the risk that investments in process improvement fail.



Lessons learned

The success of an assessment is based on two drivers:

Results acceptance from assesses and management.

Cost acceptance from management.

To ensure both, an assessment must have the following features

ISO 15504 compliant process model with process dimension and capability dimension

Tailoring the assessment due to the needs of the assessed organisation

Provide support for Guided, Workshop and Self Assessments

Provide trained and experienced assessors with interviewing, listening, negotiation and judgement skills.

Provide reports with detailed findings and improvement recommendations based on software industry best practise.

Provide an improvement planning method

Benefits must be clearly addressed

- To tangible benefits
- Intangible benefits

Most of the existing SPU cannot calculate tangible benefits so intangible benefits are the most important issue.

Assessment method must have features that prove that expected intangible benefits are reached.

Intangible benefits can be estimated from assessment reports and customer feedback.

Literature

Strategy white papers of Bootstrap 4.0 (Bootstrap Institute 2002)

Service Definition and Marketing Support 2002 (SQS AG)

Strategy white papers of Bootstrap 4.0 (Bootstrap Institute 2002)

Service Definition and Marketing Support 2002 (SQS AG)

After the Appraisal: A Systematic Survey of Process Improvement, its Benefits, and Factors that Influence Success (Dennis R. Goldenson; James D. Herbsleb; Aug 1998)

Benefits of CMM-Based Software Process Improvement (James D. Herbsleb; Anita Carleton; James Rozum; Jane Siegel; David Zubrow; Aug 1994)

Process Improvement Matrix: A Tool For Measuring Progress Toward Better Quality (Janis Livingston; Kellye Prosise; Robert Altizer

Human-Oriented Improvement in the Software Process (Khalid Sherdil; Nazim H. Madhavji)

Measuring and Improving Process Capabilities – Best Practices (Jyoti M. Bhat; Anoop Kumar)

Return on Software Process Improvement (Ken Frazer)

Measuring Softtware Process Improvement: There's More To It Than Just Measuring Processes! (David A. Reo; Nuria Quintano; Luigi Buglione)

Software Engineering Management Practices – Software Engineering Measurement and Analysis Initiative (Carnegie Mellon Software Engineering Institute)

Benefits of CMM-Based Software Process Improvement: Executive Summary of Initial Results (James Herbsleb; Anita Carleton; James Rozum; Jane Siegel; David Zubrow; Sep 1994)

Measuring the Value from Improved Predictions of Softwae Process Improvement Outcomes Using Risk-Based Discount Rates (Warren Harrison; David Raffo; John Settle)

SPICE: An Empiricist's Perspective (Khaled El Emam; Dennis R. Goldenson; Aug 1995)

Author CVs

Short Curriculum Vitae

Name	Peter Bölter
Date of Birth	April 8th, 1959
Graduation	Dipl.-Kfm. (holder of a Masters degree in commerce)
Contact	SQS AG, Stollwerckstr. 11, 51149 Cologne (Germany), Tel: ++49-2203-9154-0
Function	Senior Executive Business Unit "Business Development" Head of department "Telecommunications", project manager, management consultant
1979-1982	2.5 year industrial trainee programme including 6 months in USA
1988	Diploma in commerce, University of Cologne
Since 1988	Consultant, project manager and head of department at SQS AG, Cologne Work areas: ■ Software testing, especially accpetance testing of business software ■ Test management ■ Project management ■ Introduction of configuration management ■ ISO 9000 ff, assessment of current status and analysis of required status, setting up a quality management system including quality management handbook and procedures. ■ Quality management systems, quality assurance, especially training and consultancy ■ Strategic IT Management ■ Development of test software ■ BOOTSTRAP-Assessments (Lead Assessor conform ISO 15504)
1998-2000	Director Competence Center Quality Management and Configuration Management
Since April 1999	Director of the BOOTSTRAP Institute
2000-2001	Senior Executive Business Unit: Telecommunications, E-commerce and Public
Since Jan 2002	Senior Executive Business Unit: International Market and Business Development

Specification and Test of Forms - All in 1!

Anne Mette Jonassen Hass, DELTA

Abstract

The user interface is of paramount importance in modern applications, not least Web-applications. This is the users first, if not only impression of the application; in the end of the day the user interface is the application to the end-user. The credability of the application is closely connected to the user interface, which therefore must be consistent, reliable, and correct.

Documenting and testing a user interface can be a huge task in modern applications, where the user interface is based on highly complex windows, and usually a lot of them.

The method presented in this paper provides a way to document the requirements elicited for example through scenarios and usability assessments in a set of tables in such a way that this documentation may be used directly as basis for design and development, basis for test, and basis for support/help desk work.

Keywords

User interface as state machine

Functionality specification

Functionality test

Table driven specification and test

1 Introduction - From Requirements to a Complete System

All software development is about realizing the future users' requirements in a satisfactorily working system. Development is done according to a development model, i.e. a series of activities being performed in a specified order, and each being based on a specified input and producing a specified output.

There are many different development models in use today, the most common perhaps some kind of iterative or agile model. For all models the activities are more or less the same, the difference is to be found in the way the activities are arranged and repeated in relation to each other.

1.1 Development According to the V- Model

The basic activities in any software development model are shown below, indicating how the activities may rely on other activities for input in the form of output from other activities.

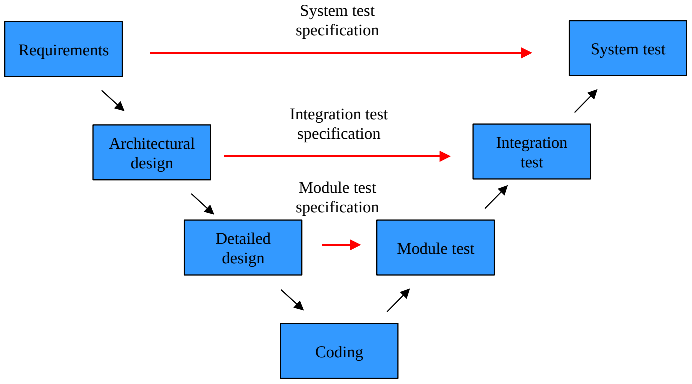


Figure 1: Development Activities Arranged according to the V-Model.

The principles described in this paper are valid no matter which development model you use.

1.2 Quality Factors

The quality of a system depends on how well it fulfils the requirements. To make the work with requirements a little easier to grasp, several quality models have been defined. A quality model consists of a number of quality factors, and may be seen as a checklist for requirements or a table of contents for a requirements specification.

The standard ISO 9126 defines 6 toplevel quality factors, namely:

- functionality (including security)
- reliability
- usability
- efficiency (including performance)
- maintainability
- portability

There are many other ways to categorise quality factors, but this one is not too bad.

It is important to consider all relevant quality factors when working on the specification of requirements.

1.3 FIRST in Relation to the Development

The method for **F**ormbased **I**ntegrated **R**equirement **S**pecification and **T**est - FIRST - to be introduced in this paper is not a silver-bullet. It is does not solve all your problems, but it is a piece among others in the development jigsaw. And an important piece!

The FIRST provides a table-driven method for the description of the functionality of a system seen from the user interface. The tables defined for capturing the functional requirements may also be used directly as part of the basis for design, for test (in combination with general test cases), and for support of the system when in use. The tables kill four birds with one stone, so to speak.

2 The Principles in FIRST

The user interface is of paramount importance in the modern applications, not least Web-applications. This is the users first, if not only impression of the application; in the end of the day the user interface is the application to the end-user. The credability of the application is closely connected to the user interface, which therefore must be consistent, reliable, and correct.

The definition of the functionality of the system and hence the system test of the functionality should therefore focus on the system seen from the user interface.

2.1 User Interface - from a Technical Point of View

From a technical point of view the user interface of any system may be regarded as a gigantic state machine, as illustrated below.

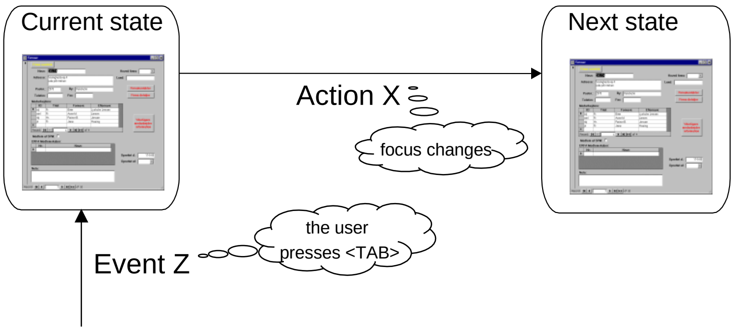


Figure 2: User Interface Regarded as a State Machine

At any point in time the system is in a state. When an event occurs the system responds with an action which changes the state from the current state to the next state.

2.1.1 State Characteristics

A state in a user interface is characterized by:

- the form in focus
- the field in focus
- the actual data presented in the form

- the data stored in the entire system
- the data handling in progress (the system mode)

2.1.2 Event Characteristics

An event in a user interface may for example be:

- typing or deletion of a character in a field
- hitting a key or issuing some other type of command that
  - changes the focus from one field to another
  - changes the focus from one record to another
  - changes the focus from one form to another

Furthermore there may be other events like batch jobs or electrical impulses.

2.1.3 Action Characteristics

An action is what happens after an event, for example:

- validation of data according to the type of the field, when the field looses focus
- further validation, if any, for a field when it looses focus
- validation to be performed when a record looses focus
- failure reaction to a validation that fails

or, if no failure:

- change of focus dependent on the event
- change(s) somewhere else in the system dependent on the new contents of the field or the record loosing focus

2.2 Complete State Machine Specification and Test

In order to describe a system's functionality seen from the user interface as a state machine, you need to describe all possible states, all possible events, and all possible actions. Some of the elements describing the states, the events, and the actions are generic, i.e. valid across all forms or across a large number of similar forms in the system, while other elements are specific to a specific form.

Documenting and testing a system like this can be a huge task in modern applications, where the user interface is based on highly complex forms, and usually a lot of them. If we consider the number of different data handled by even small systems and the number of fields we can see that the number of states becomes extremely large. Likewise the number of ways to navigate between fields and records, i.e. the state transitions becomes extremely large. And we have to test that for each possible state all events cause the correct action(s) and lead to the correct next state. Not to mention invalid transactions, transition pairs, triplets etc., etc…

We need to be able to establish an overview of all the possible states in the system and all the possible state transitions, and also to establish a way to determine the test coverage we obtain during the test, since it is obviously impossible to obtain a complete coverage.

The FIRST method presented in this paper provides a way to document the functional requirements in a set of tables in such a way that this documentation may be used directly as basis for design and development, as well as as basis for test. The tables may even be used as the basis for support/helpdesk work for the running system.

The figure below provides an overview of the tables.

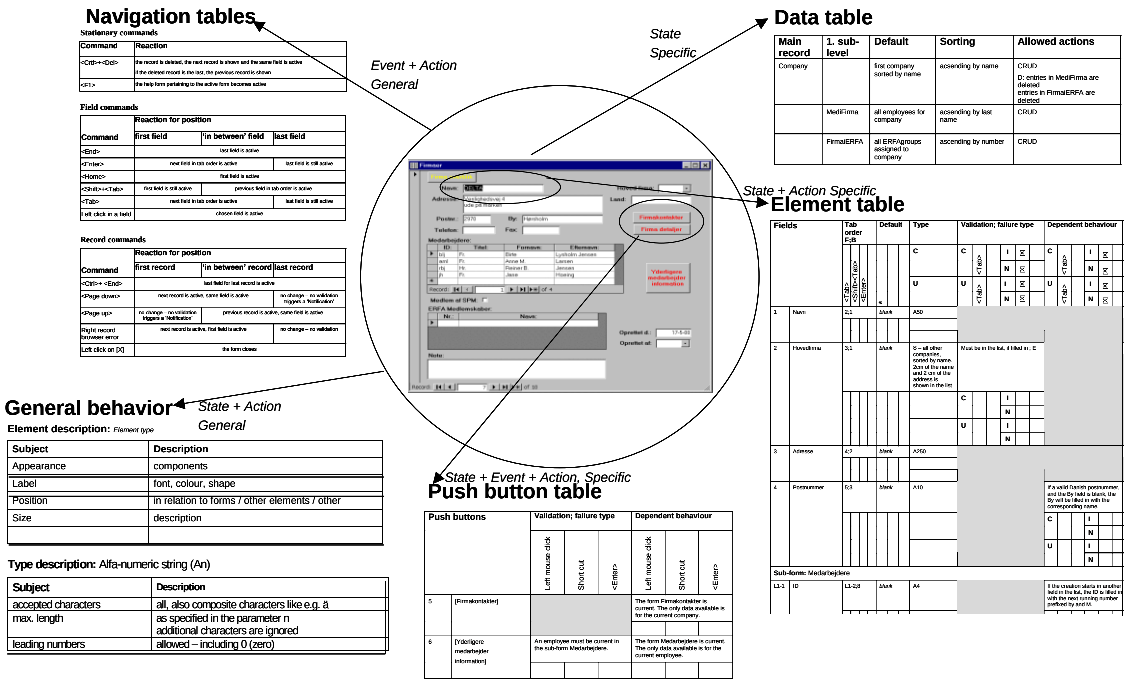


Figure 3: FIRST Table Overview

Each of the types of tables is described in details below.

3 The FIRST Tables used for Specification

The idea in FIRST is, that the information needed to describe the user interface state machine is kept in tables. This approach has several advantages, e.g. tables provide a very good overview of the information, tables help in keeping the focus on one thing at the time during specification, tables show very clearly where information is missing, tables facilitate consistency, and tables provide a lot of implicit information in their very structure.

Some people do not like working with tables, but experience has shown that the FIRST tables are worth the effort.

The tables are divided into two main groups, namely those for the general system-wide information for states, events, and actions, and those for the information for specific forms.

3.1 General System-Wide FIRST Tables

The general system-wide information for the possible states are related to the types of forms used in the system and to the allowed elements in the forms, such as fields and tables, used in the system. The information for the events are related to the commands used in the system, and the action information is related to the commands as well as to general behaviour for data handling modes, such as creation, and for reactions to error situations.

3.1.1 General State Information

The user interface of a system consists of a number of definable types of forms. First of all these types will have to be determined and described. Types may be: Main data form, Popup – modal (where the rest of the system is locked), Popup – nonmodal (where the rest of the system is still accessible), and Messagebox. Usually not many more types are needed. The descriptions are filled in to tables like the one shown below.

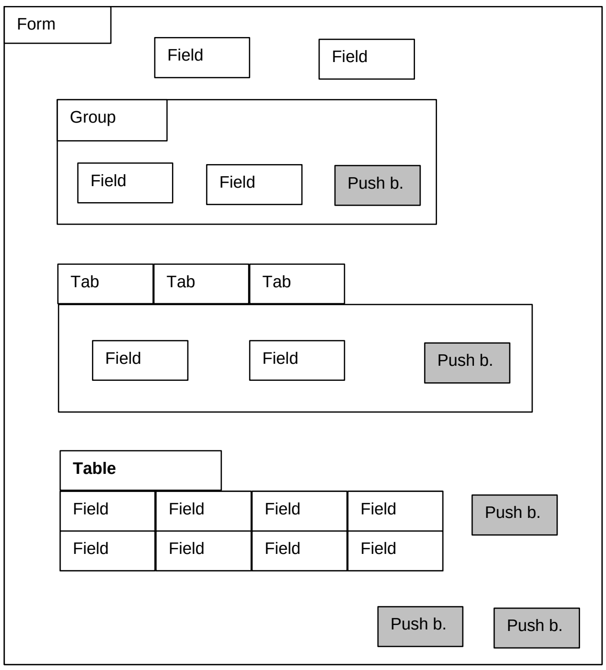
Type description:*–Type*

Subject	**Description**
Titel line	color, contents
Menu linie	position, contents
Tool line	Position, contents
Position	Relative to the screen/ other elements
Size	Change allowed / not allowed

Figure 4: Form Type Description Table

The forms contain elements. Elements may be in classes, like Fields, Groups, Push buttons etc. For each class several types may exist, e.g. there may be string fields, number fields, date fields etc. Element classes, and types must be determined and described a long with the allowed hierarchies.

The figure below shows an example of element classes and hierarchy for a typical windows system.




```
Form
  |
  |-----Field(s)
  |
  |-----Group(s)
  |         |
  |         |-----Field(s)
  |         |-----Push button(s)
  |
  |-----Table(s)
  |         |
  |         |-----Field(s)
  |         |-----Push button(s)
  |
  |-----Tabs
  |         |
  |         |-----Field(s)
  |         |
  |         |-----Group(s)
  |         |         |
  |         |         |-----Field(s)
  |         |         |-----Push b.(s)
  |         |
  |         |-----Table(s)
  |         |         |
  |         |         |-----Field(s)
  |         |         |-----Push b.(s)
  |         |
  |         |-----Push button(s)
  |
  |-----Push button(s)
```

Figure 5: Element Classes and Hierarchy.

Details for element types are described in tables similar to the one shown above in Figure 5.

3.1.2 General Event Information

The general event information is related to the navigation in the system. Within forms it is possible to navigate by commands (here keyboard-clicks and mouse-clicks), and form-specific push buttons. It may also be possible to navigate via menus, but this is left out of this description for space reasons.

For general events all commands will have to be listed. They must be split into 3 classes according to what general state transition they will cause and the type of validation they will trigger. The classes are shown in the table below.

Command class	Focus change	Validation triggered
Stationary commands	the focus does not change	none
Field commands	the focus change from one field to another	the field, that is being left
Record commands	the focus change from one record to another	all fields in the record being left

The description of the commands is found below, as these are closely related to the general action information for the system.

3.1.3 General Action Information

The general action information may also be called the general behaviour of the system. This is related to the commands as well as to the data handling modes, such as creation, and the reactions to error situations.

In the following a few examples of the descriptions in the command tables are shown. Each command will appear in only one place. The tables provide the specification of the action for each command is issued, and what type of validation that is connected to the command.

Stationary commands

Command	Reaction
<Crtl>+<Del>	the record is deleted, the next record is shown and the same field is active if the deleted record is the last, the previous record is shown
<F1>	the help form pertaining to the active form becomes active

Field commands

Command	Reaction for position		
	first field	**'in between' field**	**last field**
<End>	last field is active		
<Enter>	next field in tab order is active		last field is still active
<Home>	first field is active		
<Shift>+<Tab>	first field is still active	previous field in tab order is active	
<Tab>	next field in tab order is active		last field is still active
Left click in a field	chosen field is active		

Record commands

Command	Reaction for position		
	first record	**'in between' record**	**last record**
<Ctrl>+ <End>	last field for last record is active		
<Page down>	next record is active, same field is active		no change – no validation triggers a 'Notification'
<Page up>	no change – no validation triggers a 'Notification'	previous record is active, same field is active	
Right record browser error	next record is active, first field is active		no change – no validation
Left click on [X]	the form closes		

It may look trivial, but when you get everybody involved – developers, support people, and users – to sit down and go through all possible ways of navigation, it is not simple at all. But it is very useful to get a description of this.

Some commands react differently when issued inside an element in a form. Tables like the above must be provided for these commands for the situation where they react differently.

Examples of the specifications for the data handling modes are not included here, but tables like the one shown in figure 4 must be provided for each handling mode, e.g. creation of record, creation of sub-record (in a table form element), deletion of a record, etc. etc.

When a validation fails the system will act in some way. The types of possible actions must be identified and specified. The following table provides an example of some types and their actions.

Failure type	Type	Reaction
Error	E	a message box appears with a 'reasonable' error message the system is locked when the [OK] button is pushed the system returns to the state it was in before the validation was triggered
Notification	N	a 'reasonable' message appears at the bottom left corner of the form the message disappears when a command is issued or a key pushed
Warning	W	a message box appears with a 'reasonable' error message the system is locked when the [OK] button is pushed the system goes on according to the command that triggered the validation

3.2 Form-Specific FIRST Tables

For each form in the system the state and action information must be specified in a set of form specific tables. The set consists of three forms, namely forms for data information, for element information, and for push button information. Push buttons are also elements, and the information for these could be kept together with the other elements, just like the information could be split for other elements or as a special case. Experience shows that this split is the most convenient.

Each table in the set is described below. For the tables it is required that each cell is either filled in or greyed out – otherwise the specification is not complete.

The examples below are related to the example form shown here. The form is in Danish, but that should not the principles from being explainable.

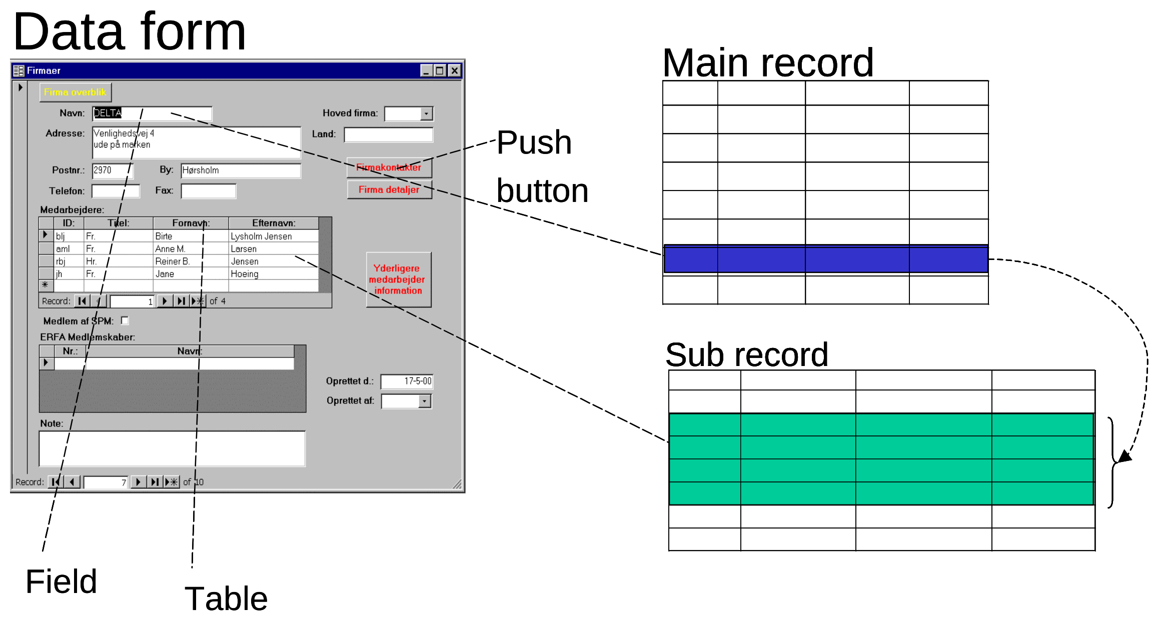


Figure 6: Example Form

3.2.1 *Data Table*

Each form handles a main record, and possibly a number of sub-records in tables. Information about the data is specified in a table like the one shown as an example below.

Form: Firmaer

Main record	**1. sub-level**	**Default**	**Sorting**	**Allowed data handling**
Firma (Company)		first company sorted by name	ascending by name	**CRUD** D: entries in Medarbejdere are deleted entries in ERFAmedlemsskaber are deleted
	Medarbejdere	all employees for company	ascending by last name	CRUD

Figure 7: Data Table

In the table above the allowed data handlings are C: Create, R: Read, U: Update, and D: Delete.

3.2.2 *Element Table*

The elements present in the form (except for the push buttons) are described in the elements table, of which an example is shown below.

The colomns in this table are, from left to right: unique number, labeltext, taborder forwards and back-wards, default when a new record is created, element type (as defined above), validation to be per-formed in addition to validation related to the element type, and the action.

Elements		**Tab order F;B**	**Default**	**Type**	**Validation; failure type**	**Action**
1	Navn	2;1	*blank*	A50		
2	Hovedfirma	3;1	*blank*	S – all other companies, sorted by name. 2cm of the name and 2 cm of the address is shown in the list	Must be in the list, if filled in ; E	
3	Adresse	4;2	*blank*	A250		
4	Postnummer	5;3	*blank*	A10		If a valid Danish postnummer, and the By field is blank, the By will be filled in with the corresponding name.
Table:	Medarbejdere					
L1-1	ID	L1-2;8	*blank*	A4		If the creation starts in another field in the list, the ID is filled in with the next running number prefixed by an M.
End of table						

Figure 8: Element Table

Action occurs when correct data is entered into a field and this data has effects in other places in the system. It might be that the data appears in another field on another form, that the data causes fields to appear or disappear, or that values in other fields are calculated on the basis of the data.

The specification of the action may be very large and complicated in some cases. This may frighten the analysts filling in the forms, but if they cannot describe the action how do they expect the developers to code it and the testers to test it? The action must be described, if there is any!

An element table may be quite big if the form it describes is big. That is just too bad – even big forms will have to be described. Take it one element at the time and work your way through the table, it is worth it.

3.2.3 Push Button Table

A form usually also have push buttons for various events and their pertaining actions. These must be specified in a table like the one shown below.

Push buttons		**Validation; failure type**	**Action**
5	[Firmakontakter]		The form Firmakontakter is current. The only data available is for the current company.
6	[Yderligere medarbejder information]	An employee must be current in the sub-form Medarbejdere ; E.	The form Medarbejdere is current. The only data available is for the current employee.

Figure 9: Push Button Table

Push buttons also behave as record commands.

4 The FIRST Tables used for Test

In principle you have to test that all state transitions work correctly. This means that for each form for all combinations of data you have to test for each element that the correct action(s) occur(s) for each possible event. For example that the validation of each field is performed correctly for all the ways to trigger a validation, i.e. for all field commands with the field in focus and for all record commands both with the field in focus and with all the other fields in focus; both during creation and update.

If you calculate the number of test steps this leads to, the result is astronomical. It is an impossible task, both to specify the test and to perform it, and it will be necessary to focus on the most important areas to reduce the task.

The tables used for the specification of the system as described above are used as the basis for the test. The person responsible for test can decide what to test, and can get an overview of what have actually been tested as the test is being performed and when the test is over.

Test cells are added to the element table and the push button table, so that they provide the information about what to test, and also a way to register directly in the table what was tested, by marking of the relevant cells. Thus it is possible get an overview of the coverage by examining what is marked and what is not; and an idea of how the test is going by examining which cells are marked OK and which are marked with a reference to an incident report. A small example of an extended element table is shown below.

Element		Tab order F;B	Default	Type	Validation; failure type	Action
		<Tab> | <Shift><Tab> | <Enter>	*	C U	C | <Tab> | I [X] | N [X] U | <Tab> | I [X] | N [X]	C | <Tab> | I [X] | N [X] U | <Tab> | I [X] | N [X]
1	Navn	2;1	*blank*	A50		
2	Hovedfirm	3;1	*blank*	S – all other companies, sorted by name. 2cm of the name and 2 cm of the address is shown in the list	Must be in the list, if filled in ; E	
					C | I N U | I N	

Figure 10: Extended Element table

It is indicated in the element table, which commands must be tried and some cells are left open, so that the tester can decide which other commands to try. More mandatory commands to try may off cause be specified – the table can be expanded in size to cover for more state transitions.

Generic test cases are written on the basis of the general tables.

A short test script should be provided for each of the forms, based on the data information, to provide the guide for testing using the test tables and the generic test scripts. The following figure illustrates a small part of the test.

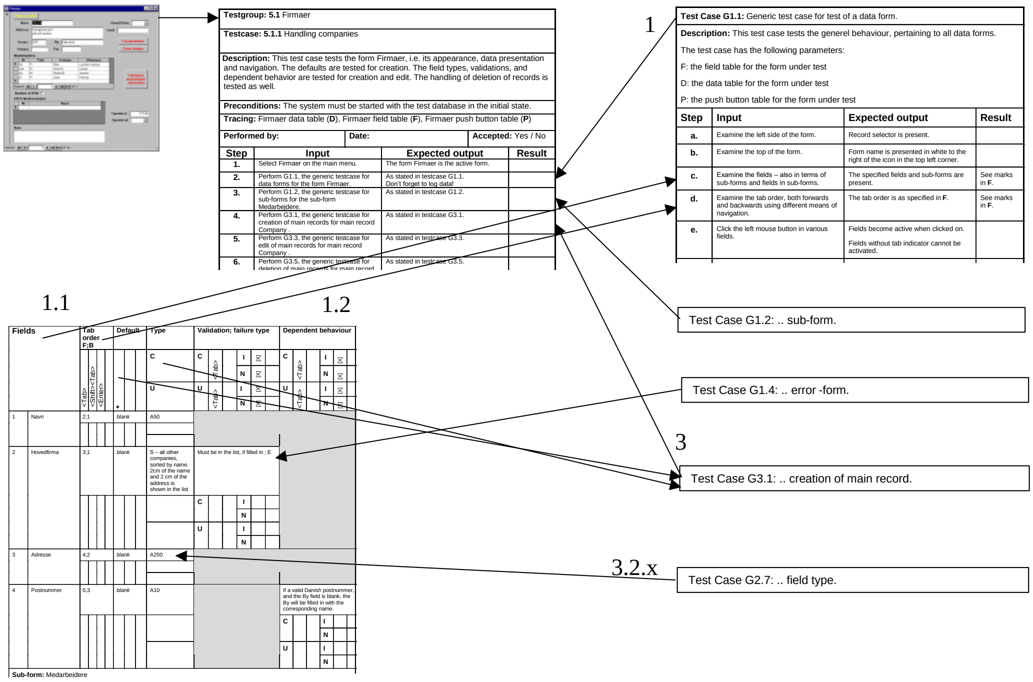


Figure 11: Overview of Test

The amount of work involved in performing a full test for a form may be deceptive when you first look at the test script for the form and the test tables, i.e. it looks like less work than it really is. The form specific test script refers to generic test scripts, which refer to the test tables. These refer indirectly to other generic test scripts. It is important when planning the actual test performance that this is understood.

5 *Experiences*

The FIRST method has been used in a number of different compnaies on very different systems, including an Access-based system, a number of Web-applications, a system with a characterbased interface etc.

The general experience is, that specifying a forms based system is a lot of work. This has nothing to do with this method, but is a fact of life. The method however highlights this fact, but it does it early in the specification phase and not far into the coding activity when it is too late to do anything about it.

The tables provide a way to specify the enormous amounts of states and state transitions in a Windows system in a very compact way. Describing each state and each transition separately would be a very time and paper consuming exercise. Describing the functionality in plain text is apparently not so time consuming, but it is very often incomplete and much less precise and consistent, and doing it in a complete, precise, and consistent in plain text is more time consuming than using tables.

The tables may be a little hard to understand when the analysts, developers, and testers are first presented with them, but once they get the hang of it, the tables reduce the need for very verbose specifications and test scripts. The tables are much easier to maintain than long and complicated texts.

The tables provide an easy overview both of what has been specified, of what has not yet been specified, of what to test, and of what has been tested.

The extended use of tables has a positive impact on the design. The design will be more structured as the analyst, the designers, and the testers thrive to extract and define the generalities in the behaviour of the system, and the user interface is more likely to be uniform through out the system.

The support people may use the tables to facilitate the support job. The tables offer an easy way to determine if a reported incident is in fact an error in the system or if a misunderstanding or mis-expectation causes it made by the user.

The use of generic test cases and tables means that the test specification is a lot easier to write and to maintain than classic test specifications. Changes are only to be implemented in one place.

Even if the generic test scripts are prepared manually they will provide a solid ground for an easier automation of the test, if that is wanted or needed. We have not had the opportunity to try this yet, but we would very much like to be able to in the future. Some preliminary studies are very encouraging.

6 Literature

None

7 Author CV

Anne Mette Jonassen Hass

Mrs. Anne Mette Jonassen Hass, M.Sc.C.E., has more than 20 years experience in IT. She has been involved in all aspects of software development: analysis, design, coding, test, quality assurance, and management. Mrs. Hass has worked in various types of business such as hospitals, the oil industry, telecommunication, hardware producers, and the space industry, in Denmark, Norway, England, and France.

Since 1995 she has worked as a consultant in Software Process Improvement, assessments, configuration management, and test. Mrs. Hass has been involved in several process improvement projects in different companies. She also performes third party test and validation of software, especially safety critical software. Mrs. Hass is a certified BOOTSTRAP V.3.0 lead assessor, having performed more than 30 BOOTSTRAP assessments in Denmark, Poland, and Canada for companies of all sizes and in many different branches.

Mrs. Hass has resently published a book about professional configuration management.

Process guides as software process improvement in a small company

Nils Brede Moe[1], Torgeir Dingsøyr[1], Tore Dybå[1], and Trond Johansen[2]

[1]SINTEF Telecom and Informatics
[2]FIRM

Abstract

Many small companies have problems making use of software process improvement methods and techniques, which were originally developed for larger companies. This is due to their organisational structure, existing software process improvement models and to the markets in which small companies operate. In this paper, we look at how a small company changed their software development from a chaotic to a more organised manner, by introducing an electronic process guide. We describe the problems developers and managers experienced in the company, characteristics of the electronic process guide that was introduced, and the resulting changes in the software development process. Finally, we describe why FIRM succeeded. The key success factors of FIRM can be used as recommendations to small companies when introducing electronic process guides.

Keywords

Electronic process guide, software process improvement, action research, small software organisation

1 Introduction

Software process improvement (SPI) has attracted a lot of attention over the recent years from industry and the research community. It is concerned with changing the way we organise software development to make the development process more effective. The goal of SPI is to achieve improvements in one or several of the following: Software Quality, production cost and development time.

Software development is performed in a rather ”immature” fashion in many companies, and problems of late delivery and unsatisfactory are not uncommon. The need to improve the software development process applies to all companies.

However, in the literature on software process improvement, the main focus has been on establishing more formal development processes – something that is applicable mainly to large companies. We believe that small companies also have a need to improve, but need to improve in a different manner.

In this paper, we discuss how FIRM – a small company that develops software for market research - has developed and used an electronic process guide, in order to improve the quality of their software development process. First, we discuss the SPI problems that are specific to small companies. By small companies, we mean companies with less than 50 developers. Then we define what we mean by an electronic process guide, describe the company FIRM in more detail, and discuss the status of software development in FIRM before and after the process guide was established. We conclude by indicating which of the actions in the work on the process guide were successful, and present what alternatives the company has for further development of the process guide.

1.1 Process Improvement in Small Organisations

What is special about process improvement in small organisations? Such companies have been described as “creative”, “energetic” and “at times, hectic” [1], and are usually companies we would place in level one on the CMM (Capability Maturity Model) [9] scale. Two problems with improvement in this kind of companies are, as Brodman and Johnson [2] wrote: “small businesses are faced not only with a lack of resources and funds required to implement many of the practises stated in the CMM, but also with the task of basing their process improvement initiatives on practises that do not apply to a small business and smaller software organisations”.

Demirors and Demirors [6] divide problems into three groups. They claim that small organisations have difficulties related to the:

- *Organisational structure* – small companies often lack quality conscious personnel, have a limited number of personnel and funds, may have a “hacker culture” in the organisations and are often immature when it comes to other processes than the software-specific ones as well.
- *Software process improvement models* – the models lack guidance for small companies – what to prioritise and how to proceed. Such companies can also lack knowledge required to work with quality issues. Further, models for software process improvement fail to focus on the benefits of small organisations.
- *Market, where small organisations operate* – small software companies are usually developing software for fewer customers than larger ones, and operate closer to their customers. When the needs of the customer(s) change, the small company also has to change the software. Also, the return of investment on the software is lower when having fewer customers.

Of course, strengths in small companies are that it is easy to involve everyone in the company in initiatives. Hence it is much easier to change the practise than in larger companies. The challenge is, to balance the refinement of the existing skill base with the experimentation of new ideas to find alternatives that improve on old ideas [10].

1.2 The Role of Process Guides in SPI

Effectively disseminating process knowledge to process participants is crucial in any SPI effort – whether it is based on an analytical approach, such as the QIP ([6] and [7]), or a benchmarking approach, such as the CMM [9]. In any case, process participants need effective guidance when process conformance is important, when a process changes frequently, when new personnel join a project, and so forth.

Traditionally, this has been the realm of large organizations, and the way of describing and communicating processes has focused on printed guidebooks, standards, and procedure handbooks. However, such handbooks are more often seen as dust collectors than SPI facilitators, and especially so in smaller companies. There could be many reasons for this. In our experience, the main reason for this situation is that most handbooks are too large, too complex, and too expensive to maintain.

For process guides to be useful, they must not only be tailored to the specific needs of individual companies, but also be made available on the company's intranet. This way, searching and navigation will be made much easier and it will also be easier to integrate the process guide with the concepts of an Experience Factory [6], providing easy access to supplementary information such as templates, checklists, examples, and experience packages. Thus, the traditional process handbook shifts from a bulky pile of paper to a flexible on-line structure allowing easy access to all relevant information.

A process guide can be seen as a structured, workflow oriented, reference document for a particular process, and exists to support participants in carrying out the intended process [8]. Whether in the form of a printed handbook or an electronic version, a process guide should include the following basic elements:

- *Activities*: descriptions of "how things are done", including an overview of the activities and details regarding each individual activity.
- *Artefacts*: details regarding the products created or modified by an activity, either as a final or intermediate result of the activity or as a temporary result created by one of the steps.
- *Roles*: details regarding the roles and agents involved in performing the activities.
- *Tools and Techniques*: details regarding the tools and techniques used to support or automate the performance of an activity.

Based on these elements, Kellner et al. [8] have proposed a set of basic requirements and design principles for electronic process guides (EPGs). Most importantly, an EPG should provide all the information elements and relationships contained in a good paper-based process guide. In addition, it should capitalize on diagrams, tables, and narrative to provide an effective user interface. Also, it should make extensive use of hyper-links to support flexible navigation and direct access to supporting information such as examples and templates.

In the remainder of this paper, we will show how FIRM has analysed its processes, suggested improvements, and implemented an EPG as the basis for further improvements in its software processes.

2 Research Method

The research reported in this article is a case study of FIRM, which was performed in a large Norwegian research project called Process Improvement for IT Industry (PROFIT), where many companies cooperate with research institutions and universities in improvement activities. The collaboration is based on finding common improvement and learning goals, and working together to obtain the goals. The communication between contact persons in the companies and researchers is through meetings, telephone calls, and e-mail discussions. In addition, the researchers have gathered data through the following activities in FIRM:

Semi-structured interviews – we have interviewed seven developers, two employees in the quality department and one general manager in the company about the status of development work before and after improvement activities. The interviews lasted 20-30 minutes, and have been recorded and transcribed in full for the analysis.

Post mortem Analysis – we have organised structured group discussions (or Postmortems, see [4]) on strengths and weaknesses in the improvement initiatives, and on the process model. Both developers and the quality department participated in these discussions, and the researchers wrote detailed minutes.

Informal survey – seven system developers and two general managers were asked about their usage of the process model in an internet-based survey. A 5-point, bipolar Likert Scale was used.

The analysis of this information was done by categorizing answers to questions and other information. We use quotes from the interviews in the presentation of the results.

This research method is a kind of action research [5], where the researchers and participants from the companies have common goals: To improve software development in different manners, and learn from that experience. Potential problems with this kind of research are that it can easily be biased, in that everyone is interested in reaching the goals that are set up. Thus, we do not know if the same results would be achieved with another set of researchers, with other people from the company, or with another company in the same situation. But this kind of research is a way to get interaction with companies in a way that would not be possible if it was not so much in the company's interest.

3 Company description

FIRM (Future Information Research Management) was established in 1996 by the former owner of Feedback Research, which was one of the leading market research companies in Norway. The business idea was to revolutionise the market research process by launching a software/web based tool that automated the marked research processes. The result was the product Confirmit.

At present, five out of the 10 world’s largest market research companies are FIRM customers. In addition, FIRM has close to 200 customers operating in industries other than market research. Such customers include PriceWaterhousCoopers, J.P. Morgan Chase Bank, Dow Jones, Economist, Financial Times and AirMiles. Today, FIRM has 60 employees (15 in the R&D department) in 5 offices: Oslo, New York, San Francisco, London and Stockholm.

FIRM’s mission is to empower organizations to maximize Return on Relationships (ROR) by continuing to provide the world’s best survey and dialog solution.

Information and Research Management – (IRM) is the concept of integrating an individual feedback tool (Confirmit) with the enterprise's information systems (Figure 1). IRM develops value added individualized dialogues and feedback with customers, prospective customers, employees and other stakeholders.

Existing information stored in databases (IM) can be used to have an individual dialogue with e.g. employees (RM) through Confirmit in order to add value to the Management Information system (Figure 1).

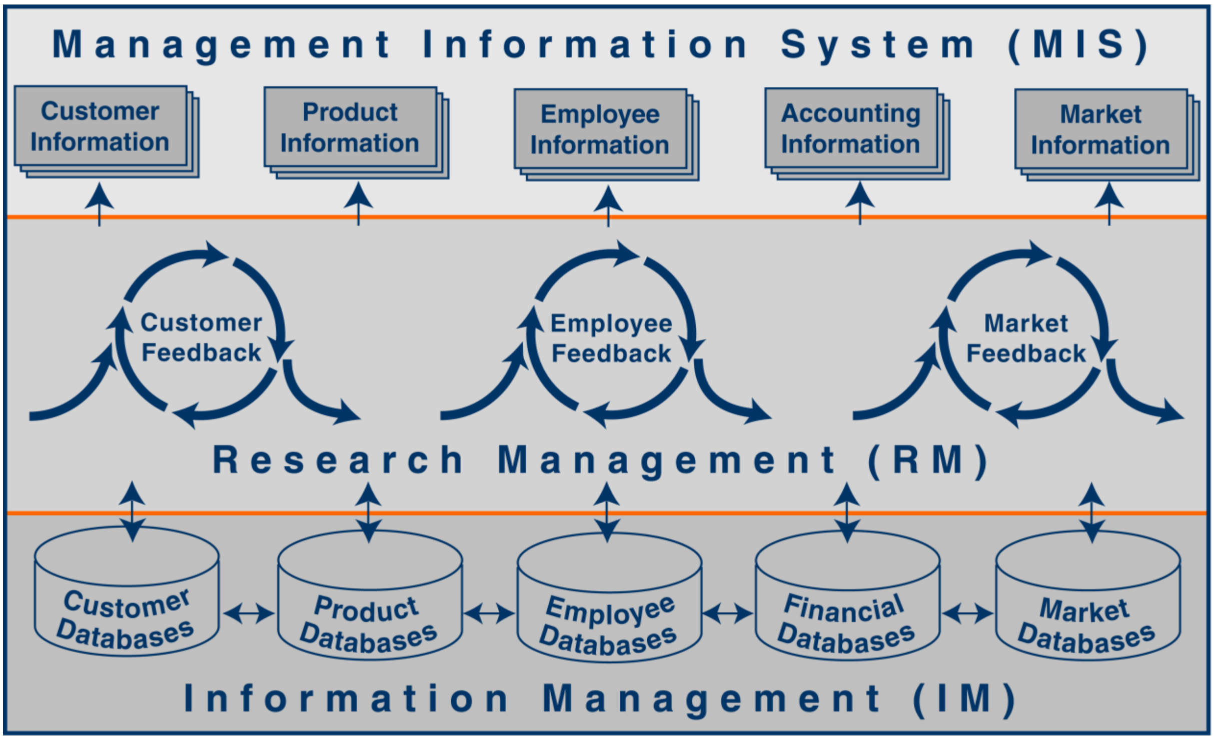


Figure 1: Information Research Management

4 Use Of Process Guides in FIRM

First, we will describe how the situation was in FIRM before they started with SPI activities. Then, we shortly describe the activities done to improve the development process in FIRM, before we describe the electronic process guide defined by FIRM and the results of introducing this guide.

4.1 Starting Point: "reach as far as possible using the shortest possible time"

In the autumn of 1999, FIRM's system development department organised an internal seminar. The intention of this seminar was to discuss the future of the development department in Firm. This was needed because Firm was confronted with two big challenges:

- Doubling their numbers of developers.
- Start selling the finished product, including upgraded versions and technical support.

Up to this point, FIRM's strategy was: "to reach as far as possible using the shortest amount of time". A general manager said: "When we were only a few developers, and we all could sit around the same table, things were OK. The way we work today will not work when we hire more people". At this point, the developers experienced the development as: Ad-hoc, stressful, no coordinating of work, no documentation on the product, no time for testing, and a product with low quality.

One of the first new system developers said:

> "The software was not documented, and only a few persons knew the code thoroughly. There were no routines for system development activities, and every release resulted in a chaotic situation".

Another developer said:

> "Another problem with not having a process is that you easily got squeezed between several projects because everything had the highest priority. Thus, there was no time to do the tings you really ought to do".

One from the quality department said:

> "All the knowledge was inside the head of the developers. When you read the two first sentences describing level 1 in CMM, this describes how FIRM was at this stage".

Because of lack of time and resources nothing was done to solve the problems in the development department until August 2000. The main prioritised problems to be solved were:

- Establish a more professional development process. A professional process may lead to a more structured working day, and higher job satisfaction among the employees.
- Deliver a product with higher quality. This is required in order to be evaluated as a serious supplier in a market with demanding customers, the product must be of high quality.
- Bring in more resources (both developers and computer equipment).

4.2 Activities

The process improvement activities started with establishing a quality department, and was followed up by performing an informal CMM assessment, and finally, establishing a process model for software development in the company.

Quality department – three people were allocated to the quality department in the company, to work with software process improvement activities such as improving the test process, and also as project managers.

Informal CMM assessment – an informal CMM assessment was performed, which placed the company on level one. This was done to get an objective view of the status on the software processes.

Introduction of process guide – a pragmatic electronic process guide was developed by the quality department with help from some developers and a CMM consultant. The guide is available on the company intranet, and is lean and pragmatic. A quality manager said: "the most important thing is what the model contains, not the model itself". He also said they "started with issues on level three before we finished all issues on level two – so we have not followed the CMM recommendations completely". Another employee in the quality department commented: "The focus has moved from getting a CMM classification to getting a process that functions internally, and I think that is perfectly all right".

We will describe the process model more detailed in the following.

At this point, 17 persons were working in the R&D department.

4.3 Processes defined

FIRM developed an electronic process guide (Figure 2), and placed it on the intranet. When a new project is started, a process model with templates (project plan, requirement specification, design, post mortem analysis etc.) is automatically generated for the project. The sub processes (Figure 3) in the model can be accessed from the top level. All the documents belonging to a project are accessible through the model.

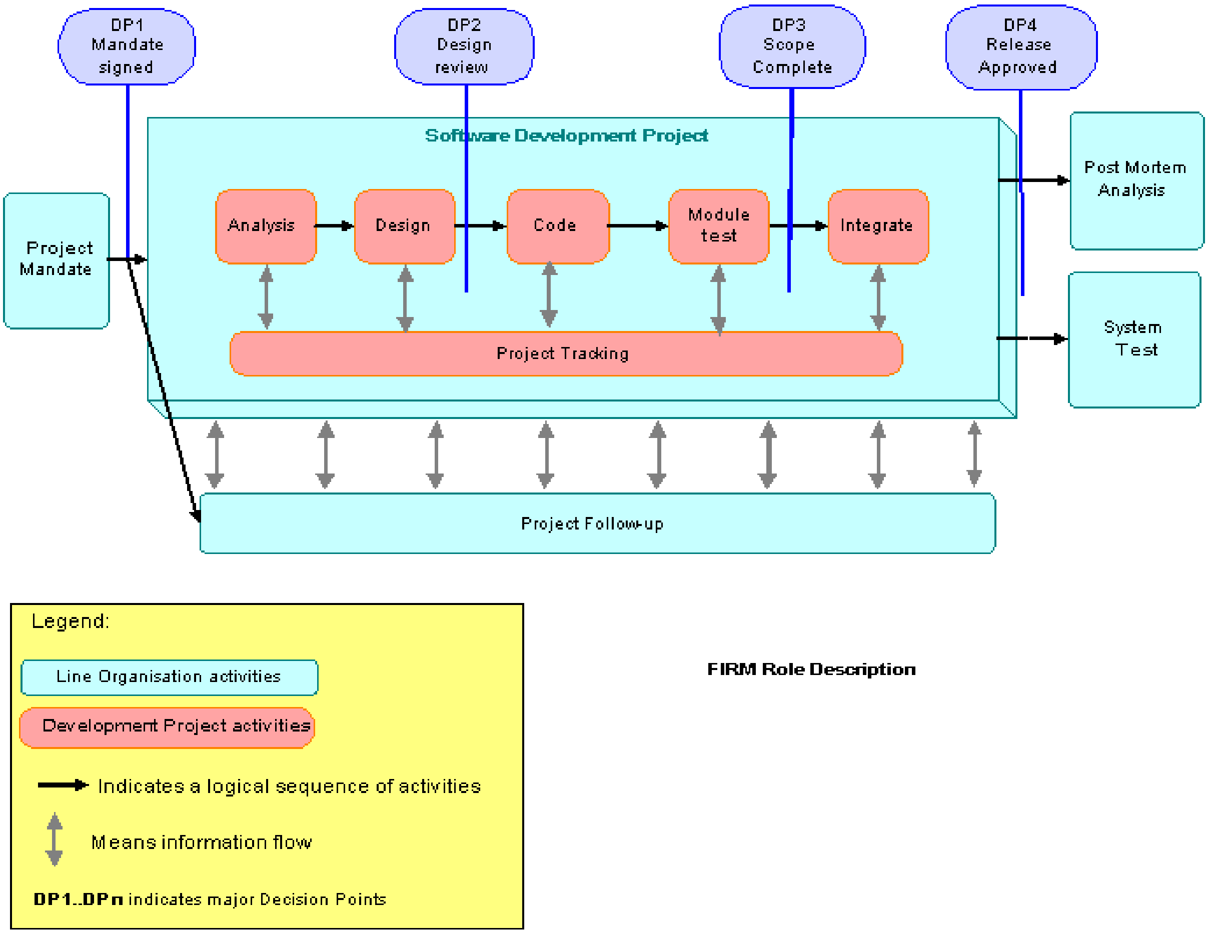


Figure 2 The electronic process guide on the intranet.

The electronic process guide includes the following elements:

- *Activities*: Process model with the process descriptions. Each process explains, “how tings are done” through a workflow description.
- *Artefacts*: Required and optional input and output documents to each process.
- *Roles*: Description of the roles involved in performing the activities.
- *Tools and Techniques*: Examples and references to manuals.

The electronic process guide should be interpreted the following way:

The line organization is responsible for agreeing on a project mandate, continuous project follow-up, system testing, and post mortem analysis. The rest of the activities are development activities.

The main purpose of the project mandate is to act as an agreement between the line organization and the project with respect to resources and deadlines. Project follow-up is performed on weekly project management meetings. All finished projects are subject to a post mortem analysis managed by the Quality Assurance (QA) group. In such a meeting, all project members are supposed to participate and discussing experiences from the project. The results from this meeting are valuable input to future projects. If a project is part of a major release, its functionality will be included in a system test.

The development project activities are organized according to a iterative waterfall approach. In the analysis process, all requirements, both functional and non-functional, are collected and analysed. The major output of this process is the Software Requirement Specification (SRS), which is signed by both FIRM and the customer. By clicking on the Analysis box in the electronic process guide, a thorough description of this process is presented with a step-by-step walk-through and all necessary templates for input and output documents. In the Design process, the requirements are analysed in more details, and a functional and technical design is developed. The design typically includes use case diagrams, class diagrams and sequence diagrams. The Code process involves the actual implementation of the design. The coding is performed according to a coding standard, which, of course is accessible from the Code box in the electronic process guide. The Code process also involves code review. All development projects in FIRM undergo thorough module testing. Test conditions covering all requirements in the SRS are written, and based on these, test cycles that verify them are written as well. Both a guideline on how to write test conditions and test cycles, and a test manual that describes how the test documentation is implemented and how the test is executed, is described in the electronic process guide. The final activity for the development project is integration. All development projects in FIRM has its own view of the code base on which the new code is implemented. This development view must be merged with the main view for the software product, on which new versions of the software is built. Most of the development project activities are closed by going through a checklist.

This is how the first part of the code process appears on the intranet.

FIRM Process description

Process Name: Code process
Process Owner: Configuration Manager

Purpose

The purpose of the code process is to ensure that coding is handled in a unified way. It is important that the coding standard is followed to allow different developers to understand code not written by themselves. The test conditions and cycles are crucial for the testing of the system before release.

Roles and Responsibilities:

Role	Responsibility
Project Manager	PX
Test Responsible	I
System Developer	X

Input documents

Requirement Document, Design Document, Coding Standard

Workflow

1. Create Test Conditions

- Prior to coding test conditions should be written.
- The sum of all test conditions shall give a complete description of the system that is tested. A test condition is on the form "if <action> then <expected results>", where <action> represents an action a user performs on the system, and <expected results> is the corresponding expected result to that action.
- The design document should be used when writing the test conditions.
- Test conditions for the module test should be written on the project called MT_confirmit in TestDirector.
- Test conditions should be organised in a logical way according to the existing module structure in confirmit.
- For a more detailed description see Test Manual, chapter 4. "Adding test conditions".

Figure 3 The first part of the code process

4.4 Status today: Less stress and less fire fighting

Today, February 2002, the EPG has been in use for over a year, and a lot of new processes has been implemented.

The developers now experience the development as more mature and more formal, with better structure and with a realistic time schedule. They used a common development model to produce a product with fewer errors, and they have satisfied customers.

One of the system developers said:

> "Now it is much less stress and less fire fighting. It is less stress even though we do a lot more today than before, and we have more customers who all give us a lot of input".

Another developer said:

> "We have found the way that works for us. I think that is very good".

Yet another developer said:

> "Without having a defined process it would not be possible to run all the projects we do today. We can also relax more because you know that issues are taken care of, and there are few surprises".

Another developer said:

> "In my former job we wrote a lot of documentation, which was never used. This is not the case here".

One from the quality department said:

> "Today you can feel more safe about the projects reaching the goals on schedule"… "When we delivered (version) 6.0 on time, we got mails from managing directors in some of the companies we delivered to. They gave us big credit. They said: Finally there was a software company that delivered on time".

The developers said the following about the EPG: Gives a good overview, simple to use, a place to put all the project documentation, increases and secures the quality in the project, results in more efficient use of resources, and offers a common work-practice.

Especially the analysis, design, post mortem analysis, test and code phases worked well.

An informal survey was performed to document the developer's relationship to the SPI work and the electronic process guide. The main findings were:

- Everybody felt they were involved in the SPI work, and that the goal and content of the SPI activities were very well communicated
- 75 % felt the EPG was very easy to use. 25% felt it was neither easy nor difficult to use.
- 25% percent used the EPG often. The rest used it from time to time. Some developers did not need the EPG in periods.
- 45% felt that the tools and techniques were easy to access through the EPG. The rest found the tools and techniques neither difficult nor easy to access from the EPG.
- Everybody agreed that the EPG leads to higher quality, and that it makes it easier to take over other developers work.
- It is a common understanding that the work gets a bit more bureaucratic but the creativity is not reduced.

The development of the EPG will not stop. The developers want to improve the following:

- It must be possible to tailor the guide to a specific project. The guide does not support the many small projects in FIRM today.

- The model does not support experience sharing.
- Some of the processes are not described deeply enough.
- The user interface of the EPG can be improved.

The EPG will be developed through periodic reviews of the guide. But FIRM has to make cure their development does not become to document driven. One developer said: ” We must be careful so we do not get to document driven, and end up using all the time on administration and not on real work. We are doing well now, but we are getting closer to the limit of how much documentation we can handle.”

5 Conclusion

FIRM has changed from a chaotic to an organised placed to work. There are fewer bugs in their product, they achieve higher productivity, they deliver on time, and the developers are more satisfied now than they were before. The main reason for this is the development and implementation of a standard work process. The process is implemented using an electronic process guide. Implementing an electronic process guide is a very difficult task, with a significant risk of failure. However, the implementation went well at FIRM, and the developers use the guide and are satisfied with it. FIRM succeeded because:

- The EPG is simple. Development of the process guide started with the key processes, and the process descriptions are short. There is extended use of examples.
- The management has focused on process improvement.
- Everybody has been involved
- The guide is easily accessible on the company intranet.
- The EPG is integrated into a project status system, to provide support for process tracking.

Future work will focus on integrating the EPG with the concepts of an experience factory, to provide easier and more flexible access to templates, checklists, and concrete project examples. Work will also be done to enhance the system in such a way that the EPG can be instantiated and tailored to the specific needs of individual projects. The use of special purpose tools for process modelling, such as Spearmint, is a possible approach.

6 References

[1] K. Kautz, H. W. Hansen, and K. Thaysen, "Applying and Adjusting a Software Process Improvement Model in Practise: The Use of the IDEAL Model in a Small Software Enterprise," presented at International Conference on Software Engineering (ICSE), Limerick, Ireland, 2000.

[2] J. G. Brodman and D. L. Johnson, "What Small Businesses and Small Organizations say about the CMM," presented at 16th International Conference on Software Engineering (ICSE), 1994.

[3] O. Demirors and E. Demirors, "Software Process Improvement in Small Organizations," 1998.

[4] A. Birk, T. Dingsøyr, and T. Stålhane, "Postmortem: Never leave a project without it," *IEEE Software, special issue on knowledge management in software engineering*, 2002.

[5] D. J. Greenwood and M. Levin, *Introduction to Action Research*: Sage Publications, 1998.

[6] Basili, V.R. (1989) Software Development: A Paradigm for the Future, *Proceedings of the 13th Annual International Computer Software & Applications Conference (COMPSAC)*, Keynote Address, Orlando, FL, Sept.

[7] Basili, V.R. and Caldiera, G. (1995) Improve Software Quality by Reusing Knowledge and Experience, *Sloan Management Review*, Vol. 37, No. 1, pp. 55-64.

[8] Kellner, M.I., Becker-Kornstaedt, U., Riddle, W.E., Tomal, J., and Verlag, M. (1998) Process Guides: Effective Guidance for Process Participants, *Proc. 5th Int'l Conf. on the Software Process: Computer Supported Organizational Work*, Lisle, Illinois, USA, 14-17 June, pp. 11-25.

[9] Paulk, M.C., Weber, C.V., Curtis, B., and Chrissis, M.B. (1995) *The Capability Maturity Model: Guidelines for Improving the Software Process*. Reading, Mass.: Addison-Wesley.

[10] T. Dybå (2000) Improvisation in Small Software Organizations, *IEEE Software*, Vol. 17, No. 5, pp. 82-87.

7 Author CVs

Nils Brede Moe

is a research scientist at the Department of Computer Science at SINTEF Trondheim, Norway. He joined the department in 1998. He holds a M.Sc. degree in Informatics from the Norwegian University of Science and Technology in 1998. He has published papers on health informatics (telemedicine and homecare), users cantered design and process improvement.

Torgeir Dingsøyr

is a research scientist at the Department of Computer Science at SINTEF Trondheim, Norway. He wrote his doctoral thesis on “Knowledge Management in Medium-Sized Software Consulting Companies” at the Department of Computer and Information Science, Norwegian University of Science and Technology. He has published papers on knowledge management in software engineering, case-based reasoning and on software engineering education. In 2001, he spent half a year as a guest researcher at the Fraunhofer Institute for Experimental Software Engineering in Kaiserslautern, Germany.

Tore Dybå

received his M.Sc. degree in Computer Science and Telematics from the Norwegian Institute of Technology in 1986 and his Ph.D. degree in Computer Science from the Norwegian University of Science and Technology in 2001. He has worked as a consultant and a researcher both in Norway and in Saudi Arabia. His research interests include empirical software engineering, software process improvement, knowledge management, and organizational learning. Dr. Dybå is currently a senior scientist at Department of Computer Science at SINTEF and a visiting research scientist at Department of Software Engineering at the SIMULA Research Laboratory.

Trond Johansen

received his M.Sc. degree in Marine Technology from the Norwegian Institute of Technology in 1997, with main thesis on how to utilise IT in Marine operations. He has worked as Product and Process -Manager within software development for the last 3 years. Main focus for his work has been: software process improvement, software testing, project management etc. Mr Johansen is currently QA & Process Manager in a software company; Future Information Research Management AS - FIRM.

ISO/IEC 15504 Requirements for Compatible Software Process Assessment Models

Marion Lepasaar, Timo Mäkinen

Abstract

Recently many widely known software standards have been subject to change, ISO/IEC 15504 among them. The most significant requirements this emerging international standard describes are the need to establish a complete and unambiguous mapping between a compatible model and the process reference model of ISO/IEC 15504-2. The requirements for ISO/IEC 15504 compatible assessment model are described through structural analysis of ISO/IEC 15504-2 and illustrated with examples from ISO/IEC 15288 and ISO/IEC 12207.

Keywords

Process reference model, process assessment model, ISO/IEC 15504

1 Introduction

The complexity of man-made systems has increased to an unprecedented level. This has led to new opportunities, but also to increased challenges for the organisations that create and utilise systems. These challenges exist throughout the life cycle of a system and at all levels of structural detail. [9]

One of the challenges for software process management is its need for a more structured approach. The attempts to automate the software process are motivated by the need to improve the quality and productivity of software engineering work. When we can reduce a task to a routine procedure and then mechanise it, we not only save labour but also eliminate the source of human error – which is the most effective way to improve productivity. Framing an automation strategy requires knowledge of what is needed, awareness of what is feasible and a long-term commitment to investment in software process improvement. [5]

The interest in software process improvement has created dozens of international software process models and standards like ISO 9000 series [6], BOOTSTRAP [1], CMM [2] and ISO/IEC TR 15504 [8]. It is important that standards are kept current and maintained so that they wouldn't become less pertinent to working conditions and their enforcement progressively less practical [5].

Recently many widely known software standards have been subject to update, like ISO/IEC 12207 [7], CMMI [3], ISO 9000 series, to mention a few. Among them is the ISO/IEC TR 15504, often referred to as SPICE, which is currently being updated by the joint technical committee of the International Organisation for Standardisation (ISO) and International Electrotechnical Commission (IEC). The aim of the emerging standard of ISO/IEC 15504 is to set the requirements for constructing a compatible assessment model.

In this paper we visualise the requirements for a ISO/IEC 15504 compatible assessment model by describing the structural elements of ISO/IEC 15504. We also illustrate the structure of the emerging standard with examples from ISO/IEC 12207:1995/FDAM 1 and ISO/IEC CD 15288 FCD. In doing so we show a way in which conformity of assessment models can be found.

The goal of this paper is to illustrate the ISO/IEC 15504 requirements for compatible Software Process Assessment Models through its structural analysis.

The paper is divided into four chapters. The first chapter gives an overview and background of ISO/IEC 15504 and describes the structure of the emerging standard. The second chapter describes the structure of Process Reference Model with examples from ISO/IEC 12207 and ISO/IEC 15288. The third chapter describes the structure of the Measurement Framework of ISO/IEC 15504, illustrated with examples also from ISO/IEC 15504. We have used the Unified Modelling Language's [12] class diagram notation to describe the structure of the models. The fourth chapter draws the conclusions about the standardised requirements of SPICE concerning the compatible assessment models.

2 ISO/IEC 15504

ISO/IEC TR 15504 is also known as SPICE, the term, which originally stands for the initiative to support the development of an international standard for software process assessment. In 1998-9 the International Organisation for Standardisation published the ISO/IEC TR 15504. An international standard ISO/IEC 15504 should be published in 2002-3 by ISO [11]. The aim of the standard is to provide a shared approach for process assessment and lead to a common understanding of the use of process assessment for process improvement and capability evaluation.

The primary importance of ISO/IEC 15504 is that it sets requirements for constructing an assessment process and a compatible assessment model that addresses unique needs of an industry or an organisation [4]. The most significant requirements that the emerging international standard describes are the need to establish a complete and unambiguous mapping between the compatible model and the process reference model of ISO/IEC 15504-2, and to develop a mechanism for translating the outputs from an assessment into the standard process profiles defined in ISO/IEC 15504 [10].

The ISO/IEC TR 15504 consists of nine parts, where Part 2 and Part 3 are normative. Part 2 is directly aligned to ISO/IEC 12207 providing an overall contextual framework for software lifecycle processes, and the process dimension of the reference model is closely mapped to this framework. Part 5 provides an exemplar model for performing process assessments that is based upon and directly compatible with the reference model in Part 2. [8]

In the emerging International Standard of ISO/IEC 15504 there will be only five parts, as a result of the substantial simplification and reduction of the earlier Technical Reports. The only normative part of the standard is part 2, which provides a framework for process assessment and sets out the minimum requirements for performing an assessment in order to ensure consistency and repeatability of the ratings. [4]

Part 2 of ISO/IEC 15504 contains all normative components of the emerging standard. Other parts of this international standard contain guidance that will provide a more detailed understanding on the subject. Part 1 describes the concepts and vocabulary of the ISO/IEC 15504. Part 3 provides guidance on performing an assessment and Part 4 describes how the assessment results can be used. Part 5 is an exemplar process assessment model aligned with and satisfying all the requirements set in Part 2. The integration of the old documents into the new document sets is shown in figure 1. The larger boxes represent the documents of the emerging standard and the smaller ones the current ISO/IEC TR 15504 document set. [4]

ISO/IEC 15504 can also be divided into a set of guides and models. Part 2 describes the reference model and Part 5 an exemplar assessment model. The rest of the five parts are guides of how to use and interpret these models. In our article we will describe the reference model, illustrating its structure with examples also from ISO/IEC 122207 and ISO/IEC 15288.

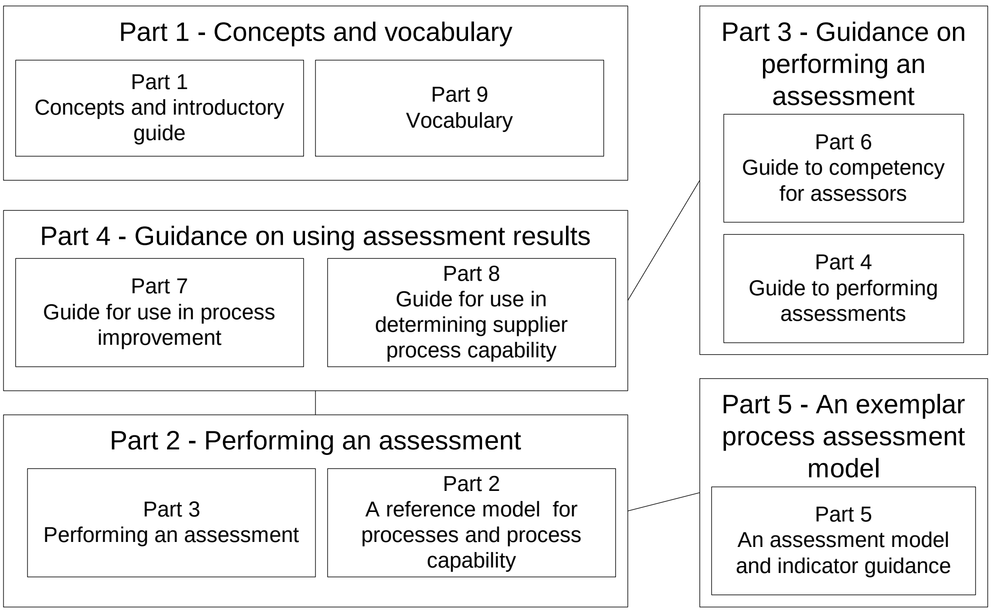


Fig. 1 – The integration of old documents into a new document set of ISO/IEC 15504

ISO/IEC 15504 defines the minimum set of requirements for performing an assessment that will increase the likelihood that results are objective, impartial, consistent, repeatable and representative of the assessed process. In order to increase the repeatable attribute ratings for assessed processes the ISO/IEC 15504 Part 2 sets out the requirements for assessment conformant with this international standard. The requirements help to ensure that the assessment output is self-consistent and provides evidence to substantiate the ratings. Fig. 2 below shows the logical organisation of the major elements of the process assessment process. [4]

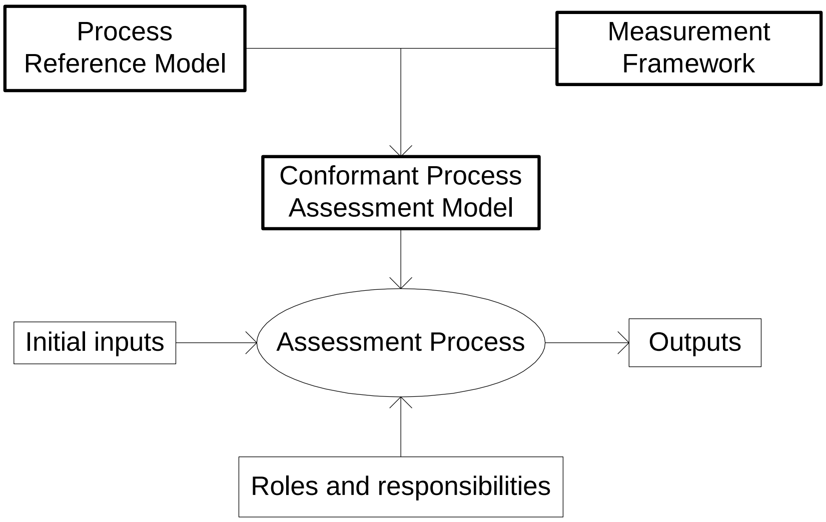


Fig. 2 - The major elements of the assessment process [4]

An integral part of conducting an assessment is to use a process assessment model constructed for that purpose, derived from a process reference model and compatible with the requirements of measurement framework. The measurement framework for the assessment of process capability is defined in ISO/IEC 15504-2. A process reference model provides definitions of processes in a life cycle described in terms of process purpose and outcomes, together with an architecture describing relationships between the processes. One of the process reference models in accordance with requirements of ISO/IEC 15504-2 is ISO/IEC 12207. A conformant process assessment model embodies the core characteristics that could be expected of any assessment model, which claims to be consistent with ISO/IEC 15504-2. An exemplar process assessment model has been described in ISO/IEC 15504-5, containing good software engineering and management practices to be considered when interpreting the intent of the process reference model. [4]

One way to look for the conformity of an assessment model is to illustrate the structural elements of that model and compare them to the requirements of ISO/IEC 15504-2. In order to do that we first need to visualise the structure of the requirements set in SPICE.

In the following chapter we describe the requirements of Process Reference Model in Unified Modelling Language class diagram notation. We will also give an example of how the requirements of the reference model have been met based on ISO/IEC 15288 FCD and ISO/IEC 12207:1995/FDAM 1.

3 Process Reference Model

Process assessment model has two dimensions, a process dimension and a capability dimension. The process dimension is provided in an external Process Reference Model, which defines a set of processes, characterized by statements of process purpose and outcomes. The normative requirements for Process Reference Model have been described in ISO/IEC 15504-2.

The elements of Process Reference Model can be divided into two groups based on the nature of the requirements they describe: the elements describing the requirements for the reference model and those describing the requirements of process descriptions. Process Reference model has been illustrated in Fig. 3. The elements *context* and *community of interest* together with their associations illustrate the requirements for the reference model itself and elements *process*, *purpose*, and *outcome* with their subclasses and associations illustrate the requirements of process description. A model claiming conformity to SPICE must satisfy both of these sets of requirements.

Requirements for Process Reference Model are the *declaration of domain* and *scope* of the process reference model, a *description of the relationship* between the process reference model and the *context of its intended use*, *actions taken to achieve consensus* within the *community of interest* of the model. The *descriptions of the processes* are the fundamental element of Process Reference Model. There is another set of requirements for the processes. There has to be a *description of the relationship between the processes* defined in process reference model. Processes are described in terms of its *purpose* and *outcomes*, where the outcome statements describe either a *work product*, a *change of state* or *meeting of specified constraints*.

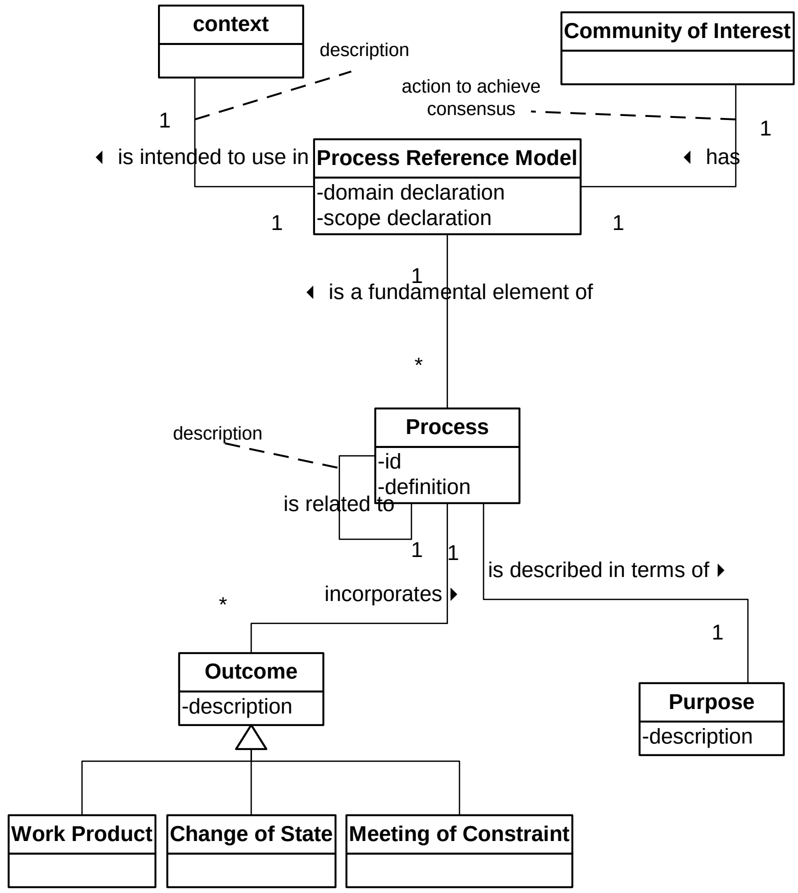


Fig.3 – Elements describing the requirements of Process Reference Model

ISO standards follow the same structure in their introductory part of the standard although they vary in terminology.

In ISO/IEC 15288 the requirements for Process Reference Model have been met in the chapters, which describe the purpose and the field of application of the standard. The domain declaration could be that this International Standard establishes a common framework for describing the life cycle of systems created by humans. The scope is defined through: this standard concerns those systems that are man-made and are configured with one or more of the following: hardware, software, humans, associated processes, procedures, facilities and naturally occurring entities. This standard can be used in the context of establishing business environments and supporting process assessment. The community of interest is described through the standard applying to both supplier and acquirer roles of systems. Although all the requirements for Process Reference Model have been met in ISO/IEC 15288, one has to analyse the introduction well to find all the elements. [9]

In ISO/IEC 12207 the requirements for Process Reference Model are more clearly described. The domain is declared by a sentence, which says that this standard establishes processes, activities, and tasks that are to be applied during the acquisition of a system that contains software, a standalone software product, and service, and during the supply, development, operation, and maintenance of the software. The scope is given through limitations of what the standard does not intend to prescribe.

Context is described through the listed fields where the standard can be applied. The community of interest is described under the term audience and focuses on the roles to which this standard has been written for. [7]

Thus, we can say that the requirements the ISO/IEC 15504-2 sets for the Process Reference Model are met by these two ISO standards.

An example of process description from ISO/IEC 15288 is given in figure 4 illustrating only the normative requirements described in ISO/IEC 15504-2 for Process Reference Model.

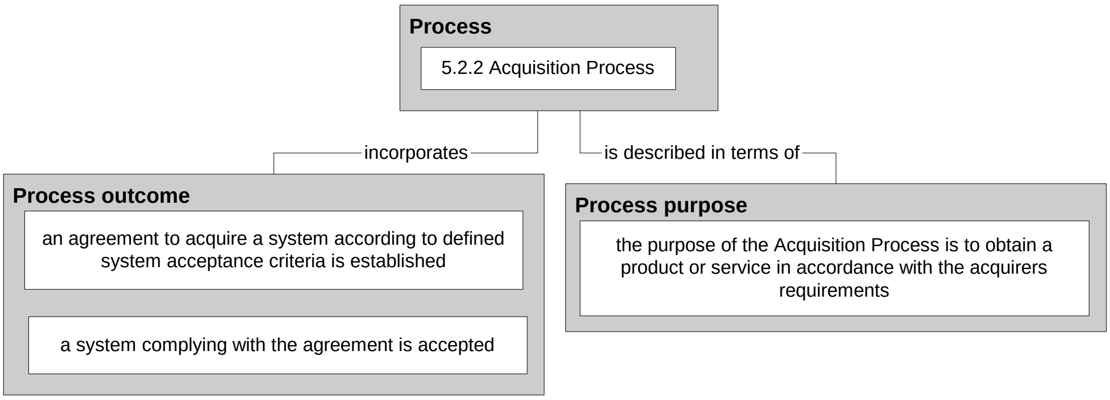


Fig.4 – An example of process requirements based on ISO/IEC 15288

The elements of the acquisition process from ISO/IEC 15288 (fig.4) are almost identical to the process elements of ISO/IEC 12207. Processes are described in both standards in terms of process purpose and process outcomes. ISO/IEC 15504-2 says that a process outcome can be one of the following types – production of work product, a significant change of state or meeting of specified constraints. In Fig. 4 there are two outcomes shown for acquisition process. The first outcome includes the outcome types of work product and the meeting of constraints. Work product is the *established agreement* and the meeting of constraint is *according to defined criteria*. In the second outcome *an accepted system* belongs to the type of work product and *the agreement* is the type of meeting of constraints. Although it is not required that the process outcome type was explicitly stated, it would help in finding compatibility of models when ISO/IEC 15504-2 specified how these outcome types contribute to the models.

Another requirement of processes is the description of the relationships between the processes. This requirement applies only when two processes have something in common. Nevertheless, it was surprising to see that in ISO/IEC 15288 there was not a single relationship between two processes and only a few of them in ISO/IEC 12207.

In ISO/IEC 15504-2 there is another requirement saying that a given set of process outcomes has to be sufficient and necessary in order to achieve the process purpose. It is very difficult to determine whether a set of outcomes for a process is both sufficient and necessary for achieving the purpose of the process.

4 Measurement Framework

The other dimension of the process assessment model is the capability dimension. Capability dimension has nine process attributes that are grouped into six process capability levels that define an ordinal scale of capability that are applicable across selected processes. The measurement framework for process capability has been defined in ISO/IEC 15504-2 (Fig.5). The measure of capability is based upon a set of process attributes. Process attributes are used to determine whether a process has reached a given capability. The attribute outcomes help to measure the attributes, as they indicate the full achievement of process attributes. The elements of the measurement framework have been given in Fig. 5 and an example of these corresponding elements has been given in Fig.6.

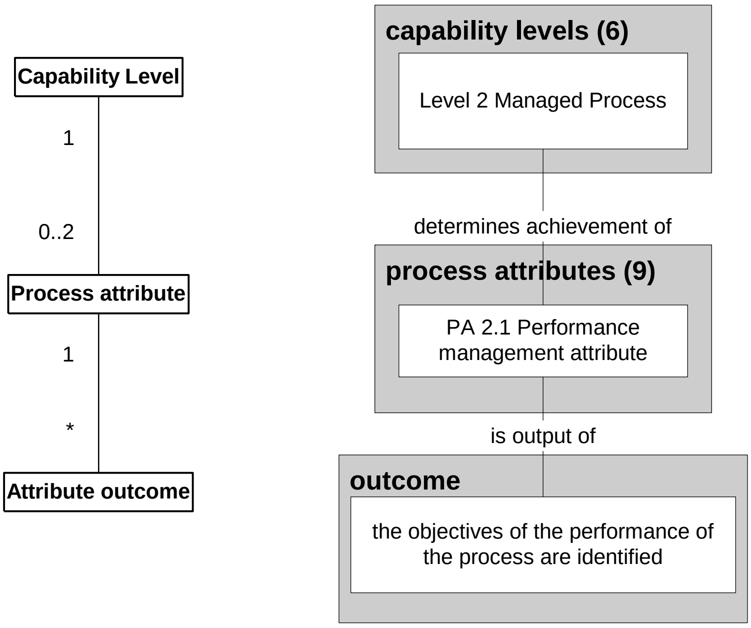


Fig.5 – Elements and an example of the measurement framework described in 15504-2

The measurement framework of the emerging standard has remained the same as it was in the ISO/IEC TR 15504.

5 Conclusion

In our article we analysed the requirements of the emerging standard of ISO/IEC 15504. ISO/IEC 15504-2 sets requirements for a compatible process reference model and provides a measurement framework for process capability.

As a result of the structural analysis of ISO/IEC 15504, we found that some requirements listed there leave flexibility for different interpretations, e.g. in determining whether an outcome is sufficient and necessary to achieve the purpose of the process. We have illustrated the requirements for compatible assessment model with examples from the ISO/IEC 12207 and ISO/IEC 15288 to see how they meet these requirements. The requirements have been met in both of the standards.

In the future we aim to analyse the structure of a compatible assessment model of the emerging standard ISO/IEC 15504. Structural analysis of process models is one way to see the compatibility between the requiremenets set in ISO/IEC 15504 and the detailed process models.

6 Literature

[1] Bootstrap Institute. http://www.bootstrap-institute.com/, 18.09.2001

[2] Capability Maturity Model for Software. http://www.sei.cmu.edu/cmm/, 20.09.2001

[3] Capability Maturity Model Integration. http://www.sei.cmu.edu/cmmi/general/genl.html, 16.09.2001.

[4] FCD 15504-2 – Software Engineering – Process Assessment – Part 2: Performing An Assessment, ISO/IEC JTC1/SC7, 2001

[5] Humphrey, W. S., Managing the Software Process, Addison-Wesley, Reading, MA, 1989

[6] ISO 9000. http://www.iso.ch/iso/en/iso9000-14000/iso9000/iso9000index.html, 10.10.2001

[7] ISO/IEC 12207:1995/FDAM Information Technology – Software life cycle processes, International Organisation for Standardisation, 2001

[8] ISO/IEC 15504-1:1998 Information Technology – Software Process Assessment – Part 1: Concepts and introductory guide

[9] ISO/IEC CD 15288 FCD – Systems Engineering – System Life Cycle Processes, ISO/IEC JTC 1/SC, 2001

[10] Rout, T., and Tuffley, A. SPICE and CMMI: Conformance of the CMMI Models to ISO/IEC 15504, in Proceedings of the International Conference on Software Process Improvement and Capability Determination, Venice, 2002

[11] Software Quality Institute. http://www.sqi.gu.edu.au/spice/, 10.10.2001

[12] Unified Modelling Language http://www.omg.org/technology/documents/formal/uml.htm, 12.09.2001

7 Author CVs

Marion Lepasaar

M.Sc. Marion Lepasaar - Ms. Lepasaar graduated MSc from Tallinn Technical University in 2001 and is currently working as a researcher for Pori School of Technology and Economics. She has studied SPI of small software organisations and compared SPIN activities in Finland and Estonia. Her research interests focus on Software Process Improvement and Assessment including SPICE, CMMI and technologies like UML.

Timo Mäkinen

Mr. Mäkinen has studied Computer Science at Tampere University of Technology and graduated Master of Science in 1991. He has worked with software development, quality systems and project management in software industry. He has been working as a consultant and training program manager in SataSPIN project to improve software processes with small software companies. He is a trained SPICE assessor and a reviewer of international software engineering standards. His current research interests include software process modelling; process oriented software development. His teaching responsibilities at PoSTE include data management and software engineering. He is a PhD student at Pori School of Technology and Economics.

Using a Knowledge Survey to plan Software Process Improvement Activities – a Case Study

Espen Frimann Koren and Siw Elisabeth Hove
espen.koren@simula.no, siweh@ifi.uio.no
Simula Research Laboratory, Martin Lingesvei 17, 1325 Snarøya, Norway

Abstract

An important success factor for software process improvement is to identify the areas of the software process with the largest improvement potentials, and focus the improvement work on these areas. It is important to involve the employees in the improvement activities. Furthermore, recent research suggests that the exploitation of existing knowledge in an organisation is another important success factor for software process improvement. Consequently, it might be effective to focus improvement efforts on areas that exploit existing knowledge. This paper reports from a case study where we conducted a knowledge survey to identify the process improvement areas with high improvement potentials. A questionnaire was distributed to the developers of the organisation. The developers described their level of knowledge on general software engineering areas such as methods and programming languages. They also described how important they felt those areas would be in the future. Then, the resulting data were analyzed using simple statistical techniques. The results of the case study suggest that, when combined with other qualitative information, the knowledge survey was useful for identifying improvement areas that exploit existing knowledge. Furthermore, conducting a knowledge survey may have a positive impact on the improvement effort since the survey involved the employees.

Keywords

Software Process Improvement, Planning, Knowledge Survey, Case Study

1 Introduction

There are two schools of software process improvement. The model-based school stems from the work at Carnegie-Mellon Software Engineering Institute which developed the Capability Maturity Model (Paulk, Curtis et al. 1991) – CMM – in the late 1980s. The model-based approaches are focused on certification and comparisons with a model of process maturity. The pragmatic approaches have the philosophy that in the total process there are activities that function very well, activities that work but have some improvement potential and activities that do not work and need to be either replaced or improved considerably. In both schools however we can find almost the same methodology for conducting improvement efforts. Zahran (Zahran 1998) talks about "stages for making software process improvement happen":

- Launching software process improvement. During this stage the concept of software process improvement is introduced to management and technical staff convincing them of the benefits of improving the software process.
- Implementing software process improvement. During this stage an approach to software process improvement is selected and the initial assessment and planning are conducted.
- Institutionalising software process improvement. Establishing a culture of continuous improvement in the organisation.
- Measuring the benefits and ensuring alignment with business goals.
- Continuously improving the process.

These stages are independent of whether you belong to the model-based school or not. The differences come to play in the implementation stage. In Norway there has been a tradition for the pragmatic approach through two research projects – SPIQ (Software Process Improvement and Quality) and PROFIT (PROcess improvement For the IT industry). The project SPIQ resulted in a handbook (Dybå, Wedde et al. 2000) for software process improvement. The method described in the handbook is pictured in figure 1.

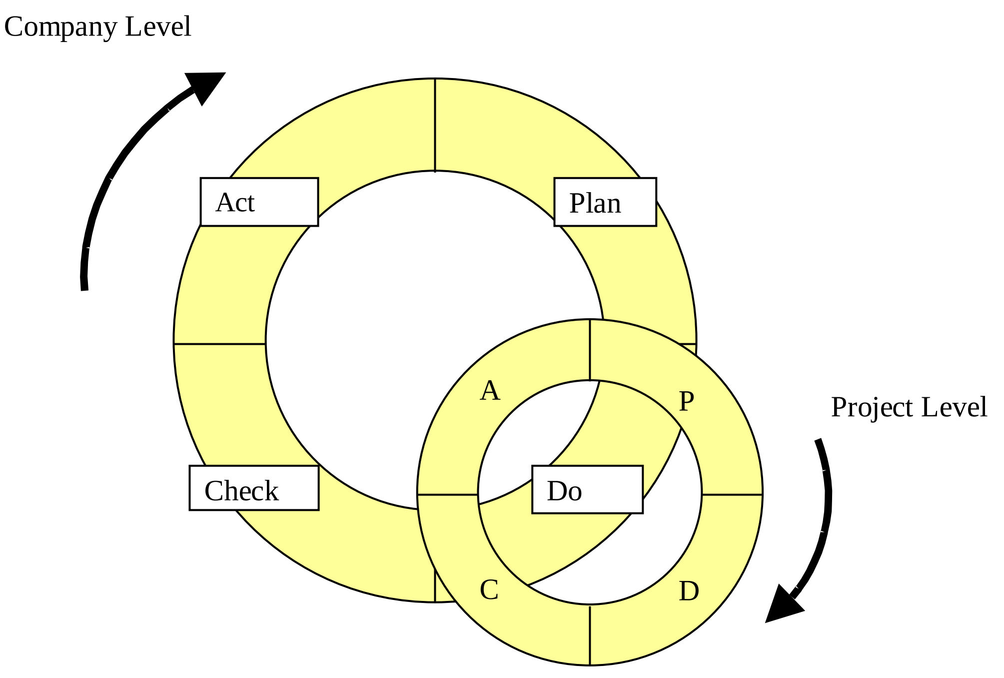


Figure 1. A software process improvement method

The approach is to do various assessments on the organization level during the "Plan"-phase. Then in the “Do”-phase we effectuate one or a few improvement tasks in one or a few projects. The “Check”- and “Act”-phases are institutionalisation of the improved activities.

This paper reports on various tasks conducted in the “Plan”-phase in a Norwegian company. Some of the work is reported in (Koren 2001). The motivation behind the survey reported here is mainly the results found in the doctoral dissertation of Tore Dybå (Dybå 2001). He found statistically significant positive effects on the success of software process improvement from four of the following six factors, and partly support from two of the factors:

- Business alignment
- Involved leadership (partly)
- Employee participation
- Concern for measurement
- Exploitation of existing knowledge
- Exploration of new knowledge (partly)

Especially the exploitation of existing knowledge matches well with the pragmatic approach to software process improvement in that knowledge of the current process and situation is used to identify the process areas with largest improvement potential.

The study reported in this paper had its origin in a wish from the quality manager in the company. He had an idea illustrated in Figure 2.

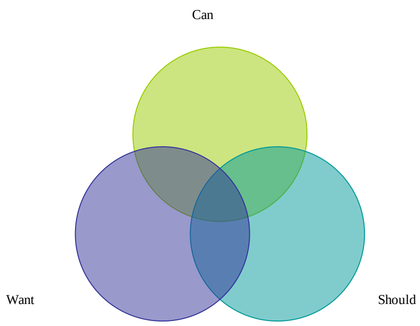


Figure 2. The differences in knowledge between what the employees can, what the employees want and what knowledge the company would benefit most from

The rationale behind Figure 2 in this context is that by finding the knowledge of the staff and also the staff’s predictions for the future need of knowledge one can deduce a direction of improvement by which the staff ideally would like to follow. If one at the same time could collect predictions of the most beneficial knowledge for the company, it is possible to plan for instance education of the staff, what profiles to look for when hiring new people, etc.

We claim that this kind of information is useful for planning improvement actions. Suppose a set of potential improvement areas and compare it with the staff’s predictions for the future. The areas that overlap with knowledge that the staff feels is important, are potentially easier to improve, because we could anticipate better involvement and by the results of Dybå have a better chance of success.

The last influence on the work is the notion of incremental improvement. Zahran (Zahran 1998) mentions improvement strategies where only a part of an organisation is subject to changes or where only a subset of potential improvement areas are considered in a stepwise model of improvement. We have summarised this in Figure 3.

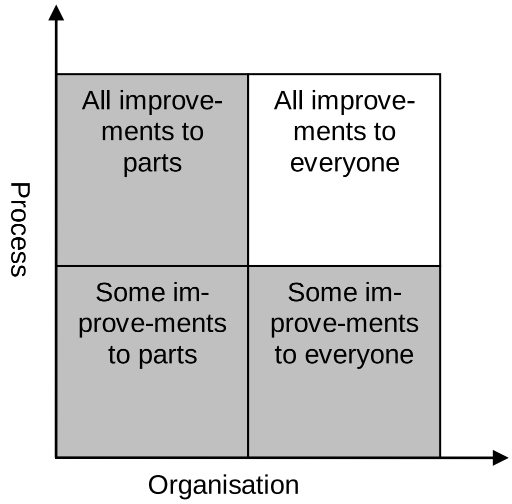


Figure 3. Various degrees of incremental process improvement.

Based on this we developed a questionnaire that was sent to the developers in the organisation. The collected data were put into Minitab and we used statistical tests and functions on the data to look for trends and issues. We were not in a position to use the data to test hypothesis, but using statistical functions we could find whether there were any patterns in the knowledge of the developers. We could see if there were any correlation between the level of knowledge of an area and the perceived notion of future importance. We could find in which areas the developers were most knowledgeable and which areas they regarded as most important for the future. In the context of planning SPI activities we regard this kind of information as important. It may be more beneficial to start with activities that are related to areas where the company has much knowledge since the education of the staff may be minimised. It may also be more beneficial to start with activities that are related to areas the staff think are important in the future, since the developers probably are more motivated. This survey is also a mean to let people express their meaning and contribute to the improvement of their organization.

The remainder of the paper is organised as follows. Section 2 describes the questionnaire and the data collection process. Section 3 describes the explorative data analysis. Section 4 concludes and discusses further work.

2 Method

The work started with a discussion around Figure 2. The purpose of the survey was to outline a picture of the current knowledge in the organization and to get an overview of what the employees regard as important in the future. We decided that a questionnaire would be suitable for this purpose. We limited the inquiry to software engineering. It means that we did not ask about specific techniques and technologies tied to telecom or hardware design and development. The questionnaire had six main areas of interest:

- Methods
- Formal languages, like UML and SDL. We used the term “formal” to distinguish these from regular programming languages, not to claim that UML is a formal language.
- Databases, which cover data modelling techniques.
- Programming languages
- Component technologies
- Platforms

In order to concretise these areas further, we specified technologies belonging to the main areas. For methods, for example, we specified the following technologies:

- Object-oriented analysis
- Object-oriented design
- Testing
- Requirement management
- Use cases
- User interface design
- Planning
- Estimating
- Project management

We also had a field in the questionnaire named “tools”. Here we wanted the respondents to write the names of the tools they had experience with in the areas of interests. The respondents should quantify their knowledge about the various technologies from 1 (no or very little knowledge) to 5 (very good knowledge). Then they should express how important they think each technology would be in the future on a scale from 1 (not important) to 5 (very important). We also wanted to know where they got their knowledge from, in the company, outside the company or at seminars or education.

The questionnaire was first sent to the six people that participated in the interviews reported in (Koren, 2001) for comments. Some of them responded and we considered their comments when we made the final draft. Then it was emailed to 58 developers. In the email we explained the purpose of the questionnaire, and offered a price to one drawn from the ones that responded. We hoped that this would increase the response rate. It was also stated that the information would be treated confidential. They got five working days to answer.

The first day we got eight answers. Eight out of 58 is a little bit under 14%, which we did not consider to be very good. After some days the person responsible for the process improvement in the organization had received some rumours that the purpose of the questionnaire was not well understood. In light of the internal turbulence that the organization was facing, it was not very strange that the developers were sceptic. We decided to post a new email where the purpose of the survey was explained further. This time we emphasized the research aspect rather than the improvement aspect. After this email we got 14 new answers giving a total of 22 answers, which is almost 38%.

It seems that the questionnaire was rather well understood. We did only receive two comments. The first comment was that the scale for future importance lacked the option “Don’t know”. The second comment was that technologies specific to telecom were missing. We got the same comment before it was sent out, but we decided to ask only for general software engineering themes.

We examined all the answers and structured it in Microsoft Excel. We also made an overview document. That document and the individual questionnaires form the basis for the statistics further in this paper.

A brief overview of the respondents’ education level follows:

- Corresponding to master degree: 14
- Corresponding to bachelor degree: 8

3 Explorative data analysis

We got 22 answers out of 58 developers (38%). The figures 4 and 5 show the distribution of the answers.

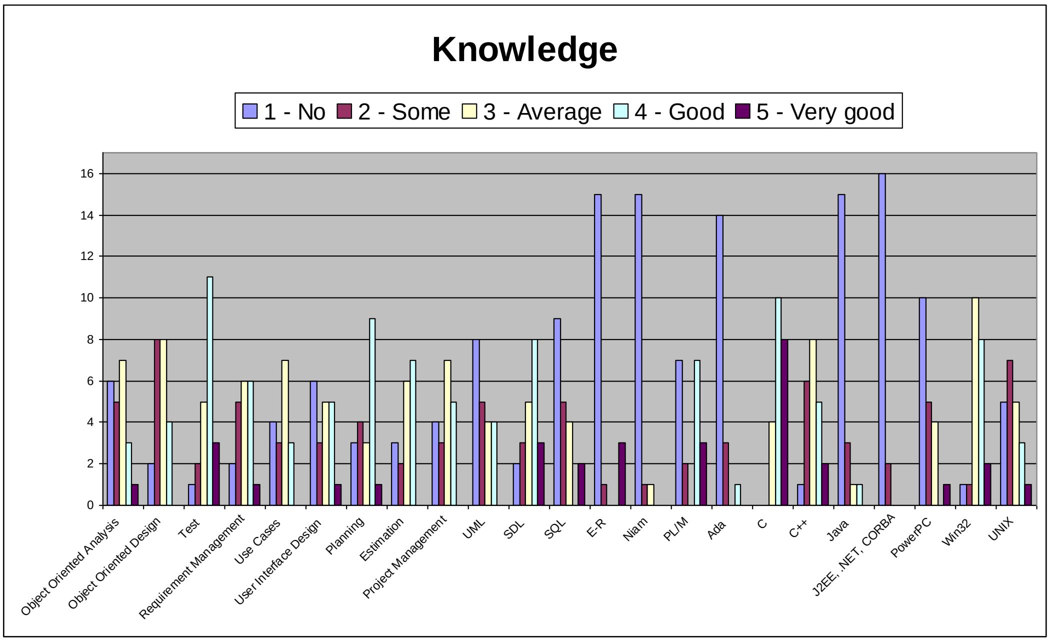


Figure 4. The distribution of the knowledge of the areas

Figure 5 shows the distribution of the future importance of the same areas.

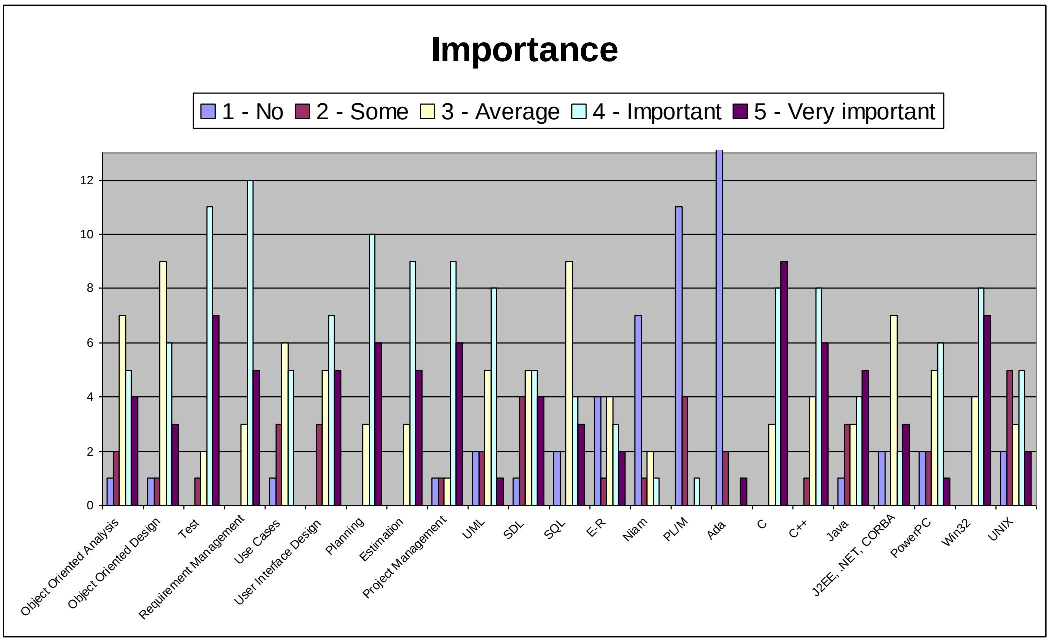


Figure 5. The distribution of the future importance of the areas

It is difficult to compare these two diagrams. You can get an understanding of how the knowledge is distributed and to a certain degree what is regarded as important and what is not. To get a better picture, we compare the medians of the various areas both the knowledge medians, the importance medians and the change in the medians. Figures 6 and 7 show the median of the answers plotted with 90% confidence intervals:

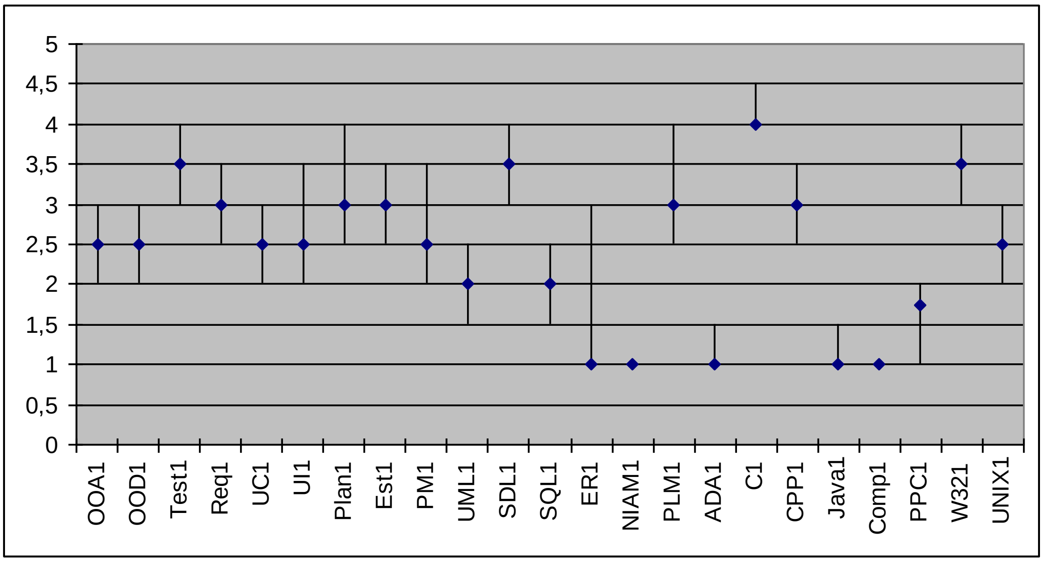


Figure 6. Knowledge - Medians with confidence intervals

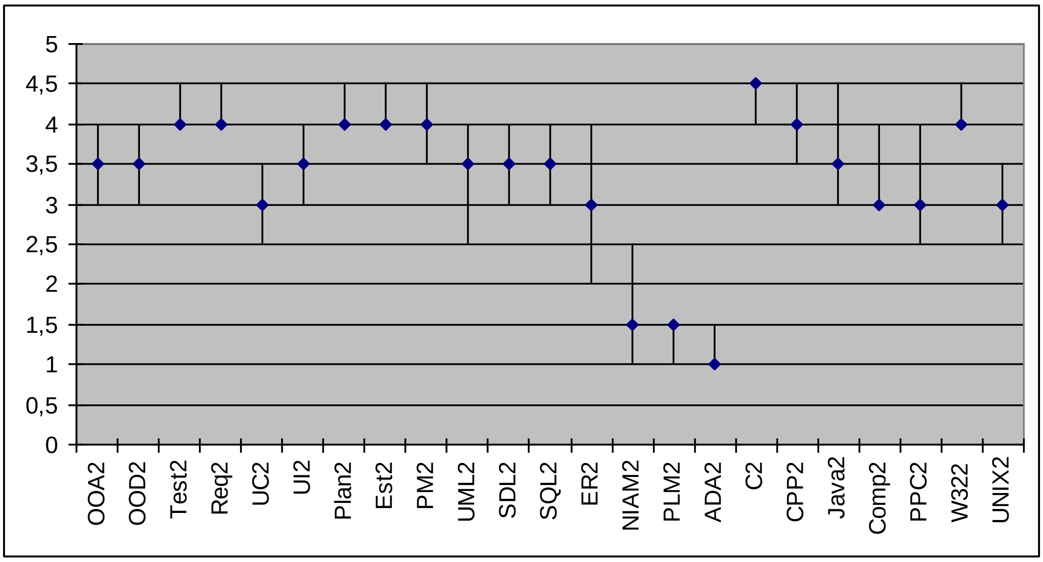


Figure 7. Importance – Medians with confidence intervals

We see that only in four areas they feel they have more than average knowledge (Test, SDL, C and Win32), while on twelve areas they have less than average knowledge. There are fifteen areas which they regard as more than average important in the future, and only three areas are regarded as less than average important.

From this we can propose some areas that it might be successful to start improving. The areas with high expected future importance might be easier to improve since the staff will have personal interest in the improvement.

Another interesting analysis is to find any correlation between the level of knowledge and the level of future importance. The most interesting correlation would be that the higher the knowledge of an area the higher the perceived importance for the future will be. We could then expect that an educational effort in an area before the improvement would increase the probability of success. Using Minitab we compute the linear correlations between the knowledge and the importance in all areas. Table 1 below summarises the correlations:

Area	Direction	P
Object Oriented Analysis	**+**	**0.005**

Object Oriented Design	**+**	**0.041**
Test	-	0.779
Requirement Management	+	0.509
Use Cases	**+**	**0.095**
User Interface Design	+	0.119
Planning	+	0.314
Estimation	+	0.769
Project Management	+	0.415
UML	+	0.290
SDL	+	0.633
SQL	**+**	**0.068**
E-R	**+**	**0.012**
Niam	-	0.519
PL/M	**+**	**0.079**
Ada	+	0.167
C	**+**	**0.012**
C++	**+**	**0.052**
Java	**+**	**0.074**
J2EE, .NET, CORBA	+	0.411
PowerPC	+	0.188
Win32	**+**	**0.077**
UNIX	+	0.302

Table 1. Correlations between the knowledge and the importance for all areas

In the table we have left out most of the statistical details and only shown the direction of the correlation. A "+" means a positive correlation, that is the level of importance grows with higher knowledge. The value in the "P"-column is the P-value computed by the tool that is used to determine how confident one can be on the estimated parameters.

A third interesting analysis we show is the factor analysis in Minitab. The factor analysis looks for degrees of covariance between a set of variables. Table 2 show factor analysis results for the nine method areas in the questionnaire:

Area	Project activities	OOAD	Requirement Management	User Interface Design
Object Oriented Analysis	0.162	**0.918**	-0.232	-0.121
Object Oriented Design	0.04	**0.981**	-0.108	-0.02
Test	**0.629**	0.436	-0.407	-0.126
Requirement Management	0.425	0.077	**-0.664**	-0.439
Use Cases	0.264	0.402	**-0.768**	-0.166
User Interface Design	0.362	0.116	-0.196	**-0.895**
Planning	**0.837**	0.055	-0.265	-0.418
Estimation	**0.884**	0.065	-0.183	-0.389
Project Management	**0.71**	0.181	**-0.585**	-0.014
Variance	2.7896	2.2167	1.7462	1.3787
%	Var	0.31	0.246	0.194

Table 2. Factor analysis

The interesting point of this analysis is that it seems to be covariance between testing, planning, estimation and project management. All these areas are technology independent and traditional project activities. In figure 7 we see that all four areas are regarded as important (the confidence intervals vary from 3.5 to 4.5). The knowledge in these areas (see figure 6) vary around average (confidence intervals from 2 to 4), but with testing as the area they know most about and project management as the area they know the least about. This is not very strange since we mainly asked developers, which may not have much project management experience. In the case of process improvement we may regard these four areas the same. They are regarded as important so improvement efforts related to these areas should be purposeful. There is no evidence in the analysis that education would increase the perceived importance.

With OOAD we can see strong covariance and also in table 1 we see that there are significant correlation between the knowledge and the importance. From figures 6 and 7 we see that the knowledge is a little bit under average and the perceived importance is a little bit above average. Compared to the project activities only project management overlaps a bit on importance. The other three areas are more important (no overlap of the confidence intervals). In the case of process improvement we may regard these two areas as the same. There is a possibility that going through with education before starting any introduction of OOAD may be purposeful because we see a correlation to the perceived importance. The areas are not regarded as important as the project activities, so they should be introduced later than the project activities.

We can also find a covariance between requirement management and use cases. From figures 6 and 7 we see that the knowledge of requirement management is round average, but a bit below average for use cases. For importance there is a significant difference in that requirement management is regarded as more than important (confidence interval is 4 to 4.5), but use cases is regarded as average important. Table 1 show that by educating people in use cases will increase the perceived importance. With respect to process improvement, we see that introduction of use cases as a way to improve the requirement management could be a purposeful improvement effort. Whether to do it together with the project activities or not is difficult to decide out of these analyses.

The last factor is the user interface design. Traditionally the company has not been very concerned with issues regarding user interfaces since their products delivers telecommunication infrastructure. The administration systems are developed using Oracle Developer with primarily automatically generated user interfaces. In the new organisation they also develop terminals to be used on the infrastructure. The terminals could perhaps benefit from focusing on user interface design. However, the survey suggests that user interface design is regarded as a little bit above average important, so it would not be the first area to improve.

3.1 Threats to Validity

There are weaknesses with this kind of analysis. It has a very theoretical approach. Even though the statistical result shows that some areas might be better to start with, one can not say for sure that there are the best areas to focus on at current time. We have based our survey on knowledge and on perceived future importance, maybe a focus on marked trends or technologies used by competing companies would have been more fruitful. The fact that the response rate was 38 % might reduce our results' validity. Despite this, we claim that our results, build upon the developers knowledge and assumption of the future, gives a sufficient basis for doing statistical analysis in order to point out actual areas to focus process improvement on.

Because we did not ask the respondents which department they work in, we were not able to make any analysis of the competency profiles of the various departments. The competency profiles presented here may therefore not be representative for all departments. Some of the competencies asked for may not be relevant in one or more of the departments. Also we cannot be sure whether a majority of the respondents are from the same department. The competency profiles presented may therefore belong to one, or a few, departments and will not reflect the competencies in the organisation as a whole. These are issues that will be relevant to address in future cases.

The things we asked for in our questionnaire is mostly relevant to this case only, but for instance CMM is heavily based on questionnaires and using similar statistical analyses may give valuable input to the planning phase.

4 Conclusions and further work

The survey was motivated by the following:

- A belief that incremental improvement is easier to conduct and easier to achieve success with than model-based improvement
- That employee participation in the improvement process has a positive effect on the success.
- That exploitation of existing knowledge has a positive effect on the success of the improvement efforts.

Using a survey to get an overview of the software engineering knowledge in the company will both facilitate employee participation and exploitation of existing knowledge. This should have a positive effect on the success of the improvement initiative. The relation to incremental improvement is that the results of the survey may pinpoint areas where improvements should be conducted. The areas that are found using this survey is based on what the employees believe are important as opposed to using a predefined framework like CMM, where the identified areas of improvement are based on some standard.

The explorative method using statistical analyses seems to point out areas that may potentially be more successful to start improving than others. At least we can use it as a way to prioritise the activities to be able to plan incremental process improvement. In this study we started out by finding the software engineering knowledge among the developers and how important they felt that different areas would be in the future. Since the data also is a picture of the experience and knowledge in the organisation, we think the exploitation of the information is purposeful. Furthermore the survey is a way to involve the developers in the improvement process. As recent research shows, these two factors have a positive effect on software process improvement.

We have used statistical analysis in a very explorative way. We have been using correlations between knowledge and importance to show how high knowledge of an area might give a higher perceived importance for the future. Factor analysis has been used to show the covariance between different areas of software development. This can be used in the planning of improvement efforts. For instance improvement of the estimation methods could perhaps benefit from improving the planning methods at the same time, since we found evidence that these two areas are perceived as similar.

There are several things to do following this study. The most interesting would be to actually use the study as a guide for implementing improvement tasks. We could for instance define it as an experiment conducting improvement activities both based on the potential successful areas from this study and from another area simultaneously. If we find that such studies are relevant, the method could be elaborated.

5 Acknowledgements

We thank the participants answering the questionnaire. Many thanks also to the quality manager for asking us to go through with the survey and for cooperating while working with the questionnaire and the paper. We also would like to thank Erik Arisholm for valuable input and motivation.

The work is founded by The Research Council of Norway through INCO (Incremental and Component based software development) and PROFIT (Process Improvement in Norwegian IT Industry).

6 References

Dybå, T. (2001). Enabling Software Process Improvement: An Investigation of the Importance of Organizational Issues. Doctoral Dissertation, IDI. Trondheim, NTNU.

Dybå, T. e., K. J. Wedde, et al. (2000). SPIQ Metodehåndbok.

Koren, E. F. (2001). Information collection and investigation for software process improvement - a case study. Generative and Component Based Software Engineering, Erfurt, Germany, Springer.

Paulk, M., B. Curtis, et al. (1991). Capability Maturity Model for Software, SEI.

Zahran, S. (1998). Software Process Improvement. Practical Guidelines for Business Success. Harlow, England, Addison Wesley.

7 Author CVs

Espen Frimann Koren

Espen Koren is a Ph.D. student at Department of Informatics, University of Oslo and at the Software Engineering Group, Simula Research Laboratory in Oslo. Before he started the Ph.D. studies he worked as a software designer and developer for seven years in Norwegian consulting companies. His thesis topic is on the introduction of component-based, incremental development processes in industry.

Siw Elisabeth Hove

Siw Elisabeth Hove is a master student at Department of Informatics, University of Oslo and at the Software Engineering Group, Simula Research Laboratory in Oslo. Her area of interest is incremental and component based software development.

PATTERNS - Patterns to Adopt Knowledge Based Solutions to Software Management Problems

Elixabete Ostolaza (Elixabete.Ostolaza@esi.es)
Nuria Quintano (Nuria.Quintano@esi.es),
Teodora Bozheva (Teodora.Bozheva@esi.es), Giuseppe Satriani (Giuseppe.Satriani@esi.es)

Organisations, both profit and non-profit, tend to "re-invent the wheel" when they come to management issues. In fact, previous experiences (even their own) seem very different (in form and context) from the current problems, so much so that the previous experiences do not appear to be valid in the problems solving process. In addition to the problem of validity of existing knowledge in new business situations, information on previous experiences is often available in different places, which increases the difficulties in sharing within the organisation or among different ones.

The solution provided by the PATTERNS project develops facilities for the introduction of the embedded organisational knowledge into business process. Specifically, PATTERNS allows the introduction of the "learning from past experience" practice into the problems solving process.

Keywords

Knowledge Management, Knowledge Representation, Software Process Improvement, Software Project Management, Software Engineering.

1 Introduction

According to Davenport and Prusak [1], **knowledge** is "a fluid mix of framed experience, values, contextual information, and expert knowledge providing a framework for evaluating and incorporating new experiences and information. In organisations, it is often embedded not only in documents or repositories but also in organisation routines, processes and norms". Moreover, knowledge specifically related to Information Technology (IT) can be found in many different past experiences and is also highly dependent on the context of that particular experience, so much so that the previous experience does not appear to be valid in the problem solving process. Therefore the knowledge is very difficult firstly to recognise it, then to extract it and finally to package it in usable forms to accelerate the "learning from experience" process. This situation is worsened by the general reluctance of the organizations to share the knowledge gained in their daily work. PATTERNS aims to overcome these constraints by modeling that knowledge to determine common characteristics and so derive cases that contain the contextualised knowledge, and the rules to be applied to store and retrieve them for being used in solving new problems.

Recognising knowledge as an organisational asset is the first step to consider its management and its investment so important as for other common used assets. However, knowledge management involves a process of transforming the tacit knowledge, located in the minds of the experts, into explicit knowledge represented in an understandable and useful form and available to be used by a broad set of individuals. The knowledge must be represented in a form that is really interpreted as useful information. As Liebowitz [8] says "Good representation is essential if the knowledge is to be considered as valuable to the knowledge worker". For this reason, it is necessary to code the knowledge and express it in formal language to explicit it, and so it can be shared by many knowledge worker. The PATTERNS project will use the patterns language [7] to packaging and transferring of knowledge useful for recognising a problem.

In the current society, there is a trend to shift from stand-alone systems to networked information and processes that facilitates more and more the knowledge sharing. Moreover, improvements in information technology make it easier to collect, store, and distribute any type of information. However, to be effective, knowledge workers need to be able to understand and act on that information. On the other hand, an huge variety of information is now available to a large number of users with very few mechanisms of filtering. PATTERNS faces to both aspects, to facilitate the knowledge sharing but, at the same time, to guarantee that inexperienced people could increase their ability to learn, by taking advantage from past experiences and by exploit only the new knowledge available that fits to their needs. For achieving these two objectives, PATTERNS will create an architecture based on CBR (Cased Based Reasoning) technology that permits to identify similarities between a current working situation and past experiences in problem solving, and so can retrieve the most adequate cases according to the users needs. Moreover, this architecture is supported by Intelligent Agents (IA) to spread the learning process across different organisations and to consider end-user behaviours.

The PATTERNS project aims to create an internetworked architecture of Knowledge Centres[1] (KC) and distributed end-users, from which organisations can benefit so as to become knowledge driven and learning organisations. PATTERNS will provide a dynamically adaptive architecture that will allow organisations to increase their knowledge capacity, capitalise on that knowledge by transform it into business processes and manage the distributed knowledge embedded in management practices.

2 How PATTERNS Was Conceived

[1] KC = a Knowledge Centre could be:
- an organisation with available experiences (explicitly collected or implicitly contained in experts' brains)
- an organisation that facilitates the access of end-users to the PATTERNS network

PATTERNS is a project funded by the Information Society Technology (IST) programme of the European Commission (EC). The project consortium included the following European organisations: European Software Institute - ESI (Spain), INDRA Sistemas (Spain), YANA Research (Italy), TEKKVA Consult (Denmark) and Software Technology Transfer Finland - STTF (Finland).

The PATTERNS idea has gone evolving from 1996 together with the ESI expertise in relation to corporate repositories. The higher expertise ESI reached creating and managing these repositories the higher value ESI obtained analysing the information contained in these repositories. The main cornerstones of the ESI expertise in corporate repositories are the following:

- VASIE

When ESI acquired the leadership of the VASIE-2 project (ESPRIT/ESSI N. 24199) it was not totally aware of the value that could be obtained from it. This project is charge of disseminating, through a public database based on the WWW, the final reports of the process improvement experiments (PIEs) funded by EC in ESSI programme (IVth framework programme). A summarised explanation of the VASIE structure is described in *Figure 1*.

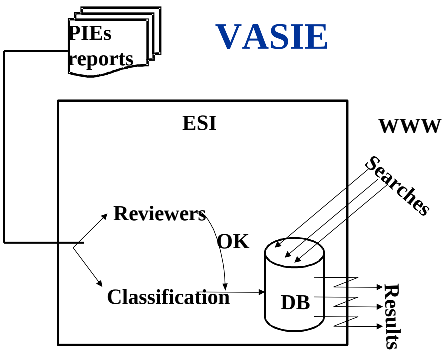


Figure 1 - VASIE structure

As long as the project was progressing and taking into account the type of VASIE users, the type of searches performed and the results obtained by another ESPRIT project (PERFECT[2] – Project Number 9090), ESI perceived the big potentiality of the value contained in past experiences to solve current Software Process Improvement (SPI) problems. Under this project, ESI has been the catalyst of the European experiences of SPI, which has provided it the opportunity of participating in the ESPINODE[3] initiative, and so reach even more expertise in this subject.

As consequence, ESI defined in 1998 the following long-term objective: "**To be able to provide effective and accurate solutions to software engineering problems, based on previous and available experiences"**.

- RoE - The ESI knowledge Management infrastructure

The possibility to provide an effective and accurate solution based on previous experiences depends on the quality and quantity of the available experiences: the larger is the experience base, the closer will be the solution for the user. To enlarge its knowledge base, ESI decided to define and create a Knowledge Management infrastructure [2], [3] to: identify the sources of knowledge, collect and describe the identified knowledge, validate it, insert it in the knowledge base, transfer it and maintain its value. During this process ESI had to face up to different problems that are also the most common problems currently existing in the Knowledge Management field:

- **PROBLEM 1**: An experience just as it is, is not directly useful for solving problem.
- **PROBLEM 2**: Difficulties in experience sharing due to confidentiality problems or due to lack of available time.
- **PROBLEM 3**: To successfully transfer the knowledge, it should be felt as useful to solve a problem. The adoption of a solution depends on the possibility to adapt it for a specific context. As consequence, to reach the ESI long-term objective is mandatory to solve two new problems: ***experience adaptation and experience contextualisation***.
- **PROBLEM 4**: Experience base dimension (and quality) remains always a critical factor.

Each problem listed above can be handled by PATTERNS through the consortium expertise described

[2] PIA Experience Factory – The PERFECT Handbook

[3] ESPINODE - ESSI PIE nodes has been an European network of competent partners in order to stimulate, promote, and foster the use of Software Best Practices all over Europe

the following.

1 - Roadmaps for software engineering practices (SEP) and Patterns of Solutions

The analysis of the experiences available in VASIE made by ESI as contribution to the ESPINODE initiative has allowed to show commonality and variability among the actions performed in different experiences. Consequently, a certain number of patterns of solution for a specific Software-Engineering Problem (SEP) or SEP Roadmaps [4] have been identified. This achievement, together with the ESI expertise in domain modelling (usually applied to software reuse), has allowed to approach the knowledge management according to an innovative perspective: **"the abstraction of patterns of solutions from available experiences"**. This new focus contributes to handle with **PROBLEM 1** and **PROBLEM 2** mentioned above, because the abstraction process distils knowledge from different experiences; moreover, in this way an higher level of anonymity is guaranteed.

2 - CASE vs. Experience

ESI considers that the use of cases instead of single experiences, would have provided a valuable solution to **PROBLEM 1**, in fact:

- From each experience or from a sub-set of similar experiences could be derived a "case".
- As Watson [5] has defined, a "case" is a contextualised piece of knowledge representing an experience.
- Each SEP Roadmap can be then seen as "**a solution to a problem in a context**".

This innovative approach handles also with **PROBLEM 2** because the "case" derivation process is an abstraction process. Further, according to the Watson's definition, a "case" contains (see ***Figure 2***) a solution for a problem in a context; it then contains all the elements for a successful transfer (**PROBLEM 3**).

CASE =	A SOLUTION	TO A PROBLEM	IN A CONTEXT
	Piece of Knowledge	User Need	Filter

Figure 2 - Content of a CASE

3- Experience sharing with members and with other European companies

An institutionalised process at ESI is the experience sharing with its members. During an experience sharing session performed in 1999, it realised that INDRA Sistemas, S.A. was facing to similar problems in its intent to transfer the software engineering knowledge available at the end of each internal project, among different departments. The need to overcome common problems, together with the consciousness to potentially shares and then adds value to its own knowledge base (another contribution to solve **PROBLEM 2** and **PROBLEM 4**), leaded both organisations to join the efforts and the expertise to define a common and marketable solution. For this reason, INDRA and ESI became the Knowledge Centres (KC) of PATTERNS.

4 - Interface between human and artificial resources

To foster knowledge sharing (**PROBLEM 2**) the communication channel (WWW and Internet) is the critical point but it can't be the only solution: a common interface and a compatible knowledge bases (domain modelling for obtaining cases) should support it. The management of informal communication between man and machine is a important element of the architecture, which will be solved by PATTERNS through an innovative idea, this is to integrate a Natural Language Process (NLP) interface to the mentioned architecture. The option of describing a problem by using natural language has many advantages among them;

- users will be able to interact with the system in their national language, if they want to; this will avoid misunderstandings and ambiguity
- the system and the user will able to "negotiate" exchanging information in order to reach an agreement about the topic of discussion
- users are not expected to know anything about the internal organisation of the system.

The expertise of Yana Research about the application the language-engineering techniques to industrial systems is a guarantee for obtaining the interface required by the system. Yana has already developed tools and skills implementing software systems for the translation of natural language expressions in temporal logic formulae and for linguistic quality control of software requirements document. The logic behind NLP is very close to CBR one: this allows Yana to provide its expertise also as developer of the CBR tool. Yana skills are completed by the expertise in Internet software development environments.

3 PATTERNS Architecture

The solution proposed by PATTERNS allows users to obtain a rapid solution to a context-based problem. This solution will be provided according to the knowledge available locally at each Knowledge Management (KC) and to the knowledge available in the network. PATTERNS will apply the general architectural solution (see ***Figure 3***) to the specific context of organisations and end-users dealing with Software Management practices. The choice of this domain has been motivated by the knowledge available in the project partners. Moreover, PATTERNS will explore the application of novel and proven technologies to define a solution that could support the transformation of entities into “smart” organisations by:

1. Modelling the KC domains of management practices, to determine common characteristics and the rules to be applied, according to the value assumed by these characteristics. This aspect will lead entities to foster the change into knowledge driven organisations.
2. Defining a learning architecture at each KC level that allows local users to explore analogies between a current working situation and past experiences, and to use them in solving a new problem by recognising its similarity with a specific known problem.
3. Creating an internetworked structure supported by intelligent agents, to spread the learning process across different organisations and by considering end-user behaviours, to overcome the local organisation’s boundaries, the resistance to experience sharing and the difficulty of taking advantage of them, due to a variety of contexts and different learning cultures.
4. Providing a dynamically adaptive architecture based on Case Based Reasoning (CBR) technology that permits similarities to be identified between a current working situation and past experiences in problem solving.

The principal elements of the global architecture shown in the ***Figure 3*** are:

- the **Knowledge Centres (KC)**, where local knowledge repositories are active;
- **Internet**, that is the communication mean for remote search and connection;
- the **Other Sources** of information/knowledge that could be useful for integrated extra elements for obtaining a more complete solution.

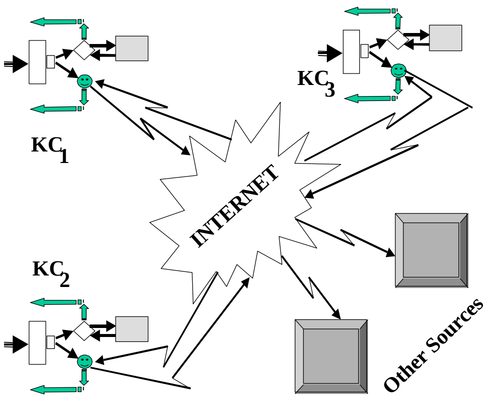


Figure 3 - Global Architecture

The solution proposed by PATTERNS allows a high level of portability. The local architecture is based on Intranet technology; the remote connection is done through Internet and specific intelligent agents; the structure for domain modelling is independent from the domain to be modelled.

At the local level, that is, in each KC, PATTERNS will implement the architecture shown in the ***Figure 4***. The principal elements of the local architecture are:

- the **Front-End (FE)** that will accept **Problem** description in **Natural Language,**
- the **CBR-Tool (CBR)** that will search in the Modelled Domain a similar solution to a problem,
- the **Intelligent Agent (IA)** that in parallel of the CBR-Tool will start a remote search in other KCs belonging to PATTERNS and in Internet, based on similar "behaviours",
- **Database of cases (DBC)** or knowledge base: each KC should identify the explicit knowledge available and the related knowledge sources. Essentially, three different knowledge source types could be identified: tacit or implicit knowledge available in experts, explicit knowledge embedded in documentation, explicit knowledge expressed in terms of best practices and lessons learned. Each KC will then formalise the available knowledge and/or will implement the necessary actions to transform the implicit knowledge into explicit. The formalised knowledge will be represented in a set of different "Cases" and it will become the corporate knowledge base of each KM.

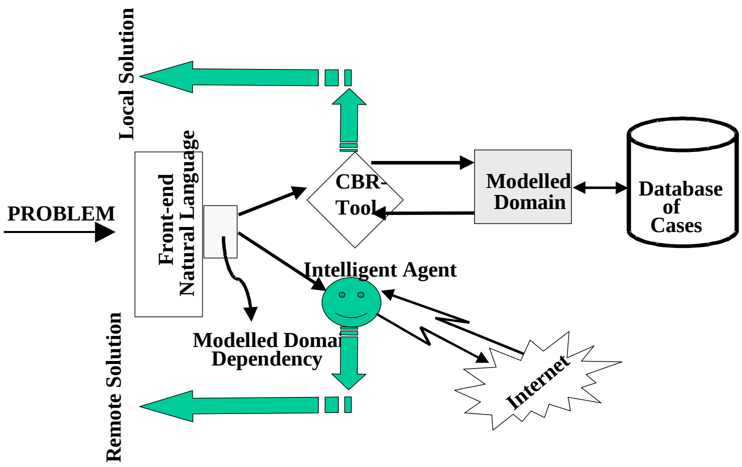


Figure 4 - Local architecture

- the **Modelled Domain (MD)** that contains the formalisation of the Software Management domain of experiences available in the KCs. A graphic representation of a Modelled Domain is shown in ***Figure 5***, which represents how the modelled domain operates; based on a set of domain characteristics different rules are activated that obtain as result actions to be taken.

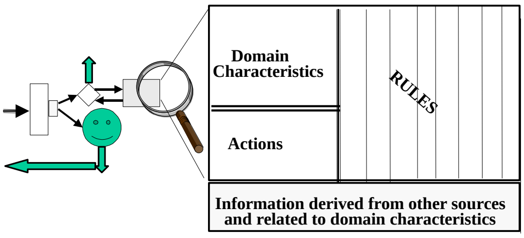


Figure 5 - Modelled Domain

4 Knowledge Centres - Domain Modelling and Patterns Derivation

The knowledge Centres (KC) involved in PATTERNS guarantee the provision of knowledge about Software Management practices. But the knowledge existence is not sufficient to obtain knowledge useful, it is necessary to extract it and package it in usable forms. One of the first things that any science or engineering discipline must have is a vocabulary for expressing its concepts and a language for relating them together. A pre-requisite for defining a language is to identify the domains the language will model. It has been demonstrated in other application domains, that patterns can be a mechanism for domain modelling, for packaging and transferring of knowledge and for recognising a problem, based on the available solutions/experiences. As Richard Gabriel states in *Patterns of Software: Tales From the Software Community* [6]: "Each pattern is a three-part rule, which expresses a relation between a certain context, a certain system of forces which occurs repeatedly in that context, and a certain software configuration which allows these forces to resolve themselves." Moreover, Christopher Alexander in *A Patterns Language* [7] provides this definition: "Each pattern describes a problem that occurs over and over again in our environment and then describes the core of the solutions to that problem in such a way that you can use this solution a million times over without ever doing it the same way twice." Each pattern is, then, a rule that expresses a relationship among a certain context, problem and solution. Using the pattern form, the description of each case tries to capture the essential insight that it embodies, so that others may learn from it, and make use of it in similar situations.

Each case derived will be stored in the Database of Cases, which is the base where the CBR tool will search the most similar cases to each problem purposed. The Database of the Cases should be organized into a manageable structure that support efficient search and retrieval methods. A balance has to be found between storing methods that preserve the richness of cases and their methods for accessing and retraining those cases that match against the problem purposed.

5 Case-Based Reasoning (CBR)

From a technological perspective, patterns are strongly related to CBR technology that helps to solve problems by storing, retrieving and adapting past situations or cases. In case-based systems a "case" is usually a contextualised piece of knowledge representing an experience [5]. It contains the past lesson that is the content of the case and the context in which the lesson can be used.

The mental process to describe CBR is typically represented by a cyclical process (see ***Figure 6***) that receives as input the current problem description and:

- RETRIVE the most similar case(s).
- REUSE the case(s) to attempt to solve the problem.
- REVISE the proposed solution if necessary.
- RETAIN the new solution as a part of a new case.

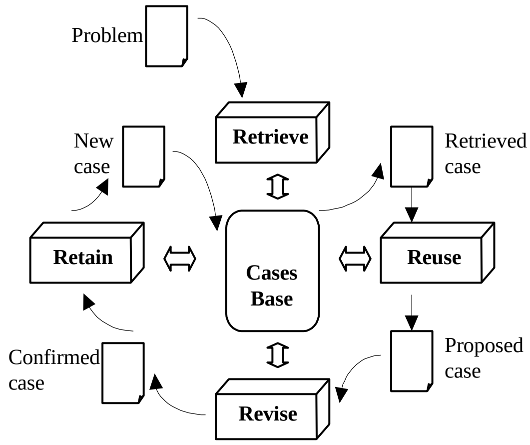


Figure 6 - CBR cycle

CBR makes direct use of past experiences in solving a new problem by recognising its similarity with a specific known problem and by applying its solution to find a new solution for the current situation. An important characteristic of this technology is its capability for learning from the available knowledge, generating new cases. The procedure followed is complex but efficient; A new problem is matched against the available cases and one or more similar cases are retrieved. A solution suggested by the matching cases is then reused and tested for success. Unless the retrieved case is a close match, the solution will have to be revised, producing a new case that can be retained [5]. The capability of learning of the CBR tool is based on the use of heuristics, that allow modifying previous cases to fit new cases according to the user feedback collected after revising phase.

6 Natural Language Processing (NLP)

Any application dealing with real world has to tackle the problem of handling informal communication. Tackling this problem corresponds, from a different point of view, to the effort of allowing a more natural way of communication between human and artificial resources. Generally, the real problem for Natural Language Processing (NLP) based application is domain modelling and instantiating the link between domain knowledge and linguistic knowledge. PATTERNS offers a really challenging line of research and development since, independently from the application of a NLP engine, it addresses both the problems of domain modelling and reasoning over the domain. In other words, it will try to define how to develop the knowledge base which represents formally a given domain. Our idea is to use this knowledge base as the (formal) representation of the context underlying ongoing (informal) communication.

The knowledge base will contain a set of frames representing different situations that can occur in a given domain. The user will be able to describe to the system the current domain and a situation to be managed. The system will retrieve a matching or a similar scenario in its database and return it to the user as a possible solution strategy. NLP provides to users the advantage of interacting with the system in the national language; it doesn't force users to use pre-defined list of choices for describing a problem; and it and the user will able to "negotiate" exchanging information in order to reach an agreement about the topic of discussion.

7 Intelligent Agents (IA)

According to Liebowitz [8] an Intelligent Agent (IA) should be capable of adapting to user habits and preferences using learning techniques, also they should decide when to help to user, what to help the user with, and how to help the user through its automated reasoning. And finally, an IA should can collaborate with other agents, exchanging information and service, and thereby solve problems that cannot be solved alone. The main characteristics of an IA are: Autonomous, because perform its tasks without the direct human intervention; Goal-oriented/Reasoning, due to that the agents do not simply act in response to their environment, they exhibit goal-directed behaviour; Self-starting, by the capability of deciding when to act by sensing environment changes; Temporal continuous, by its continuously running to monitor environment changes; Personality, because the agents have well-defined, believable personality that facilitates interaction with the human users.

IA in the PATTERNS context are software units that scout, index and retrieve information to and from knowledge databases, and, at the same time, they are able to "learn" and enhance their search and retrieval capabilities and to provide to new cases creation. Further more, IA can "learn" the users' pattern of operation, this characteristic provide a high added value because the IA can provide additional information obtained from the CBR. The method that IA follows consists of recording the purposed problems and the case(s) chosen by each user, so the IA can offer other users with the similar problems references about solutions (cases) chosen by other user by solving the same problem. At finally, the IA can identify the lack of knowledge and take preliminary steps to supply needed information and knowledge before PATTERNS users ask for it. In fact, IAs also continuously communicate among them to mutually transfer the knowledge acquired. These characteristics guarantee a high level of knowledge sharing. The aim of IA in the PATTERNS context is to provide additional research and support for complementary knowledge not found by the CBR, which analyses the modelled domain at the local level.

8 Conclusions

The solution provided by PATTERNS is based on the integration and specialisation of already existing information society technologies and methodologies for an Intranet/Internet based software solution, written in Java to guarantee fully portability to different technical platforms. This make PATTERNS a leading solution in IT field in which it will be directly used. In fact, it provides a possible way to foster IT companies transformation into "smart" organisations: knowledge based and learning. This trend in the Information Society leads to the introduction of strategies that bring together people, processes and technology, transforming the company culture into one that values learning and sharing. This cultural change is an essential first step for any company aiming to get the best from knowledge management.

Another new user requirement is gaining more and more importance within the Information Society; "How can I take advantage of the vast knowledge available, to find the information I need at the moment that I need it!". PATTERNS proposes to decentralise the knowledge but to centralise the knowledge management. In this way users would have an homogeneous user interface to search across the available knowledge bases without having to know where the content is located.

Summarising, PATTERNS offers a new method of work that enables both individuals, but mainly, organisations to innovate and be more effective and efficient in their work. In this way, organisations will be able to increase their competitiveness by including past experiences in the current problem solving process. So, exploring and validating novel technologies, applications, architectures and practices PATTERNS aims to support the transformation of profit and non-profit entities into "smart" organisations: "knowledge driven, internetworked, dynamically adaptive to new organisational forms and practices, learning as well as agile in their ability to create and exploit the opportunities offered by the digital economy".

9 Literature

[1] Davenport, T and Prusak, L: Working Knowledge; Cambridge, MA: Harvard Business School Press, 1998.

[2] Marina Blanco, Jose Ángel Díez, Pedro Gutiérrez,Leire Markaida, Giuseppe Satriani (1999). Software Process Improvement Driven by Business Goals: The role of experiences. Sixth European Conference On Software Quality: Software Quality-The Way To Excellence (Vienna, Austria, April 12-16, 1999)

[3] Marina Blanco, Pedro Gutiérrez,Leire Markaida, Giuseppe Satriani (1999) Repository

of Expertise Guide. European Software Institute. ESI-1999-TR-017

[4] Marina Blanco, Jose Angel Díez, Pedro Gutiérrez, Leire Marcaida, Giuseppe Satriani: Roadmaps for SPI; Submitted to IEEE Software magazine, (1999).

[5] Ian Watson: Applying Case Base Reasoning: techniques for enterprise systems; Morgan Kaufmann Publishers, Inc., 1997.

[6] Gabriel Richard: Patterns of Software: Tales From the Software Community; Oxford University Press, 1997.

[7] Cristopher Alexander: A Patterns Language: towns, buildings, constructions; Oxford University Press, 1977.

[8] Jay Liebowitz: Knowledge Management Handbook; CRC press LLC, 1999

10 Author CVs

Elixabete Ostolaza (Elixabete.Ostolaza@esi.es)

She is Project Leader at the European Software Institute since March 1998. She is currently PATTERNS project (IST- 2000 – 25120) coordinator and partner manager in ESERNET project (IST-2000-28754). She has been leading several projects related with SofTQM technology (ENABLER, BEST) and Software Process Improvement such as BIG-TTM (guide based on SPICE to reduce Time to Market), SPISO (SPICE compatible model that merge SPICE and ISO 9001 to provide a roadmap based on SPISO to achieve ISO 9001). She is a competent SPICE assessor, EFQM evaluator and ISO 9001 auditor.

Elixabete Ostolaza was previously given a grant by the Basque Government and the European Social Fund to take part in several ESPRIT projects (LOQUI, PRINCE, ARCHON) related with Artificial Intelligence, afterwards she was Project Manager in the Information System for Alliances and Subsidiaries Department of Telefónica, S.A. She was then Manager of Development and Maintenance of Information Systems in Directorate General XVI of the European Commission.

Elixabete Ostolaza received her degree in Computer Science from the University of the Basque Country and her Master degree in AI from the University of Aberdeen.

Nuria Quintano (Nuria.Quintano@esi.es)

She works at the European Software Institute in the *Strategic Software Process Improvement Unit*. Nuria recently started cooperating in the *Good Practice Management* line of the unit she belongs to, where the PATTERNS project fits in. Her work since she is in the Institute is mainly related with the participation in the development of strategic improvement and management programs, as well as consultancy actions primarily built around the *ESI's Balanced IT Scorecard technology (BITS)*. She is currently obtaining her Master in Electronic Business Administration from ESEUNE business school (Getxo, Spain). She received her degree in Computer Science from Deusto University (Bilbao, Spain).

Teodora Bozheva

Educational Degree: M.Sc. in Computer Science from Sofia University, Bulgaria.

Experience:

She started as an assistant professor in Artificial Intelligence at the Sofia University. She also performed research activities in the field of neural networks and their application for natural language comprehension. During the last 6 years she was a project manager of software development projects at the branches of two German companies in Bulgaria. She took part in the Software Process Improvement project (SW-CMM®) at her former company and was in charge for its implementation in the Bulgarian subsidiary. Recently she joined ESI and is currently working on SPI and e-Learning projects.

Giuseppe Satriani (Giuseppe.Satriani@esi.es)

He is presently a Consultant at the European Software Institute. He was involved in the following European initiatives: VASIE (ESSI 24119), AESOPIAN (TE-2001), RISKDRIVER (ESPRIT 27890); he is also involved in a local initiative called Knowledge Management structured network.

He is competent SPICE assessor, EFQM evaluator and SEI Qualified Instructor for the CMM.

Giuseppe Satriani was previously an Engineer at the System-Test department at Olivetti SpA, after which he was Project Leader in the Quality department, with responsibility for quality system definition and application within the development department. Afterwards, he was responsible for marketing activities (product promotion and suppliers' management) at the Add-On Business department.

Giuseppe Satriani received his BSc degree in computer science from Pisa University.

Information Quality and Process Improvement

János BAJZA, DSS Consulting Ltd.
janos.bajza@dss.hu
Miklós BIRÓ, Budapest University of Economic Sciences and Public Administration
miklos.biro@informatika.bke.hu

Abstract

It was first suggested by John von Neumann in his "First Draft" in 1944 that instructions for the computer–previously entered on punched paper tape, or by plug boards–can be stored in the computer's electronic memory as numbers and treated exactly the same as numerical data. Ideas related to both instructions and data have gone a long way since Neumann's seminal paper. Instructions turned into software systems, and software process improvement is recognized as the essential means of reliably achieving software quality expectations. Data on the other hand is more generally called information whose quality is becoming of utmost importance in the Information Society where an unprecedented number of people can ubiquitously access an unprecedented amount of information. It is no surprise by consequence that on the way to overcoming the software crisis, we are about to entering the era of information crisis. This paper summarizes concrete experiences with the devastating effect of poor information quality, and besides highlighting the once more winning process improvement approach, offers a solution to improving the quality of already existing information repositories.

Keywords

Information Quality, Process Improvement, Data Cleansing, Total Quality Management

1 Introduction

The experiences and solutions described in this paper originate from the Hungarian company DSS Consulting Ltd. whose mission is to help its customers achieving economic success through the effective utilization of business data. This includes building data marts/data warehouses, data analysis applications, conducting data mining projects, consultancy, etc. One major problem DSS had usually had to face right at the beginning of its projects was that the data provided by the customer for processing was defective in several ways. Required items were missing from a considerable part of the data fields, other fields contained unacceptable values. Deeper analysis revealed that even data, which seemed to be correct at first sight, was often inconsistent with the content of a related field. The more serious these problems were the bigger was our concern for the desired satisfaction of our customer without first handling the data quality problems. For example, in a customer segmentation project, in which data mining techniques were used to reveal hidden relationships between past customer behavior and expected interest in new services or products, the analysis was based on the historical data cumulated through years of servicing the clients. If the provided data are erroneous and do not reflect the real facts and processes of the past, the “discovered“ relationships, rules are based on these false or incorrect data values, and may trigger erroneous decisions. This and similar problems led DSS Consulting to the recognition that, in such situations, extra work has to be invested to filter out likely or evidently incorrect data before concentrating on the original tasks. Naturally, we required help from our customers, since they were supposed to best know their data, and be able to help deciding about their correctness. This process resulted in the unplanned use of resources on both sides, which had a negative effect on the overall success of the whole project right at the beginning. However, this sacrifice had to be made to avoid larger problems later.

2 Information Quality Management (IQM) and Improvement

Based on our experience, we are now convinced that unless the customer is aware of its importance and implements processes to ensure information quality, the data maintained in the databases is loaded with errors. Unfortunately, the common situation is that quality management is not treated with the priority it deserves. Surprisingly, companies maintaining large customer and product databases are not aware of the costs of relying on non-quality data; they only recognize the problems when these data explicitly inhibit them from completing a task they feel necessary to do. For example, they may find that there is no certain answer to the seemingly simple question “How many customers do we have?” The relationship between customer records and real customers turns out to be unknown. DSS, as a consultancy firm aware of facing data quality problems, is now encouraging its partners to start information process improvement projects in order to help them exploit their data assets more effectively.

Simultaneously to our own activities, similar problems were identified and initiatives started by others, the most recognized of whom became Larry P. English with the methodology called Total Quality data Management (TQdM®) [English, 1999]. We closely studied TQdM®, invited Mr. English and decided to adopt the methodology briefly summarized below.

The figure below presents the main processes of TQdM®, and their relationships.

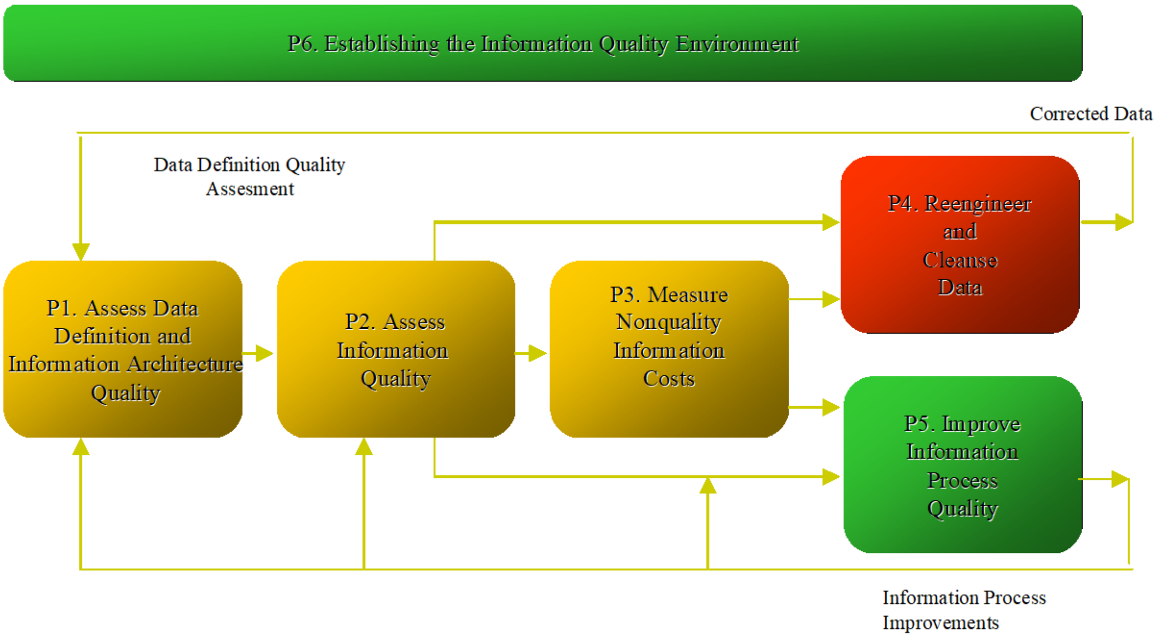


Figure 1. Total Quality Data Management, TQdM®

The practical steps of our IQM projects are partly based on the result of theoretical work, and partly on our own experiences gained in our projects. Our methodology can be considered as a tailoring of TQdM®. In its present state, it concentrates on processes P1, P2 and P4. Instead of P3, we currently institutionalize the process called "Create the Information Quality Standard". Our software tool, Quality Monitor primarily backs these processes. We are nevertheless aware of the importance of the measurement in P3 and we intend to implement it to improve the capability of our processes.

Below is a compact, simplified explanation of the processes mentioned on the figure above.

2.1 Assessing Data Definition and Information Architecture Quality

Questions we raise at this point:

What kind of data does the customer (think to) have? What do they expect their data empower them to do? Is the data really suitable to back the business processes? How is the data organized in tables? Are there any redundancies (the same data maintained in different fields)?

In this process step, we need:

- Access to the data structures we have to examine, so we can assess the data architecture.
- All written and verbal information the customer can provide concerning the definition of the data.

What we usually get:

- Data model: description of data tables, relationships
- Name of fields, data type, data length, precision
- A short definition of the fields

What we usually don't get automatically, but ask for:

- Glossaries that define the meaning of the fields more precisely, the format of the data to be entered, the domain of the accepted values, and any constraints.
- Application user guides that explain the processes in which the data is created and modified.
- Access to the application itself, used by people who create and modify the data.
- Opportunity to meet the representatives of every unit of the organization, who create, modify or use the data.

To all this, we add:

- Our own expectations and criteria regarding the data that we have derived using our experience. These may appear to be new to the customer, or may be known, but not explicitly stated.

2.2 Assessing Information Quality

Based on the results of the previous process, we run test scripts on all the fields to be analyzed. When done, we compile a document entitled Information Quality Assessment, which contains the following information about all the data fields analyzed:

- field's physical name,
- field's screen name, as it appears in the native application
- definition,
- value domain (possible values),
- rules for completing,
- relations if any, to other fields,
- total number of fields,
- number and percentage of non-empty fields,
- number and percentage of empty fields,
- number and percentage of correct values,
- number and percentage of incorrect values.

The results should only be regarded informative, since we don't have all the information yet to decide about what is to be considered correct or incorrect. In order to make a decision, we need precise definitions, which at this point are usually not available. It is the most important aim of the next step to fix the exact meaning of the fields in an unambiguous way acceptable to the customer and useful to the business.

2.3 Creating the Information Quality Standard

The Quality Standard is a document, which contains the precise definition of all data elements, and all the requirements that the data has to comply with. It is based on the knowledge gained during the information quality assessment and on the additional information we get from data stakeholders including their final approval. This approval can take a very long time to obtain, and may require several meetings with the people involved, because at this point they start to feel the responsibility of defining something which may not have been defined so far. The simple fact that we insist on getting answers to questions that have not been raised before helps understanding the importance of precise and unambiguous data definition.

The Quality Standard as a document is similar to the Information Quality Assessment. The differences are:

- Definitions are clear and unambiguous.
- The concrete number and percentage of correct/incorrect fields is not included, since the standard is independent of the concrete data tables.
- All criteria, that the specific fields have to meet, are listed. These criteria can then be programmed and implemented using our Quality Monitor system.

Just as software quality characteristics are hierarchically structured in the ISO/IEC 9126 standard, data quality attributes are classified according to the following dimensions identified by [Wang, Strong, 1996]:

- Intrinsic Data Quality (Accuracy, Believability, Objectivity, Reputation);
- Contextual Data Quality (Value-Added, Relevancy, Timeliness, Completeness, Appropriate Amount of Data);
- Representational Data Quality (Interpretability, Ease of Understanding, Representational Consistency, Concise Representation);
- Accessibility Data Quality (Accessibility, Accessibility Security).

Another classification is defined by [Redman, 1996]:

- Values (Syntactic accuracy, Semantic accuracy, Currency, Timeliness, Completeness of values, Internal Consistency, External Consistency);
- Formats (Usability, Comprehensability/Universality, Precision/Capability to represent Null Values, Flexibility, Memorization Efficiency).

The systematic consideration of the above attributes can help the specification of the requirements documented in the Information Quality Standard.

2.4 Information Process Improvement and Data Cleansing

In our practice, so far the focus has been on data cleansing. It is still hard to convince the customer about the necessity of information process improvement. The customer will require that its benefits are thoroughly explained and its return on investment is unquestionably proven. The need for data cleansing is more obvious, so we concentrate on this first. During this process, the customer gains experience with the benefits of having better quality data and this may trigger the decision to try to improve the information process.

Data cleansing is comparing the data fields to the requirements stated in the Quality Standard, and eliminating all non-conformances (deviations). The steps are as follows.

2.4.1 Creating the Data Error Report

Based on the Quality Standard, we produce a list of all the data fields that contain error(s). We call this the Data Error Report. The Data Error Report is inspected by the data stakeholders to see if the practical outcome of the comparison of the data to the Quality Standard really reflects the wish of its authors. If anything in the report is not approved (because for example a data value which has been marked wrong turns out to be correct), the Quality Standard has to be revised, and an updated Data Error Report has to be produced.

2.4.2 The Data Cleansing Process

Based on the approved Data Error Report the actual cleansing work can be performed effectively, in a focused way. The correction of most errors needs human interaction. However, considering the huge amount of data that has to be usually corrected, it is essential to seek ways to automate at least part of the work. In some cases this is feasible. Quality Monitor is conceived to support both manual and automated (batch) error correction. In any case, the data cleansing process will take a considerable time (ranging from at least a few weeks to several months). During this time the data is also altered independently of the cleansing process. New records are added, old records are modified in the source systems. As time passes by, the Data Error Report becomes more and more obsolete, and it is necessary to recreate it. This should be done periodically, from once a week to once a month, depending on several factors. By comparing versions of the Data Error Reports produced at known points in time, the cleansing process can be monitored.

2.4.3 The De-Duplicating Process

One very important aim of data cleansing is de-duplicating, eliminating superfluous instances of records that represent the same entity in the database. In order to help this process, we create a Duplicates Report, a list of record groups that contain “similar” records. The definition of “similar” is composed with de-duplicating rules. For fast and effective de-duplicating the key data values (for example, in case of humans name, place of birth, date of birth etc.) used for identification should be as error free as possible. This is why eliminating data errors should happen prior to de-duplicating. However, in a real-world environment the data will never be absolutely error-free, and de-duplicating should eventually take place. Even in a slightly error infected database, good tools for defining similarities between records will produce record groups most of which will prove to be real duplicates after human inspection.

3 Quality Monitor

DSS Consulting uses its custom developed software–Quality Monitor–to back the building processes TQdM®. Below, we give a short description of the capabilities of this tool.

The underlying principle of Quality Monitor is that data should be extracted from the source systems, loaded into Quality Monitor’s own database, where the processing, the compilation of the Data Error Report and the Duplicates Report is performed. No direct updating of data in the source systems is done. This approach precludes any security questions. All data correcting should be done either manually by trained and authorized personnel, or by loading separately compiled update tables, that contain the correct values to replace the incorrect values. Again, trained and authorized personnel should load the update tables following a security approval. Almost needless to say, all actual duplicates removal will be done either manually or by separate tools developed outside of Quality Monitor.

The modules that make up the Quality Monitor (QM) system are discussed below.

3.1 QM Server

This is the main component of the system, which offers several functions. These can be best described together with the processes that they are designed to serve.

The first function can be used to design and create a data structure into which the data extracted from the source systems will be loaded, and another one in which it will be processed. The first structure is normally just a set of tables, identical to the ones in the source system and without any constraints that, in case of an error, could stop the loading process. With an external loader, we first load the data into this structure. The other data structure can be identical to that of the original system, but we can also create a slightly modified, or totally new structure, which represents our opinion of the appropriate architecture to be used. We also apply the necessary constraints. Then we transfer the data from the first structure to this second one, and if necessary, do some pre-processing without modifying the state of quality of the data. (E.g. trim trailing spaces, but only if trailing spaces are not considered an error.)

The next function is used to compose and maintain the quality rules. The quality rules are embodied in SQL scripts written by, or under the supervision of the quality expert (a person who fully understands the rules set by the Quality Standard). There are plenty of handy functions that help the compilation and management of the quality rules. The main categories of data errors that the rules can detect are the following:

- Missing data: a required field is empty.
- Format error: the data is not in the required format.
- Length error: the data contains more characters than allowed (by business rule, not data length of field).
- Invalid data value: the value falls outside of the valid domain.
- No information: the data cannot be interpreted.
- Extra information: the required data is present, but extra information is also found.
- Integrity error: two data values are correct by themselves but conflict with each other.

A sophisticated set of tools is available for authoring de-duplicating rules. Running the de-duplicating rules will produce a list of record groups, each of which contains suspected record duplicates. These so-called generated duplicate groups will be listed in the Duplicates Report. De-duplication can be done effectively only if the data used for matching is of good quality.

The next function is used to run the quality rules and the de-duplicating rules against the data to be processed. Running the quality rules results in the Data Error Report, also created in QM Server. Each error record contains the primary key, the value of the erroneous field, the error message, and–for informative purposes and based on the parameters set in QM Server–additional fields of the affected record.

Error correcting and de-duplicating will be carried out using the QM Client and QM Batch components.

Because QM Server works on data extracted at a certain time, the Error Report and the Duplicates Report will gradually become obsolete. Therefore, with certain periodicity, the data extraction should be repeated, and the rules should be re-executed.

3.2 QM Reports

Prior to starting actual data cleansing, it is necessary to inspect the Data Error Report. The QM Reports module gives access to the report, so we can examine as many error records as we wish. The module offers handy functions to filter and search through the error records, of which usually there are plenty. After thorough inspection we can judge whether running the quality rules gave the expected results or not. In the latter case we should modify either the quality rule in question, or if this still proves to be correct, eliminate the error that is responsible for the problem in the corresponding SQL script.

3.3 QM Client

The people who will process the Data Error Report and the Duplicates Report use this component.

QM Client provides access to the Data Error Report from which the operator can choose an error record to process. The operator locks the error record for exclusive use and inspects it. After understanding what the error is, he/she has to find the correct data value in a trusted source. It is a business decision–not a technical one–what source will be considered trusted: e.g. service contracts, in case of a customer database. If the correct data is found, it should be entered into the appropriate field *using the native application*. Setting the status value and unlocking the record conclude processing the error. The possible status values are:

- Unprocessed: The operator was not able to correct the error. This is the default value.
- Corrected: The operator successfully corrected the error.
- Nothing to correct: The operator's opinion is that the data in question is correct.
- Impossible to correct: The operator's opinion is that the data in question is erroneous, but it is impossible to correct it. (Because the correct data is no longer accessible, for example.)

Along with the status, the actual date and time, and the operator's name are logged.

The status represents a valuable feedback to the error correcting process. Evaluation of the status helps the process converge towards the desired error-free database.

In the first case, the error remains in the Data Error Report, and later anyone (including the original operator) can attempt to process it. In the second case, the successfully corrected error is not displayed any longer on the QM Client screen. When we extract source data and run the quality rules against it the next time, the same field may have the same type of error again. QM Server can detect this by keeping the previous Error Report, and comparing it to the newly generated error records. In this case, the field gets a "Still incorrect" status, which should draw attention to the fact that the correction attempt was unsuccessful. Without handling this case, the unsuccessful processing could be repeated many times wasting valuable resources.

QM Client also gives access to the Duplicates Report. This report contains the record groups that have been matched by the de-duplicating rules. Processing an entry means checking out (locking) the group, and either accepting it, or breaking it into smaller groups (or single items) as necessary. Just like in the case of error correction, the operator needs accurate and trusted information to carry out this work. The end of the process is a Duplicates report with all the corrections that the human operator found necessary to perform on the machine-generated report. The actual de-duplication of the records should be done in the source system, based on the accepted list of duplicate groups.

3.4 QM Batch

The task of this component is to produce update tables, which can be loaded into the source system correcting several (maybe several thousand, or even more) errors at a time, saving considerable manual work. Update tables can be used to correct only specific sets of errors; these sets however usually contain many data elements, so a significant improvement is made.

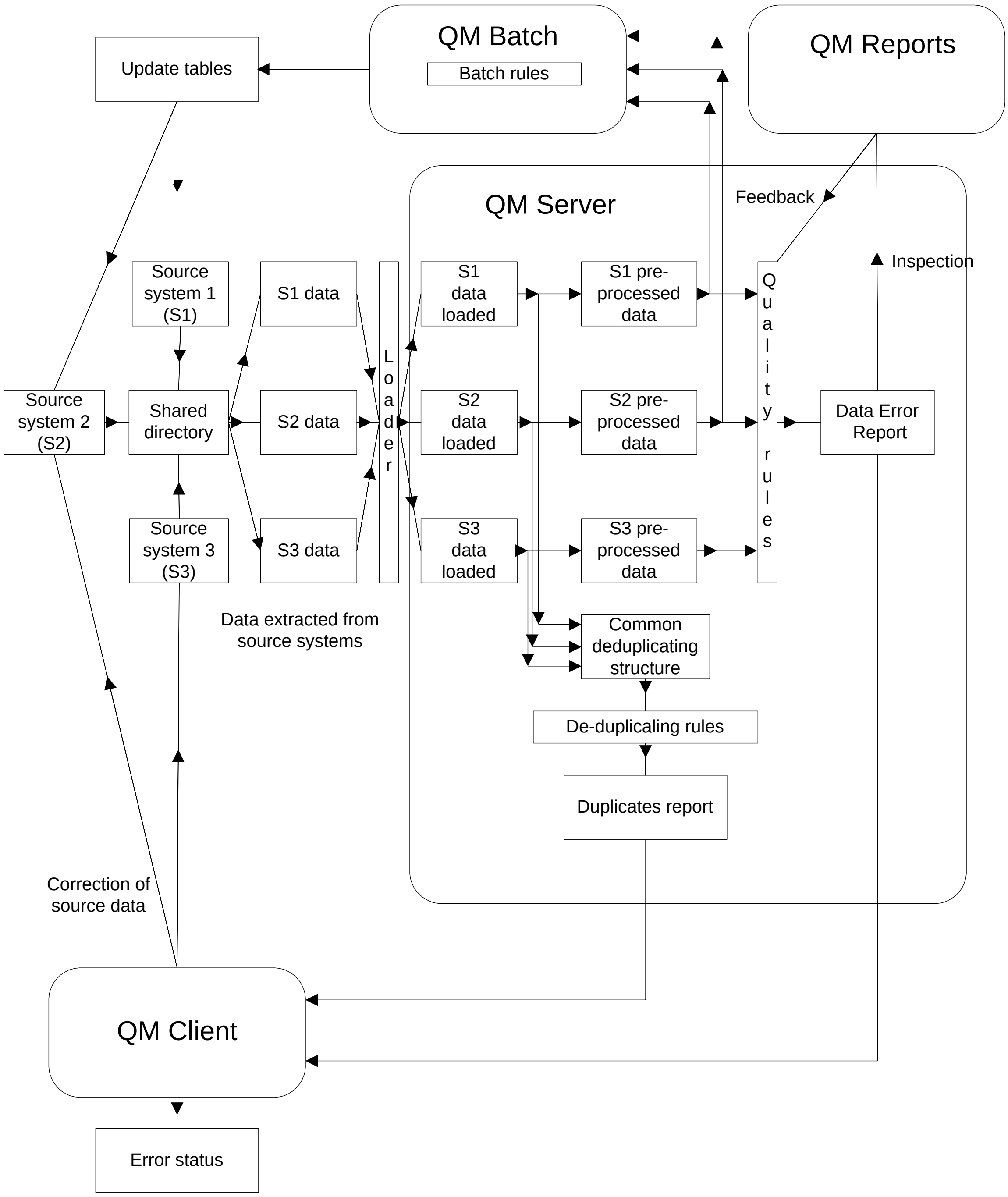
Based on the Data Error Report, a skilled user can point out error types that are good candidates for batch cleansing. Examples are

3.5 QM Progress

In a large organization holding huge amounts of data, data cleansing is usually a timely process, involving many people from different organizational units. QM Progress helps keep track of the process, by providing statistics and forecasts from historical data. Also, there are several control functions that proved to be useful. For instance, an operator makes a mistake and sets the status of an unprocessed error to "Nothing to correct". If a supervisor inspects a report called "Error records set to <Nothing to correct> during February" produced by QM Progress, and the supervisor reveals the mistake, he can reset the status to "Unprocessed". Otherwise this error would disappear from the data cleansing process. Another example: if we find out that the script corresponding to a quality rule produced incorrect results using the control functions of QM progress, we can delete all error records from the Data Error Report that have been produced by the erroneously implemented script.

Naturally, only trained and authorized people should have access to the control functions of QM Reports.

The Quality Monitor architecture



4 Conclusion

We started by highlighting that information quality could only be achieved by improving the processes that create data, and finished by describing the first step of this improvement, which is data cleansing of already existing information repositories. We must stress however, that data cleansing alone attacks only the symptoms of the information quality problem, which will resurface unless the data creation processes are not improved themselves [English, 1999]. The Quality Standard, introduced in section 2.3 has two roles. One is to set the rules for data cleansing; the other is to become an organization-wide standard that will guide the data creation and manipulation processes further on. If this standard is put into force, understood and observed by all, the organization is on the right track towards quality information. Of course, the Quality Standard should be revised and updated as changes in the organization's environment and activities render it necessary. However a team of experts should do this consciously, not all individuals who have access to the data–similar to legislation.

When DSS started to develop the QM system, we had relatively little time to produce a tool that worked. So we started developing the QM Server and Client modules, as these seemed to be the essential components. As we gained more experience in a real-world project, these modules were refined, and enhanced. The need for batch cleansing emerged, because the number of data fields to be corrected turned out to be more than initially expected, and the organization could not afford to provide the human resources required. Unexpected ratios of "incorrectly performed manual corrections" (human error) lead to the need of introducing more sophisticated status handling. The scope of the data cleansing project called for a tool to manage and predict the demand on resources: QM Progress. So although the QM system seems to be in the focus of our article, we would like to emphasize that it embodies the quality management process–planned in theory and shaped by practice.

Our experience with primarily telecom and banking organizations shows that even data cleansing is not started without a specific critical situation, which makes it unavoidable. The simple need to carry out the everyday tasks more effectively is usually not enough to encourage an IQM project. It is still not widely recognized that non-quality data constantly causes losses in profits even if this has no spectacular consequence.

The history of the software process movement shows that convincing managers about the long-term business benefits of process improvement is not an easy undertaking [Biró, Tully, 1999], [Biró, Messnarz, 2000]. However, this is the path we consider our mission to follow regarding information quality, in order to contributing to a better quality of life in the Information Society.

5 Literature

[Biró, Tully, 1999] 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).

[Biró, Messnarz, 2000] 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.

[English, 1999] English, Larry: Improving Data Warehouse and Business Information Quality. New York: Wiley Computer Publishing, 1999.

[Redman, 1996] Redman, T.C. *Data Quality for the Information Age*, Norwood, MA: Artech House, 1996.

[Wang, Strong, 1996] Wang, R.Y.; Strong, D.M. "Beyond Accuracy: What Data Quality Means to Data Consumer," *Journal of Management Information Systems*, 12 (1996) pp.5-34.

Author CVs

János BAJZA

János BAJZA has been a senior consultant at DSS Consulting Ltd. since the founding of the company. Following the setup of the information infrastructure of the new company he started working in the Information Quality Management group. He has participated in several IQM projects as a senior consultant.

He has an MSc. in Electric Engineering from the Technical University of Budapest. He is fluent in Hungarian, English and Spanish.

He started his professional career at the Computer and Automation Institute of the Hungarian Academy of Sciences as a research worker. He worked on several computer network-oriented projects and assignments. He took part in the development of X.25 network based software products, and in the installation and trouble-shooting of these systems throughout Hungary. As an employee of Hungarnet Association (the Internet service provider for the whole academic community in Hungary) he held courses on Internet applications. Finding interest in word processing, later he worked as the Hungarian Project Manager in international projects for the localization of commercial word processing packages. He was first involved in database oriented projects in MIS Consulting, Ltd., where, while using data mining techniques, he first became aware of the quality problems that can hinder the processing of data. In DSS Consulting, together with his colleagues, he is dedicated to tackle these problems using a methodology and corresponding software tool, which has already proved to be effective in several successful projects.

Dr. Miklós BIRÓ

Dr. Miklós BIRÓ is a professor at the Department of Information Systems of the Budapest University of Economic Sciences and Public Administration with 25 years of software engineering and university teaching (including professorship in the USA), and 15 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 Bootstrap and SPICE (Software Process Improvement and Capability dEtermination - ISO/IEC 15504) assessor. He gave 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 initiated and managed the Hungarian participation in numerous European multinational projects and organisations committed to software process improvement (European Software Institute, Bootstrap Institute). He was the initiator and head of the Information Society Technologies Liaison Office in Hungary for the European Union's 5th Framework Programme. He was expert consultant for Hungarian and international organizations (European Commission, Hungarian Committee for Technological Development, Investment and Trade Development Agency of Hungary, Hungarian Airlines, UNIDO, International Software Consulting Network ,...). He played 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.

He has numerous publications in international scientific and professional journals (Software Process Improvement and Practice , Software Quality Professional , 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 author of Hungarian and English language book chapters 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.

Enhancing Software Engineering using Team-working Techniques: A Risk Management Scenario

John Elliott and Lisa Tipping, QinetiQ UK

Abstract

The software engineering profession has significantly progressed over the last few decades. However, the resultant quality of software is still heavily dependent on the way that software developer's are selected, trained and managed. This is because software development involves innovation and creativity, human factors, management techniques and organisational behaviour. One feature of people management is team-working (i.e. how team roles are expected to communicate). Team-working significantly contributes to the success of software projects yet team-working techniques have received little attention compared to the software process paradigm. This paper describes a trial of team-working techniques and argues for more emphasis and research on the integration of team-working techniques into software process definition and guidance (including EC process based standards). A software risk management scenario is used to demonstrate current team-working technology and techniques, and to highlight the measurable benefits that are expected for software customers and developers. This paper presents some early findings of the EU supported TEAMwork project.

Keywords

Team-working; Development environments; Software; Risk management; Roles; Processes

1 Introduction

The software domain has been driven towards an 'engineering' approach in response to ongoing concerns about software product quality. An engineering approach concerns the discipline and understanding of software production to reduce software product variablity. There have been advances in key areas such as computer science, software process definition, engineering based methods and much emphasis on the semi-automation of software production.

The automation themes have attracted significant research investment since the 1980's. Early research had two major objectives: to develop engineering methods and tools to support component lifecycle processes and to create integrated project support environments (IPSEs). IPSEs had the objective of facilitating the sharing of project information and supporting the data exchange between component tools. The IPSE developments were significant and they generally were based on process and information paradigms. However, they did not gain popularity for many reasons. One reason was that they did not adequately address the needs of teams and their members.

In the 1990's, attention turned to refining the process-oriented quality assurance paradigm to be applicable to software. This also led to the development of software measurement and assessment techniques, much of which focussed on process assessment, for example CMM and SPICE. In the latter part of the 1990's, the weaknesses with process based management controls, although remaining a valid contribution, became evident as more attention was being given to human factor issues relating to design/creativity and also to understanding human frailities and motivation. The concept was that improved methods and tools might evolve to account for such human factors; this evolution is still relatively immature. Furthermore, techniques for software individuals [Humphrey 1997] and teams [DeMarco and Lister 1999] have been slowly evolving.

The software industry is now at a stage where process understanding is fairly advanced and process based method and tools are more sophisticated and continually evolving. The performance of the people aspects of software projects and organisations (i.e. teams and individuals) are starting to be addressed more seriously, all with the aim of reducing software product variability and predicability.

This paper describes current 'work in progress' research and trials that are being undertaken to improve software project success by enhancing support for software teamworking. This work is being undertaken by the TEAMwork project consortium, supported by the EU's IST programme, see www.euteamwork.net.

2 TEAMwork Project Overview

TEAMwork is an EU project in the Information Societies Technologies (IST) Programme [IST 2001]. The objective is to increase the capacity of distributed organisations to operate in an e-working environment. TEAMwork's objective is to develop and apply a novel and integrated solution for distributed e-working organisations through the innovative and creative combination of technology, methodology and a skills solution. The project will use a configurable technology platform (NQA) that can be evolved and adapted to user needs for flexible working and effective coordination and planning in e-working environments. The methodology (BESTREGIT) will set out management and team-building processes and organisational practices for effective management and motivation of distributed e-working teams. The skills solution will identify the societal impact (based on user feedback) on those working in a distributed organisation.

TEAMwork's integrated solution for e-working in distributed organisations will be designed to operate across member states and regions in Europe. The solution is being applied and adapted in a series of trials in three different user-domains: software engineering, research network and public service domains. The results of the trials will be disseminated across the community.

The participants in TEAMwork are: The Technology Network Ireland (TecNet) from Ireland (lead partner), International Software Consulting Network (ISCN) from Ireland, Fundaciaon General Valladolid (Fguva) from Spain; Computer Automation Institute of Hungarian Academy of Sciences (MTA SZTAKI) from Hungary; University of Maribor (UniMar) from Slovenia, Danube from Austria, Hyperwave (HWR&D) from Germany, QinetiQ from the UK, GMV from Spain, Ausbildungs-Partnerschaft Sudosterreich (APS) from Austria and BisDato from Austria.

This paper describes the early results from QinetiQ's studies within the software engineering domain.

3 Software Engineering and Team-working

Software team-working is essentially:

Identifying and defining software project roles and their communication activities. Communication will involve each role sending, acknowledging and responding to information, often dictated by sequencing rules. Each role will be performing some or all activities of a defined process using available and chosen methods and tools. Roles may be performed by one or more named individuals who may also perform more than one role.

There are a number of important observations about this definition of team-working:

- Team-working is all about 'who communicates what with whom and when' after having completed one or more activities. Hence the emphasis is on 'roles' rather than process.
- Team-working can only be effective if the needs for communication are well defined and are also implemented in a disciplined and consistent manner.
- There is considerable variation in implied role definitions across associated process based standards, sectors, organisations and projects.

- Role based communication represents an essential feature of inter-process flows (as in process or workflow modelling techniques), and is often hidden or not strongly related to people and their jobs.
- Information flows between roles should be capable of being standardised in many cases and thus require 'template' definitions to define chosen document contents.
- Roles may be performed at one or more distributed locations creating an extra communication complication associated with all 'virtual' (or distributed) organisations (geographical e-working).
- Roles may be performed by different organisations representing separate business enterprises or by specialist teams within the same enterprise (logical e-working).
- The status of a project may be captured by the state of each communication (e.g. status is completed).
- Process based activities, for mandatory or discretionary use, act as guidance for the roles in generating or checking individual or integrated outputs.

The problem of addressing team-working characteristics to reduce project risks and variability is clearly challenging. Furthermore, team-working needs to be integrated with process activities and other human factor and organisation aspects. Team-working communication will arise across the whole spectrum of the software lifecycle process and hence there are many potential areas within software development where improved teaming and communication techniques will benefit projects.

An issue is how to decide which development processes will benefit most from team-working techniques. This improvement investment decision may be made using a goal-based approach such as BESTREGIT [O'Leary et al 2001]. This methodology is the result of a previous EU Leonardo project BESTREGIT and was designed to help organisations structure their business, set business goals, improve team-work and infrastructure, and build a quantitative feedback loop to learn and change in an objective way. The methodology is composed of four steps: initiation, goal analysis, teamwork analysis and experimentation. Initiation concerns defining the role of a process improvement professional that will be responsible for facilitating change. Goal analysis concerns creating a mission statement, and evolving business and work goals from that mission. The teamwork analysis concerns identifying roles, work products, templates and infrastructure analysis. Finally, in the experimentation stage, personnel are assigned to roles, experiments are selected and prioritised, measurements are made and lessons are learnt and disseminated.

The team-working concepts and the BESTREGIT method was used to select and rationalise those activities, as scenarios, of a typical software engineering organisation whose software projects would most benefit from enhanced team-working techniques.

4 Why Risk Management?

There were two considerations for selecting risk management as an appropriate process area and project scenario for which to develop team-working techniques and tools. These considerations were a typical software organisation business and the overall software lifecycle process. The business and process areas were analysed using the goal-based BESTREGIT method. The purpose was to highlight one or more key area(s) for which team-working trials may be determined so that a role-based team-working environment could be created and subjected to user evaluation.

4.1 Software business analysis

A 'typical' software business provides a range of business services and products. These include:

- Software development: bespoke software or software products (e.g. tools).

- Software assessment: process, people and project evaluation and measurement.
- Specialist support services: project, quality and risk (including safety) management; process assurance; systems and software engineering; requirements management; and training.
- Innovation services: addresses best practice, business and technical excellence and assessment models, and promotes, disseminates and researches software-related activities.

These services arise across market sectors, (e.g. defence, railways, communications) and technical system capability, (e.g. real time systems, e-commerce systems, process capability). Software businesses need to adopt a 'dual use' approach to their products, processes and people across different market and technical domains for business viability reasons. The choice of exploitable team-working scenarios within different business situations needs to account for such dual use needs.

The application of team-working techniques to the software domain will need to take account of the diversity of sector process standards Major standards that are influencing software process harmonisation include: ISO/IEC 15504 for process assessment (SPICE), DO 178B for whole life software cycle processes in aerospace and ISO 9000 for general product quality areas. There are used in TEAMwork's software domain studies.

4.2 Software process analysis

The 'underlying' software process for the TEAMwork study was derived from the ISO/IEC 15504 (SPICE) standard on process assessment. The main aim of SPICE is to provide a framework from which assessment methods may be tailored to meet specific process, project or organisation assessment needs. SPICE offers guidance into acquisition, development and assessment processes. The underpinning development processes are described within a reference model contained in the ISO/IEC 15504 documents, and this provides a consensus about what processes definitions should apply to software domain businesses. The processes have been defined in alignment with ISO/IEC 12207:1995, Information Technology - Software Life Cycle Processes. The reference model groups the processes in the process dimension into three life cycle process groupings, which contain five process categories, according to the type of activity they address. These are structured into primary life cycle, supporting life cycle and organisational processes and are shown in Figure 1.

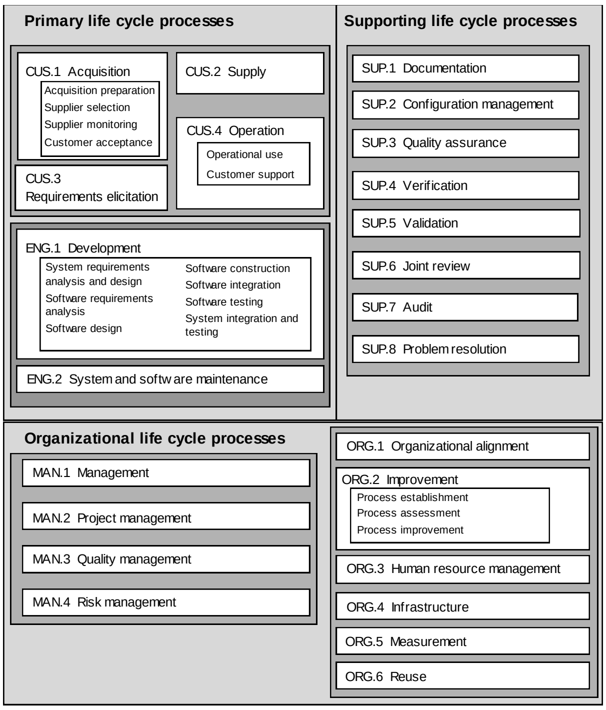


Figure 1 ISO 15504 Reference Model: SPICE Process Architecture

SPICE processes are only described at a high level providing objectives and general descriptions, (not activities, inputs and outputs). However, SPICE provides a template for process definitions in terms of process attributes. Consequently, the application of team-working studies to the software domain requires more detailed information about the processes than is provided by SPICE. The variation of process detail across generic and domain specific standards provides a challenge when deriving a representative approach to devise reusable and configurable team-working techniques.

4.3 Scenario and process selection

An integrated team-working approach to software management was considered as a major opportunity for exploitation. Software management also maps onto the SPICE reference model, in terms of the customer-supplier, organisation, management, engineering and support processes. The main management areas were identified as follows:

- Project Management: high level processes are well defined but common sources of difficulties are project size and complexity. Distributed teams and organisations increase the complexity.
- Risk management: a key part of project management and should address all business, project, and technical risks related to stakeholder interests.
- Requirements management: this is really part of a technical activity but it is also critical to ensuring quality software systems, and hence very exploitable if adequately integrated into a project management context.
- Quality management: this is part of a requirements statement and closely related to acceptable risks and to defining a project's approach (e.g. processes), and combines quality assurance (e.g. procedural) and technical (e.g. fitness) management perspectives.

In terms of generic applicability, software risk management was chosen as the major scenario to be explored further and be enhanced by team-working techniques. This is because a risk management based process improvement offers the maximum exploitation value across the software domain; software is both important in many systems but also a major risk area [Van Scoy 1992]. This is further enhanced by an increase in the distributed working needs and the vital need of effectively involving all relevant staff in supporting risk management activities, irrespective of where they may be located. Applying team-working techniques to risk management will increase rigor, and hence its effectiveness on projects. The control of projects will improve because the communication problems normally associated with distributed working environments are reduced.

4.4 Risk management and team-working

A novel and unique team-working approach to risk management is being developed. The risk management scenarios being developed as part of the TEAMwork project are founded upon a role based framework. Cooperatively, project stakeholders work together using a distributed e-working environment to aniticipate and avoid problems by managing project risk. Specifically, the risk scenarios establish a working environment where all stakeholders in a project have the ability and motivation to look ahead and handle risks before they become problems. This is accomplished through a comprehensive and practical set of processes, roles and deliverables. The team-working approach ensures that risks are proactively managed and that all risks are cost effectively managed throughout the lifecycle of the software development. The approach becomes 'instituationalised' and provides management at all levels with the information required to make informed decisions on issues critical to project success.

In pursuit of a broader integrated approach to software management, any study of team-working in risk management has to recognise the following:

- The business and project management context must be defined (a wider project scenario) and this will affect risk cultures, perspectives and tolerances.

- This context needs to also include the approach to quality and requirements management, and any key relationships (different sub-scenarios).
- Requirements management (that influences the project risk and complexity) will provide significant information for the risk management scenarios.

5 Development of Risk Management Environments

The concept is to introduce role based (not process based) team-working environments to support software management, initially to be illustrated as a risk management scenario. The environment will enable the communication expectations of defined risk management roles to be executed with the use of an user-friendly automated system. The development of a risk management environment requires:

- The development of a role model for risk management
- The implementation of the role model to be translated into a software component for use in an existing role based communication platform technology.

The role models will be described using a simple, user-friendly notation. The existing role-based technology is Network Quality Assurance (NQA), and has been derived from the previous EU HYMN project [O'Leary et al 2001]. It is an adaptable and flexible platform for creating effective team-working system solutions to operate across Intranet and Internet networks to support distributed e-working. More details about NQA can be found at www.iscn.com. The development, and user trials, of this evolving risk management environment is currently 'work in progress' and the remainder of this paper concentrates on the role model aspect of team-working based risk management environments.

5.1 Risk Management Purpose

'The purpose of the Risk management process is to identify and mitigate the project risks continuously throughout the life cycle of a project. The process involves establishing a focus on monitoring of risks at both the project and organisational levels' [ISO/IEC 15504].

Chapman and Ward [1997] describe the essential purpose of risk management is to improve project performance via systematic identification, appraisal and management of project-related risk. A focus on reducing threats or adverse outcomes ('negative risk') misses a key part of the overall picture. An aim of improving performance implies a wide perspective that seeks to exploit opportunities or favourable possibilities ('positive risk'). Risk management is an activity that is essential to apply throughout the whole lifecycle (addressing negative threats and positive opportunities), and not just during the initial planning phase (when negative threats to success are examined and mitigated).

5.2 Risk Management Process

There are many risk management processes defined, but all of them have some key processes in common. The risk management scenario develops and refines the SPICE [ISO/IEC 15504] process using the generic risk management process advocated in BS 6079-3:2000, as in Figure 2. This BS6079-3 model has been enhanced with a more comprehensive approach advocated in Project Risk Management: Processes, Techniques and Insights [Chapman and Ward 1997]. The UK MoD approach [DBMS 1998] was also examined. It should be noted that there is no EC wide generic risk management standard at this time. These process variations are compared below.

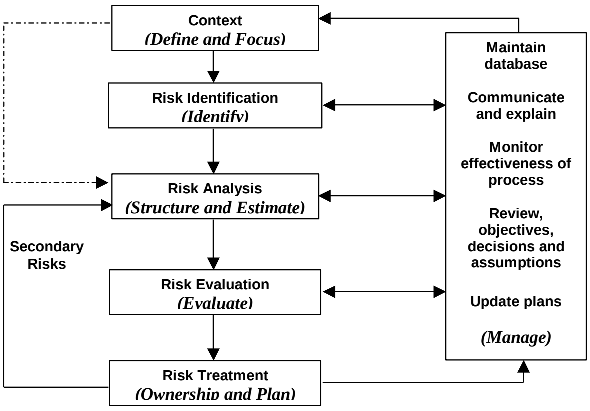


Figure 2: **Modified from BS6079-3: 2000, Project Management – Part 3: Guide to the management of business related project risk**

SPICE	BS 6079-3	UK MoD	Chapman & Ward
	Context		Define
Scope and risk management strategies			Focus
Risk identification	Identify	Identify	Identify
Risk analysis	Analysis	Analysis	Structure
			Ownership
			Estimate
	Risk Evaluation		Evaluate
		Planning	Plan
Define, apply and assess risk metrics Take appropriate action	Manage	Management/Control	Manage

5.3 Risk Management Roles

A team-working analysis produced a set of roles and their actions from an evaluation of SPICE and risk management processes. A simplified presentation of the role model for risk management has been subdivided into two scenarios: Risk Planning and Risk Control (to monitor the risks identified in planned activities). These graphical role models are in Figure 3.

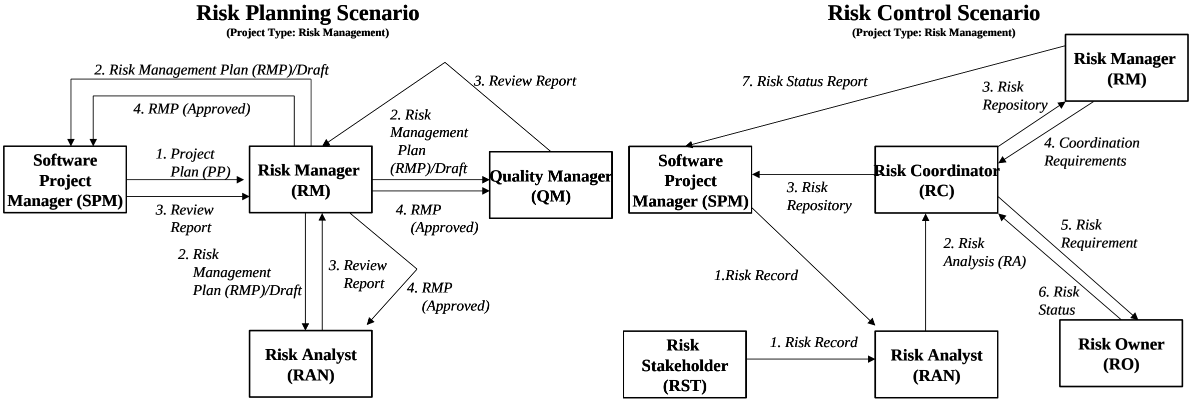


Figure 3 **Risk Management: Role Models for Risk Planning and Control Scenarios**

These models show roles and information flows as documents. Some documents may be standard templates for reuse and consistency within risk management activities, e.g. for plans or review reports.

The roles identified are:

- **Software Project Manager (SPM):** Defines the project scope and is responsible for project risk management. Approves risk analysis and risk treatment. Produces and updates the project plan.
- **Quality Manager (QM):** Approves the project and the risk management plans.
- **Risk Manager (RM):** Defines and implements appropriate risk strategies (and metrics), and documents them in a risk management plan. Facilitates the identification of risks and their treatment. Manages and reports risk status. Produces and updates the risk management plan. Directs the Risk Coordinator (RC) with coordination requirements.
- **Risk Analyst (RAN):** Approves the risk management plan. Facilitates the identification of risks and analyses those risks.
- **Risk Coordinator (RC):** Collates and combines the risk information from different risk analysis perspectives, and stores the information in an accessible and shared repository. Reports the status of risk management to the Risk Manager (RM).
- **Risk Owner (RO):** Owns and takes responsibility for the management of specific risks. Suggests possible treatments. Reports status of the risks. Takes action where required.
- **Risk Stakeholders (RS):** Identifies risks and possible treatments. Risk Stakeholders may include customer, users, senior management, software project manager, project team members and other engineering groups.

There are a number of information work products (documents) involved, including: Risk Management Plan, Review Report, Risk Record, Risk Repository, Risk Requirements, Risk Status, and Risk Coordination Requirements. Standard templates are being designed to capture the document structures for ease of use and consistency.

These scenario-based role models will be used to create a risk management environment. This environment and its constituent role models will be refined within QinetiQ's user trials. The associations between roles, work (process based) activities and results are now described.

5.4 Risk Management Roles and Work Products

The details of the team-working analysis of the risk processes are captured in the table below. This analysis illustrates the links between roles, flows and work products for each risk process.

Name of Process	Roles	Flows	Work Products
Context	➢ Software Project Manager (SPM) ➢ Customer (RST) ➢ Users (RST) ➢ Technical Manager (RST) ➢ Risk Manager (RM) ➢ Quality Manager (QM)	**Inputs:** ➢ User requirements ➢ Customer and organisational communication ➢ Contract **Outputs:** ➢ Terms of reference (could be defined in a statement of work or project plan) ➢ Risk management plan (RMP)	A clear, unambiguous, shared understanding of all relevant key aspects of the project and risk management process (RMP). **Terms of Reference** ➢ Project objectives ➢ Project scope ➢ Project strategy **RMP** ➢ Who is doing RMP for whom? ➢ What is the scope of the RMP? ➢ What resources and timeframe? ➢ What models and methods?

Name of Process	Roles	Flows	Work Products
Identify	➢ Software Project Manager (SPM) ➢ Risk Manager (RM) ➢ Risk Coordinator (RC) ➢ Risk Analyst (RA) ➢ Risk Stakeholders (RST)– customer, user, software project manager, technical manager, subcontractor etc.	**Inputs:** ➢ Terms of reference ➢ Risk Management Plan ➢ Brainstorming ➢ Checklists ➢ Interviews **Outputs:** ➢ Risk list, log or register ➢ Risk classification or taxonomy	➢ All key risks identified ➢ Understanding of threats and opportunities ➢ Classified ➢ Characterised ➢ Assumed response (at least one)
Analysis	➢ Software Project Manager (SPM) ➢ Risk Manager (RM) ➢ Risk Coordinator (RC) ➢ Risk Analyst (RA) ➢ Risk Stakeholders (RST) – customer, user, software project manager, technical manager, subcontractor etc.	**Inputs:** ➢ Terms of reference ➢ Risk Management Plan ➢ Risk list, log or register **Outputs:** ➢ Analysis scheme ➢ Updated risk list, log or register with additional information about likelihood and impact ➢ Generation of a set of pictures of graphs, and defining associated mathematical models (where appropriate) that capture all the key relationships.	➢ A clear understanding of the implications of any important simplifying assumptions between risks, responses and project plan activities. ➢ A basis for understanding which risks and responses are important. ➢ Estimates of likelihood and impact in scenario or numeric terms, the latter including identification of assumptions or conditions.
Evaluate	➢ Software Project Manager (SPM) ➢ Risk Manager (RM) ➢ Risk Coordinator (RC) ➢ Risk Analyst (RA) ➢ Risk Stakeholders (RST)– customer, user, software project manager, technical manager, subcontractor etc.	**Inputs:** ➢ Risk list, log or register **Outputs:** ➢ Evaluation criteria ➢ Prioritised and ranked risk list, log or register including go/no-go/maybe criteria. ➢ Revised project plan (if appropriate)	➢ Diagnosis of all important difficulties and comparative analysis of the implications of responses to these difficulties. ➢ With specific deliverables such as a prioritised list of risks, or a comparison of project plan and contingency plans with possible difficulties and revised plans.
Treatment	➢ Software Project Manager (SPM) ➢ Risk Manager (RM) ➢ Risk Coordinator (RC) ➢ Risk Analyst (RA) ➢ Risk Stakeholders (RST)– customer, user, software project manager, technical manager, subcontractor etc. ➢ Risk Owner (RO)	**Inputs:** ➢ Risk list, log or register ➢ Project plan ➢ Risk management plan **Outputs:** ➢ Revised risk management plan ➢ Risk list, log or register containing the risk assessment including ownership information	➢ Risk assessment containing a list of threats, prioritised, assessed in terms of impact and a range of options for treating the risk (including proactive and reactive contingency plans). ➢ Risk management plan ➢ Detailed task breakdown and resource requirements ➢ Policy for risk allocation issues: scope the policy (why, who and what?) ➢ Operational contracts: details of the approach, the instruments, and the timing.

Name of Process	Roles	Flows	Work Products
Manage	➢ Software Project Manager (SPM) ➢ Risk Manager (RM) ➢ Risk Coordinator (RC) ➢ Risk Analyst (RA) ➢ Risk Stakeholders (RST)– customer, user, software project manager, technical manager, subcontractor etc. ➢ Risk Owner (RO)	**Inputs:** ➢ Risk list, log or register ➢ Project plan ➢ Risk management plan (including risk metrics) **Outputs:** ➢ List of risks/response issues whose status has changed and which require management attention ➢ Exception reporting of significant events ➢ Updated objectives, decisions and assumptions ➢ Updated project and risk management plans	➢ Diagnosis of a need to revisit earlier plans, and to initiate re-planning as appropriate. ➢ Exception (change) reporting after significant events, associated with planning.

6 Benefits of Applying TEAMwork to Risk Management

The main benefit of using the TEAMworking approach is that it provides a systematic approach to implementing processes that is directly linked to project (in terms of cost, quality and timescales) and business objectives (e.g. increasing profitability and market share). Risk management is a key activity to achieving business and project goals. Furthermore, team-working represents an important enhancement to conventional risk management to help meet such goals.

The BESTREGIT methodology requires that a mission statement and goals be defined for an organisation. A clearly defined mission statement determines and communicates what the organisation is trying to achieve and provides a vision for the future. The benefits of clearly defining goals are:

- Attention is focused on the contribution the products and services make to the business and society.
- Criteria for success (and how the criteria will be measured) must be identified.
- The future role of the organisation is identified in terms of likely business opportunities, where the organisation fits, what society needs and what the organisation can contribute.
- Weaknesses are recognised.

Scenarios are constructed from the goals. The scenarios allow the logical consequences of processes to be explored and defined. The scenarios identify roles and work products. The benefits of defining scenarios are that people understand what is expected of them, what work products they must produce, when they must produce the work products and in what format. For the organisation, this means:

- Encourages team-working. Successful teams are characterised by effective communications between all team members [Higuera et al 1994]. Risks, problems, and crises often occur when communications break down in an organisation.
- That goal and priorities are clearly defined.
- A free flow of information exists among project stakeholders.
- A formalised communication structure exists so that stakeholders have access to the information they require.
- The value of each stakeholder contribution is recognised.

- Software development can be viewed within the larger systems-level definition, design and development.
- Less rework is required.
- Templates reduce the amount of rework and clearly define what must be produced and by whom.
- A continuous improvement cycle is created that is characterised by routine improvement to the scenarios.

In addition to the above benefits, the TEAMwork approach formalises communication and encourages the sharing of knowledge about risk areas. This facilitates the identification and analysis of risk. Furthermore, the risk management scenario prioritises and focuses attention on important project and business issues and objectives, integrates risk management into project management activities, and integrates risk management with the process improvement cycle.

Software process improvement is comprised of technology, work processes and people management. Monitoring and tracking critical items through risk management can achieve balance in all of these areas. In summary, if people are given proper leadership, and are trained, supported and managed, they are capable of performing astounding feats [Humphrey 1997]. The TEAMwork approach helps to facilitate the implementation of this vision.

7 Summary and Way Forward

The QineitQ user trials in the software domain are concerned with applying the BESREGIT methodology to define team-working requirements, and using the derived software environment (based on the NQA platform tool) to capture the risk management scenario. Specifically, QinetiQ has used the ISO 15504 (SPICE) standard [ISO/IEC 15504] as a reference model as well as complementary civil and defence standards.

The selection of a business area and scenario was related to the exploitation prospects. The key question is 'Will a future tool or environment using a particular scenario be adequately attractive in providing a demonstrable cost-effective solution to commonly known problems of planning, communication and coordination within the software domain/industry in general?' From this BESREGIT methodology evaluation, the risk scenario emerged because effective risk management is dependent upon good communication. The scenario has been captured in the form of role models. Communication underlies the TEAMwork methodology and is the foundation for team building scenarios in a distributed e-working environment; risk management is a context specific scenario.

For the future user trials, creating the team-working based risk management environment involves translating the initial risk management role models into scripts for the generic NQA platform environment. The user trials will reuse and 'configure' this risk management environment to be tailored to different software business and project situations. Configuring means: Refine exact document flows and templates between roles; Assign people to roles; and Set up email addresses for communication purposes.

At the end of the TEAMwork project, the environment will be ready for use to support risk management throughout the project life, enabling key roles to interact irrespective of their location, and be prompted to ensure that essential communication occurs. In addition, exploitation opportunities for a team-working based risk management tool will have been identified.

For further information, or to provide feedback, please contact the authors (jjelliott@qinetiq.com or lctipping@qinetiq.com).

8 Literature

BS 6079-3:2000 — Project management – Part 3: *Guide to the management of business related project risk*. BS 6079-3:2000

Chapman and Ward 1997	Chapman C and Ward S, *Project Risk Management: Processes, Techniques and Insights*. Wiley 1997
DeMarco and Lister 1999	DeMarco T and Lister T, Peopleware: productive projects and teams, Dorset House Publishing Co., NY, 1999
DPMG 1998	*Managing Risk*, Defence Procurement Management Guide, DPMG/MULT/310 Issue 1.0, November 1998
Higuera 1994	Higuera R, Gluch D, Dorofee A, Murphy R, Walker J and Williams R, *An Introduction to Team Risk Management*, version 1.0, Special Report, CMU/SEI-94-SR-1.
Humphrey 1997	Managing technical people: innovation, teamwork, and the software process, Addison Wesley, 1997
IST 2001	TEAMWORK, Annex 1, Description of Work, Information Society Technologies (IST), 2001
ISO/IEC 15504	SPICE: ISO/IEC TR 15504-2: 1998(E). Information Technology, software process assessment. *Part 2: a reference model for processes and process capability and Part 5: An assessment model and indicator guidance*
O'Leary et al 2001	O'Leary E, Foley B, Messnarz R, Nadasi G, Bernhard R, *Experience with Team-Work Concepts for Virtual Organisations*, Proceedings of the EuroSPI 2001
Van Scoy 1992	Van Scoy Roger L, *Software Development Risk: Opportunity not Problem*, Technical Report CMU/SEI-92-TR-30, September 1992

9 Author CVs

John Elliott

John Elliott is a Research Manager within the UK QinetiQ's Systems and Software Engineering Centre. John has a MSc in Information Systems Engineering, and is a Member of the British Computer Society and the Institute of Statisticians. He is currently leading various research activities associated with systems and software product quality, measurement, critical systems and risk management. He is QinetiQ's leader and board member within the EuroSPI consortium. He has spent over 25 years specialising in advanced system and software engineering, in particular, where quality and integrity is vital. John has experience as manager, developer, consultant and assessor of software-based systems. Prior to joining QinetiQ in 1997, he led two separate software assurance groups at UK consultancy companies where he led system assessments and state of art research programmes in safety critical systems. John has published and presented papers in many software, safety and security conferences/workshops.

Lisa Tipping

Lisa Tipping currently works as a software process improvement professional for the Systems and Software Engineering Centre (SEC) at QinetiQ. Lisa has a BSc in Applied Physics, a MSc in Software Engineering and a MPhil. The title of the MPhil is 'Facilitating the Development of a Modern Electronic Design Process using Soft Systems Methodology (SSM)'. She is a Chartered Physicist and a Member of the Institute of Physics, and a Chartered Engineer and a Member of the Institute of Electrical Engineers. Her role involves a variety of process improvement activities including: the development of a SEC Capability Maturity Model capability, advises software projects about software quality assurance and occasionally performs internal audits. She is the chairperson of the QinetiQ wide software metrics group. Lisa gives software related training courses, and contributes to the systems engineering activities. Lisa has previous experience of implementing a software metrics programme within the software development department of a major financial institution, as part of a CMM process improvement team. Before this, Lisa implemented an ISO 9001 TickIT quality system within a blue chip engineering company.

Performing Process Assessment to Improve the Supplier Selection Process: an Experience in Automotive

F. Fabbrini, M. Fusani, G. Lami, E. Sivera

Abstract

To succeed in the motor vehicle market place, the ability to employ new technology in all aspects of the business is crucial, since today's modern vehicles base their reputation mainly on performance and quality: increasing competition creates demands for more features, quality and reliability at the same time, yet with strong pressures for delivering with less cost and reduced timescales. But as functionality increases so does complexity, and as project schedules get shorter the pressure on software development costs gets stronger: with a larger and larger part of the functionality in the modern motor vehicle controlled by software, both the motor vehicle manufacturer and the supplier need to address these issues.

A project will be presented, carried out by Fiat Auto in co-operation with the Centre for Software Certification of the Italian National Research Council, to set up a methodology supporting the management of software projects and suppliers. Problems peculiar to this application field are outlined along with possible consequences and ways-out.

Keywords

Process Assessment, Supplier Selection Process Improvement, SPICE Reference Model.

1 Introduction

The past four decades have witnessed an exponential increase in the number and sophistication of electronic systems in vehicles. The growth of electronic systems has had implications for vehicle engineering and the resulting demands on power and design have led to innovations in electronic networks for automobiles [1]. Just as LANs connect computers, control area networks (CANs) connect a vehicle's electronic equipment and facilitate the sharing of information and resources among the distributed applications. A typical vehicle can contain several CANs, operating at different transmission rates, to manage a car's "comfort electronics" (like seat and window movement), or to run more real-time critical functions (like cruise control, antilock brakes and engine management). Other applications that use electronics to control a system (the so-called X-by-wire solutions), rather than mechanical or hydraulic means, are responsible for an ongoing revolution in vehicle electronics architecture and require more specialized and reliable control networks. Multimedia devices in automobiles and interconnecting facilities over the Internet demand networks with extensive bandwidth, while other applications require wireless configurations [2]. Vehicles are becoming more like PCs, allowing for a number of plug-and-play devices and creating the potential for significant growth in automotive application software [3]. With more than 85% of the functionality in the modern motor vehicle already controlled by software, both the motor vehicle manufacturer and the software supplier need to take action to face quality issues related to the management of software projects.

Suppliers have not been slow to take action. A number of companies are already using process assessment techniques as a basis to identify areas for improvement in their processes both to meet their business needs and the demands of their customers. There is a general understanding that action has to be taken if they are to continue to make business.

The motor vehicle manufacturers, as a general trend, have also started to take pro-active action to address the situation in a number of ways; by focusing on software capability of the supplier in the supplier evaluation process; making provision for contractual demands with respect to software quality; performing supplier software capability assessments both before and during contract performance; and asking for the supplier to implement process improvement plans, when needed.

To set up a methodology supporting the management of software projects and suppliers, Fiat Auto started the ESCAPE (Electronics Software Capability Evaluation) Project, in co-operation with the Centre for Software Certification (CCS - an independent organism of the Italian National Research Council that performs evaluation and certification activity in Information Technology), with the following main goals:

- To improve the software suppliers selection process.
- To provide Fiat Auto with methods to determine the risks associated to a software supplier.
- To improve the software development process of suppliers, helping them to detect possible weaknesses and risks in specific processes, to define improvement paths and to provide tools for verifying the results of improvement actions
- To achieve a better control on the software development project and on the quality of the resulting product.

2 The ESCAPE Project

The need to evaluate the Process Capability of the Suppliers of FIAT Auto was discussed between the CCS and FIAT Auto in year 2000. The main goals and reasons for undertaking this evaluation were:

- To derive a “capability” and “risk” level for each software supplier.
- To improve the supplier selection process of FIAT Auto by using criteria based on the derived supplier “capability” and “risks”.
- To improve the control of the supplier’s software development process and of the quality of the resulting products.
- To identify weaknesses and strengths of the supplier’s software process.
- To identify possible improvement opportunities in the relations between the customer and suppliers.

With the aim to achieve the above targets, the ESCAPE project was started by FIAT and, in the framework of this project, cooperation with CCS was established.

The first step of the ESCAPE project was to decide the reference model to use to perform the suppliers evaluation. The traditional reliance on Quality Systems Standards such as ISO9001, QS9000 [9] and ISO/TS 16949 [10] has not provided sufficient confidence in the software area. The motor vehicle manufacturers, like others in the defence and aerospace industries, have now turned to international standards for software process assessment, based on ISO 15504 [4] and/or the Capability Maturity Model (CMM), as a means to identify and control risk and to assess the software capability of suppliers [5], [6], [7]. Some claimed features of the ISO 15504 standard proved to be crucial in selecting the approach to perform the software capability assessments [8]:

- Software-oriented approach
- Applicability over a wide range of application domains, businesses and sizes of organizations
- Output as process profiles at different levels of detail
- Comparability, reliability and consistency of results

- Independence of organizational structures, life cycle models, technologies and development models
- Adaptability of the assessment scope to cover specific processes of interest
- Re-usability of assessment results, both for process improvement and capability determination

Another important factor that supported the decision to use SPICE as the assessment methodology is the launch of an initiative by the Procurement Forum (www.procurementforum.org) with the principal European Car Makers, their assessors and representative bodies to address the problems related to software assessments in automotive. In the framework of this initiative, a Special Interest Group has been founded with the aim to design a special version of the SPICE model (called Auto-SPICE) tailored on the needs and peculiarities of the automotive business area. In fact, the focus on software capability determination by means of software process assessment has already in use provided significant business benefits, but at the same time has highlighted the scale of the potential problem, particularly with suppliers of safety-critical embedded software system components. Whilst the immediate short-term benefits are clear for the motor vehicle manufacturer, in the near term, without consensus on commonality of approach, suppliers face multiple assessments from multiple manufacturers using different model and consuming resources that put additional pressure on delivery times. Therefore the choice of SPICE as the Reference Model to adopt for the Supplier's software process assessment has been corroborated by the existence of this European trend towards the use of the SPICE Model is in act in automotive.

FIAT Auto and the Centre for Software Certification are part of this Special Interest Group and are currently participating in the works.

2.1 Assessment purpose and scope

The assessment purpose is to evaluate the software process capability of suppliers in order to improve the general project management during the development phases. In order to achieve a trade-off between performing a wide and comprehensive assessment that should provide many indications on the way a supplier conducts its own software development process and the need to respect budget limitations, FIAT Auto identified some critical areas concerning software with the aim to concentrate the assessment effort on them. The principal critical areas that was been identified are listed below:

- Relationships between the customer (FIAT Auto) and suppliers
- Extent of Requirement Analysis (by the supplier's side)
- System design capability (where the system is intended as the ECU - Electronic Control Unit)
- Software design capability
- System integration and testing capability
- Project management capability

Consequently, the scope of the assessment has been defined matching the criticalities identified by FIAT Auto with the processes in the SPICE reference model. The final scope of the assessments is composed of five processes that in the following are indicated according to the SPICE terminology:

- CUS 3: Requirements Elicitation Process
- ENG 1.1: System Requirements Analysis and Design Process
- ENG 1.3: Software Design Process
- ENG 1.7: System Integration and Testing Process
- MAN 2: Project Management Process

3 Assessment Activities

The first step in the ESCAPE project was the selection of the suppliers to involve in the assessments and in each assessment the selection of the project (or process instances) to be considered in order to collect evidences of the process capabilities. The general policy followed gave priority to including those companies that are currently involved in new projects with FIAT Auto, and those projects with FIAT Auto that are enough advanced to provide sufficient evidence at assessment time.

The activities strictly related to the assessment were divided into four main phases:

- *Preliminary meeting*: an introductory meeting was held at Fiat Auto at the beginning of the operational phase with representatives from the companies involved in the assessments, with the purpose of presenting the SPICE approach, of reviewing the assessment purpose, agreeing the scope and discussing the constraints, introducing the assessment activities and a provisional assessment plan. At this stage, particular care was taken on informing the suppliers that process assessment does not disclose sensitive information about the techniques used in software development nor details on proprietary software and algorithms. In fact, the assessment method intends investigate only on knowledge, experience, skill, confidence, benefits, resources allocation and management.
- *Assessment preparation*: each assessee was sent a questionnaire - to fill and return before the on-site visit - to gather preliminary information on the processes. Furthermore, some documents describing the purpose and the topics to be investigated and the way the assessment should be conducted was sent too, to help the assessee to prepare for the assessment. These preliminary activities allow to save time during the on–site visit and to make of the assessment more effective and efficient. At this stage, a non-disclosure agreement was signed by the assessment team members.
- *On-site activities*: during the on-site visit (that was taken about 3-4 working days), and initial briefing was held aiming at recalling the assessment purpose, scope, constraints and model. Then, information was gathered by means of presentations, document analysis and interviews. To better assure assessment repeatability and results comparability, checklists were developed to be used as guidelines for the assessors.
- *Results derivation*: after the on-site visit the gathered data was validated and analysed and, for each assessed process, each process attribute was rated. Then, these ratings were used to derive the capability profiles and the capability levels of the assessed processes. Finally, a detailed report of the whole assessment was prepared including the detailed ratings and the final capability profiles. These results, along with the indication of improvement opportunities were sent to the company and to the sponsor (FIAT Auto).

4 Results

In this section we present the results in terms of Capability Profiles of the processes assessed. For confidentiality reasons the name of the Companies where the assessments have been conducted has been omitted and the projects have been indicated as Pi (with i = 1, … , 10).

The figure 1 shows the detailed capability profiles of all the processes assessed. The capability profile is the collection of the rating achieved by each process attribute of the Spice Model. According to the SPICE terminology and rules, an attribute is rated in a four-value scale (N=Not Achieved, P = Partially Achieved, L = Largely Achieved, F = Fully achieved). Furthermore, the bold red line determines the achieved capability level of the correspondent process.

		P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	Capabilty Level
CUS 3	PA 4.2	N	N	N	N	N		N	N	N	N	**4**
	PA 4.1	N	N	N	N	N		N	N	N	N	
	PA 3.2	P	P	P	P	L		L	P	L	L	**3**
	PA 3.1	P	P	L	P	P		P	P	P	P	
	PA 2.2	L	L	P	P	L		F	L	F	L	**2**
	PA 2.1	L	L	L	L	L		F	F	F	L	
	PA 1.1	F	F	L	F	F		F	F	F	F	**1**
ENG1.1	PA 4.2	N	N	P	N	N	L		N	N	N	**4**
	PA 4.1	N	N	P	N	P	P		N	N	N	
	PA 3.2	L	L	L	L	L	L		P	L	L	**3**
	PA 3.1	P	P	L	P	P	L		N	P	P	
	PA 2.2	F	F	L	L	L	F		P	F	F	**2**
	PA 2.1	L	L	L	F	F	F		F	F	F	
	PA 1.1	F	F	F	F	F	F		L	F	F	**1**
ENG1.3	PA 4.2	N	N	P	N	P	L	L	N		N	**4**
	PA 4.1	N	N	L	N	P	P	N	P		N	
	PA 3.2	L	L	L	L	L	F	L	F		L	**3**
	PA 3.1	P	P	L	L	L	L	L	L		P	
	PA 2.2	L	F	F	F	L	F	F	F		F	**2**
	PA 2.1	L	L	F	F	F	F	F	F		F	
	PA 1.1	F	F	F	F	F	F	F	F		F	**1**
ENG1.7	PA 4.2	N	N	P	N	P	L	N	N		N	**4**
	PA 4.1	P	P	P	N	P	P	P	P		N	
	PA 3.2	F	L	L	L	L	F	L	L		L	**3**
	PA 3.1	L	P	L	P	L	F	P	P		P	
	PA 2.2	L	L	L	L	L	F	F	F		F	**2**
	PA 2.1	L	L	F	L	F	F	F	F		L	
	PA 1.1	F	F	F	F	F	F	F	F		F	**1**
MAN2	PA 4.2	N	N	N	N	N		L	N	N	N	**4**
	PA 4.1	N	N	N	N	N		P	P	N	P	
	PA 3.2	P	P	P	P	P		L	L	P	L	**3**
	PA 3.1	P	P	P	L	P		F	L	L	L	
	PA 2.2	P	P	P	P	L		L	F	F	F	**2**
	PA 2.1	P	P	L	L	P		F	L	F	F	
	PA 1.1	L	L	F	F	F		F	F	F	F	**1**

Figure 1. The complete capability profile of the process assessed

Each grey column means that the correspondent process was, for some reason, excluded from the scope for that project.

From the data above some general indications may be derived on the process capabilities of the automotive software suppliers belonging to the sample we have considered. In fact, as the Figure 2 shows, the average ratings of the assessed processes provide some high level indications.

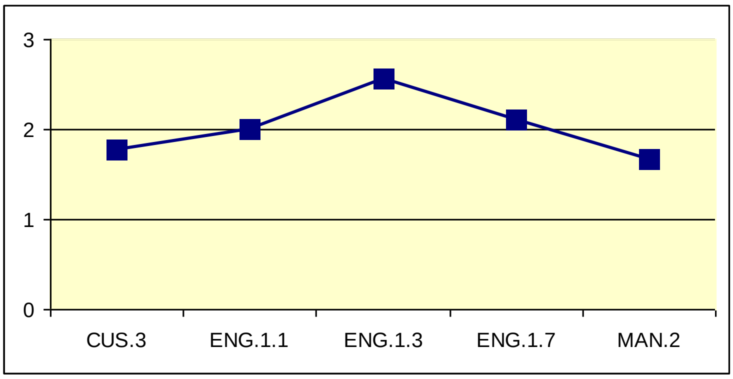


Figure 2. Average ratings

5 General Considerations

The average capability level achieved by the technical processes (ENG.1.1, ENG.1.3 and ENG.1.7) is higher than for the other processes. From this outcome it is possible to infer that the weaker areas are not related to the capability to perform the technical tasks (see the capability level of the engineering processes) but are related mainly to the managerial issues and the relations with the customer.

From the assessments results, some general issues emerge. The principal issues will be discussed below together with their consequences and the related improvement opportunities. These issues identify also research directions to provide answers to real needs of the automotive industry

5.1 Facing the New Challenges in the Automotive Software

A few years ago the automotive development environment was still almost totally oriented to the system (intended as mechanics and electro-technical issues). In the last few years electronics and software have pervaded the automobiles and the automotive companies have to face this new challenge. The answer to the demand for an extensive use of electronic and software solutions is still inadequate, as it comes from an environment (including both customers and suppliers) that is not enough prepared for the transition - for historical, cultural and technical reasons.

Consequences:

- The companies (at the top management level) are aware of these problems and are introducing (or have already introduced) new development process models with considerably high costs.
- Several problems usually arise (consistently with the low capability level achieved by the CUS.3 process) about the mutual and unique understanding and agreement of the requirements.
- The customer, which has the most difficult task (since it has to maintain a comprehensive point of view on the whole automobile product), is not completely aware of its role in the acquisition process and of the importance of its co-operation in it.
- A similar situation may be found also within the supplier’s organization between the departments dealing with the system and those dealing only with software components. This is most true in those companies that produce “traditional” components with embedded software.

Improvement opportunities:

- To define reliable, available and competent interfaces between customer and supplier in order to facilitate their relations.
- To increase the effort for identifying and managing the risks related to interpretation problems and poor involvement of one of the partners in the acquisition process.
- To encourage initiatives among carmakers and software suppliers aimed at the definition of common schemes and standards (as for example Auto-SPICE).

5.2 Platforms vs. Ad Hoc Solutions

Automotive software is embedded in subsystems delivered in very large numbers (by the millions), with hard constraints in terms of resource optimization (e.g. memory size, computing power). Furthermore, the automotive software suppliers deliver their products to almost all the European carmakers, but they have to differentiate quality and performance characteristics of the products, to meet quality and cost requirements of the different carmakers.

Consequences:

- The supplier tries to impose its own platforms, i.e. products based on the same architecture but with adaptable components. This kind of solution allows the supplier to save many resources.
- The customer tries to get ad hoc designed software with the aim to maintain the full ownership of both the project and the marketing policies and reduce the dependence from a particular supplier.
- Code minimization techniques have been taken up again in order to save memory occupation.

Improvement opportunities:

- Both the customer and the supplier are scarcely aware of the risks related to the solutions adopted to cope with this problem (both the ad hoc and platform solutions may present significant drawbacks in the maintenance and upgrading phase). These risks have to be clearly understood and a trade-off between efficiency and resource utilization has to be achieved and managed.

5.3 The Modularity-related Issues

The electronic final system is characterized by a strong modularity since it is composed of integrated subsystems (typically ECUs)

Consequences:

- Serious interoperability problems may arise (typically a stand-alone ECU runs correctly but behaves unpredictably when integrated with the others).

Improvement opportunities:

- To adopt techniques for the requirements and interfaces specification and analysis in order to be able to better evaluate and manage their completeness.

6 Conclusions and Next Steps

This paper describes an initiative (the ESCAPE project) on software process assessment in automotive sponsored by FIAT Auto and performed by the Centre for Software Certification of the Italian National Council of Researches. So far, the performed assessments on a selected group of FIAT Auto’s suppliers has been oriented to capability determination (in order to provide criteria for selecting suppliers by means of a comparable set of process capability profiles).

The results of this experience have pointed out some general considerations about the development and acquisition processes in automotive. These results provide useful indications both for the customer - FIAT Auto - (to better understand the capability of its suppliers and to include this information in the supplier’s selection criteria) and for the suppliers (to identify improvement opportunities) and for CCS (to identify research directions for contributing to the improvement of the development process in the software industry). In particular, a demand for new techniques for specifying and analysing requirements aimed at avoiding or reducing incongruity, ambiguity and incompleteness emerges

Next steps of the ESCAPE project will be the completion of the first round of assessments by performing the assessments of a new group of suppliers. Then, further rounds of delta-assessments will be planned and performed to verify that the identified improvement opportunities have been implemented.

7 Literature

[1] Leen G., Hefferman D., Dunne A., “Digital Networks in the Automotive Vehicle”, IEE Computer and Control Eng. Journal, Dec. 1999, pp. 257-266.

[2] Kassakian J. G., “Automotive Electrical Systems: The Power Electronics Market of the Future”, Proc. Applied Power Electronics Conference (APEC2000), IEEE Press, 2000, pp. 3-9.

[3] “Development Guidelines For Vehicle based Software”, The Motor Industry Software Reliability Association, 1994. Published by MIRA. ISBN 0952415607.

[4] ISO/IEC TR2 15504. “Information Technology – Software Process Assessment: Part 1–Part 9”, ISO, Geneva, Switzerland, 1998.

[5] Paulk M., “Top-Level Standards Map: ISO 12207, ISO 15504 (Jan 1998 TR) Software CMM v1.1 and v2”, Draft C (available at http://www.sei.cmu.edu/pub/cmm/Misc/standards-map.pdf), Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, 1998.

[6] Hailey V., “A comparison of ISO 9001 and the SPICE framework”. In “*SPICE: The Theory and Practice of Software Process Improvement and Capability Determination*”, El-Emam K, Drouin JN, Melo W (eds). IEEE CS Press, 1998.

[7] Jung H-W, Hunter R., “The relationship between ISO/IEC 15504 process capability levels, ISO 9001 certification and organization size: an empirical study”, *Journal of Systems and Software, n.* 59, 2001, pp. 43–55.

[8] El-Emam K., Jung H-W, “An evaluation of the ISO/IEC 15504 assessment model”, Journal of Systems and Software, n. 49, 2001, pp. 23-41.

[9] Quality System Requirements (QS-9000 Third Edition) Version 03.00, DaimlerChrysler, Ford Motor Company and General Motors Quality Publications, March 1998.

[10] ISO/TS 16949- Quality Management Systems - Automotive Suppliers - Particular Requirements for the Application of ISO 9001:2000 for Automotive Production and Relevant Service Part Organizations, 2002.

8 Author CVs

Fabrizio Fabbrini

He obtained his degree in Computer Science from the University of Pisa, Italy, in 1974. Since 1975 he has served as a scientific researcher at the Institute for Information Processing (IEI) of the Italian National Research Council (CNR), where now he is Senior Researcher and coordinates the Software Laboratory of the Centre for Software Certification. Fabrizio Fabbrini’s present activity is focused on Software Quality, and more precisely on the development of methodologies and standards for the assessment and the evaluation of software products and processes, with particular attention to Software Engineering Standards and Software Certification. Software Process Assessment & Improvement, Software Verification & Validation, Computer Security & Data Privacy represent the main fields of application of such research activities.

Mario Fusani

He obtained his degree in Electrical engineering from the University of Pisa, Italy, in 1971. Since 1973 he has served as a scientific researcher at the Institute for Information Processing (IEI) of the National Research Council (CNR), where now he is Senior Research. His present activity is focused on Software Quality, including the development of methodologies and standards for the assessment and the evaluation of software products and processes. Since 1999 he has been the Scientific Coordinator of the Centre for Software Certification of the Italian National Research Council.

Giuseppe Lami

He obtained his degree in Computer Science in 1994 from the University of Pisa, Italy. He participated in the “Leonardo da Vinci – Software Quality Evaluator” European project (1997-1999) and in the “Software Product Evaluation and Certification” project founded by the European Space Agency (1998-1999). Since year 2000, he has been working at the Istituto di Elaborazione dell’Informazione (Institute for Information Processing) of the Italian National Council for Research. His research interests are: software process assessment (the SPICE ISO15504 model), Requirements Engineering (natural language requirements quality evaluation), Software Testing (coverage methods). Since 1998 has given lectures in Fundamentals of Computer Science at the Faculty of Telecommunication Engineering of the University of Pisa. Since 2001 he has been a PhD student in Informatics Engineering at the University of Pisa.

Edoardo Sivera

Electronic Engineer (1990 – Politecnico di Torino). He worked (1991-1994) at the Fiat Research Centre analysing and using tools based on formal models for the definition of automotive functional requirements. Since 1994 he has worked in Fiat Auto, collaborating to the development of the architecture of integrated electronic systems based on serial networks (CAN protocol). Since 1998, he has worked on the definition of standards for the communication on serial networks (based on CAN protocol), used in all Fiat recent vehicles. Since 2000, he has been the responsible of the Fiat Auto “Standard and Methodology Software” group. The main objectives are: to manage the development of new equipments containing software; to define a software architecture for these electronic equipments; to develop and use methodologies to analyse and validate the automotive functional requirements.

Practical implementation of risk-based test

Kai Ormstrup Jensen
DELTA, Denmark
Karsten Bank Petersen
WM-data, Denmark
Ebba Ehlers
UNI•C, Denmark

Abstract

This paper describes how to manage the acceptance test of a 10000 Function Points application, delivered from a subcontractor. The test strategy used is based on risk assessment, where the test effort is focused on finding and removing the most risky errors seen from a business point of view. We illustrate how this situation was managed and introduced into an organisation unfamiliar with risk-based test.

An administrative software system for planning, monitoring and reporting of resources used for National Institutes for Social Educators was ordered by the Danish Ministry of Education. Project management of the system development was performed by UNI•C, an IT organisation under the Danish Ministry of Education. The system was developed by a subcontractor, WM-data. A test team was set up to handle the communication with the subcontractor and perform the acceptance test using representatives from the end-users organisation. DELTA was assigned to the test team to take care of the Quality Assurance activities and to support the organisation with information about how to plan, monitor and evaluate the test process.

Keywords

Risk Based Testing, Test Planning and Management, Test Monitoring, Test Metrics, SPI, Collaborating, Customer Satisfaction.

1 Project type

This project was a fixed price project where WM-data developed an administrative software system for planning, monitoring and reporting of resources used by app. 400 users at the 32 National Institutes for Social Educators in Denmark. The development of this application was based on an application for a similar business area developed earlier by WM-data for the Danish Ministry of Education.

The development of the application was split into 5 deliverables. Each deliverable introduced new functionality into specific business areas. Some of the new functionality was based on reuse from the previously developed system, while other functionality was developed specifically to this application.

The first 4 deliverables were only used for test purposes. They were received from the subcontractor and installed in an environment similar to the one on the institutes. These deliverables were never installed in the institutes. Only the fifth and final deliverable was installed in the institutes. The relative size of each deliverable can be seen in figure 1.

The application was developed using the Oracle Developer tool with PL/SQL as the primary program-

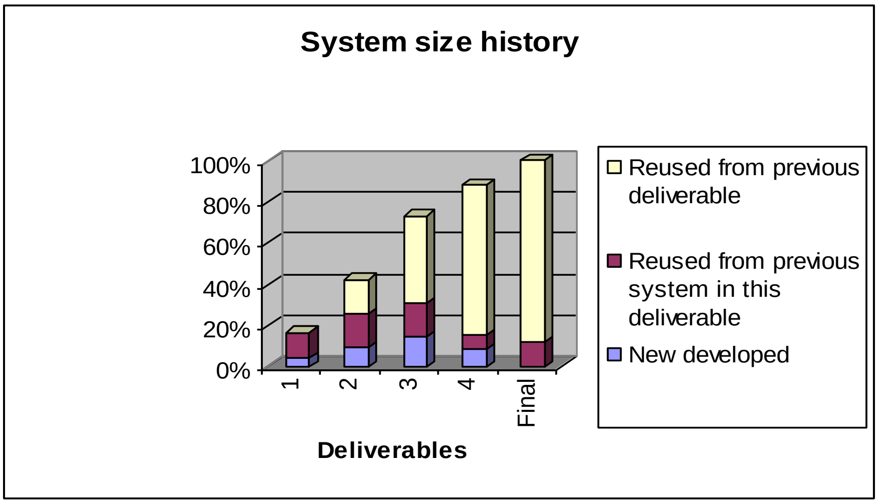


Figure 1 - The historical development in system size

ming language. A repository was used for development and maintenance of database tables and triggers.

The final delivery comprised 578 components: 314 screen forms, 189 reports, and 75 batch jobs.

2 Test of the first deliverable

A test core team that was allocated 100% to handle communication with the sub-contractor and plan the acceptance test was set up. They were assisted during test performance by end-users that had been involved in the specification and design activities and therefore had good knowledge about the application. The testers, especially the ones from the end-users organisation, were unfamiliar with test and this delivery was their first real-life experience with test of a large system. Besides testing the functionality this deliverable was used to evaluate the test approach, test organisation and the test tools.

The first deliverable which mainly contained reuse stuff from the original application was seen as the “easy” part of the total application. The result after 72 hours of testing was that 213 problems were identified. The number of generated problem reports was influenced by the weak test experience of the testers, and the fact that the application was new in this application area. This indicated high risk for a large schedule overrun and that the system would not fulfil its purpose.

3 Rising awareness that a new test strategy is required

The high number of problem reports generated in the first deliverable made it clear that a new test strategy was needed as it would be impossible, with the available resources, to reach an acceptable quality by continuing testing in the way used during the first deliverable.

Apart from the lack of quality in the first deliverable, two other items were to be supported by the new test strategy:

- The test resources were limited, and mainly drawn from the end-user organisation.
- Dissatisfaction with a number of other software systems delivered to other public organisations had generated a very high degree of attention on the quality of software systems in general.

Test of the second deliverable was performed at the time when the new test strategy was considered. It was tested using the same testers as in the first deliverable, they now had more knowledge about testing and this made them focus more on real problems, than during deliverable 1. The subcontractor also had improved their development and test processes, based on the experiences from the first deliverable.

The largest factor in the improvement was that both customers and supplier agreed to increase the test and was willing to go new ways. This improved the quality of the second deliverable dramatically. The test organisation was changed between deliverable 2 and 3. The original test organisation was a part of UNI•C. Acceptance test of the remaining deliverables was still handled by UNI•C and was still involving the end-users but also more experienced testers were assigned to the project.

4 The new and improved test strategy

Based on the experiences from the first deliverable it became clear that customer satisfaction would be the key to a successful project.

Customer satisfaction with a new software system is based on these success criteria:

- Confidence that the system correctly supports important functionality in accordance with expectations.
- Confidence that the system can handle the daily operations effectively and efficiently.
- That it is obvious to the user when the system fails, and that valuable data is not corrupted if an error occurs.

Based on these considerations a risk based test strategy was selected. The strategy allowed the test team to focus the test on identifying problems, which would most seriously harm the success criteria for the application.

5 Implementation of the new test strategy

Implementing a risk-based test strategy requires answers to the following two questions:

- Where are the risky areas, seen from a business point of wiev?
- How is it ensured that the application areas with the most annoyingly bugs are found?

The first question was answered during a risk workshop as described later. The second was handled during the user acceptance test, by evaluating the quality of the delivered software, using the results from the module and system test. Furthermore, monitoring the problem report density during user acceptance test as described later supports this.

5.1 The Risk Worksop

To identify the critical work processes that required special attention in the acceptance test, a 2 days workshop was arranged with 9 representatives from the institutes that were going to use the system daily. These representatives included end-users that were going to use the application daily and leaders of the relevant business areas of the institutes. Furthermore, the 6 persons in the core acceptance test team participated in this workshop.

The overall goal of this workshop was to ensure an alignment of the end-users' and testers' expectations to the system, in order to make sure that the right things were tested and to ensure that the institutes had the right expectations to the system when receiving it.

Involving the end-users allowed them to transfer knowledge to the core test team about how the system would be used and about the consequences of failures of the system. They were also able to influence the depth of test inside each business area.

The way the workshop was structured was inspired by a concept for making effective decisions based on quality and acceptance [1]. This concept contains 4 phases:

- What do we know?
- What can we do?
- What do we want to do?
- What do we do!

The content of the individual phases was structured and handled as described by Barry Boehm in "Software Risk Management" [2].

The tailoring to risk mitigation based on test activities gave the following 4 phases in the workshop:

- What risks do we see in this project?
- What can we do to handle these risks?
- What will we do? – Which (test)activities will we do inside each area?
- The common decision about how to prioritise the test effort inside each area.

The first 3 phases of these phases were handled in 4 groups, where each group contained testers as well as end-users. Between each phase the groups met in plenum and discussed their results before they returned to the groups to work with the next phase. The fourth phase was handled in plenum. The result was an agreement on the priority of the 14 most critical business items.

5.2 The practical implementation of the test strategy

The following parameters had to be taken into consideration during test:

- The quality of the software received from the subcontractor.
- The prioritisation of risky areas as identified at the risk workshop.
- The actual results of the performed tests.

The documentation describing the results of the performed test by the subcontractor was used to adjust prioritisation of the test based on the agreed prioritisation from the risk workshop. The supplied documentation included metrics for hours of testing and numbers of bugs found and fixed during module and system test inside each application area.

The practical test of each deliverable started with a smoke test to make sure that no parts of the delivered software contained serious problems. The smoke test was performed across all components of the total received deliverable. After this test, the additional test focused on the agreed business risk areas, adjusted by experiences from the test executed by the subcontractor.

The results of the testing activities were continuously monitored and the test activities adjusted to get more information about the most risky business areas. This was based on the agreed priority and the defect rate for the individual areas of the application, as well as the problem report rate inside each risk area. The strategy used is illustrated in figure 2:

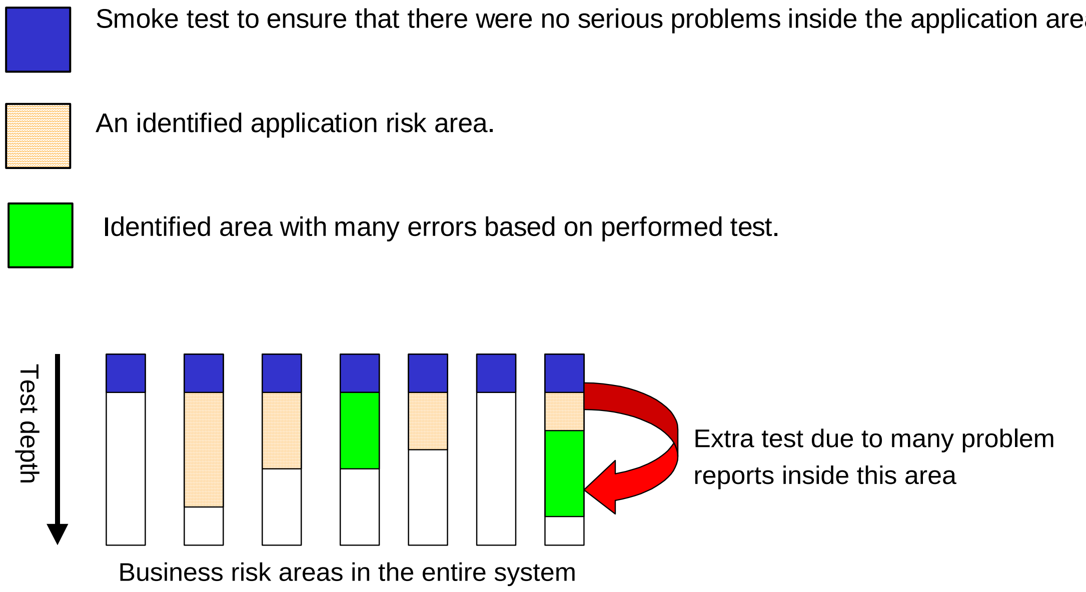


Figure 2 - The implemented test strategy

The knowledge about testing metrics were continuously increased and more metrics were added to get more information about the quality of the tested software from deliverable to deliverable. Monitoring of the used resources was evaluated once a week and a report was produced by the QA responsible based on the reported figures on planned and performed test inside each functionality area as well as the number of problem reports in each of these areas. The problem reports were also directly related to the identified business risk areas.

6 Improvement of subcontractor test

The quality of the first deliverable was not satisfactory. This fact was clearly indicated by the test results. The subcontractor had to increase the quality of the subsequent deliverables. The subcontractor initiated a range of improvements.

The focus on test activities was increased significantly:

- Already established test procedures were followed more closely.
- A sufficient amount of resources were allocated to the test activities.
- The progress of test activities was monitored carefully.

In a few cases, the deliverance of program components with poor quality was postponed to the subsequent deliverable.

Furthermore, the quality of new and changed program components being tested was enhanced. This was mainly due to project group members improving their qualifications and experience with respect to:

- Knowledge of the business area of the customer.
- Knowledge of the previously developed system.
- Knowledge of the development tools.

Each of these improvements did not impose radical changes to the subcontractor project. It was rather a matter of improving processes throughout the remaining project. As a result, the quality of the development and test processes was gradually raised to an appropriate level, which is indicated by figure 3, showing error rates for subcontractor module and system tests for each deliverable. One of the shortcomings of the development of deliverable 1 was that error rates were not measured.

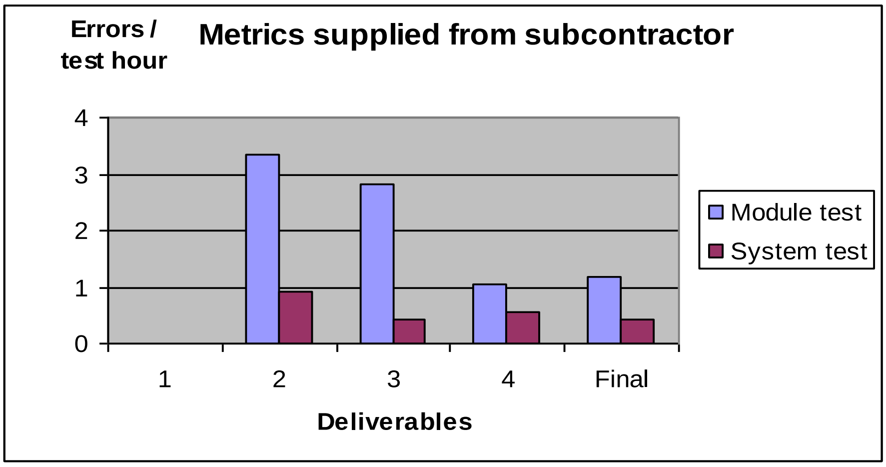


Figure 3 : Results of the Module and System test executed by the Subcontractor

7 *The overall results of the acceptance test*

During the acceptance test the number of problem reports as well as the number of hours used to test the delivery was monitored. The overall test results are shown in figure 4 below.

It can be seen that there is a large decrease in the ratio of problem reports per test hours from deliverable 1 to deliverable 2. This decrease is directly related to an increase in the quality supplied by the subcontractor and more experience about the application by the testers. The fact that the application now is more “complete” and can be tested by complete business cases in the way they occur at the institutes decreased the possibility that testers make errors, this may also have influenced the test result positively. After deliverable 2 the number of problem reports stabilises at a level around 0,5 problems report per test hour. The slightly higher number of problem reports per test hour found in deliverable 2 compared to the other deliverables has two causes. Firstly, it included new functionality for planning the usage of resources like teachers and rooms etc. in the institutes. This functionality had not been available as an integrated part of the previous system, and because of this the individual institutions had used different systems to handle these planning functions. The specification of the planning module had been difficult because it is a very new and complex area to integrate with the remaining administrative functions and there had been changes in the group involved in the testing of this functionality and consequently, many changes occurred to the functionality during the test session. Secondly, this deliverable also included several new interfaces to other parts of the system, which contributed to the increased number of problem reports.

Deliverable	No. of problem reports	Test hours	Problem reports / test hours
1	213	72	3,0
2	334	361	0,9
3	224	441	0,5
4	294	436	0,7
Final	197	409	0,5
Sum	1262	1719	0,7

Figure 4 : The results of the acceptance test

8 Test results related to agreed business risks

As described earlier, the test strategy was based on the identified and agreed risk areas that were identified and prioritised at the risk workshop as described above. During test planning the identified risks provided the background for selection of the tests inside each functionality area. The tests were described and performed in a way that supported the agreed business risks. The testers received detailed test cases related to a specific risk area but they were not informed what risk they were testing for. When they had finished these test cases, they were requested to test the application using their own application knowledge inside the business area under test.

The module and system test performed by the subcontractor should test each part of the application to the same coverage level. Because of this the business risk areas were never communicated to the subcontractor, to make it sure that the module and system test not was focused on business risks.

The overall result of the acceptance test was that 80% of the problem reports were related to a risk area. Because the risk areas are related to business areas, one problem report can have relations to more than one risk area.

9 Experiences from the first year in operation

In February 2001, the system was brought into pilot production and daily use in 3 of the 32 institutes. An unusually detailed training of the end-users supported the introduction of the system. The need for training is well known in this environment, but the result from the acceptance test and the fact that 34% of the problem reports, in the final deliverable, were related to usability of the system, made it very clear that detailed training was needed. In September 2001 all the 32 institutes used the new system on a daily basis. The response from the customers was that they were satisfied with the system. More than 50% of the money reserved for support from the subcontractor was not used. The first full-scale use of the system occurred in October 2001, where all the institutes had to report financial data to the Ministry of Education. Only 5 of the 32 institutes used other ways to handle their reporting. Their reason for not using the new system was internal and <u>not</u> related to the quality or functionality of the new system.

9.1 Defect Detection Percentage

During the 15 months of operation, 3 releases of the system were launched. These releases were all planned and included minor additions of new functionality. No serious failures occurred and the number of problem reports was very low. The number of defects found in the 3 versions was very low in relation to the number of defects found before release. The defects detect percentage, defined as defects found during testing divided by the total number of defects including the ones found afterwards, are shown in figure 5 below.

Release	Date	No. of Installed locations	No. of problem reports before release	Total test resources before release - hours	No. of problem reports after release	DDP - Defect Detect Percentage	Time period for DDP - months
1.0	January 2001	4	1262	1719	13	99%	3
1.1	May 2001 App size inc. 6,5 %	32	1367	1845	101	93%	6
1.2	November 2001 App size inc. 3,1 %	32	1672	2083	124	93%	6

Figure 5 - Defect Detect Percentage (DDP)

Please observe that defect here is defined as a problem report. Problem reports include faults as well as proposals for new functionality in the application. Proposal for new functionality was evaluated regarding the total budget for the project and the business need for it, before the relevant ones were included as problem reports.

The time period for measurement of the **D**efect **D**etect **P**ercentage is short, only 3 to 6 months, but when comparing the measured results to other similar results, the result is remarkable. E.g. Casper Jones [3] reports that companies with little or no control over the testing process tend to have a **D**efect **D**etect **P**ercentage of only 53%-63%. Jones also reports that simply introducing well-known testing stages can increase a team’s defect-removal efficiency to 74%-87%. The figures reported by Casper Jones are based on a 12 months period.

On this type of application it is not possible to get data for a longer period than 6 months, because the whole application is updated on all institutes when a new release is available, and that happens approximately each 6th month.

When looking on two periods of 6 months (release 1.1. and 1.2) the **D**efects **D**etect **P**ercentage is twice calculated to 93 %. 126 hours were used to test the first release that included 6,5 % of new functionality. The second release, included 3,1 % of new functionality, was tested for 238 hours before release. These test hours were added to the previous number of test hours to get the total number of test hours before release. This principle has been used to calculate the number of problem reports and test hours in all deliverables and releases.

10 Lessons learned

The lessons learned during this complete acceptance test of a full-scale real-life project is related to introduction of acceptance test in an organisation unfamiliar with risk based testing, as well as how to monitor and communicate test activities in a way that can be understood by people without test knowledge. An other important lesson learned is how to create alignment of expectations between end-users and testers.

The most important lessons learned are listed here:

- The first difficult part was to realise and accept that a full-scale test was not possible and then take the consequences of it.
- The risk workshop ensured a proper alignment between end-users and testers' understanding of the system.
- Risk-based test was a successful strategy for this project.
- Risk-based test required continuous evaluation of the test results to be successful.
- Metrics are crucial for the evaluation of the software quality.
- The number of defects found in use is very low.

The monitoring of the customer satisfaction with the project by monitoring the number of problem reports for more than a year after the first release has been a good reminder to the project team as well as a good way to communicate this successful project to management. This paper and the presentation of it also is an important part of creating awareness of the good results created in this project.

11 Literature

[1] "Creativity and Innovation Yearbook" Nils Bech, 1991, ISBN 90-6743 222 9.

[2] "Software Risk Management", Barry W. Boehm, IEEE Computer Society Press, 1993, ISBN 0-8186-8906-4

[3] Jones C. 1996. Quantifying Software Process Improvements. Proceedings at SP '96

[4] SPICE **S**oftware **P**rocess **I**mprovement **C**apability d**E**termantation -World Wide Web: www.spiceworld.hm/www/SPICE.WORLD.html

[5] Bootstrap - World Wide Web: www.bootstrap-institute.com

[6] EFQM European Foundation for Quality Management - World Wide Web: www.qpronline.com/EFQM/index.html

[7] DTF The Danish Software Technology Forum - World Wide Web: www.datatekniskforum.dk

12 Author and company CVs

Ebba Ehlers - UNI•C

She has a M.Sc. in Food Science from The Royal Veterinary and Agricultural University, Copenhagen. She worked in the area of study administration in higher education's from 1979-1996. From 1996 to 2000 she has been working at the VUE-project on a study administrative software system aimed at universities in Denmark for their planning, monitoring and reporting - from 1997 as the project manager.
Since 2001 she has been working in UNI•C as a project manager for a similar administrative software system for National Institutes for Social Educators. As project manager she is responsible for software functionality and software testing.
She can be contacted at UNI•C, Vermundsgade 5, DK-2100 Copenhagen Ø, Phone +45 35 87 88 89, Fax +45 35 87 88 90,
E-mail: ebba.ehlers@uni-c.dk, World Wide Web: www.uni-c.dk

Karsten Bank Petersen - WM-data

He has a M.Sc. from University of Aarhus, Department of Computer Science.
He has worked for WM-data since 1983 mainly in the areas of software development, project management, quality management, and process improvement.
For the last 10 years he has been quality manager of the Project Center of WM-data in Denmark. The Project Center develops, delivers, and maintains administrative IT systems to customers' specifications. The projects are often large, and mostly based on fixed price contracts.
As a quality manager, he works with project quality, project evaluations, development models and methods, software process improvement, tender reviews, etc.
He can be contacted at WM-data, Hermodsvej 22, DK-8230 Åbyhøj, Phone +45 44 78 40 00, Fax +45 87 44 44 87, E-mail: kapet@wmdata.com, World Wide Web: www.wmdata.com

Kai Ormstrup Jensen - DELTA

He has a B.Sc. E.E from the Engineering College of Aarhus. He is an examined assessor in SPICE 3.0 [4], Bootstrap 3.0 [5] and EFQM [6]. He also has fulfilled the requirements of the examination in Software Testing developed by The British Computer Society and certified by the Information Systems Examinations Board. In 1999 he wrote the DTF report “Automation of Software Test” [7].
Previously he worked for Brüel & Kjær (1982-98) on different projects using integrated product development. The projects he participated in were embedded projects and in these projects he worked primarily with software and digital hardware. In 1986 he became software project manager with responsibility for the software functionality and interfaces to other parts of the product. From 1995-98 he participated in two ESSI PIE’s. One project concerning reduction of software errors using experience-driven test efforts. The other project was about reduction of errors introduced in the requirement phase of the project.
In 1998 he came to DELTA where he is currently working as a consultant in the area of software testing and test improvement. He also participates in software and hardware process assessments, process improvement, effective and efficient product development etc.
He can be contacted at DELTA Danish Electronics, Light & Acoustics, Venlighedsvej 4, DK 2970 Hørsholm, Denmark, Phone +45 72 19 40 00, Fax +45 72 19 40 01, E-mail: koj@delta.dk, World Wide Web: www.delta.dk

UNI•C

UNI•C is a government institution under the Danish Ministry of Education with a staff of approximately 260 employees with offices in Copenhagen, Lyngby and Aarhus. Last year UNI•C's annual turnover amounted to DKK 272 millions.
UNI•C develops and provides knowledge about IT in the entire educational sector and collaborates with research communities in Denmark and abroad. UNI•C provides ground-breaking IT solutions for the Danish business community as well. The majority of the staff is computer scientists or engineers, but there is also staff with educational or other scientific backgrounds. UNI•C's staff is active in the national as well as the international arena.
UNI•C is divided into two major units: Education, and Business and Research Services. Education specialises in all aspects of IT integration: connecting schools to the Internet and creating educational material, web sites, and IT tools. UNI•C makes training programs for teachers in the use of IT in education and provides technical as well as educational support. The most essential specialist areas for Business and research services are infrastructure, IT security, statistics and analysis, high performance computing and e-learning.
Administrative systems is a unit in UNI•C Education responsible for development of administrative systems for vocational schools, national institutes for social educators and agricultural colleges. The systems merge a long series of administrative functions so that the different institutions need only one single system in order to get a full overview of information on finances, working hours, student data, course participant data, examinations and educational courses.
UNI•C Administrative Systems is located at: Vermundsgade 5, DK 2100 Copenhagen Ø, Denmark, Phone +45 35 87 88 89, Fax + 45 35 87 87 90, World Wide Web: www.uni-c.dk

WM-data

WM-data is one of the Nordic area's leading IT companies. Group sales for 2001 amounted to SEK 11975 millions and the number of employees is approximately 7000.
Our business concept is to provide "customer benefit through a complete range of IT-related services". By "complete" we mean that we offer solutions in all areas of information technology. In the concept of "IT-related services" we wish to emphasise the importance we place on expertise and service in developing successful solutions based on the application of information technology.
Behind us we have more than 30 successful years marked by involvement in a wide range of assignments in both public and private sectors. We know how companies and organisations work and what today's IT possibilities are. We can combine this knowledge with our insight into the various information systems and applications to provide the customer with a solution which both supports his business and assists its development.
We either revitalise the customer's existing systems, introduce and customise standard systems or develop an entirely new solution.
Our solutions naturally embrace software, hardware, computers, data links and telecommunications as important components of the finished product. But only people can apply knowledge, provide service and take responsibility. Only people can create the final result.
That is why we foster close, long-lasting customer relations. Our aim is to ensure that the customer is satisfied and that our own personnel are motivated through their feeling of involvement with the customer's own development.
WM-data has app. 100 locations in Sweden, Denmark, Norway, and Finland. World Wide Web: www.wmdata.com

DELTA

DELTA Danish Electronics, Light & Acoustics is an independent self-governing knowledge centre for advanced technology, affiliated to the Danish Academy of Technical Sciences (ATV). DELTA employs approx. 280 people in 5 departments (Electronics Testing, Software Engineering, Microelectronics, Light & Optics, and Acoustics & Vibration) providing services for every aspect of product development. The turnover is EUR 28 millions. DELTA's objectives as an ATV institute is to transform knowledge and experiences from both industry and universities to products and services that can be provided to the industry e.g. as consultancy support. DELTA Software Engineering is a division within DELTA with a Software Technology department. DELTA has a long-standing commitment to software engineering in general and specifically in the area of software best practice.
DELTA Software Technology employs 10+ consultants, all with a high level of experience within software engineering and software test directly involved in software best practices, offering consultancy to companies or teaching courses via our extensive course department (more than 80 courses a year). In addition to this DELTA has a responsibility for disseminating knowledge within the field of software engineering. The Danish SPIN group is an experience exchange group within Software Technology Forum, which has existed for 20 years with more than 70 members from Danish software industry. The Danish SPIN group meets 3 – 6 times per year and is chaired by DELTA as are the other 9 experience exchange groups within Software Technology Forum.
DELTA Danish Electronics, Light & Acoustics, is located in a number of locations in Denmark, Norway, Sweden, and England, the headquarter is located at: Venlighedsvej 4, DK 2970 Hørsholm, Denmark, Phone +45 72 19 40 00, Fax +45 72 19 40 01 World Wide Web: www.delta.dk

The Role of Scope Manager as Link Between Engineering and Management Processes

Pekka Forselius, Risto Nevalainen, STTF Finland

Abstract

The paper will be distributed at the conference and later included in the on-line EuroSPI library.

Keywords

SCOPE, Measurement, Processes

Adopting Statistical Process Control For The Software Process: An Experiential Learning

Rajesh Gupta
Sayam Computer Services Limited (India)

Abstract

More and more high maturity organisations are adopting statistical process control methods to improve the capability of their processes. At Satyam Computer Services Ltd., India – a SW-CMM Level 5 organisation, the drive to adopt SPC techniques was launched in early 1998. The purpose of this paper is to briefly explain the steps taken at Satyam, to establish and institutionalise the use of SPC technique for its software processes. The paper begins with a brief review on Statistical Process Control (SPC) techniques. The paper describes the various challenges faced by the organisation in applying the SPC techniques. The author, by virtue of being associated in this organisational endeavour, presents an insight as to how they were resolved and the results of implementation. The paper also illustrates the benefits, which the organisation has gained over the period. Also, some issues with which the organisation is still grappling with are discussed.

Keywords

High-maturity, SPC, statistical process control, software

1 Introduction

Robert Frost, the famous English poet, once wrote:

> "Two roads diverged in a forest wood,
>
> and I, took the road less traveled by,
>
> and that has made all the difference."

These words, trigger two streams of thoughts for me: First, the organization that chooses to adopt statistical process control (SPC) for planning and managing software projects, find that their ability to plot and interpret the control charts has made all the difference - in attaining its organizational objectives. Second, the path that leads to the unknown terrain of applying concepts of SPC in the software process, as part of an improvement program, requires vision, tenacity and innovation. Certainly, this path is less traveled by.

The purpose of this paper is to briefly explain the steps taken at Satyam Computer Services Ltd., India – a SW-CMM Level 5 organization, to establish and institutionalize the use of SPC techniques for its software processes. The paper begins with a summary explanation of SPC and goes on to describe the various challenges faced by the organization in applying the techniques: how they were resolved and the results of implementation. The paper concludes with the benefits derived by the organization and lessons learned in adopting the SPC techniques, which would be of immense use to the organizations that are keen on adopting these techniques for software processes.

Satyam Computer Services Ltd. (India) (hereinafter called Satyam), is a totally integrated IT solutions provider, engaged in application development and maintenance, systems integration, and engineering services (CAD/CAM/CAE). It specializes in providing customized IT solutions for industries in the areas of manufacturing, financial services, insurance, transportation, telecom, healthcare, power etc. It was assessed at SEI CMM Level 5 in March 1999, when it was among the first ten companies in the world to attain Level 5.

In the course of its journey to Level 5, by late 1997, the organization had established a basic measurement system. Each member of the software project team was reasonably clear as to what data he needed to collect and at what frequency. Measures were identified for each project; they were properly collected and analyzed, but the recipients of these reports hardly looked at them. Because neither the data nor the resulting report would help the managers to make decisions; the manager, in fact, did not see any value in those data, in which several man-hours were spent each month. The organization, as a consequence, was still in the 'reactive mode'.

There were numerous challenges, when it was decided to adopt the application of the SPC technique in the organization. Some of them were manifested in the form of questions, such as:

- Is it possible to apply the concepts of SPC to software process?
- How can I apply SPC techniques to my software processes, so that my manager can understand it and use it effectively?
- The software process is performed by people, and not machines. Hence, the very act of analyzing a software process might change the human behavior in dysfunctional ways.
- Can SPC technique provide us a timely feedback on process performance?
- How will the use of SPC make my customers satisfied?

Other issues in adopting the SPC techniques in Satyam were:

- Myths and misconception on using the SPC to deal with these problems were prevalent in the organization (See box below)
- Lack of ability to interpret control charts appropriately and draw possible reasons for variances.
- Monitoring of processes was difficult
- Resistance to taken on additional 'burden' to use of SPC techniques, for analysis of processes.

Some common myths and misconceptions about SPC, as prevalent in Satyam:

- SPC is required only when you decide to go for Six sigma methodology, and not otherwise.
- It is not possible to apply the concepts of SPC to the software processes.
- It is very cumbersome to use SPC for processes, other than manufacturing, and benefits are often not commensurate with the 'pain' associate with the use.
- SPC is another 'topic of the month' and it will fade out next month.
- The aim of SPC is to 'stabilize' a process whereas we need to 'improve' our processes, hence the technique is not suitable for us.

Further, there were a number of technical issues in using the statistical techniques for a software process like: Do we need a stable software process in today's rapidly changing, high tech environment? Or, does a capable process always add business value to the organization? Some of the other apprehensions were primarily based on wide variation in skill and competence level of different software professionals. Studies of the differences between software professionals have ranged from 10:1 up to 100:1, with 20:1 differences being fairly in common (Curtis, 1990; DeMarco, 1999)

Nevertheless, the organization made a determined attempt to apply SPC techniques to the software processes, across the entire gamut of projects, even though questions continued to be raised on the issues of specific processes, measures, and the exact statistical techniques that would provide significant business value to the organization. The primary reason for such a decision was the realization that the use of SPC techniques would help management to predict a result within a reasonable limit, by establishing bounds on process performance (in terms of attributes like cost, quality, and schedule). The SPC techniques would help to understand the 'reliability' of human actions, and determine the 'voice of the process', which would lead to discovery of 'voice of project' and then the 'voice of the organization'. Understanding the patterns of processes would help the management to discern the causes of variation therein, and take suitable (preventive) measures. Moreover, SPC would help the management shift their focus from a defect detection system based on final inspection (i.e. reactive mode) to a defect prevention system (Grabov, 1998). The adoption of SPC was based on the business need of the organization for defect prevention. Using SPC would not only be practical, beneficial, and appropriate to software development but also would improve understanding and execution of software processes (Gibson, 1998). Besides, it was found difficult for Satyam to attain higher maturity levels i.e. Level 4 and Level 5 of SW-CMM, without adopting SPC techniques effectively.

During the journey of SW-CMM Level 5, sometimes around early 1998, the Chairman, B Ramalinga Raju, conveyed his decision, in a key note address to senior executives in the organization: "…… **all** our management decisions have to be based on data…. what we have demonstrated is our capability of basic quantitative management ……. Now we need to move further…. towards robust statistical control methods"

2 Statistical Process Control (SPC)

SPC is a statistically based analysis of a process and measurements of process performance which aims to identify 'common' and 'assignable' causes of variation in the process performance, and maintain process performance within limits.

Traditionally, using SPC tools always have implied the use of seven basic tools for statistical process control (Ishikawa, 1986). These are (i) Flowcharts (ii) Scatter diagrams (sometimes called scatter plots) (iii) Histograms (iv) Pareto analysis (v) Cause and effect diagrams (also known as fishbone diagrams or Ishikawa charts) (vi) Run (trend) charts, and (vii) Control charts. Later on, other tools like affinity diagrams, stratification were added to the list. But using SPC always implies the necessary use of control charts (Paulk, 2000).

Control charts provide a statistical method for distinguishing between variation caused by 'common causes' – that are natural and inherent to the process and whose results are common to all measurements of a given attribute – and variation caused by 'assignable causes' – that are due to events which are not part of the normal process. The common causes of variation are due to normal or inherent interaction among the process components (people, machines, material, environment, and methods). Variation in process performance due to common causes is random, but it varies within predictable bounds. The process performance in such a case is characterized by a stable and consistent pattern over time, and the process is called "in control" or "stable" or "predictable". On the other hand, assignable causes (also called special causes) indicate sudden or persistent abnormal changes to one or more of the process components. These changes are not inherent part of the process and specific to some transient circumstances only, which can be due to abnormal changes in things such as inputs to the process, the environment, the process steps themselves, or in the way in which the process steps are executed. They markedly affect the process performance and its outputs, and the process, in such a state, is said to be an "out of control" or "unstable" process.

Depending upon the types of measurement data – whether it is variables[1] or attributes[2] type, there are different types of control charts available for the software processes:

- Average (X-bar) and range (R) charts
- Average (X-bar) and standard deviation (S) charts
- Individuals and moving range (XmR) charts
- Individual and median moving range charts
- Moving average and moving range (MAMR) charts
- c charts
- u charts
- Z charts

The X-bar and R charts apply to variables data, as do the X-bar and S charts. The c, u and Z charts apply to attributes data, provided the underlying data distribution conforms to the Poisson distribution. The individuals and moving range charts may be used for either variables or attributes data.

[1] Variables data (sometimes called discrete or measurement data) are usually measurements of continuous phenomena like: elapsed time, effort spent, years of experience, memory utilization, CPU utilization, cost of rework.

[2] Attributes data (sometimes called count data) occur when information is recorded only about whether an item conforms or fails to conform to a specified criterion or a set of criteria. Attributes data almost always originate as counts like: the number of defects found, the number of source statements of a given type, the number of lines of comments in a module of given lines, the percent of projects using formal code inspections.

3 Application of SPC Techniques In Satyam

The adoption of SPC techniques in Satyam can be easily divided in two phases: The first phase began with a 'big bang' approach throughout the organization in early 1998, and was marked with confusion among the users about the basic technique. The confusion was primarily due to lack of awareness and ability to prepare an appropriate control chart or to interpret the results and use it for making timely decisions. The situation slowly improved and after over two years of its adoption, we moved the second phase of implementation, when many of the methods were 'institutionalized' in the organization. Today, invariably all projects in Satyam use SPC techniques. While some use it to ensure just the stability of the process while others use it to improve the capability of various processes. The paradox is that when we launched the initiative, we had a set of questions for which we sought answers. Now when we believe, we have reasonable answers to them, we find that we have another set of equally intriguing but important issues waiting for an early resolution.

At Satyam, it did not take much time to realize that in order to adopt the SPC techniques successfully, the managers at all levels had to be trained on the techniques. Not only should they know the intricacies involved in collection, verification and validation of data, they needed to know the techniques of analysis (e.g. which control chart would be appropriate for a set of data?) and now to interpret the results (e.g. when do I need to act?). People at different levels across the organization were identified for 1-day, 2-day and 3-day training courses on SPC, depending upon their role and requirements. An external faculty, who was conversant with both software processes and SPC techniques, were invited to impart the training. Further, project managers and project leaders, who took the training, were encouraged to pass on their learning to other team members in the project.

The training helped people to dispel initial apprehensions they had about the technique and its applicability in the software processes. It was aimed at providing a desired level of understanding for all the team members in a software project. Further, the members of the SQA group were identified as 'champions' to lead this initiative in their respective business units, and they were given extensive training on the subject.

Enthused with the new initiative in the organization and the training received, each project manager wanted to deploy the SPC techniques in his software project. Two months after the launch of this new initiative, over 60% of software projects had made an attempt to deploy the SPC techniques, despite being at different stages of lifecycle. Though, not every one of them was clear as to why he was doing so. The initial euphoria declined, after some problems were identified:

- Despite training, there was a difference of opinion on deciding as to which control chart would be appropriate for which set of measures in the project.
- Attempts were made to put most of the metrics in the software project under SPC, without realizing the benefits it should give.
- What should be done to those processes, which are not yet stable?
- How should we deal with processes with infrequent iterations?
- Lack of data quality

The next part of this paper describes as to how the organization dealt with such problems.

The use of control charts for different sets of measures triggered a widespread discussion and brainstorming amongst team members of the same and different projects. The member of the SQA group attached to these projects facilitated the discussions. Numbers of fortnightly/monthly review meetings were organized between the project team and the concerned SQA group, which were solely devoted to the SPC discussion. The SQA member had assumed the responsibility to put all the learning, doubts and clarifications in the form of FAQs, in the online Quality Management System (QMS) at Satyam, which could be accessed by any user. Learning was encouraged with other members of the SQA group and other project members through information sharing in both formal and informal sessions. There can be no denying that one of the reasons for successful adoption of the SPC technique is the ability to maintain strong communication network amongst the users. Regular review on the progress on this initiative by the top management was another booster to this effort.

3.1 Selection of Control Chart

Over a period of several discussions, it was learned that all of the control charts mentioned previously could be used to draw information about the software processes, especially the average (Xbar) and range (R) charts, XmR, c, u, p and Z charts. The areas of application for them were cost, earned value, defects per build, staffing etc. XmR charts, however, offered the widest applicability. Many of the software measures are attributes type (like the number of defects per page, the number of defective items found, percentage of non-conformances in the output of a phase, the number of source statements of a given type). For data such as these, the individual and moving range (XmR) charts are used to examine the time- sequenced behavior of process data. Many software professionals recommend the XmR charts for count data (Wheeler, 1998). XmR charts do not have any distribution requirement, unlike other attribute charts (like c and u charts, which require that process behavior have an underlying Poisson distribution; or p or np charts which require processes to follow binomial distribution). Moreover, XmR charts allow us to avoid the issue of sustaining sub-group homogeneity of measurements taken over wide intervals of time, and therefore can be used for periodic data (Florac, 1997). XmR charts assume standard deviation (sigma) is constant across all observations, and are applicable irrespective of whether area of opportunity is constant or variable. Nevertheless, there was an issue of control limits becoming wider, and control charts becoming less sensitive to assignable causes when containing non-homogenous data. Also the results of XmR chart are not reliable when the events are so rare that the counts are small and values of zero are common Despite this, for many software processes, it was found more appropriate to plot individual data points rather than the averages of sub groups. It was agreed that individual data for XmR chart would be the 'standard Satyam control chart', although other charts, which take advantage of knowledge about the statistical distribution of data, were gradually introduced when appropriate (e.g. the u chart was used for measures like 'system failure per day', defects per thousand lines of source codes, in some projects, when empirical evidence indicated the process to follow Poisson distribution and are of opportunity was not constant). Use of one type of control chart (XmR) was recommended initially to reduce confusion amongst users. Gradually, other charts were introduced when their underlying assumptions were met and users in the organization had gained some confidence in using charts.

3.2 Selection of process

It was realized that it did not make sense to apply SPC to analyze every software issues in the organization. It is unreasonable to expect that every software process should be managed using control charts (Paulk, 2000). What is important is to identify which are the critical processes that would help in meeting the goals of the project or organization and what are the attributes of the processes or products that require improvement. To begin with, we decided that attributes related to product quality, process duration, product delivery, and process cost that would be common to many software processes need to be statistically controlled. At Satyam, the processes selected for SPC analysis are related to the following measures:

- Effort Estimation Variance
- Schedule Estimation Variance
- Delivered Defect Density
- Productivity
- Defect Removal Efficiency.

3.3 Selection of level of process

Process capability may be determined theoretically at any level: for the entire organization or product line or at project or team or at individual level. The issue was then raised of appropriate level of process at which SPC should be applied. Application of the technique at too low a level loses the value addition; at too high a level of process the technique loses its meaningfulness. At too low a level, say at sub-process level, improving capability at the sub-process level need not yield the optimal solution at the project or organizational level. On the other hand, at too high a level of process, the variability is so high that one may not get any insight into the process. For example, the head of a business unit in Satyam wanted to use SPC techniques for earned value parameters at his division (aggregate) level, to know how his division was performing on this parameter. The problem was that at any point in time, over fifteen projects, all of different sizes and complexities, would be running under him. On plotting the data, he found that there was so much of variability in data that he could hardly get any clue as to why his charts indicated such a wide variation. Wider control limits meant poor predictability for him. (Figure 1.0)

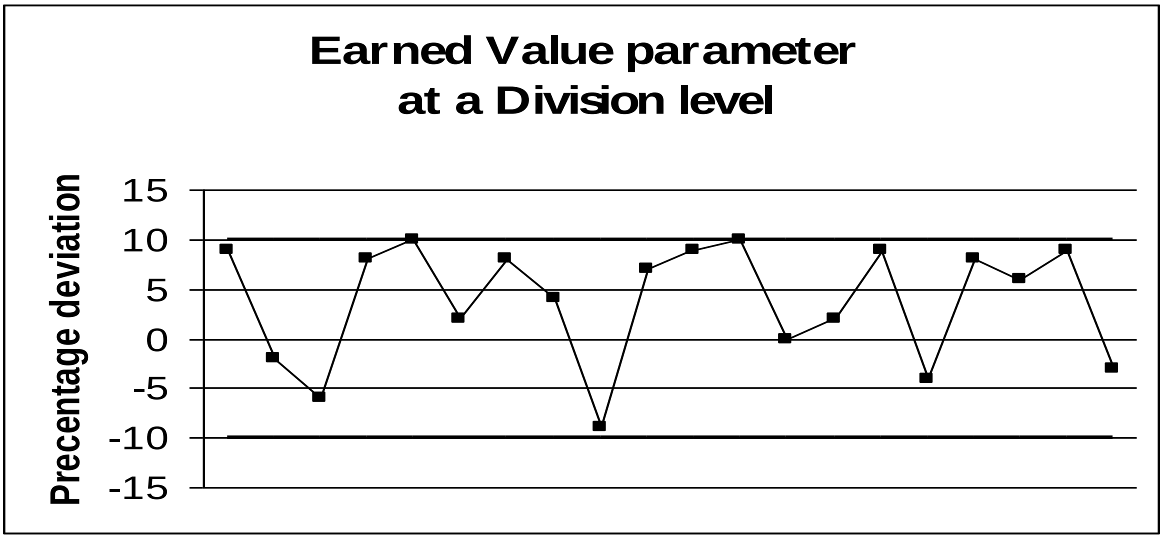


Figure 1.0: shows the relevant earned value parameter at a division level ie at the aggregate level of 15 projects, looked stable. The wide variation did not give any insight.

In another instance, the Project Manager of a Health Care project at Satyam put the 'total no. of defects found in each component inspection' on the control chart. He was very happy to find that the process was stable until he decomposed the inspection process performance data into various types of defects. On decomposition, he realized that control charts for five out of seven types of defects indicated the presence of assignable causes, and, hence the process for finding these defect types was not stable

In many cases, it was found that it is only at sub-process levels that true process management and improvement can take place, because there is far too much variation in the overall process to be helpful in identifying possible actions for improvement. Sometimes, smaller programs were better able to use SPC due to the analyst's end-to-end knowledge of the program. Thus, it was difficult to arrive at one solution that would be universally valid. It was, therefore, left to the judgment of individuals to apply SPC at the level where decision-making occurs and there is a clear value addition in the use of the technique.

3.4 Dealing with unstable process

Cognizance was also taken that many of the software processes in some business units were still not stable. The problem was: should the SPC be used in such cases? Some of the processes, like requirements definition, software architecture, and many of the management process take relatively more time to get stabilized. The software processes that stabilize first are design, code, and test, since there is usually an adequate amount of inspection data to apply SPC. Without stability and the related knowledge of what a process can do, it is not possible to distinguish signals in measured values from the random noise that accompanies them. In fact, in such a case we have no basis to recognize assignable causes. Therefore, efforts were undertaken to ensure that process behaviors get stable for a time period sufficient enough to detect any unusual behavior, before SPC could be applied on them. For example, during application of SPC techniques on coding process in an Insurance Project in a new business unit indicated data points beyond the Satyam's prevailing control limit for effort variance of 30%. The result is shown in the below (Figure 2.0):

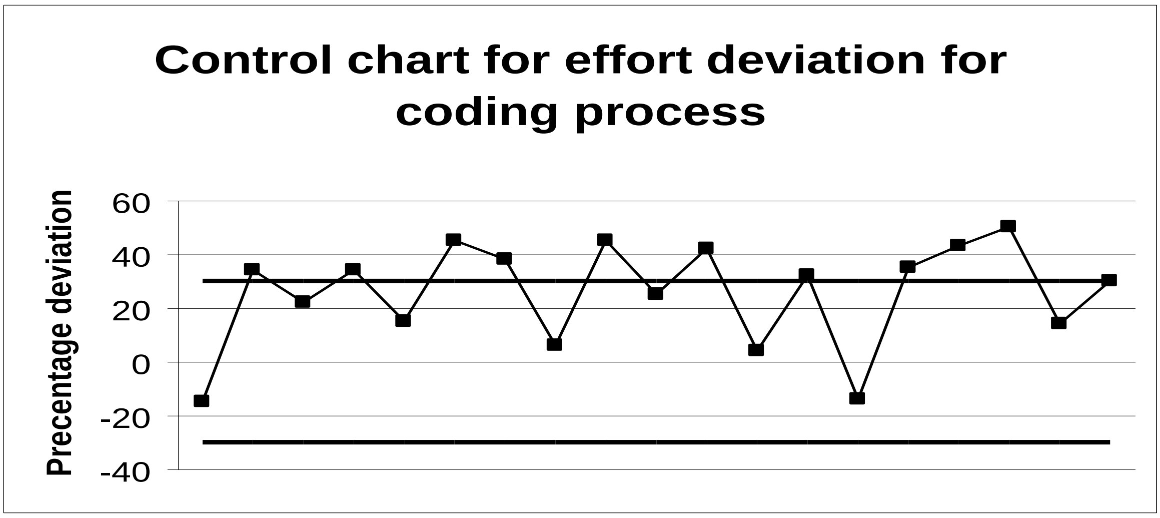


Figure 2.0: An unstable coding process in a new business unit.

On investigation, the problem of inconsistent and unpredictable results was found due to low software coding experience of the developers as well as non-availability of a critical resource. As a result, the task was not performed thoroughly. The code and test error rates decreased substantially as a consequence of developers gaining experience and writing the codes with due concentration. The issue of non-availability of required resource was escalated and back up resource was provided.

3.5 Problems associated with data

In many projects, when we began applying the technique, we identified the problems associated with data. These were in the form of:

- Missing data, e.g. what software version is in use for which application; documents in database lack number of pages; number of non- conformances were reported but since when?
- Found many suspect values for some attributes; coarse and imprecise data due to rounding, taking measurements too frequently
- Highly aggregated data, which led to wide variability in the control limits
- Data collected at phase end or on monthly basis that was too late for real time control.
- For some processes, a high volume of data was available (at the sub process level) which required to be categorized - by product family, application domain and in other such useful groups, for an effective control and comparison purpose.
- For some processes, there were not sufficient data: small projects often had insufficient iterations; mature projects in maintenance mode often had releases only once or twice a year; pilot projects often produced too few data points.

For many of the above problems related to data, there was no way but to dig out more details to improve the 'quality' of data or start afresh. Then the issue cropped up as to how much data, to start with, would be required to construct a control chart. It was agreed upon that instead of not using SPC altogether on limited data, it would, at least, be useful to calculate and plot the limits, and seek out the assignable causes that the data might suggest. The control charts could always be updated when more data become available. The caution one needs to exercise in such a case is in interpreting the initial results. Limits computed by less than 20 points may not be reliable. (Florac, 1999). Use of SPC on processes with infrequent iterations needs to be validated.

The problem of insufficient granularity was resolved by ensuring that the data used for control charts have a resolution that is smaller than the process standard deviation. In Satyam, we follow a rule of thumb: we ensure that the set of points that are plotted always take on values that range over at least 5 possible levels of discreteness. Increasing the number of levels was accomplished by measuring more precisely, by decreasing the frequency of measurement to allow for variation to occur between measured values, or by increasing the size of sub groups to allow for more variation within a sub group.

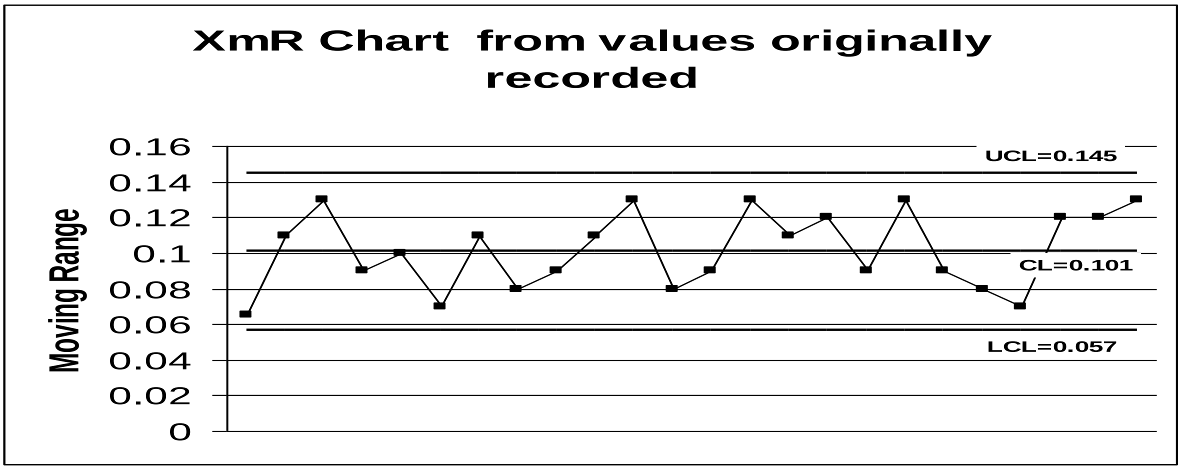


Figure 3.0: Ensure that the data used for control charts has a resolution smaller than the process standard deviation, to resolve the problem of insufficient granularity.

3.6 Informal stabilization

Many times it looked that the defined processes used by the projects was not consistently implemented or measured, as required. Analyzing the data suggested of stratification or mixing of data from different sources. In such a case, the data obtained were "informally stabilized" by removing the known assignable of variation. The data were simply examined graphically before placing it on a control chart and "informal stabilization" – as Paulk put it, would occur invariably (Paulk, 1999).

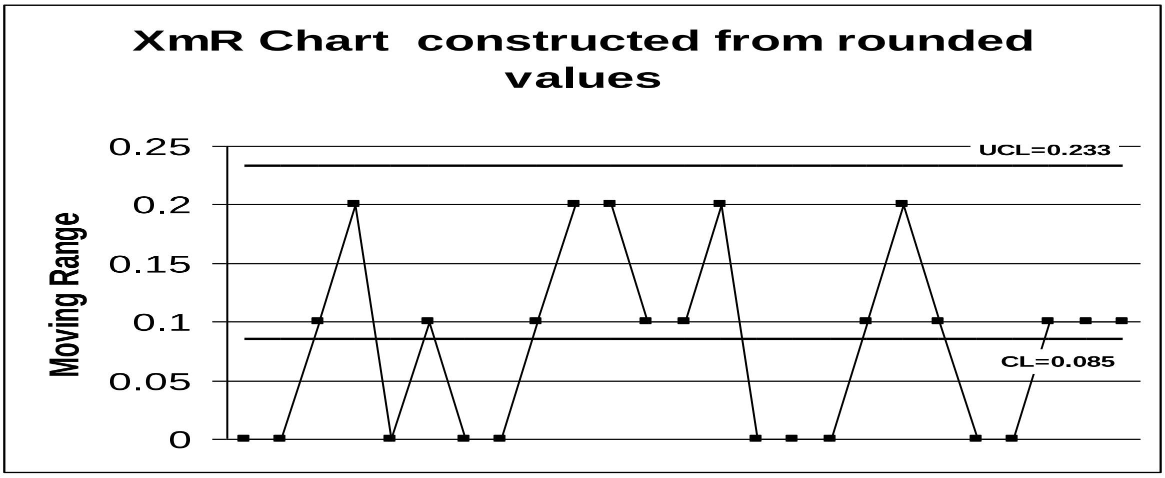


Figure 4.0: Stepped appearances are because of insufficient granularity – only three possible levels.

3.7 Selection of tests for detecting 'out of control' situations

Different literature suggested different tests for detecting out-of-control patterns on control charts e.g. Nelson (Journal of Quality Technology, 1986) suggests eight types of tests for this purpose. Initially, there was confusion among users as to which of these tests should be used. It was not easy to comprehend all the tests at first go, for them. Initially, we agreed on a simple rule for detecting a signal (possible assignable cause) as Test 1: a point outside the 3- sigma control limits, for majority of the cases. Later on, additional tests (Carleton, 1997) were used:

- **Test 2:** At least two out of three successive values fall on the same side of, and more than two sigma units away from, the center line
- **Test 3:** At least four out of five successive values fall on the same side of, and more than one sigma unit away from, the center line
- **Test 4:** At least eight successive values fall on the same side of the centerline.

Other tests of detection (Figure 5.0) found in the literature make the control chart more sensitive to signals, but leads to unnecessary false alarms in many cases. The decision of the detection rule was purely based on risk and economic-trade offs.

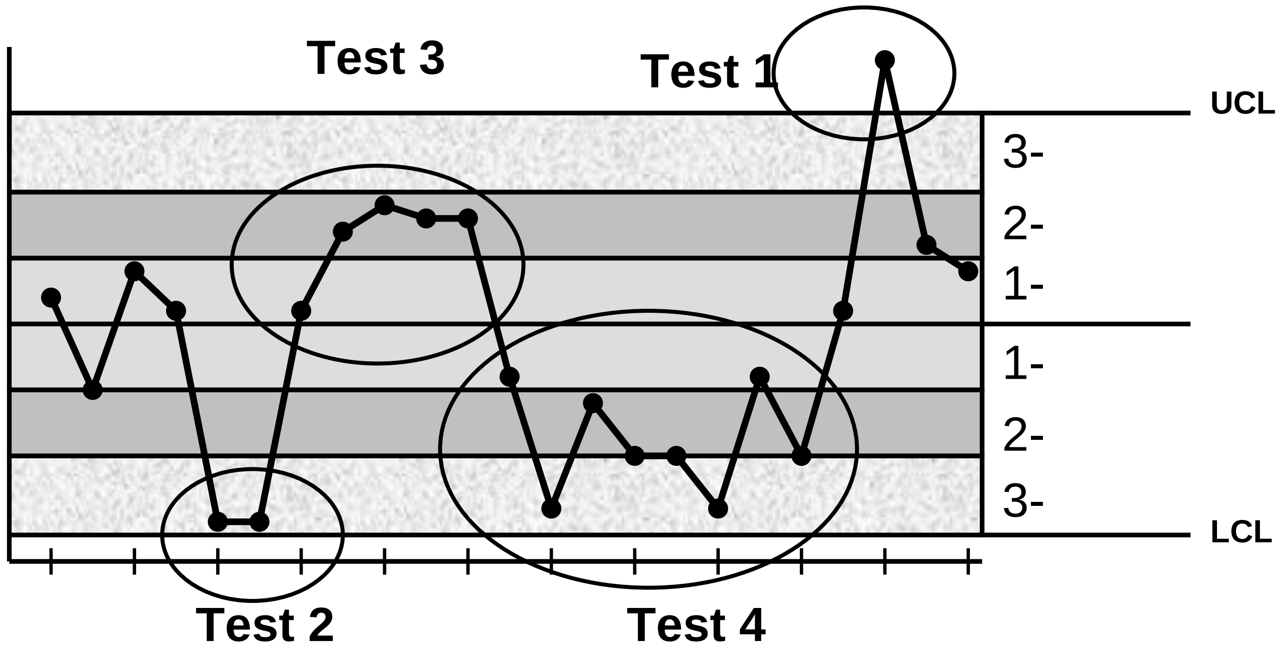


Figure 5.0 : Tests for detecting 'out of control' situations

4 Benefits

New challenges have kept on pouring in every day in using the SPC technique. They are being resolved in a routine manner, using the collective judgments of all concerned. What started, apparently, as a directive from the top management, has now become 'institutionalized' practice in the organization. The organization has, in turn, begun reaping the benefits in terms of:

- Change in the mindset of software developers and managers from defect detection system to self-discovery mode. There is a marked transition in their approach and attitude from being in a reactive mode to being proactive mode. The focus shifted towards preventing a problem, before it occurs.
- Substantial improvement in predictability of various process and project related data like: software quality, earned value, cost and schedule. The range of variation has been reducing with each passing year.
- Interpretation of the SPC results facilitated improvement in various processes, between February 1999 to December 2001:
    - Effort Estimation variance improved by 20%
    - Schedule Estimation variance improved by 12%
    - Productivity improved by 4.5%. The marginal improvement is attributed to the learning curve due to technology and domain application shifts over a period of time.
    - Delivered Defect density reduced by 12%
    - Defect Removal Efficiency improved by 10%

- Improved employee morale: The result of annual ‘associate satisfaction survey’ carried out in the organization indicates a marginal improvement in employee morale (~ 4%).
- An attempt is underway to understand human behavior. As a human being, one will err. The preliminary results of the SPC techniques has given some good insight as to how often they will err, which root causes could be eliminated or minimized, and what corrective or preventive actions need to be taken to desensitize our processes to human error.

5 Lessons Learned

During the course of applying the SPC techniques successfully for last several years at Satyam, some significant lessons were learned:

- A few major steps were taken at the management level, which played a crucial role in making the SPC program successful:
  - Provide adequate training to the all users of the SPC techniques.
  - Arrange regular reviews by top management on the progress of the initiative
  - Clear communication and regular interaction amongst users contributed immensely to sorting out implementation and interpretation issues. The assigning of a SQA personnel to each project to clear any doubts of the users expedited implementation; putting FAQs on SPC techniques on Quality Internet web-site provided online clarification to many doubts; networking amongst SQA personnel and members from other projects lead to sharing of knowledge, which expedited the adoption of SPC techniques at Satyam.
- Many people had suggested initially to test –pilot SPC for a selected group of projects. It was argued that based on their initial success stories, the buy-in for the techniques would be higher. Instead, Satyam adopted the “big-bang” technique and implemented use of SPC organization wide at one go. Contrary to the expectations of many, it proved successful. The success stories on a few projects motivated others to try harder.
- Don’t try to put all metrics under SPC – the shoe doesn’t fit to all. SPC should be applied at the level where decision-making occurs or in those areas that are a matter of concern for the project/organization.
- Look for indications of enhanced quality or better performance, not just problems. The emphasis should be on reducing the range of variation rather than plotting the points on a control chart.
- Programs stay in observation mode for extended periods before stabilizing. Statistical techniques are useful to apply on stable processes
- One needs to observe some caution in interpreting the results of control charts, with lesser amount. of data.
- The control chart is a form of instrument like a thermometer, which will provide you with data to guide decisions. It is not the panacea to all software process problems.
- And at last but not the least: Understand your data, understand your data and understand your data before you put them on the control charts, otherwise be prepared to get unexpected results. Understanding how the data is defined and collected is crucial to know what the data represents. Consistency in data collection and reporting is imperative. Aggregating and comparing data within and across projects require care and planning. Most of software data come from individual software projects and are specific to the particular software process in use, both in terms of their definition and contexts in which the measured results are interpreted. Differences among software process require change in analysis or the way data are collected and aggregated. \

6 Miles to go…. Still

It is a matter of satisfaction that the SPC has taken deep roots in the organisation. We had many issues, when we initiated the using the control charts. Many of them kept on getting resolved with time. However, our kitty of questions has not become smaller. At Satyam, we are still looking for an answer to many questions, like:

- A concern is being raised on using the measurement data to evaluate the performance of employees. Will the use of SPC to measure individual behavior (paying attention) not change the behavior (Howthorne effect) and 'temper' with the data before putting in the control chart?
- Does having a stable, predictable process has any meaning in today's environment, where everything changes rapidly? Do we want "all" processes to be stable? Does the objective of making a process stable not contradict the concept of continual improvement?
- Do we need to serve our customers with processes operating at + 3-sigma level whereas his requirements can be met at lower sigma level? Does it add any business value to the organization to operate process at higher capability level? Are the benefits to the organization operating with process on target with reduced variance, commensurate to the investment made by the organization on doing so?
- Challenges are still there to resolve issues like building better operational definitions (for better consistency and reproducibility), dealing with confusing thresholds and control charts.
- How can SPC be applied in a value added way without all levels of management being trained and involved
- There are statistical techniques other than control charts that can add value. Which of them can be better applied to software processes? And how ?

Clearly, the list never ends. One solution leads to another question, which requires further answer. We are already working on many of them. With our determination and dedication, we will find answers to all of them – one day. Till then, here we move on, and as Robert Frost, had put it:

> "The woods are lovely, dark and deep,
>
> But I have promises **to** keep,
>
> And **miles to go** before I sleep,
>
> And **miles to go** before I sleep! .

7 Literature

- Carleton Anita D and Paulk, Mark C.(1997) “Statistical Process Control (SPC) for Software Tutorial, Software Engineering Institute, Pittsburgh.
- Curtis, Bill, (July/ August 1990), “ Managing the Real Leverage in Software Productivity and Quality, “American Programmer, Vol 3, No. 7, pp 4-14.
- Demarco Tom and Lister Timothy (1999) , Peopleware, 2nd Edition. Dorset House, New York Press , NY
- Florac William A. and Carleton Anita D. (1999) Measuring the Software Process : Statistical Process Control for Software Process Improvement, Addison –Weseley, Reading , MA
- Frost, Robert (1874-1963) Stopping By Woods On A Snowy Evening, The Poetry Of Robert Frost, Houghton Miffin Company.
- Gibson, John E. (April, 1998) “ Is the Software Industry Ready for Statistical Process Control?” a white paper. Gathersberg, Md: Lockheed Martin Mission Systems.
- Grabove, Pavel (May 1998), “SPC’s Effectiveness Prediction and Analysis” American Society for Quality, 52nd Annual Quality Congress.
- K. Ishikawa, (1986), “Guide to Quality Control”, Asian Productivity Organization, Tokyo, Japan,
- Nelson, L.S. (1986), “The Schewart Control Chart – tests for special causes”. Journal of Quality Technology, 16(4), p 238
- Paulk Marc C. (October, 1999), “Towards Quantitative Process Management with Exploratory Data Analysis,” Proceedings of the Ninth International Conference on Software Quality, Cambridge, MA
- Paulk, Mark C (2000), “ Applying SPC to the Personal Software Process”, Software Engineering Institute Pittsburgh.
- Shewhart, Walter A. (1993): Statistical Method from the View point of Quality Control. Washington D.C. : Graduate School of Department of Agriculture
- Wheeler, Donald J and Poling, Shiela R. (1998) Building Continual Improvement. Knoxville, Tenn: SPC Press.

8 Author CV

Mr. Rajesh Gupta, MBA, PMP is a Process Consultant (Quality) at Satyam Computer Services Limited, an Indian based SW-CMM Level 5 organization. He holds B. Tech degree from Indian Institute Of Technology (IIT) Kanpur and MBA degree from Xavier Institute of Management (XIM) Bhubaneswar where he won Gold Medal for securing the top rank in the batch. He has over twelve years of experience in the field of Project Management and Process Management. He has presented numerous workshops and training sessions for many clients frm a wide variety of industries.He is a certified Project Management Professional (PMP) by Project Management Institute (PMI) USA, and has earned the Certified Quality Manager (CQM) designation from American Society of Quality (ASQ) USA.

He is co-author of the new model on IT enabled Outsourcing Services, called e Services Capability Model (e^{SCM}) http://itsqc.srv.cs.cmu.edu/escm/index.html), developed by Carnegie Mellon University, USA. He is a Lead Auditor ISO 9001 and is due to be authorized by CMU as Lead Evaluator e^{SCM} . Presently, he, as a visiting scholar to CMU, is working on the development of other Appraisal methods for e^{SCM}.

Besides, he has already published three papers on quality related topics in national publications.

Process-oriented effort estimation of software projects

Ursula Passing, Susanne Strahringer

Department of Information Systems
European Business School
Oestrich-Winkel, Germany
{ursula.passing, susanne.strahringer}@ebs.de

Abstract

A new, process-oriented effort estimation approach for custom software development projects is presented. It provides a methodological framework to support expert judgements and analogy estimating, the most popular estimating techniques in practice, in order to make estimates transparent and repeatable. The proposed approach is based on a standard decomposition of the software development process. Three dovetailed methods are provided on how to estimate the items in the decomposition, depending on the time of the estimation, the availability of project details, the desired estimation accuracy and the desired prescriptiveness of the estimation method.

Keywords

Effort estimation, software development process, Rational Unified Process, software cost, project management.

1 Introduction

Effort estimation continues to be an important issue in custom software development project management today. Although researchers have been working on estimation models for decades, subjective and unstructured estimation practices, such as the so-called expert judgements, still seem to prevail in day-to-day effort estimating.

The effort estimation approach proposed in this paper provides a methodological framework to support expert judgement and analogy estimating, the most popular estimating techniques in practice. The aim is to make estimates transparent and repeatable.

The proposed approach is process-oriented: The effort is estimated using a standard decomposition of the software development process. The Rational Unified Process serves as an example development process. The effort is estimated based on the activities performed in a project.

The rest of the paper is organized as follows: Section 2 deals with a short introduction of existing estimation techniques. Then, the fundamental principles of process-orientation in effort estimation are explained. In Section 4, the new estimation approach is presented in detail. Finally, the approach is discussed and conclusions are drawn.

2 Existing estimation methods

The existing effort estimation techniques can be classified into three types: Expert judgement, analogy estimation and algorithmic estimation models.

Expert judgement denominates estimation methods that are solely based on judgements of people who have extensive knowledge in the project-relevant area.[Boe81]

Analogy estimation is in some respect a systematic form of expert judgement.[Shep96] To estimate the effort for a new project, a similar completed project is identified and its actual cost is used as a basis for the estimate of the new project.[Boe81]

In algorithmic models, the development effort is estimated as a function of variables representing the most important cost drivers in the project. Usually, the variables are identified by correlation analysis of data on completed projects.[Boe81, Hee92] A more recent algorithmic approach are neural networks. They also work with combinations of parameters, but use artificial intelligence learning algorithms in order to identify the relations between them.[Gray97] Algorithmic estimation models typically use the estimated size of the software to be developed in the project as the key parameter. Examples of algorithmic models are COCOMO [Boe81, Boe00] and Function Point Analysis (FPA)[Alb83].

In practice, expert judgement and analogy estimation are the most frequently applied estimation methods, algorithmic estimation methods seem to be rarely used.[Hee92, Led93, Hee99]

The estimation techniques can be be fundamentally distinguished by whether they do or do not rely on a decomposition of the project.[Hug96] Decomposition means splitting the project up into smaller parts (components), and adding the component estimates up to form the overall estimate. This is also called bottom-up estimating.[Boe81] The contrary is usually called top-down estimating: The effort is estimated for the whole project at once.[Boe81]

Decomposition reduces the uncertainty inherent in the estimation by reducing the complexity of the estimation task – the probability of gross estimating errors is reduced because the sum of the errors in the component estimates will likely be smaller than the error in an overall estimate. Moreover, the additive connection between the component estimates limits error propagation. Top-down estimating may allow faster estimates but involves higher uncertainty.

Whereas expert judgements are often based on a decomposition of the project (project components are estimated and then added up), analogy and algorithmic estimation techniques usually operate top-down (whole projects are compared and estimated).[1]

All existing estimation methods are traditionally product-oriented, i.e. the product to be developed in the project serves as basis for the estimation: Experts estimate project effort based on information on the system to be developed. In analogy estimating, similar projects are identified by similarities in the product (e.g., system size, programming language, etc.).[Shep96] Algorithmic models are based on an estimate of the system size and other product-related factors (e.g., product complexity, user friendliness, etc.).[Boe81, Boe00, Alb83]

In contrast thereto, the estimation approach presented in this paper uses a process-oriented bottom-up approach, i.e. the activities gone through to create the product serve as the basis for the estimation. In the following Section, the idea of process-orientation is explored in depth.

3 Process-orientation in effort estimation

The fundamental reasoning behind process-oriented effort estimation is that the effort needed to complete a project is not induced by the software product itself, but by the activities that are performed to produce it. In fact, the development of a product can involve very different amounts of effort, depending on how the development is organized.[Hull02]

Prerequisite for applying a process-oriented approach to effort estimation is an implemented software development process model: The process model defines *who* is doing *what when* and *how* to reach a certain goal.[Jac99]

Process-orientation has various implications on effort estimation:

As a methodology to support expert judgement, the approach proposed here involves a decomposition of the project before the actual estimation takes place. Generally, there are two possibilities how a project can be decomposed: Either it is decomposed into the products to be delivered (components, interfaces, documentation etc.), or into the activities necessary to build the product.[Boe81] In order to reduce estimation effort and make estimates more comparable, it would be desirable to have a standard decomposition for all the company's projects instead of setting up an individual decomposition structure for each project. However, the product view would hardly allow a standard decomposition because a company's projects (i.e., the systems to be built) are too heterogeneous. Process-orientation however makes such a standard decomposition possible: In every software project, more or less the same activities are performed, irrespective of the system to be produced. Therefore, the development process can be broken down into a standardized set of process components for estimation, i.e., workflows and, more detailed, activities.[Tau80] The process component estimates are summed up to yield the total effort estimate. The standard process decomposition must only be modified when the development process in the company is modified.

A standard decomposition has to support estimations at different stages in the project lifecycle. Therefore, it must provide different levels of detail: At the very beginning of the project, an estimate can only be rough, because little detail on the project is known. As the project proceeds, more and more detail is available – if the later project estimates are to be more accurate than the rough up-front estimates, they have to be more detailed. Generally, different levels of detail are possible in process- as well as in product-oriented decomposition structures, but the process view offers certain advantages:

In a process-oriented standard decomposition, all levels of process detail can be included in the structure right at the time the decomposition is created because the process model clearly defines what activities will be performed during the project. The more detailed levels will not be used until later in

[1] An exception is FPA: The size estimate in FPA is obtained by a decomposition of the system into functions (external inputs, outputs, etc.). However, the adjustment factors relate to the system as a whole (reusability, operational ease, etc.), and the estimating result is still an estimate of the whole project.[Alb83] FPA can therefore be considered a hybrid method.

the project, but they keep the estimator aware of what has to be done. This means that a process-oriented decomposition is stable: The decomposition structure will not change during the project, only the level of detail is varied. Even when there are major redirections in the project scope (e.g., additional products are to be included), this will only lead to certain activities being repeated but does not change the decomposition structure itself. E.g., when an additional product component has to be implemented, the implementation activities are performed once more. A stable decomposition provides better planning reliability and makes estimates comparable.

A product-oriented decomposition on the other hand cannot contain all levels of detail right away because all the product details are not yet known at the time when the decomposition structure is created. Such a decomposition is less stable, it will grow throughout the project. Redirections in the project scope may cause major changes in the decomposition structure. E.g., the above mentioned additional component would have to be added in the decomposition structure. This instability complicates project planning and makes estimates less comparable.

The process view also implies that, as opposed to most algorithmic models, an estimate of the system size is no prerequisite for estimating the project effort; the influence of the size estimate on total effort is reduced significantly. This is an advantage because the estimation of the system size is very difficult, especially in early project phases, and represents the main source of estimation errors in algorithmic estimation models.

Furthermore, process-oriented effort estimation directly supports project management: Activity-specific estimates enable the project manager to compare estimated and actual effort on an activity level which allows for better project control and helps to meet the targets.

4 *A process-oriented estimation model*

4.1 Overview

In this Section, the proposed process-oriented effort estimation approach is presented in detail. Figure 1 gives an overview on the model.

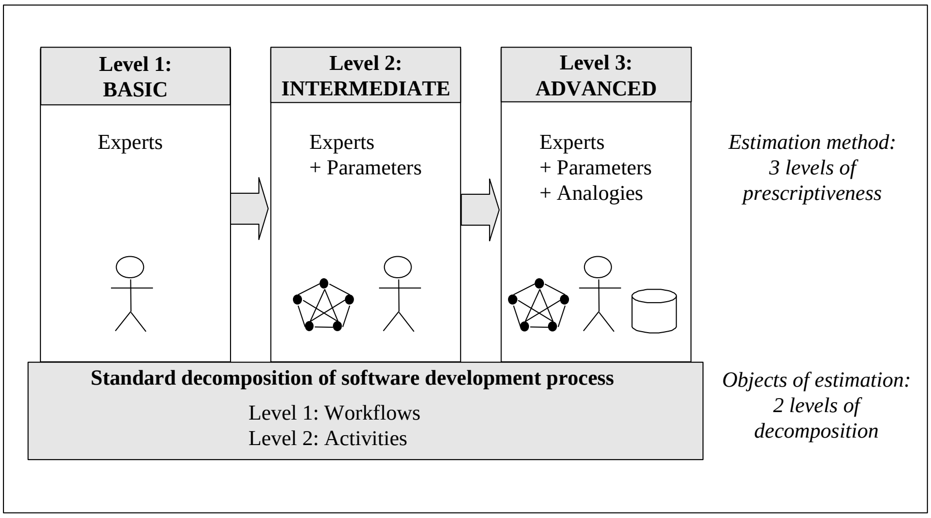


Figure 1: Model overview

The estimation approach is fundamentally based on a standard process decomposition. The decomposition defines *what* is to be estimated: The items in the decomposition are the objects of estimation.

Three estimation methods build on this decomposition, involving increasing levels of prescriptiveness: A basic, an intermediate, and an advanced method.[2] An estimation method defines *how* the objects are estimated. A method's level of prescriptiveness indicates to what extent the method prescribes a repeatable estimation procedure which produces estimates that are transparent and comprehensible. The result is a "2x3" model structure: There are two levels of detail in the decomposition, and three levels of prescriptiveness in the estimation method.

The model essentially relies on expert judgements. Expert judgements have often been discredited as subjective because they lack a methodological foundation and do not produce repeatable and comprehensible estimates. This criticism also applies to subjective analogy estimating, i.e., estimating with (subjective) experience from completed projects.[Hug96] But, given their easy applicability and widespread use, it has been frequently suggested that expert judgements, supported by a transparent and repeatable procedure, would be welcomed as the most appropriate estimation technique in practice.[Park94, Hug96] The aim of the proposed estimation model therefore is to provide such a methodological foundation for expert judgements, in order to make estimates transparent and repeatable.

4.2 A standard decomposition of the development process

As shown in Figure 1, the foundation of the proposed estimation approach is the standard decomposition of the software development process. The decomposition defines the objects of estimation (*what* is to be estimated?): In this case process components, i.e. workflows and activities.

During the development of the model, the Rational Unified Process (RUP) will be used as the underlying process model, but generally, any other software development process model could be used. The RUP is a process model for object-oriented software development projects and is based on the Unified Modeling Language (UML).[Kru98]

The RUP represents a standard breakdown of the software development process into workflows and, more detailed, into activities. This standard decomposition is used throughout the estimation model, at varying levels of detail: The first level of the decomposition are the workflows – the RUP defines nine of them for the software development process: Business modeling, requirements, analysis and design, implementation, test, deployment, project management, configuration management, and environment. The second level of detail is the activity level: Each workflow is composed of eight to fifteen activities.

The estimation model incorporates these different levels of detail because during the lifecycle of a project, estimates of differing levels of detail are made. With two levels of decomposition detail, it is possible to estimate those parts of the project on which sufficient information is available on an activity basis, and the parts that are still rather unclear on a workflow basis. That is, the decomposition and with it the whole estimation model can be used flexibly at any time of estimation.

The result of the decomposition is a table which includes the workflows and activities the software development process is composed of. Figure 2 shows an extract, the RUP requirements workflow: This workflow is composed of 13 activities. The artifacts (i.e., products) and roles (i.e., responsibilities) are included as supplementary information for the estimator. Depending on the time of the estimation, the requirements workflow can be estimated either on the workflow or on the activity level – although it can be assumed that even for early estimations (e.g., for a project proposal), it will probably be possible to estimate the requirements workflow directly on the activity level. This is different for other workflows, e.g., implementation: At this early time of estimation, there would probably not be enough information available yet to estimate on the activity level.

The estimation methods described in the following Section provide the procedures as to how the workflows and activities are to be estimated. Whereas the *Basic* estimation method operates on the stand-

[2] This structure resembles Boehm's COCOMO models which also provide basic, intermediate and advanced levels. However, whereas Boehm's levels refer to the level of detail in the estimate (from a rough estimate on the system level to detailed estimates on the component level), the basic, intermediate and advanced levels in the proposed approach refer to the quality (repeatability, transparency) of the estimation procedure.[Boe81, Boe00]

ard decomposition as introduced above, the *Intermediate* and *Advanced* methods use an enhanced standard decomposition, additionally including workflow and activity effort drivers.

WF	Activities	Roles	Artifacts
Requirements	Capture a common vocabulary Elicit stakeholder requests Develop requirements mgt. plan Develop vision Manage dependencies Find actors and use cases Structure the use case model	System analyst	Glossary Stakeholder requests Requirements mgt. plan Vision Supplem. specification Requirements attributes Use case model
	Detail the software requirements Detail use cases	Requirements specifier	Software requirements Use case Use case package
	Model the user interface Prototype the user interface	User interface designer	Actor Boundary class User interface prototype Use case storyboard
	Prioritize use cases	Architect	
	Review requirements	Requirements reviewer	

Key:
mgt. = management; supplem. = supplementary; WF = workflow

Figure 2: Standard decomposition of the RUP Requirements workflow

These effort drivers are intended to facilitate the mental process of inferring the effort needed for a given process component. They indicate what significantly influences the effort of a particular process component and therefore give an impression of the component's "size". To be really helpful in the estimation, it is important that the effort driver is easier to quantify than the process component as a whole. Therefore, the effort driver must either be a measurable quantity, or easier to estimate than the process component itself. In the following, the effort drivers are also simply called (size) parameters.

An example illustrates the nature of the effort drivers: Assuming the appropriate level of decomposition for estimating the RUP requirements workflow is the activity level, an activity to be estimated is "Elicit stakeholder requests" (see Figure 2). To support the estimator in assigning an effort to this activity, a possible "size" parameter is the identified number of stakeholders. The number of stakeholders involved in the project may either be known very early, then the parameter would be measurable. Otherwise, it can relatively easily be estimated by considering the scope of the project. It is assumed that estimating the effort needed for eliciting the stakeholder requests is facilitated if it is known whether the project will involve three or 25 different stakeholders.

Our research on identifying meaningful parameters that drive process component effort and are easier to quantify than the component effort itself is still ongoing. Preliminarily, about 20 parameters have been identified, with one parameter influencing several workflows or activities. Most parameters are of quantitative nature, i.e. amounts like number of stakeholders, number of use cases, number of interfaces. Almost all quantitative parameters are UML classifiers, i.e. elements in UML diagrams. This ensures a clear definition of the parameters and improves the general applicability of the estimation model. There are also several qualitative parameters which have to be assessed by ratings, like staff experience or how well organized the client is. All parameters together characterize the project.

However, it is not intended to establish a functional relationship between a parameter and the component effort. The parameters merely serve as hints and it is the task of the estimator to infer the component effort from the parameter value. This has been designed so by intent, in order to make the model applicable to different environments and not to tailor it to a particular empirical data basis.

For the complete requirements workflow, the quantitative effort drivers are shown in Figure 3. For a better focus on the parameters, the roles and artifacts have this time been omitted from the extract.

WF	*Workflow parameters*	**Activities**	*Activity parameters*
Requirements	*# stakeholders* *# business use cases* *# use case packages* *# interfaces* *# user groups / # actors* *# requirement "packages"*	Capture a common vocabulary	*# stakeholders*
		Elicit stakeholder requests	*# stakeholders*
		Develop requirements mgt. plan	--
		Develop vision	*# stakeholders*
		Manage dependencies	*# requirement "packages"*
		Find actors and use cases	*# business UC*
		Structure the use case model	*# UC packages, # actors*
		Detail the software requirements	*# UC packages, # req. "packages"*
		Detail use cases	*# UC packages*
		Model the user interface	*# UC packages*
		Prototype the user interface	*# UC packages*
		Prioritize use cases	*# UC packages*
		Review requirements	*% of above*

Key:
mgt. = management; req. = requirements; supplem. = supplementary; UC = use cases, WF = workflow

Figure 3: Parameter structure in the RUP Requirements workflow

The effort drivers in the "Workflow Parameters" column relate to the workflow as a whole and provide support for estimations on the workflow level. All workflow parameters are UML classifiers, except requirement "packages" (rough requirements). In the "Activity Parameters" column, possible effort drivers for the various activities are shown, most of them again taken from the UML. They help in estimations on the activity level. No parameter has been identified for "Develop requirements management plan". It is assumed that planning and management activities can more easily and more appropriately be estimated by lump sums or percentages. Similarily, "Review requirements" is estimated as a percentage of the rest of the requirements workflow.

In the case of the requirements workflow, the activity parameters are approximately the same as the workflow parameters, only that they are attributed to single activities. In other workflows, e.g. the implementation workflow, the workflow parameters differ more strongly from the activity parameters because at the time of a workflow-level estimate, less information is available on the implementation tasks than at the time of an activity-level estimate. Therefore, the workflow parameters have to be quantifiable earlier in the project than the activity parameters.

Whereas the quantitative parameters influence certain workflows and activities, the qualitative effort drivers affect the project as a whole. They are therefore not explicitly enlisted in the requirements workflow example.

In the following, three methods are described how to estimate the process component effort.

4.3 Three effort estimation methods

An estimation method prescribes *how* the objects of estimation are to be estimated. The proposed model offers three alternatives to do so, involving increasing degrees of prescriptiveness: A basic, an intermediate, and an advanced estimation method. These three levels build on each other: The intermediate method cannot be used without having already implemented the basic method, and at the same time represents the prerequisite for using the advanced estimation method.

4.3.1 Basic estimation method

The first level is called *Basic*. The items of the decomposition are estimated directly by experts. In Figure 1, the foundation of the method on expert knowledge is symbolized by the stick figure.

The estimating procedure in the basic method is depicted in Figure 4: Based on the standard process decomposition, several experts, independent from each other, estimate the process component effort. They then decide on a consensus estimate for each process component. The consensus component estimates are added up to form the total project effort estimate. This procedure is usually called "Delphi" technique.[Boe81]

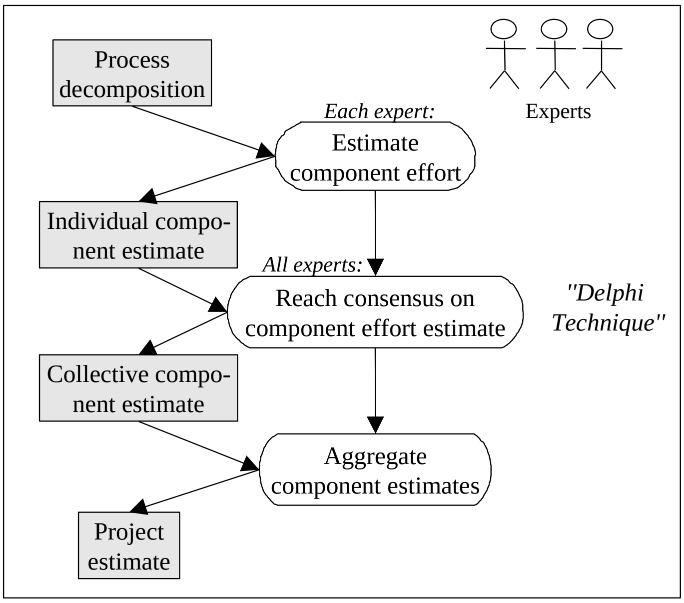


Figure 4: The basic estimation method

An example illustrates the procedure: Assuming the level of decomposition appropriate for the time of the estimation is the workflow level. The decomposition provides nine workflows. The experts estimate how much effort each workflow will involve. For each workflow, the experts discuss their estimates and agree on a consensus estimate. The nine workflow estimates are added up to form the project estimate.

The basic estimation method provides a low degree of prescriptiveness. Structure is only provided by the standard decomposition. Therefore, the method is relatively simple and flexible. At the same time, it provides basic comprehensibility and repeatability due to the standard decomposition. However, the procedure how the experts come up with their component estimates remains intransparent to outsiders. This can be remedied by a sound documentation of the estimation but still represents a weakness of the method.

If a consistent estimation method has never been used at all in an organization, the basic estimation method represents a first step into structured effort estimation that is superior to "guessing". The method is also appropriate for situations where detailed information on the project is not available yet.

The basic estimation method can be used with both levels of detail in the decomposition, i.e. also on the activity level, so that it is applicable throughout the lifecycle of a project. The more detailed the decomposition, the more accurate the estimates will likely be because the objects of estimation are smaller. But, the quality of the estimates produced with this method always depends on the estimating capability of the experts – the experts have no methodological support beside the decomposition.

If an organization is used to formal estimation approaches or if detailed information on the project is available in an estimating situation, it should be considered to seize this knowledge using a more sophisticated estimation method that provides more comprehensible and repeatable estimates and is apt to produce more accurate estimates, as described in the following.

4.3.2 Intermediate estimation method

The *Intermediate* estimation method improves the support given to the experts when estimating the items of the decomposition. This is achieved in two ways: First, the objects of estimation are structured further: Parameters are identified that drive the effort of the workflows and activities and therefore indicate their "size", as introduced above. Second, a more structured procedure is defined for how the experts' estimates should be treated to form a consensus estimate. In Figure 1, this additional support has been symbolized by a pentagonal structure for the parameter structure.

In the basic estimation method, the experts could only build on their experience when estimating the effort of a given item in the process decomposition. Now, an effort driver is introduced for each process component as an indicator of its size. As already explained in Section 4.2, this parameter is intended to facilitate the mental process of inferring the effort needed for a given process component.

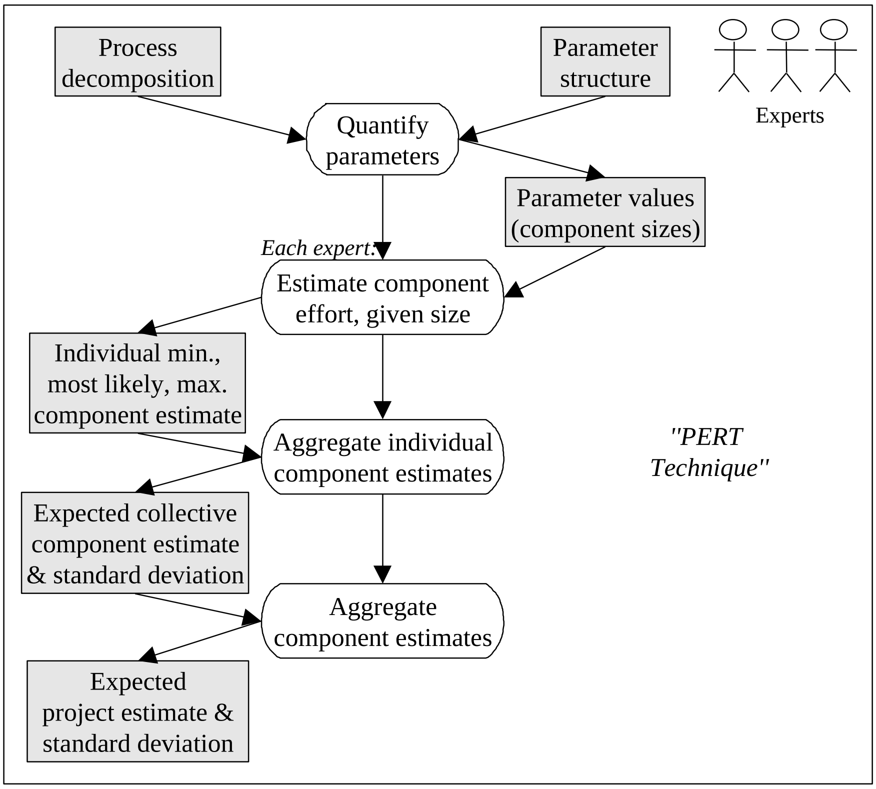


Figure 5: The intermediate estimation method

The estimating procedure, which is more repeatable and comprehensible to outsiders than the "Delphi" technique used in the *Basic* estimation method, is illustrated in Figure 5: After the parameters in the decomposition have been quantified, each expert provides a minimum, most likely and maximum effort estimate for each process component, given the "process component size" indicated by the parameter. The three estimates are weighted and aggregated among the experts to form the expected value and the standard deviation for the collective process component estimate. The collective component estimates are then aggregated, yielding an expected value and the standard deviation for the whole project estimate. The procedure is also known as "PERT" technique [Put79, Hoe99] and can be easily managed with a spreadsheet prepared in advance.

The improvements over the basic method are the following: The estimation of the process components is facilitated and the component estimate is more comprehensible due to the "size" parameter. Furthermore, a repeatable procedure is provided to handle diverging expert estimates. Moreover, the final estimate is not a point estimate, but a range indicated by the expected value of the estimate and the standard deviation; this gives an impression of the uncertainty inherent in the estimate.

The intermediate method is more prescriptive than the basic method; it increases the comprehensibility and repeatability of estimates. The method is also more complicated and may involve more time for the estimation itself. It is suited for organizations that desire better repeatability and comprehensibility of estimates. The method can be used with both levels of detail in the decomposition, it is therefore applicable at all stages in the project.

4.3.3 Advanced estimation method

The *Advanced* estimation method is a further enhancement of the intermediate method. Whereas it relies on the same structure as the *Intermediate* method (process decomposition and parameter structure), the repeatability and comprehensibility of the estimating procedure are further improved: When estimating the process component effort, the experts are supported by an analogy database. The analogy support is indicated in Figure 1 by the database symbol.

The estimating procedure in the advanced method is depicted in Figure 6: In the beginning, it is equivalent to the intermediate method. The experts estimate the effort of the process components, given the values of the "size" parameters. Whereas in the intermediate method, it was up to the expert how much effort to assign to a process component given its parameter value, this decision is now supported by historical data. The parameter values, the component estimates and actuals, as well as the estimated and actual total effort of completed projects are stored in a database. This means that analogies are provided at the process component level, not at the project level as in other estimation techniques.[e.g., Shep96]

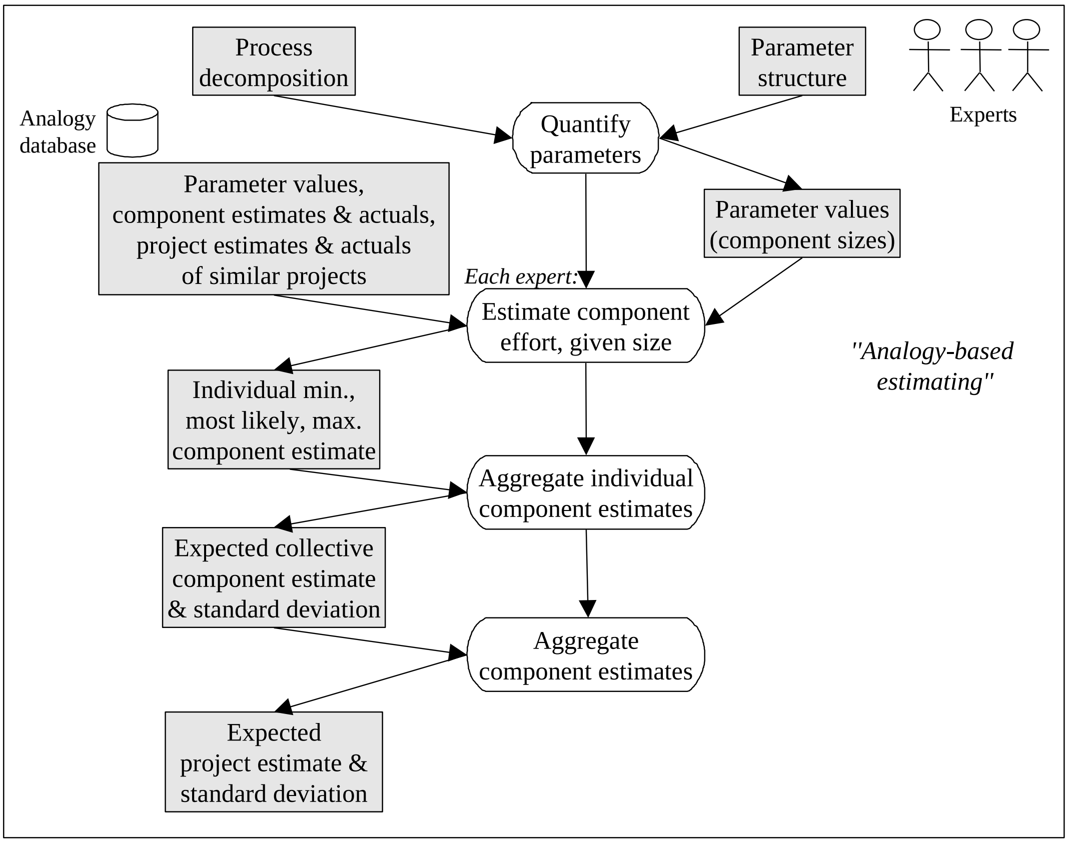


Figure 6: The advanced estimation method

A process component is characterized by its size parameter. The expert searches the database for projects with similar parameter values. The estimated and actual effort of the analogical process component give an indication of the effort required for the process component to be estimated. The expert's experience is then "only" needed for the decision whether to assign the same, less, or more effort to the new component compared to the historical project.

The remaining steps in the procedure are equivalent to the intermediate method: The experts provide three effort estimates per process component, a minimum, most likely, and maximum estimate, which are aggregated to yield an expected collective component estimate and its standard deviation. The collective component estimates are then aggregated, the result is again an expected project estimate and a standard deviation which indicates the uncertainty in the estimate.

Compared with the intermediate method, the advanced method offers several advantages: The estimation process becomes more transparent and more repeatable by the structured use of analogies because the influence of (subjective) expert experience has been reduced. Furthermore, the method enables estimators to improve their estimates from project to project because for new estimates, lessons learned from past projects can be seized. Past evidence may also reduce the standard deviation in the final aggregated estimate because it provides the experts with a common starting point.

The advanced estimation method can only be applied when a database with a sufficient number of projects has been established. As long as such a database is not available, the intermediate method can be used: The estimation results achieved with this method are entered in a database; when the project is completed, the actuals are added. The intermediate method is used until a sufficient number of projects has been recorded in the database to allow the application of the advanced method.

The advanced estimation method is the most prescriptive of the proposed methods and offers the best comprehensibility and repeatability. However, this requires preparation. It will take months or even years until the database provides enough analogy projects to really support new estimates. Therefore, the method is appropriate for organizations that desire a structured estimation method and are willing to invest time in establishing it. Again, the method can be applied with both levels of decomposition.

5 Discussion

The proposed estimation model represents a new, process-oriented approach in effort estimation modeling.

Process-orientation is a new direction in software effort estimation. It was argued that not the software system to be developed induces the effort, but the activities performed to produce it. Also, process decomposition allows a standard decomposition – this saves estimation effort, increases planning stability and makes the estimates more comparable. Focusing on processes instead of the software product also saves the estimator from estimating the software system size as a prerequisite for an effort estimate. The influence of the software size estimate on the whole effort estimate is significantly reduced – and with it the propagation of errors. Furthermore, the process decomposition enables better project control.

Moreover, the prescriptive bottom-up estimating technique in general is unprecedented in the effort estimation domain. The usual decomposition estimating techniques (micro expert judgements) do not use a standard decomposition and do not provide a methodological procedure on how the items in the decomposition should be estimated. FPA uses decomposition only for the size estimate. By providing varying levels of detail in the decomposition, the approach can be applied at different stages in the project. The potential of estimation errors is reduced by both the decomposition and the additive connection between the component estimates.

However, this also implies that an implemented software development process model is the prerequisite for applying the approach. Whereas the three estimation methods can be used with any process model, the decomposition and parameter structure have to be adapted if a process model other than the RUP is used. They also have to be adapted in case the development process is modified.

The approach seizes the knowledge of experts and provides a methodology that reduces the subjectivity of the estimates. The "2x3" structure of the model accounts for different estimation requirements and levels of experience in organizations. It allows to establish a structured estimation method from scratch and to enhance it step by step.

The model is theoretically founded, it has not been designed to fit empirical data, as is the case with most algorithmic estimation models. The model therefore is generally applicable in any software development organization, as long as a process model is available. It has been designed to account for the needs of medium-sized software development companies who cannot afford to invest in effort estimation techniques like large organizations.

Apart from the *Basic* estimation method, the estimation model provides estimate ranges rather than point estimates, which reveals the uncertainty in the estimate instead of simulating accuracy.

However, to evaluate the usability of the estimation method and the accuracy of the estimates, the proposed estimation approach has to be tested in real projects. In a first pilot implementation of the *Basic* method together with a company-specific process model in a software development organization, positive results have been achieved. But it is necessary to conduct more, especially longitudinal, studies to evaluate the usefulness of the whole approach.

6 Conclusion

A new, process-oriented effort estimation approach for custom software development projects has been presented. It provides a methodological framework to support expert judgement, in order to make estimates transparent and repeatable.

The proposed approach relies on a standard decomposition of the software development process. To account for different levels of detail available at different stages in the project, the approach provides two levels of decomposition. To meet different requirements concerning the repeatability and transparency of estimates, three different estimation methods, involving varying degrees of prescriptiveness, are offered.

The results from a first pilot implementation together with a company-specific process model in a software development organization were promising, so that the authors consider process-oriented effort estimation techniques as a worthwhile area of further research.

7 Literature

[Alb83] Allan Albrecht and John Gaffney, “Software function, source lines of code, and development effort prediction: A software science validation”, *IEEE Transactions on Software Engineerin*g, vol. SE-9, no. 6, pp. 639–648, 1983.

[Boe81] Barry Boehm, *Software Engineering Economic*s, Englewood Cliffs, 1981.

[Boe00] Barry Boehm et al., *Software Cost Estimation with COCOMO II*, Upper Saddle River, 2000.

[Gray97] Andrew Gray and Stephen MacDonell, “A comparison of techniques for developing predictive models of software metrics”, *Information and Software Technolog*y, vol. 39, no. 6, pp. 425–437, 1997.

[Hee92] Fred Heemstra, “Software cost estimation”, *Information and Software Technolog*y, vol. 34, no. 10, pp. 627–639, 1992.

[Hee99] Fred Heemstra and Rob Kusters, “Software cost estimation in the Netherlands: 10 years later”, in *Proceedings of the European Software Control and Metrics Symposium (ESCOM-SCOPE*), 1999, pp. 23–31.

[Hug96] Robert T. Hughes, “Expert judgement as an estimating method”, *Information and Software Technolog*y, vol. 38, no. 1, pp. 67–75, 1996.

[Hull02] M. Hull et al., "Software development processes – an assessment", *Information and Software Technolog*y, vol. 44, no. 1, pp. 1-12, 2002.

[Hoe98] Martin Höst and Claes Wohlin, “An experimental study of individual subjective effort estimations and combinations of the estimates”, in *Proceedings of the 20th International Conference on Software Engineering (ICSE*), 1998, pp. 332–339.

[Jac99] Ivar Jacobson, Grady Booch and James Rumbaugh, *The Unified Software Development Process*, Boston etc., 1999.

[Kru98] Philippe Kruchten, *The Rational Unified Proces*s, Reading/Mass., 1998.

[Led93] Albert L. Lederer and Jayesh Prasad, “Information systems software cost estimating: A current assessment”, *Journal of Information Technolog*y, vol. 8, no. 1, pp. 22–33, 1993.

[Park94] Robert Park, Wolfhart Goethert, and Todd Webb, “Software cost and schedule estimating: A process improvement initiative”, Special Report CMU/SEI-94-SR-3, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, URL: `http://www.sei.cmu.edu/publications/documents/94.reports/94.sr.003.html`, 1994, retrieved 12.11.2001.

[Put79] Lawrence Putnam and Ann Fitzsimmons, “Estimating software costs (Part 1)”, *Datamatio*n, vol. 25, no. 9, pp. 189–198, 1979.

[Shep96] Martin Shepperd, Chris Schofield, and Barbara Kitchenham, “Effort estimation using analogy”, in *Proceedings of the 18th International Conference on Software Engineering (ICSE*), 1996, pp. 170–178.

[Tau80] Robert Tausworthe, “The work breakdown structure in software project management”, *Journal of Systems and Softwar*e, vol. 1, no. 3, pp. 181–186, 1980.

8 Author CVs

Ursula Passing

Ursula Passing is a doctoral student at the Department of Information Systems at the European Business School at Oestrich-Winkel. She is currently working on a Ph.D. thesis on effort estimation for software projects.

Susanne Strahringer

Prof. Dr. Susanne Strahringer is chair of the Department of Information Systems at the European Business School at Oestrich-Winkel. Her research focus covers different topics in the area of object-oriented application development.

SPI - easy in theory, hard in practice

M.sc. Geir Kjetil Hanssen[1], Dr. Tore Dybå[1], Prof. Tor Stålhane[2], Dr. Torgeir Dingsøyr[1] and M.sc. Hans Westerheim[1]

1. *SINTEF Telecom and Informatics, Norway*
2. *Norwegian University of Technology and Science, Norway*

Abstract

This paper draws attention to obstacles in performing SPI and research on SPI. It describes four recent cases from PROFIT, a Norwegian research program within SPI. These cases show how some of the problems that this type of work has to deal with. The paper gives a discussion on the cause of the problems and how they could be avoided. It also gives a more general discussion on how we should use our experiences in reshaping the way we work.

The paper focuses particularly on action research and the cooperation between companies doing software development on one side and researchers studying the process on the other side.

The paper is not a comprehensive study that covers all aspects of SPI. However, it is an experience report that will be of practical use for others doing SPI and research on SPI.

Keywords

Practical problems with SPI

Action research

Company - Researchers' cooperation

Research planning and preparation

1 Introduction

SPI has over the years become a well known acronym in the software business. There is however still not a common understanding of the concept, no concise definitions are available, at least no definitions that all agree upon. But it seems like most individuals involved in software development, both on the top (the management) and on the shop floor (the developers) means that SPI is an important success factor. This opinion is based on the realization that most software companies really need to perform SPI. Both the need of SPI and the belief in SPI is present. This has resulted in numerous SPI activities. Most companies today have, at least once, performed SPI - in any other form. But despite this fact, many of these companies have experienced severe problems in doing so. How come?

This paper intends to discuss this important question. It is the authors' opinion that this is a question that has been given far too little attention in the SPI community. Look into the huge amount of literature and experiences made available over the years and you will find that most of it is focused on technologies, methodologies and ideologies. One example is the almost religious and ever ongoing discussion on what approach to SPI is the best. Is it the analytical approach (e.g. QIP [1]) or is it the benchmarking approach (e.g. CMM [2]). Another discussion goes on which methodology is the best. Is it the rigid waterfall mentality [3] or is it the iterative mentality, exemplified by the unified process [4] or Extreme Programming [5].These are only a few examples on topics given large attention. They are extremely important; in fact, they represent much of the core of SPI. But before we can discuss this we must consider the basic question on *how do we accomplish SPI and research on this subject in practice?*

Theory is great, but when the rubber meets the asphalt it often gets hot! The realization of this fundamental question is based on numerous experiences the researchers in the PROFIT research program have gained over the past few years. PROFIT is a collection of SPI initiatives in the Norwegian software industry. It is partially financed by the research council of Norway; it started in 2000 and will finish at the end of 2002. PROFIT is a joint venture between researchers from universities and research institutes on one side and several software companies (approx. 10) on the other side. The concept of this research program is that two or three researcher's works with one company trying to perform a well defined SPI-activity. The overall research model in PROFIT is action research [6]. Examples of research activities are improvement of the estimation process, definition of a new development strategy or adaptation of an off-the-shelf methodology (such as RUP).

PROFIT is in many ways a continuation of the SPIQ program (Software Process Improvement for Better Quality). This program aimed to increase the competitiveness of 12 participating Norwegian software companies, by creating an improvement environment in the companies, and introducing ideas, like the Experience Factory, and adopting it for small and medium-sized enterprises in Norway [7]. It also included pilot projects for improvement in companies, as well as discussion forums for issues related to process improvement. Further, it contained dissemination activities like conferences and the writing of a method handbook for process improvement in Norwegian [8]. This project lasted from 1997 to 1999.

So far, we have been setting our focus on the success stories when we have been publishing our research results. This is natural; after all it is success that motivates us to keep on and it is important to share these experiences with the rest of the community. However, every once and awhile we make bad experiences. Some would even say that we have made quite a lot of bad experiences, that is - obstacles we have met in our research at the companies. It is now time to dive into these problems to try to find explanations that can help us, and others, to remove or at least reduce the problems. This constitutes the central problem in this paper:

"What are the most important obstacles we must cope

with when improving software development processes?"

The discussion presented here does not build on a solely objective research process. We have not collected data for statistical analysis. What we do have is a solid base of experiences and knowledge from more than twenty case studies. Four of these are described briefly in chapter 4.

2 A difficult task

During the two first years of PROFIT we have come to realize that SPI is not an easy task at all. Despite the fact that all necessary conditions are present we have encountered many problems in performing our deed. These conditions are:

- we have extensive knowledge and practical experiences with SPI
- there is identified an obvious need for improving the software development process
- we have highly motivated organizations (at least highly motivated individuals) and highly motivated researchers

In addition to this, PROFIT contributes economically. Each participating company gets a significant amount of money besides that it also gets funding for participating at conferences and other relevant activities.

So, why do we struggle? Shouldn't doing SPI and research on SPI in these companies under these conditions be straightforward? When we started out in 2000 we thought so, but now we know better. This experience report tries to point out the most important obstacles we have tripped into. The intention is to make others in the same position aware of these obstacles. Our audience should be other researchers involved in similar activities but also individuals in software companies that work with QA and SPI. This is not performed as an objective study of the question 'how to perform SPI'. We think however that our findings and our thoughts are important and relevant as they are. We hope and believe that this will trigger similar thoughts with our soul mates that have undertaken the same mission. We continue the paper with the description of the theory that has formed the basis for our work and after this we describe our business cases. We then present the main obstacles we have met and a discussion on these, where we try to point out suggestions for avoiding or at least reducing these problems. Finally we give some conclusions.

This is not the first time someone have addressed the problem pointed out in this paper. In [9], the authors' points to several important explanations to the question of why SPI isn't used more. They say that "quite a lot of companies did not believe that SPI would solve their problems". Our experiences from the past two years indicate that this tendency is changing. The companies in the PROFIT-program have shown more interest and belief in SPI than was the case in the preceding SPIQ-program.

One concrete example of this is the assemblage of the PROFIT company participants. It took only a few months in the beginning of the research program to define all the SPI-projects and sign the contracts. We are talking about activities that occupy several man-years at each company. So it's clear, at least in our case, that the *interest* for SPI definitively is present. With this in mind, we will now try to look into the reasons why some SPI initiatives encounter problems.

3 The theory

The work in the PROFIT program builds on the work performed in the SPIQ-program. Both these research programs build on a TQM philosophy [10]. In short terms, TQM focuses on using experiences which is collected and partially analyzed in groups. Group experiences are important and TQM involve several techniques that support this. All improvement performed is directly based on experiences from the subject that is being improved.

Since PROFIT is a research program, it means that our cases represent a symbiosis of researchers and companies with real challenges. Our model has been to help the company with establishing a SPI activity, called the business project. In parallel with this, the researchers have established a research project that uses the business project as a "laboratory" for research on a particular methodology or a technique etc. The overall model is presented in figure 1.

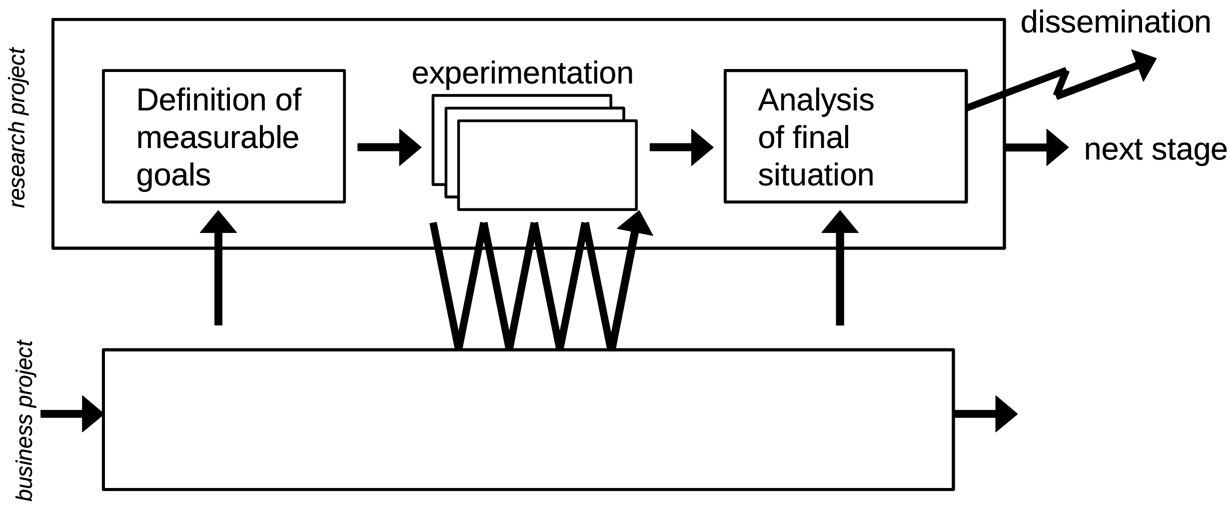


Figure 1: The PROFIT project model

This model is based on the ESSI-model [11] which again is inspired by Demings improvement wheel [10] (Plan - Do - Check - Act). The researchers interact with the business project by influencing the plan for the SPI activity. The research project collects experiences and data from the internal activities in the company; it participates in the evaluation of the results and helps in giving advice on how to institutionalize improvement. The obstacles that we identify can be classified into two groups. One regarding problems seen from the inside, that is, problems the companies have encountered regardless of the cooperation with the researchers. The other group are problems the researchers have encountered while performing the research projects. This model has directed us to work in a way that has caused some of the problems described in chapter 5.

4 The cases

This paper builds on the experiences from four SPI activities at four Norwegian software companies.

4.1 Company A

This is a medium sized software company that is organized as a local office of a large international organization. The speciality of company A is internet development and graphical design. Many of their solutions are used in marketing and branding, thus emphasizing the importance of graphical design and usability. Company A have a young and competent staff.

The SPI activity in company A was to adopt a specialized version of the Rational Unified Process. This local office received a modified version from the mother office, where RUP was extended to also handle graphical design and usability activities in the development process. The challenge was to adapt this to company A's special needs. This should be done via a series of pilot projects. The task of the researchers was to collect experiences from the introduction of this new process. The experiences would be analyzed to find improvement opportunities on the use of the new process. We also planned to evaluate the process by testing the products from the pilot projects in a specially designed usability laboratory.

The planning phase was disproportionately long. Even if the main lines of the SPI activity were clear, it took a long time to define how the researchers should interact with the pilot projects. This was partly caused by too little company knowledge from the researchers' side and partly by the company's fear of letting outsiders closely watch the introduction. Another problem in this case was that company A experienced a large loss of personnel due to hard times in the business. In two of the pilot projects the project leader was replaced, thus generating interference. The most serious problem occurred recently when the mother office announced that this local office in Norway was to be sold out of the mother organization. At the time of writing it is still not clear who the new owners will be. It is obvious that a situation like this have negative effects on both the internal SPI initiative and the according research project.

4.2 Company B

This company is the software development department in a large Norwegian company. Company B focuses on developing sales support and administrative systems. The staff is highly competent and has a long experience in software development.

The SPI activity at company B was to adopt the Rational Unified Process. It looks like the situation in company A, but company B had to do the initial adaptation themselves. According to the initial plan this was going to be done by collecting project experiences from the organization and thus to include historical knowledge and competence into the new methodology. This was going to be done in series of KJ sessions [12]. The researchers were going to take care of this experience reuse in addition to follow up the pilot projects that was going to use the new methodology.

The planning phase in this case also took a long time. It took the company a long time to define how the new methodology should be adapted and introduced in the organization. It was initially one enthusiastic initiator that started this. His colleagues showed a reserved attitude to the plans. Much of his energy and time was spent to convince and explain his ideas. After a while he got trust from the management and was able to start the planning. After a lot of preparatory work, the management wanted to remove this person from his responsibility. The reason for this was that they had expected the SPI responsible to produce a complete methodology, ready to put on the bookshelf. This was not according to the SPI responsible's view of the situation. After this, there was put some energy into explaining the totality of the work to the management. At the time of writing this situation is not entirely clarified.

4.3 Company C

This company is a large Norwegian manufacturer of communication solutions, working with hardware and software development and the integration between the two. The company has long experience in the field and have an experienced staff.

This company had previously participated in the SPIQ program and achieved quite a lot. The focus then was on where and why errors were inserted and how to change the current development process to reduce the probability of error insertion during the development process. In the PROFIT program, the SPI activity was to reuse textual requirement specifications by redesigning them using a graphical notation (UML). The system that the single pilot project was going to develop was going to be based on a similar system developed earlier, thus the focus on reuse. The research goal was to study the pilot project to see if the reuse and the new notation would cause differences in development time, error content, debugging effort and so on. To be able to do this comparison, parts of the specifications were going to be textual to form a reference.

The planning phase in this case was extremely long. This was caused by a situation where the contract was delayed. Company C could not take the risk of starting the project without the security of a formal contract. When the contract finally was signed the project got into a hurry. There was a lot to catch up with. This resulted in the decision that all the specifications were going to be expressed graphically. Thus, most of the research data was sacrificed on the company’s altar of Time-To-Market. The company still kept focus on the introduction of use cases and the whole – rather ambitious – project boiled down to the introduction of use cases. The company got a free use case course and we got the opportunity to assess the introduction of use cases into the company.

4.4 Company D

This company develops, maintains and implements a software product for the telecom industry. The software development processes is rather effective, and the developers have long experience.

The main challenges have been to control the complete life cycle of the software product. This involves handling of a target market, existing customers, design, development and implementation. The biggest challenge is to be able to separate core functionality and customer adaptation in the product from one version to the other. The main problem areas in focus of the SPI work was 1) the role of the product manager, 2) the importance of a scope document following the complete life cycle and 3) the organizational issues. The company has been running a project towards an ISO 9000 certification. The SPI work conducted by PROFIT has been organized as a separate project.

In this research project the researchers had as a special goal to meet most of the organization, this was done by conducting structured interviews with people from all departments, at all geographical locations. Even though we had an extensive contact with many of the employees we still was not able to get in touch with the upper levels of the organization (where the decision makers are). So, we had a positive dialogue with the ones that developed the solution, they had a lot of experience that was useful in this process. However we were not able to visualize the relationship between cost and benefit to the top management and thus were not able to speak their language.

5 The obstacles

The following is a list of the obstacles we have met. Some of these are obvious while others might be new to the reader. This list does not represent a complete overview of the challenges of performing SPI in practice, but we do believe that it represents some of the most important aspects to have in mind.

A. No common knowledge and understanding of the ongoing SPI-activities

SPI is an activity that involves the whole organization, and therefore also involves everybody that works there. It is the company's employees that both gives us the experiences and carry out improvement actions that come from the analysis of the collected experiences. These people are the researchers' most important team mates.

In three of our four cases (A, B and C) we had contact only with a few persons that knew about the activities. This was typically the QA or process responsible and a few project leaders. Initially this was fairly comfortable because meeting and getting to know new people always is time consuming. Having contact with only a few persons in the company also meant that we didn't have to deal with too many interpretations and views of the SPI activities being planned. During the course of the project we have experienced that this hinders that the organization, as a whole, got to know the planned SPI activities. The propagation of such information does not happen by itself. We relied on our contacts on the inside to give the information to their management and their colleagues, but that has not been the case. This has nothing to do with unwillingness. The researchers have been too vague on the importance of that everybody should at least know about the planned activities. This situation has had several negative effects on our work:

- we have not been able to establish the necessary amount of motivation that is a prerequisite for succeeding in SPI
- as the SPI activities went on and we met people for interviews or experience harvesting sessions we had to explain our intentions and activities over and over again
- having a situation where some people are informed and some are not may cause suspiciousness, which by it self is negative

The lessons to be learned from this are that the SPI activities must be preceded by information to the whole organization. It is also important to try to reveal any possibilities for internal interest conflicts and disagreements. If for instance one group of employee's show a negative attitude to the preliminary research questions and/or research methods, this *must* be handled before the "wheels starts to turn".

B. One single responsible contact person at the company

The experiences from the SPIQ program showed us that letting a group have a common responsibility could be problematic especially if there was a hard decision to be made or if the individuals in the group disagreed. Therefore the model of cooperation between the researchers and the companies in PROFIT was that each company has one single contact person. He/she is responsible for getting internal resources to accomplish the SPI activities defined. An activity can be to get people for an interview or it can be to find pilot projects to try out a new methodology. This is a fundamental function and if it fails it will jeopardize the whole project. We have mixed experiences also with using a single contact person. In some cases (A and B) we have seen that our contact, maybe unintentionally, have filtered the information between the researchers and the rest of the organization. This person may also be coloured by personal opinions and interpretations of what's going on. This has not happened out of ill will, but if you are alone, with a big responsibility in a complex setting, this is a trap that it is easy to fall into. Another negative effect of this has been that this single person has been difficult to get in touch with from time to time, simply because one person cannot be available all the time.

One obvious solution to this could be to have two responsibles at the participating company, or even better a working group with a collective responsibility and motivation. This will prevent misconceptions and it would ease the burden of each individual. This would also make it easier to reach one of the persons responsible when necessary. But with the experiences from SPIQ in mind it is extremely important that the group must take the responsibility *together*. A situation where the group members push the responsibility onto the others must be avoided.

C. Too little knowledge of the company

A company consist of individuals with individual views, individual experiences and individual capabilities. Software development is a large degree about technology, and there are a lot of technologies in use, not all equally easy to understand. A company has its own culture and history and so on. Bottom line: software organizations can be a complex matter.

Part of it can be understood by studying organizational charts, process overviews and other formal descriptions etc. Most organizations however also have a tacit side, which is as least as important as the side presented on glossy paper. To discover this tacit information it is necessary to be in contact with the individuals at all levels that actually constitutes the organization. This is a comprehensive task which requires many resources, both in time and effort. On the other hand, this is important background information when an SPI activity and possibly a research project are planned and executed. The rule is, the more we know, the better a plan we make and the fewer surprises we get. This is obvious, or at least it should be. Despite this, all our cases show that we have been too negligent of this. The reason for this is probably that we have underestimated the resources needed for doing this properly.

To improve on this in the future it is important that those who plan SPI spend enough time to learn about the organization and the processes within it that is about to be improved. To find the right therapy, you need to know both the patient and his disease.

D. An ever changing environment

Traditional SPI practices builds on four steps:

- realize the need for improvements
- plan the SPI activities
- perform the SPI activities (and make observations)
- evaluate the findings

A prerequisite for this to happen is a relative stable context:

- a stable vision and focus

- a stable company; low turnover, no buy-ups, mergers or reorganizations

We have experienced some problems with unstable environments. One company (C) got into a situation where they had to allocate all developers to projects to meet the needs of the market: the improvement focus was lost. Another company (A), which was a department in a large international organization, recently got the message that the top management had decided that they was about to be sold out of the parent company. Such events are destructive to a SPI initiative. It is easy to see this problem, but it is hard to deal with. The environment is often influenced by factors outside the company that are impossible to control. The best way to deal with such a situation is to be aware of it and to continuously analyze the situation and, if possible, take proper actions.

E. No visualization of the relationship between cost and benefit

SPI can be a large investment:

- occupies personnell
- takes time
- requires a focus beyond only software development
- interferes with projects

The motivation for performing SPI, even if it can be expensive, is that it will pay off in terms of increased competitiveness, improved product quality, increased competence etc. At the end it is the management that have to make the decision of taking the investment or not. This is really an analysis of cost and benefit. So, to make this analysis in the start-up phase and to follow up the investment it has to be possible to, in some ways, measure the effects of the SPI actions. This is not an easy task because the effects of improved processes does not necessarily directly relate to economical figures. It is therefore important that the management is aware of the value of such effects as increased competitiveness, improved product quality, increased competence etc. It is the SPI responsibles task to visualize this. If they fail to do so it may cause lack of confidence from the management, which again is an important prerequisite for succeeding in SPI.

One of our cases (B) is an example where we failed to visualize the planned effects. Without any warning we got the message that the top management wanted to release the internal SPI responsible from his task because they could not see that he had produced what they expected should be the result from the SPI activities. The management had expected a new process model documentation in the form of a printed process description that they could place on the bookshelf. The SPI responsible, on the other side, had been concentrated on establishing a motivation internally for the new methodology and was planning extensive experience harvesting sessions to build the new process description. This had not been visible to the management; instead they saw a large consumption of human resources without, in their eyes, any results; therefore their reaction.

This story shows the importance of being open about the process from the very beginning and to explain how the SPI activities eventually will lead to a benefit that is worth the investment.

6 The way ahead

Being experienced and having knowledge is of utmost importance to SPI. However, just as having knowledge about horses is not the same as an ability to ride, to improve software development processes, *knowledge, experience, and action are all mandatory*.

One of the insights from this investigation is that; while most of the software organizations in fact have knowledge of what to do in order to improve, they frequently fail in turning this knowledge into action. This is similar to the “performance paradox” observed by Cohen [13]: “Managers know what to do to improve performance, but actually ignore or act in contradiction to either their strongest instincts or to the data available to them.”

There are several possible explanations for this phenomenon. One is that the conception of knowledge as something explicit and quantifiable, leads organizations to overestimate the importance of the tangible, programmatic aspects of what competitors do, and underestimate the importance of the underlying philosophy that guides what they do and why they do it. Another reason for the observed gap between knowledge and action is that the most obvious solutions often tend to be discounted for the reason that they seem too ordinary. *The key lesson is, thus, to tackle obvious things*. Still another reason for the knowledge-action gap is that action inevitably involves a greater risk of failure than planning, decision-making, and talks. Furthermore, managers often believe that just because a decision has been made, something will happen. However, as Brunsson [14] has shown in his analysis of organizational hypocrisy, talk, decision, and action is often not connected.

There are no easy actions that can be used to close the gap between knowledge and action. However, our findings suggest that too many managers and developers alike fall into the trap of wanting to learn "how" in terms of tools and techniques, rather than "why" in terms of underlying philosophy and guidance for action. Also, since learning by doing, by definition, eliminates the knowledge-action gap, this suggests that *we should focus on theories that are grounded in industrial experience by developing practice-oriented theories*.

Also, evidence suggests that 70-80 percent of large-scale reengineering and quality improvement efforts do not produce the *desired* results ([13];[15];[16]). This suggests that rather than large-scale, complex, and abstract theories, we should develop SPI theories that are close in space and time to practice, focusing on generating short-term, incremental gains and small wins. Moreover, such approaches should be improvisational in nature, valuing an attitude of action and the importance of learning by experimentation [17]. To slightly rephrase Benjamin Disraeli, we should remember that: *action may not always bring improvement, but there is no improvement without action*.

A further implication of our experience is the realization that action research [18] is particularly well suited to advance the methodological basis for SPI. There are several reasons for this, but the most important is that action research is unique in the way it associates research and practice, so research informs practice and practice informs research synergistically. Furthermore, the action researcher is not a neutral observer, detached from practice. Rather, *the action researcher takes active part in the organization's change process*. This is also an important part of the industry's motivation to participate in SPI research projects like SPIQ and PROFIT – that the researchers actively participate with relevant knowledge and experience to help the organizations improve themselves.

Finally, results of empirical investigations in SPI have shown that organizational issues are as important in SPI as technology, if not more so [19]. This suggests that the large literature on organizational theory, developed in non-software settings, has more relevance to software development than has previously been recognized. So, while software engineering has been offered as a way of resolving the intrinsic technical problems, the organizational problems need another approach. However, *both classes of problems need to be jointly resolved to improve the process of software development* – they are co-producers of the outcome.

7 Conclusion

The software industry is not different other industries, there is always room for improvement. The rapid technical evolution the last years in such areas as internet technology and mobile systems etc. support this fact. Software development can be a complex process and improving it can be even more complex. SPI is in many cases a prerequisite for successful software development. There are numerous examples of SPI initiatives that have helped companies improve their processes resulting in more satisfied customers, deliverance on time and according to the planned budget etc. But there are also experiences in the other end of the scale, where SPI has failed - completely or partially.

One central doctrine in SPI is to repeat success and avoid failure. It is about time that the SPI community takes its own medicine. This paper shows that there are many tasks to solve. Our mixed experiences with planning improvement do indicate that we might have to think anew.

Software development is not manufacturing, it is not a mechanical process. On the contrary, software development is largely an intellectual and sociological activity, requiring the ability to deal with change above the ability to achieve stability [20]. Therefore, it's time we distance ourselves from the assumptions underlying the rationalistic, linear model of software engineering, and admit that reality for most small software organizations is a non-deterministic, multi-directional flux that involves constant negotiation and renegotiation among and between the social groups shaping the software. The sooner we admit this to ourselves, the sooner we can develop more useful models for improving the process of software development [17].

8 References

[1] Basili, V.R. and Caldiera, G. (1995) Improve Software Quality by Reusing Knowledge and Experience, Sloan Management Review, Vol. 37, No. 1, Autumn, pp. 55-64.

[2] Paulk, M.C., Weber, C.V., Curtis, B., and Chrissis, M.B. (1995) *The Capability Maturity Model: Guidelines for Improving the Software Process*. Reading, Massachusetts: Addison-Wesley.

[3] Royce, W.W. (1970) Managing the Development of Large Software Systems: Concepts and Techniques, *Proceedings of IEEE WESCON*, Los Angeles, pp. 1-9.

[4] Kruchten, P. (2000) *The Rational Unified Process: An Introduction*, Second Edition, Reading, Massachusetts: Addison-Wesley.

[5] Beck, K. (1999) *Extreme Programming Explained: Embrace Change*, Reading, Massachusetts: Addison-Wesly.

[6] Avison, D., Lau, F., Myers, M., and Nielsen, P.A. (1999) Action Research, *Communications of the ACM*, Vol. 42, No. 1, pp. 94-97.

[7] Reidar Conradi, “SPIQ: A Revised Agenda for Software Process Support”, In Carlo Montangero, editor, Proc. 4th European Workshop on Software Process Technology (EWSPT'96), pp. 36-41, Nancy, France, 9-11 Oct. 1996. Springer Verlag LNCS 1149.Dybå, T. (Ed.) (2000) *SPIQ – Software Process Improvement for better Quality: Methodology Handbook* (in Norwegian), IDI Report 2/2000, Trondheim, Norway: Norwegian University of Science and Technology.

[8] Dybå, T. (Ed.) (2000) *SPIQ – Software Process Improvement for better Quality: Methodology Handbook* (in Norwegian), IDI Report 2/2000, Trondheim, Norway: Norwegian University of Science and Technology.

[9] Stålhane, T, Wedde, K.J.: SPI - Why isn't it more used?, EuroSPI proceedings 1999

[10] Deming, W.E. (1986) *Out of the Crisis*. Cambridge, Massachusetts: MIT Center for Advanced Engineering Study

[11] http://www.cordis.lu/esprit/src/stessi.htm (accessed 2002-03-22).

[12] Scupin, R. (1997) The KJ Method: A Technique for Analyzing Data Derived from Japanese Ethnology, Human Organization, Vol. 56, No. 2, pp. 233-237.

[13] H.B. Cohen (1998) The Performance Paradox, *Academy of Management Executive*, Vol. 12, No. 3, pp. 30-40.

[14] N. Brunsson (1989) *The Organization of Hypocrisy: Talk, Decisions and Actions in Organizations*, New York: John Wiley & Sons.

[15] M. Hammer and J. Champy (1993) *Reengineering the Corporation: A Manifesto for Business Revolution*, New York: Harper Business.

[16] G. Morgan (1997) *Images of Organization*, Newbury Hill: Sage Publications.

[17] T. Dybå (2000) Improvisation in Small Software Organizations, *IEEE Software*, Vol. 17, No. 5, pp. 82-87.

[18] Avison, D., Lau, F., Myers, M., and Nielsen, P.A. (1999) Action Research, *Communications of the ACM*, Vol. 42, No. 1, pp. 94-97.

[19] T. Dybå (2001) *Enabling Software Process Improvement: An Investigation of the Importance of Organizational Issues*, Doctoral Dissertation, Norwegian University of Science and Technology, IDI Report 7/01.

[20] M.A. Ould (1996) CMM and ISO 9001, *Software Process – Improvement and Practice*, Vol. 2, pp. 281-289.

9 Author CVs

Geir Kjetil Hanssen

Geir Kjetil Hanssen (born 1969) is a Research Scientist in SINTEF Telecom and Informatics. He is a M.Sc. (Cand.Scient) from the Norwegian University of Science and Technology (1996). He works within the computer science area covering various aspects of information technology. System architecture is one of his focus areas, and he has experience in systems design and systems engineering methodology and processes. Other focus areas are transport telematics and process improvement where he assists several Norwegian software companies in improving their internal software development processes. He is working with messaging systems using XML based technology and a system for generating XML messages based on graphical message models (UML). He has experience from the IT industry, both as a systems engineer and developer and as a project leader.

Tore Dybå

received his M.Sc. degree in Computer Science and Telematics from the Norwegian Institute of Technology in 1986 and his Ph.D. degree in Computer Science from the Norwegian University of Science and Technology in 2001. He has worked as a consultant and a researcher both in Norway and in Saudi Arabia. His research interests include empirical software engineering, software process improvement, knowledge management, and organizational learning. Dr. Dybå is currently a senior scientist at Department of Computer Science at SINTEF and a visiting research scientist at Department of Software Engineering at the SIMULA Research Laboratory.

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. During 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 was back at SINTEF where he mainly worked with quality assurance and software safety and reliability. In 1997 he became professor in Computer Science at the Stavanger Polytechnic. In 2000 he became professor at the Norwegian University of Technology and Science in Trondheim where he today works full-time. During the latest decade he has been mainly been working with safety analyses of software intensive systems and measurement based process improvement.

Torgeir Dingsøyr

is a research scientist at the Department of Computer Science at SINTEF Trondheim, Norway. He wrote his doctoral thesis on “Knowledge Management in Medium-Sized Software Consulting Companies” at the Department of Computer and Information Science, Norwegian University of Science and Technology. He has published papers on knowledge management in software engineering, case-based reasoning and on software engineering education. In 2001, he spent half a year as a guest researcher at the Fraunhofer Institute for Experimental Software

Engineering in Kaiserslautern, Germany.

Hans Westerheim

Hans Westerheim (born in 1967) is a Research Scientist in SINTEF Telecom and Informatics. He is educated in Information Technology and Transportation Economy. He has worked with processes within companies both as consultant and researcher. The focus has always been “unstructured “sorts of processes, as found in the software industry and also within sales and marketing.

He participates in PROFIT and ARKTRANS as part of his work as researcher in SINTEF. The ARKTRANS project will develop a national architecture covering the multimodal area of transportation. As a support to this project he is managing smaller projects within the area of transportation. The focus is mobility and user interfaces. The PROFIT project is described in the paper.

Process Improvemenet the Hardware Way

Claus Trebbien-Nielsen
DELTA Danish Electronics, Light & Acoustics, Denmark
Kurt Frederichsen
Danfoss Drives, Graasten, Denmark

Abstract

This paper describes how a process improvement project for hardware development was performed using methodologies and experience from software process improvement, SPI, to address the improvement effort. The process areas to be improved were identified in an assessment based on a modified version of the BOOTSTRAP maturity model. The improvement effort was planned in a workshop with participation of representatives for all the improvement project's stakeholders. The actual process improvement was supported by a full time assigned project manager and practitioners, concurrently with running a product development project. The conclusion is that taking into account that hardware do have a lot more stakeholder to their processes than software and it takes more time to handle that, SPI transfers well to hardware.

Keywords

Reusing experiences, Assessment, SPI, Hardware process improvement, BOOTSTRAP

1 Introduction

Danfoss Drives A/S is one of the world's largest manufacturers of frequency converters. Danfoss Drives A/S develops, manufactures and markets frequency converters for infinite speed control of electric motors.

For several years Danfoss Drives has successfully been involved in software process improvement, SPI, based on maturity models. The results obtained in software have moved the bottleneck in the development process from software and has now put hardware into focus in terms of quality and adherence to plans. It was therefore decided to use the experiences from SPI to improve the performance of the hardware development team.

The process improvement project has used the concept of a framework of development processes as the basis for the product development effort and a modified version of the BOOTSTRAP maturity model to identify which process areas could most efficient and effectively improve the overall performance of the development team.

The process improvement project has used the concept of process based product development, i.e. product development is based on a set of processes, each focused on solving a specific task in the development life cycle. The feasibility of the existing processes were assessed with a modified version of the BOOTSTRAP maturity model, in order to identify the processes where improvement most efficient and effectively could support reaching the goals of the organisation.

2 Initiating the Project.

Prior to setting up the project, the management team has thoroughly analysed the possibilities for this new improvement project. A maturity model for hardware developed by DELTA, based on the software BOOTSTRAP maturity model, was evaluated by a number of persons in the team, before it was decided that this would indeed be a feasible way to reach the goals for the improvement project.

The main goals for the project were to:

- Reduce the development cycle.
- Improve the final product quality.

These two goals were identified to be essential in order to fulfil the company vision to be the best to meet the customers' expectation for electric motor drives at the best price-performance ratio. This meant being able to react fastest to new trends in the market and to reduce the cost of quality due to the increased economical consequence of each defect. This increase was due to reuse of components across many different products, the increased numbers of products manufactured and the extensive global distribution of these products.

The project manager from the SPI project was appointed project manager for this new project for two reasons. He had a good reputation in the organisation as a SPI project manager and he had a formal hardware background that would allow him to communicate well with the hardware development team. It was also decided to follow the same process improvement strategy that has been used successfully for software. This strategy was to dedicate resources for setting long term visions and goals, building up knowledge and running the process improvement. To implement improvements only when there is a positive need for them in running product development projects and to involve developers as much as possible, letting them decide as many "how to's" as possible, aiming more at having something useful for the project rather than having it 100% correct.

3 Identifying the Where to Improve

In order to identify which processes to improve, an assessment based on DELTA´s hardware adaptation of the BOOTSTRAP model was performed. The assessment involved both the management team and several practitioners from a number of different development projects.

Two assessors from DELTA performed the assessment. Prior to the assessment the concept of development based on processes, the model to be used and the assessment method was presented to the people in the hardware group.

The results of the assessment could quite easily be established and presented for the participants. Based on the assessment findings, the management team selected a number of processes as candidates for process improvement and these candidates were communicated to the people in the organisation. The candidates were:

- Test.
- Version and Change Management.
- Quality Management.

4 Creating a Plan for the Improvement Effort

The main prerequisite for a successful improvement project is the buy-in from all the stakeholders in the improvement project and the correct staffing of the project.

Originally, the attitude towards the improvement project in the hardware development team was that this might be necessary and an interesting challenge, but there were a lot of other necessary and interesting challenges to be involved in, too.

Therefore, it was decided, in accordance with the strategy laid out for the project, to use a workshop with participation from all the stakeholders as the vehicle for the planning process. The goals were to train the participants in the concept of process improvement, to obtain their buy-in to the project and during the workshop to identify the required profile for the staffing of the project.

The workshop had four phases answering the following questions:

- Why are these processes not working as desired?
- What can we do to resolve the problems?
- What are the consequences and what will we do?
- Who, when and how will it be done?

The time in the workshop was quite limited and the aim was therefore to establish a strategic plan for the improvement project including:

- Description of the process improvement goals, long terms as well as short terms and their relation to the strategic goals of the organisation.
- The objectives of the project including the motivation for the organisation to launch the project and the consequences of maintaining status quo.
- The assumptions and risks associated with the project including barriers for acceptance of changes in the organisation.
- The organisation of the project including required infrastructures, responsibilities of individuals and relation to other improvement activities.
- The success criteria including the metrics to be used to identify the success.

- Brief work break down structure and relations to other parts of the organisation being affected by the changes.

After the workshop the project manager elaborated this plan further into an action plan for the project together with management and his project team. This was established based on the staff profile identified in the workshop in order to ensure the project manager a good network into the organisation.

5 Carrying Through the Plan – Lessons Learned

While carrying through the plan it became clear that although the experiences from SPI were quite helpful, there were some significant differences between process improvement for software and hardware.

From an overall point of view, one can say the software development to a high degree is characterised by use of processes and abstractions with few if any physical limitations, whereas hardware is based on well established phases and modelling with rigid physical limitations. These differences most likely account for the different attitude towards making changes to processes or product, as hardware people associate making changes with a lot of unpleasant rework. Furthermore, few people in the organisation have a good understanding of software, whereas the opposite is the case for hardware and a lot of people therefore try to influence the hardware process improvement.

The experiences from the Test improvement area show that the concept of dividing tests into various levels of tests: unit, module, etc, is fairly familiar for hardware developers. They do therefore not find it difficult to make test specifications and plans for the individual test as it is done for software. The main difference is that while you in software can rely on that a component that is tested without showing any errors, will continue showing no errors when tested after it has been “manufactured” into the final product. This is not true for hardware, where e.g. the manufacturing process can have a significant effect on the amount and type of errors found in the final product. The scope of testing in hardware development therefore has to be extended deep into the manufacturing and in-use area, to ensure that the final product is robust enough to “survive” the manufacturing processes and the operation in the final working environment.

The experiences from improvement in the area of Version and Change Management also demonstrated that the solutions that are feasible for software can be used for hardware, but that some significant differences in implementing them exist between hardware and software. This difference is due to the significant increase in the number of stakeholders to the process being targeted for improvement. When working with software, relatively few people outside the software team need to be consulted about the impact of the changes. For hardware the picture is quite different. Hardware has a lot of working relations to e.g. supporting functions like ECAD and logistics, production and service. Changes to the Version and Change Management process therefore have to be negotiated with these professional groups, because they all rely on a common access to and use of design drawing, bill of material and similar artefacts. Negotiating with all of these groups was quite time consuming and very often the empowerment of the process improvement manager was challenged by these outside groups. The solution to this problem was to form a core team of the relevant managers. The objectives for this team was to be the ambassadors for the improvement project and make it clear that the decision made by the improvement team members, were supported at all levels of management.

The experiences from improvement in the area of Quality Management also showed that it is not just a question of improving the hardware development processes alone. The solutions from software can be used, e.g. analysing error reports, review, etc. However, implementing them is different. Purchasing and manufacturing need to be part of the Quality Management to obtain the desired result on the bottom line. Analysing error reports from the market, software mainly reveals errors related to the development process, whereas many of the errors related to hardware could be traced back to the purchasing or manufacturing process instead of the development process. This problem was solved introducing a more formal review process, having participants with different professional background and by placing the improvement project manager in a cross-functional quality group with the product project manager as participant, focusing more broadly on the processes involved in quality assurance and not only on the hardware processes.

6 Conclusion

Summarising the lessons learned, we can say that most of the solution to improving the processes can be reused from software, but implementing them is more time consuming and difficult due to the many stakeholders to the hardware development process. All of these have to be nursed and negotiated with before an improved process can be implemented in the organisation.

The process focus and the use of a maturity model for benchmarking the development processes seem to transfer well into hardware domain. In the beginning it seemed a little strange to the hardware team, but after having had time to become familiar with the idea, they did not have any problem in relating to it and using it in their improvement work.

The attitude of the participants to process improvement changed positively after the improvement workshop. After the workshop the team members started to motivate each other for the urgency to change the development processes and giving support to the project manager when somebody tried to question the project and its value.

The main drivers for the success of the project were:

- The full support from the management team.
- The skilled project manager with a good network in the organisation.
- The use of a workshop as kick-off for the project,
- The strategy to link the improvement project to a real product development project,

The strategy to empower the improvement team members to go for a practical solution more than a 100% correct solution.

7 Literature

[1] Bootstrap - World Wide Web: www.bootstrap-institute.com

8 Author and company CVs

Kurt S. Frederichsen – Danfoss Drives

He has a B.Sc.E.E. from the Engineering College of Aarhus. Kurt Frederichsen is technology leader for Quality Assurance methodologies and Process Improvement for both Hardware and Software Technology Centers in Danfoss Drives A/S.

He started in the Danfoss Postgraduate Rotational Programme in 1992, working with everything from HW and SW development for embedded systems to customer communication in a marketing department. In 1994 he started as a SW developer at Danfoss Drives A/S, but soon found interest in the ideas on process improvement and took the task in 1995 of starting a SW-QA group to build up knowledge and handle long-term process improvement. Based on a BOOTSTRAP assessment in 1996, implementation of basic management of the development process and quality assurance a couple of years later resulted in SW no longer was the key problem in product development. In 1998-2000 he took part in a cross division SW process improvement initiative, StepUp, run by the Danfoss group, where the primary outcome was an intranet based "Software Process Assistant" - a tool for user oriented process definition in each division.
The success with process improvement in SW, started the idea of spreading the good things experienced to other areas. In 2001 he took the task of starting a similar function for HW as he did in SW. HW process improvement was a reality!

He can be contacted at Danfoss Drives, Ulsnæs 1, DK-6300 Graasten, Denmark, phone +45 74 88 37 36, e-mail: ksf@danfoss.com, World Wide Web: http://www.danfoss.com/drives

Claus Trebbien-Nielsen

Claus Trebbien-Nielsen holds a M. Sc. in chemistry and computer science from University of Copenhagen and a bachelor degree in international business management and marketing from Copenhagen Business School.

He has been working with R&D for several years. First in the area of safety critical low volume equipment for surveillance of critical ill patients in the operation theatre and intensive care units of a hospital. Next for PHILIPS and NOKIA being involved in development of high volume, consumer product in a very competitive market, as well as in process improvement management. He is currently working for DELTA in the area of organisational change management, maturity models and integrated product development.

He is formally trained as ISO 9000 and TickIt auditor, and a SPICE, BOOTSTRAP and EFQM assessor.

He can be contacted at DELTA Danish Electronics, Light & Acoustics, Venlighedsvej 4, DK 2970 Hørsholm, Denmark, phone +45 72 19 42 62, fax +45 72 19 40 01, e-mail: ctn@delta.dk, World Wide Web: http://www.delta.dk

Danfoss Drives A/S

Danfoss Drives A/S is a division in the Danfoss group. Danfoss Drives A/S is one of the world's largest manufacturers of frequency converters. Danfoss Drives A/S develops, manufactures and markets frequency converters for infinite speed control of electric motors in three places in the world: Graasten in Denmark, Illinois and Wisconsin in USA. Danfoss Drives A/S is one of Denmark's largest electronic companies with an annual turnover of approx. 250 mio. $ and more than 1800 employees world-wide.

In additional to meeting the ISO9001 quality and the ISO14001 environmental assurance standards, Danfoss drives are designed to meet all relevant product standards including UL and CSA standards.

The mission for Danfoss Drives A/S is to be known and recognised as the specialist within frequency converters.

Danfoss Drives A/S has high attention on research and development of new technologies to ensure that the products are at the forefront of the market. New methodologies for development, modelling and simulation of the drive system is highly prioritised in:

- Design of power and control electronics
- Design of enclosure
- Development of control algorithms
- Software design
- Design for optimal EMC e.g. RFI filters
- Analysis of the drive system and the interaction with supply, motor, application and user.

The headquarter is located at: Ulsnaes 1, DK 6300 Graasten, Denmark, Phone +45 74 88 22 22, Fax +45 74 65 25 80, World Wide Web: www.danfoss.com/drives

DELTA

DELTA Danish Electronics, Light & Acoustics is an independent self-governing knowledge centre for advanced technology, affiliated to the Danish Academy of Technical Sciences (ATV). DELTA employs approx. 280 people in 5 divisions (Electronics Testing, Software Engineering, Microelectronics, Light & Optics, and Acoustics & Vibration) providing services for every aspect of product development. The turnover is EUR 28 millions.

DELTA's objectives as an ATV institute is to transform knowledge and experiences from both industry and universities to products and services that can be provided to the industry e.g. as consultancy support. DELTA Software Engineering is a division within DELTA with a Software Technology department. DELTA has a long-standing commitment to software engineering in general and specifically in the area of software best practice.

DELTA Software Technology employs 10+ consultants, all with a high level of experience within software engineering and software test directly involved in software best practices, offering consultancy to companies or teaching courses via our extensive course department (more than 80 courses a year).

In addition to this DELTA has a responsibility for disseminating knowledge within the field of software engineering. The Danish SPIN group is an experience exchange group within Software Technology Forum, which has existed for 20 years with more than 70 members from Danish software industry. The Danish SPIN group meets 3 – 6 times per year and is chaired by DELTA as are all the other experience exchange groups within Software Technology Forum.

DELTA Danish Electronics, Light & Acoustics, is located in a number of locations in Denmark, Norway, Sweden, and England, the headquarter is located at: Venlighedsvej 4, DK 2970 Hørsholm, Denmark, Phone +45 72 19 40 00, Fax +45 72 19 40 01 World Wide Web: www.delta.dk

Fritz Design GmbH

Eur SPI
European Software Process Improvement

EuroSPI
c/o ISCN Coordination Office
Florence House, 1 Florence Villas
Bray, Co. Wicklow, Ireland
Tel. +3 53-1-2 05-00 20
Fax +3 53-1-2 05-00 21
office@iscn.com · www.eurospi.net

ISBN 3-00-010074-1