EuroSPI² 2013 Proceedings

Proceedings

The papers in this book comprise the industrial proceedings of the EuroSPI² 2013 conference. They reflect the authors' opinions and, in the interests of timely dissemination, are published as presented and without change. Their inclusion in this publication does not necessarily constitute endorsement by EuroSPI² and the publisher.

EuroSPI²

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

The EuroSPI² conference presents and discusses results from **systems, software and services process improvement and innovation (SPI)** projects in industry and research, focusing on the gained benefits and the criteria for success. This year's event is the 19th of a series of conferences to which international researchers and professionals contribute their lessons learned and share their knowledge as they work towards the next higher level of software management professionalism.

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

The Dundalk Institute of Technology, Ireland, is the host of the EuroSPI² 2013 conference. Dundalk Institute of Technology is currently leading research projects to develop and trial MediSPICE implementing medical device standards industry. Thus in 2013 key notes come from leading medical industry and also an international risk management workshop is organised moderated by Dundalk IT.

EuroSPI² 20th Anniversary

In 2013 we celebrate the 20th anniversary of EuroSPI²!. For that reason we organise an extra conference day involving key notes from the past 20 years these key notes saw and influenced EuroSPI² in the past and give a vision for the next 20 years of SPI.

EuroSPI² Chairs

General & Workshop Chair	Richard Messnarz, ISCN, Austria/Ireland
EuroSPI² Marketing Chair	Miklós Biró, Software Competence Center Hagenberg (SCCH), Austria
Scientific Programme Chair	Rory O' Connor, Dublin City University, Ireland
Scientific Programme & Local Chair	Fergal Mc Caffery, Dundalk Institute of Technology, Ireland
Industrial Programme Chair	Jørn Johansen, DELTA, Denmark
Industrial Programme Chair	Risto Nevalainen FiSMA and STTF, Finland

Industrial Programme Chair	Stephan Goericke, iSQI, Germany
Industrial Programme Chair	Mads Christiansen, DELTA, Denmark
Industrial Programme Chair	Nils Brede Moe, SINTEF, Norway
Organizing Chair	Adrienne Clarke, ISCN, Ireland
Co-Organizing Chair	Eva Christof, ISCN Austria
Local Organizer	Fergal Mc Caffery, Dundalk Institute of Technology, Ireland

Industrial Programme Committee

Bachmann Volker	SIBAC GmbH	Germany
Bæk Jørgensen Jens	Mjølner Informatics A/S	Denmark
Breske Eva	Robert Bosch GmbH	Germany
Christiansen Mads	DELTA	Denmark
Daughtrey Taz	James Madison University	USA
Dekkers Carol	Quality Plus Technologies	USA
Dussa-Zieger Klaudia	Method Park Software AG	Germany
Ekert Damjan	ISCN GmbH	Austria
Fehrer Detlef	SICK AG	Germany
Fenz Andrea	Skills International GmbH	Austria
Forselius Pekka	4SUM Partners	Finland
Georicke Stephan	iSQI GmbH	Germany
Hagenmeyer Philipp	ZF Friedrichshafen AG	Germany
Hallikas Jarmo	Falcon Leader Oy	Finland
Hällmayer Frank	Software Factory GmbH	Germany
Hind Tim	AXA	UK
İbrişim Ayşegül	Turkish Standards Institution	Turkey
Johansen Jørn	DELTA Axiom	Denmark
Johansson Mika	FiSMA	Finland
König Frank	ZF Friedrichshafen AG	Germany
Messnarz Richard	ISCN Ltd	Austria/Ireland
Morgenstern Jens	-	Germany
Nevalainen Risto	FiSMA	Finland
O'Leary Eugene	EQN Ltd.	Ireland
Poth Alexander	Volkswagen AG	Germany
Renault Samuel	Centre de Recherche Public Henri Tudor	Luxembourg
Romcea Cristina	Conti Temic microelectronic GmbH	Germany
Sauberer Gabriele	TermNet	Austria
Schweigert Tomas	SQS Software Quality Systems AG	Germany
Sechser Bernhard	Method Park	Germany
Spork Gunther	Magna Powertrain	Austria

Stefanova-Pavlova Maria	Center for Innovation and Technology Transfer-Global	Bulgaria
Von Bronk Peter	Systemberatung Software-Qualität	Germany
Wegner Thomas	ZF Friedrichshafen AG	Germany

EuroSPI² Board Members

DELTA, http://www.delta.dk

FiSMA, http://www.fisma.fi

ISCN, http://www.iscn.com

iSQI, http://www.isqi.de

SINTEF, http://www.sintef.no

ASQ SW Division http://www.asq.org

Editors of Proceedings

Richard Messnarz, ISCN, Austria/Ireland

Damjan Ekert, ISCN, Austria

Jørn Johansen, DELTA, Denmark

Eva Christof, ISCN, Austria

Supporting Partners

Dundalk Institute of Technology, http://www.dkit.ie

European Certification and Qualification Association, http://www.ecqa.org

Welcome Address by the EuroSPI² General Chair

Richard Messnarz

ISCN, Austria/Ireland

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

1. An annual EuroSPI² conference supported by Software Process Improvement Networks from different European countries.

2. EuroSPI² supported the establishment of a world-wide SPI Manifesto (SPI = Systems, Software and Services Process Improvement) with SPI values and principles agreed among experts world-wide. We build clusters of experts and knowledge libraries for these values and principles.

3. Establishing a web-based experience library based on hundreds of experience reports contributed to EuroSPI² since 1994 and which is continuously extended over the years and is made available to conference attendees.

4. Establishing a European Qualification Framework for a pool of professions related with SPI and management. This is supported by Europe-wide certification for qualifications in the SPI area, exam systems, and online training platforms (European Certification and Qualification Association, www.ecqa.org).

5. Establishing a world-wide newsletter with articles from key industry and European key research associations helping to implement the SPI manifesto world-wide (newsletter.eurospi.net).

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

EuroSPI² conferences present and discuss results from **systems, software and services process improvement (SPI)** projects in industry and research, focussing on the benefits gained and the criteria for success. This year's event is the 20th of a series of conferences to which international researchers contribute their lessons learned and share their knowledge as they work towards the next higher level of software management professionalism.

A typical characterization of EuroSPI² was stated by a company using the following words:

> *"... the biggest value of EuroSPI² lies in its function as a European knowledge and experience exchange mechanism for SPI and innovation".*

A cluster of European projects (supporting ECQA and EuroSPI²) contribute knowledge to the initiative, including currently SafEUr (ECQA Certified Safety Manager), SIMS (ECQA Certified Social Media Expert), VALO (ECQA Certified Valorisation Manager), BPM- HEI (BPM for Higher Education), AQUA (Knowledge Alliance for Training Quality and Excellence in Automotive), LSSH (Lean Six Sigma for Health Care), Idea 2 Enterprise (i2E). A pool of more than 30 qualifications has been set up (see www.ecqa.org).

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

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

Welcome by DELTA, Editors of the DELTA Improvement Series

Jørn Johansen

DELTA, Denmark

DELTA has been working with Software Process Improvement (SPI) for more than 18 years including maturity assessment according to BOOTSTRAP, SPICE and CMMI. DELTA has also been a partner in the EuroSPI² conference from the very beginning 18 years ago. We are now for the 5th time the publisher of the Industrial Proceedings from EuroSPI² making it part of the DELTA series about Process Improvement.

Jørn Johansen is Senior Technology Specialist of at DELTA. He has an M.Sc.E.E. from Ålborg University and more than 34 years experience in IT. He has worked in a Danish company with embedded and application software as a Developer and Project Manager for 18 years. Mr. Johansen has been involved in all aspects of software development: specification, analysis, design, coding, and quality assurance. Furthermore he has been involved in the company's implementation of an ISO 9001 Quality System and was educated to and functioned as Internal Auditor.

For the last 20 years he has worked at DELTA as a consultant and registered BOOTSTRAP, ISO 15504 Lead Assessor, CMMI Assessor and *ImprovAbility™* Assessor. He has participated in more than 100 assessments in Denmark and abroad for companies of all sizes. He was the Project Manager in the Danish Centre for Software Process Improvement project, a more than 25 person-year SPI project and Talent@IT, a 26 person-year project that involves 4 companies as well as the IT University in Copenhagen and DELTA. Latest Mr. Johansen was the Project Manager of SourceIT an 18 person-year project focusing on outsourcing and maturity. Mr. Johansen is also the co-ordinator of a Danish knowledge exchange group: Improving the Software Development Process, which is the Danish SPIN-group. At the moment Mr. Johansen is lead editor on ISO/IEC 33014 Guide for process improvement.

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

Welcome from the Local Organization and Scientific Programme Committee Chair in Ireland

Fergal Mc Caffery

Dundalk Institute of Technology, Ireland

Welcome to the 20th EuroSPI[2] Conference in Ireland at the Dundalk Institute of Technology.

Dundalk Institute of Technology is situated between Dublin and Belfast. The Institute was one of the original network of Regional Technical Colleges set up in Ireland in the 1970s with an emphasis on business, engineering and science. The institute's research is conducted through its various research centres and groups whom are embedded across the institute's four academic schools. The Institute has both academic and research centres undertaking research in Ageing, Health, Software Engineering, Energy, the Environment, Music and Humanities, Software Regulation and Social Sciences.

The Regulated Software Research Centre (RSRC) is focused upon medical device software engineering, with a particular focus upon medical device software processes. One of the main projects within the RSRC is the development of an international software process improvement framework (Medi SPICE) for the medical device industry as a key enabler of best practice for the sector. This work involves the RSRC working closely with the international medical device standards community, the international software process improvement community and the medical device software industry. One if the key objectives of the RSRC is to help assist the growth of the Irish medical device software industry.

The RSRC is part of Lero – the Irish Software Engineering Research Centre.

Dr Fergal Mc Caffery is the local chair for the EuroSPI 2013 Conference and is the Director of the RSRC. He has been awarded SFI funding through the Stokes Lectureship, Principal Investigator and CSET Programmes to research the area of medical device software. Additionally, he has received EU FP7 research funding to improve the effectiveness of embedded software development environments for the medical device industry. He also has received Enterprise Ireland Commercialisation funding for a number of different projects.
He has published over 150 peer-reviewed conference and journal papers and is on the editorial board/programme committee for a number of leading software engineering conferences and journals. Additionally, he represents Ireland at International medical device software standards meetings and is an active member of the IEC SC62A JWG3 working group that is responsible for the International standard for medical device software lifecycle processes (IEC 62304) and also the IEC SC62A JWG7 working group that is responsible for the development of a new International standard for Healthcare Software (IEC 82304). He is also a member of the ISO SC7 WG10 working group that is responsible for the International Standard for Software Process Assessment (ISO/IEC 15504).

Contact Details:
Fergal Mc Caffery (E-Mail: fergal.mccaffery@dkit.ie)

Welcome from the ECQA President

Michael Reiner

ECQA, Austria

The European Certification and Qualification Association (ECQA) is a not-for-profit association that aims to unify the certification processes for various professions in Europe. It is joining together institutions and thousands of professionals from all over Europe as well as worldwide and offers the certification to participants for numerous professions. Currently, 27+ professions are active and some new professions are being developed right now. ECQA services are being offered in 24 countries across Europe by more than 60 ECQA members. With the help of Ambassadors the ECQA is also enhancing its activities by expanding to all over the world (e.g. USA, China, Thailand, India, Singapore, Japan etc.).

The main objective of the ECQA is to develop and maintain a set of quality criteria and common certification rules across the different regions. Therefore the ECQA ensures that the same knowledge is presented to participants across Europe and all participants are tested according to the same requirements. The knowledge to be provided and tested for certain professions is defined by experts from industry and research, who know best what the requirements of the market are and what the state of the art knowledge is within certain domains. These experts work in ECQA groups called Job Role Committees. The EQCA coordinates their work and provides the infrastructure and IT support.

The ECQA has developed a set of quality criteria, which are used for the certification of the following types of service providers: trainers, training organizations, exam organizations, and certification organizations. The aim is to ensure the same level of training and certification quality in all participating countries.

Working today means cooperating with a lot of international partners. Thus the understanding on both sides is essential. Certifications can help to better understand the different views of different professions. Out of this the ECQA Job roles aim at core competences in networking and understanding as well as concentrate on the needs of the industry.

Michael Reiner, president of the ECQA and lecturer for Business Administration and E-Business Management at the IMC University of Applied Sciences Krems, has several years of experience in the field of IT, Microsoft Office, Microsoft NAV (ERP), Knowledge Management, Business Intelligence, Web 2.0 and social networks. Moreover Mr. Reiner is member of the Microsoft Dynamics Academics Advisory Board and coordinates and participates in various EU projects.

In the last 2 years ECQA has developed towards an international certifier issuing certificates and establishing partnerships in all European countries as well as in India, South America, China, Japan and Arabia.

I wish you a good time at the EuroSPI² 2013, a lot of interesting networking partners and informatory meetings.

Contact: Michael Reiner, IMC University of Applied Sciences Krems, Austria, e-mail: michael.reiner@fh-krems.ac.at

Table of Contents

Experience Session 9: Traceability

Experience Session 10: Medical Systems I

Experience Session 11: Medical Systems II

Experience Session 12: Invited Key Topics

Managing Mobile Application Quality through an Agile Software Process

Phelim Dowling, Kevin McGrath, Gemma Power
TSSG, Waterford Institute of Technology
Carriganore, Waterford, Ireland
{pdowling ,kmcgrath, gpower}@tssg.org
http://www.tssg.org

Abstract

The prevalence of smartphones and tablet computers in everyday life is emphatic. The popularity and pervasiveness of these devices and the spectrum of applications they carry has seen mobile services cross the line from not being simply communication tools but rather the backbone of modern day living. Their growth is inextricably linked to social, economic and cultural developments. The adoption of these applications into the well understood processes and practices of software creation teams has evolved. In certain aspects the development and testing of these apps follow the standard rules for software development methods and frameworks such as Agile. In other ways they expose subtle but significant differences. This paper will examine the challenges faced by one organisation as they ensured maximum quality for their apps under the constraints of a limited budget.

Keywords

Mobile, Smartphone, Smart device, Tablet, Applications, Software Process, Testing, Agile, Verification and Validation,

1 Introduction

The International Telecommunication Union (ITU) statistics paint the picture quite clearly [1]. There are 5.9 billion mobile-cellular subscriptions worldwide - a staggering 87% global penetration. Mobile broadband subscriptions have risen by 45% annually over the last 4 years and now stand at double the number of fixed broadband subscriptions. There are 1.2 billion active mobile broadband subscribers worldwide. These figures not only confirm the existing worldwide popularity and demand for connectivity and services while on the move, they also outline the potential growth in the coming years. Just as the traditional mobile device has reached universal reliance so too the smart device will attain similar status [2]. There are still large areas in the world primed for mobile application penetration, e.g. developing economies and nations [3]. The advent of 4G capabilities will further enhance the features and benefits these applications and devices provide [4].

The opportunity for software quality exponents is evident. A massive industry of smartphone application creators and consumers already exists. The demand and up-take for these services will grow rapidly in the next few years. The delivery expectations by customers appear far more aggressive for mobile applications and so the Agile methodology is an attractive option to pursue [5].

The Telecommunications Software & Systems Group [6] has long engaged in cutting edge research

and development in technologies enabling communications and information services. Sustained through winning competitive funding at national and EU levels and driven by a staff of research engineers with proven experience and expertise, the organisation has numerous successful commercialisation spin-out successes to its name [7]. The areas of research that the group engage in include:

- Mobile Platforms and Services
- Data Analytics and Social Computing
- Security, Privacy and Identity
- Adaptive Networks and Services

All of these research units have been involved in developing applications for smartphones and tablets many of which have made it to commercialisation and market. The TSSG's Experimental Facilities Management (EFM) group supports projects in all of these units [8]. The EFM team bring a skillset and expertise in Agile software process and testing (see Table 1 below) that facilitates the smooth running and successful outcome for these projects and products:

Process	Techniques	Testing
Requirements Gathering	Test Driven Development	Functional
Project Planning	Continuous Integration	System
Agile and Scrum Methodologies	Iterative Work Cycles	Integration
Risk Management	Pair Programming	Automated
Defect Management	Code Reviews	Load/Performance
Release Management	Refactoring	Usability

Table 1: EFM Agile services and tools applied to mobile projects in TSSG

The EFM groups experience in managing mobile applications through an Agile lifecycle in a budget conscious environment is explored in the subsequent sections of this paper.

2 Mobile Project Kick-off

2.1 Requirements Gathering

The TSSG have been responsible for the successful completion of many projects requiring a mobile verification and validation stage. Over time the EFM group have developed effective agile processes to deal efficiently with the requirements gathering stage. How will end-users use the app? Which functions must be central? If user requirements are poorly specified it can prove to be a significant factor behind project failure.

For example, in the case of the PERIMETER project [9], implementing an end user based scenario at the outset aided the consortia greatly in gathering the requirements for the project and determining the testbed infrastructure that would be needed in order to demonstrate the capabilities of the application efficiently. This was largely achieved by breaking down each detailed user scenario into sub scenarios and consequently into a component and functionality level. On other projects the main actors associated with the application were identified, their goals articulated, use cases created and finally a list of user stories was generated as a basis for the project work. Redmine [10] is the tool TSSG use for

managing projects and this includes capturing all the user stories in a backlog before assigning them to iterations.

To avoid over-reliance on internally generated requirements i.e. what is believed the app users want as opposed to information from end-user requirements studies and what users actually need, bringing key user data requirements to the early planning stages is important. The Design and Usability Team in TSSG is experienced in a range of user-focused requirements gathering and analysis techniques. Appropriate methodology is selected to provide a clear understanding of user requirements to develop applications and to formulate usability goals. These can include focus groups, interviews, question-naires, card sorting, evaluation of an existing application or creating usage scenarios. Such methodol-ogy is also useful in supporting early user interface design work e.g. wireframing, which in turn can also help in clarifying important functionality early and creating a clearer application roadmap.

One of the most important pieces of information to determine at this point if not already understood is to have a clear requirement from the customer of where the app is intended to work in terms of smart-phones and/or tablets, platform, OS version, etc.

2.2   Build Setup

The build setup for mobile applications remains largely the same as for any application just keeping in mind the memory and CPU constraints. The complete test cycle is depicted below in Figure 1 and as can be seen it is an iterative process.
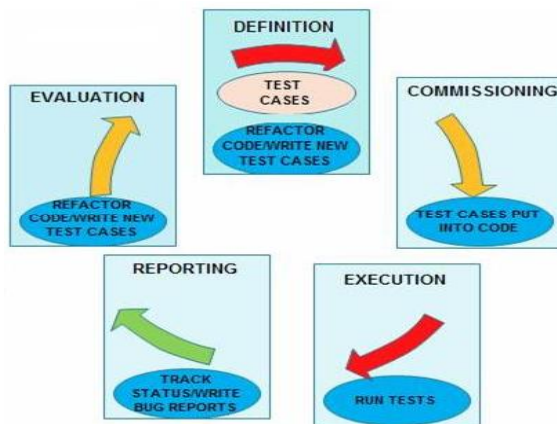
Figure 1: Build and unit test case setup

1. **Definition:** agreement of scope and functionality to be tested.
2. **Commissioning:** setting up the build and test environment.
3. **Execution:** performing the tests on the testbed at build time.
4. **Reporting:** recording test results and communicating these to the interested partners.
5. **Evaluating:** dissemination of the test results and taking the appropriate actions

In order to support the seamless continuous integration of projects the team employs tools such as Hudson [11] and Jenkins [12] for building and testing automatically, monitoring the Subversion [13] or Git [14] repository for freshly committed code as well as providing reports on the status of the builds, tests and code coverage. For unit testing, project teams have employed JUnit for Android builds with EMMA [15] used to measure code coverage. The built in testing tools in Xcode [16] have been used for iOS – there are Jenkins Xcode and Cobertura [17] plugins to set this up. Automated acceptance

testing has been setup using Frank [18] and Cucumber [19].

Build distribution is handled in a number of ways. For iOS builds the TestFlight app [20] (described in section 5) is widely used. For other less complicated deploys we have used Dropbox [21] - a shared secure cloud site where project members can easily push the tested app to their preferred mobile device. While the build tool will keep a history of all apps built, the EFM test team will maintain a matrix of where app builds are tested, i.e. on which devices, OS versions, etc. This is often hosted on the project Wiki page.

3   Test Case Creation and Execution

A set of test cases to be executed against each application is generated by the EFM team. The full suite of test cases can be executed against the devices as defined in Section 4. These test cases are based on four primary aspects or layers of application behaviour that allow us to validate the application quality: generic application tests, specific functional tests, on device performance and load tests.

3.1   Generic Happy Path for Mobile Applications

The steps outlined in this section are a guide to testing any apps created by the development team and in fact could be applied to any app on any platform. The flow outlined below describes a set of high level tests that are designed to analyse and verify general application behaviour and usability across varying device sizes, platforms, network connectivity, etc.

Install/Uninstall
- Install the app on the test device
- Ensure install and sync execute as defined
- Ensure app launches successfully following install
- Uninstall the app from the test device
- Ensure app uninstalls as defined
- Re-install app to continue testing
- If this is an upgrade, test this scenario also.

Testing App Startup
- Ensure app icon/badge is correct and looks ok on the device
- Launch the app
- Confirm correct loading screen is displayed, e.g. 'powered by TSSG'
- Verify that the expected startpage is loaded
- Monitor startup performance and execution

Device Date & Time
- Test the app with the phone at various time settings
- 12hr clock v 24hr clock
- Set the time to a day, week, month, year in advance
- Pay close attention to any date or time dependant features with these settings in place.

Device Rotation
- Test the app with the device in portrait orientation
- Test the app with the device in landscape orientation
- Execute functional tests in both modes and ensure transition between modes is seamless.

In App Advertisements
- Check with the developer if ad bars, etc. are intended for this app and what screens they should appear on.
- Verify that ads display on the expected screens
- Verify this from various different areas/functions of the app, e.g. homepage, in a story, in a map, etc.

Device Idle
- Launch the app and let the device idle and auto lock
- Confirm app is still open and behaves as expected after you unlock the device
- Verify this from various different areas/functions of the app, e.g. homepage, in a story, in a map, etc.

Network Tests
- Confirm network unavailability is handled appropriately, e.g. friendly and clear error to user
- Test various network connectivity options, i.e. wifi, 3G, etc.
- Test app behaviour when moving between wifi and 3G

Backend Monitoring
- Execute regular backend checks during app testing
- Log file monitoring, i.e. check for any unusual events in appropriate logs
- Check for error alerts, e.g. using nagios monitoring
- System vital stats, e.g. monitor cpu, disk usage, etc.

Figure 2: Generic test cases to be applied to any mobile application

3.2   Functional testing specific to the application

This part of the test specification covers testing the specific functionality of the app. Each individual apps functionality can be determined on an app by app basis through consultation with the developer and reviewing the appropriate documentation, e.g. user guide, requirements spec, etc. The requirements (discussed in Section 2) are kept to the forefront in this process. This allows an assessment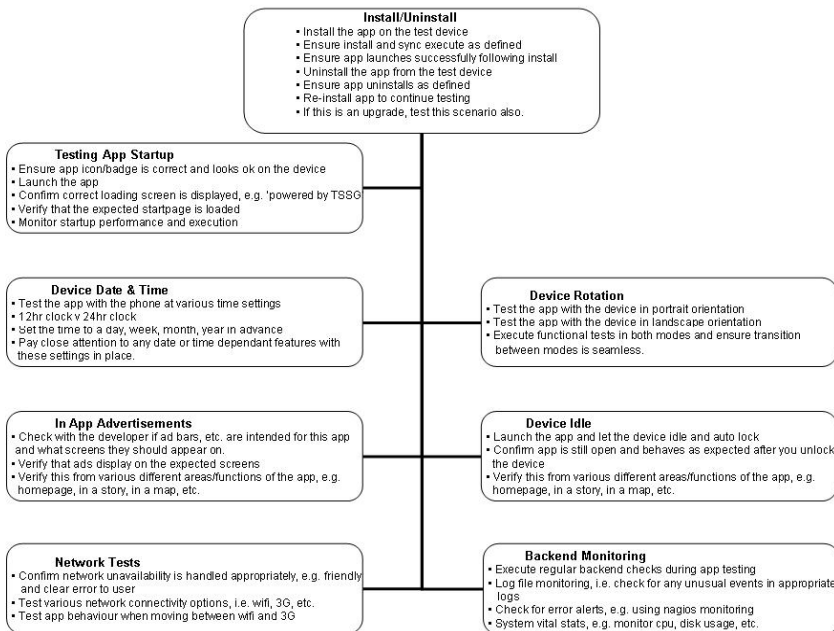 of application features and their potential implementation to be discussed from the earliest of stand-up and planning meetings. A strong collaboration with developers and their insights helps refine the testing specification.

A test specification should be created that examines every aspect of the application under test. Starting at the home page the test spec should direct the tester to visit every individual page in the app. Verify that selecting a tab, menu or link works and brings you to the expected destination. Validate all data entry functions with both valid and rogue data inputs. Verify that all text is displayed as desired and that there are no spelling or grammar issues. Ensure scrolling, back button, etc. work seamlessly.

If mapping/geographical functionality is included in the app, confirm that the map opens as designed and shows all relevant places associated with the app. It is important to verify functions like location based content and learning and this may involve site visits. Where a 'Help' option is presented ensure it is relevant and easy to use. If there is a feedback mechanism check minimum and maximum character entry and confirm feedback is received at the backend.

Certain apps use a subscription model and all aspects of this model need to be tested. Execute buying all available subscriptions. Verify that the purchase is registered at the back end. Tweak the back end data to pre-expire and expire the subscription and check appropriate warnings and reminders are presented to the app user.

Most feature rich apps will make use of API's that permit access to device functions such as the camera, accelerometer or address book. Make sure that each of these features is comprehensively tested as part of the test specification for the app.

This phase of the testing process can also benefit from an examination of the usability of the application [22] and how users with varying technical capabilities rate the app in terms of design, user-friendliness, meeting their expectations. The results of these usability tests on the project produce valuable input for further research and technical development as well as improving the application under test. In particular usability testing is crucial to determine the users' ability to correctly understand if the developer/designers intended functionality were successfully conveyed. This consequently determines whether some specific application features are incorrect or unclear and as a result need to be revisited by the developer

3.3   On Device Performance Tests

A key activity in validating the quality of a mobile application is determining the performance and responsiveness of the app on the devices and platforms it is intended to run on [23]. If the app has a memory leak it may have some unintended consequences for the device it is running on. Application crashes, unusual device behaviour associated with the app, poor responsiveness or abnormal battery drain will soon be spotted by the end user and leave a less than favourable impression, especially in app store ratings and reviews.

iOS

The tool we use for analysing app performance on iOS devices is Instruments [24]. Simply connect your phone to an Apple machine running the suite of iOS developer tools. Launch the Instruments tool on the PC and launch the app for test on the device. From here you can monitor and analyse data such as file access, memory, CPU usage. The information is presented in graphic displays making it easy to understand and detect problems.

Android

For Android testing we use Android Debug Bridge (ADB) [25]. This tool includes a client which can run on your PC, a server component which runs as a background process on your PC and a daemon which runs as a background process on each emulator or device instance. The tool allows you to install apps on a device and issue shell commands to the device e.g. 'top' to analyse cpu usage. In conjunction with ADB we use UI/Application Exerciser Monkey for sending pseudo-random events at the app to test performance.

3.4   Load Tests

While numerous commercial tools exist for performing load tests for mobile applications, the TSSG solution of choice is the Java based open source tool from Apache, namely JMeter [26]. JMeter allows us to configure thread groups to simulate multiple users accessing the same endpoint on the server/cloud side. The HTTP header plays an important role in identifying the source of a request. JMeter allows us to simulate mobile requests by sending the requests through the "user agents" of various mobile browsers. While load testing is being performed, it is important to keep tabs on system health. Refer to Section 3.1 Backend Monitoring checks.

4 Getting Access to Devices for Testing

4.1   Access to Test Devices

An obvious and basic requirement for testing an application is having the specified devices available. The application will need to be deployed onto the required devices so that the suite of test cases defined in Section 3 can be executed. The customer will have specified what platforms and devices that the application is required to run on. This will have come from their own market research or defined business requirements. A matrix of test platforms should be created that captures all of the customers needs. This may include device types (phones, tablets, etc.), OS (including versions), manufacturers and models (HTC, Samsung, etc.) and even different carriers and service providers.

The major mobile platforms that the TSSG projects have worked on include:

Figure 3: Mobile platforms tested by TSSG

As most of the projects in the TSSG start out as funded research, the budget to buy multiple devices may not be available. Generally each project has some hardware budget so over time we have established a decent collection and variation of device platforms and manufacturers. These devices can be used and shared across all future projects. However, the competition between manufacturers to release new devices and new features so frequently means that test devices have a relatively short life span before becoming redundant.

The TSSG is co-located with other software companies that also engage in smartphone application development. We have found it beneficial to build relationships with these companies not only in terms of sharing knowledge but also for sharing devices. Quite often a member of our EFM team involved in testing an application will knock on a neighbours door to borrow some devices that are not available within our own organisation. This has proved a handy and cost free method for expanding our test capability. One word of caution when sharing devices internally and externally; carefully track and log who has a device and where it is at any given point. The EFM team are the central point of contact for this in the TSSG.

On certain projects the TSSG has engaged with device retailers in our locality. For a nominal fee the retailer would allow the test team to go to their site and use a range of their display phones to test the app in question. Bringing our own SIMs, we could access maybe 15 to 20 devices in a test session.

For Nokia devices we use the Nokia Developer Remote Device Access (RDA) service. A free to use web based tool, you get access to a huge range of real Nokia devices where you can install and test applications.

A poor alternative to testing on a real device is to test the application on an emulator. While we would never certify an application based solely on testing via an emulator [27], the use of a virtual device can prove advantageous to get some testing done on an early release of the app or when a physical device is unavailable yet. All of the major mobile platforms contain easy to use emulators in their SDK tools.

4.2   Paid Solutions and Automated Tools

A tool that we have used on some projects to gain access to an even wider range of devices is Keynotes Device Anywhere [28]. This cloud based tool provides easy access from the tester's desktop to a diverse range of actual smart devices where manual test execution can be performed. The user has full access to device features such as the camera, accelerometer, power on/off, etc. Tests can be performed over live networks and Wi-Fi from any location in the world. Another advantage to this product is an add-on that will enable automated testing where you can create scripts to capture, replay and verify real user interactions across live mobile devices. This can be really beneficial for applications requiring sizeable regression testing or following continuous integration. Also, this product can be integrated with other Keynote components to provide a full mobile app lifecycle management service.

A competing product that offers similar capabilities which we have reviewed but not used is Perfecto Mobile [29]. This tool can be integrated with HP's QuickTest Professional suite of tools.

Another benefit to using a testing framework such as those described above is the ability to stay on top of the latest OS releases. With early access to any OS upgrades and an automated suite of test cases at hand you will be able to quickly assess and react to any issues caused by new OS releases.

The major drawback to using these service offerings, from a TSSG perspective anyway, is the costs associated with the purchase and subscription models available. When a project has matured and spun out as a company the finances may be available to pursue a certain level of subscription to one of these tools. However, for the most part, our projects need to reply on access to devices outlined in section 4.1.

5   Case Study - Using Agile for mobile application development in a global development environment

TSSG recently worked on a project to produce a second screen application that allowed for TV viewers to engage with their friends on social networks, allowing discussion on and sharing of content. This secondary screen involved the development of an innovative smartphone application. While the app development was performed in Waterford, the design/assets for the app along with various API's were developed in parallel at partner sites in the US. A time zone difference of -8 hours resulted in only a one hour overlap in the typical working day. The Verification and Validation (VnV) function was performed at both locations.

There are several well documented challenges for the application of Agile methods in global software development [30]. The challenge of realising the benefits of communication is a common theme that runs through the research to date [31]. How could daily communication be arranged effectively? How could informal communication that is important to Agile methods be encouraged? Traditional Agile methods were developed for collocated teams with the perception that this enabled better intra-team communication based on conventional IT projects. However there were a number of factors that made this project a good fit for Agile development techniques. These included the use of small teams, usability factors requiring constant stakeholder feedback, and the short deadline to create the mobile app.

Once the application design document was delivered a Product Backlog listing top-level requirements was agreed for the product. A daily 'Technical Scrum' was held between the developers in TSSG and our partners at the offshore sites. In these meetings, lasting fifteen minutes and held at the same time everyday, each team member was present and answered the three daily Scrum questions – What did you do yesterday? What are you going to do today? Do you have any blockers? Any stumbling blocks identified in this meeting were documented and a resolution was worked on outside of the meeting. The meetings were facilitated using the GoToMeeting [32] web conferencing software. Having a dedicated number/pin to access conference calls for the duration of the project enhanced the communication as everybody knew from the start of the project what call to join at what time. Previous experience indicated that when conference call details are regularly changed lots of participants get times and access numbers wrong resulting in less effective communication.

As the client's role in using a Scrum methodology is essential, a second non-technical daily scrum took place that included the client. Delivering a high quality app early with frequent builds strengthened trust and respect between partners making further collaboration easier. The client monitored the product backlog and acting as product owner was responsible for making entries or adjustments to the project throughout the project process.

At the end of each sprint a review meeting was held with all the stakeholders. During this meeting each team showed what they accomplished during the sprint, which in our case involved providing a demo of the integrated app using the iOS simulator with the screen shared to all participants in the meeting. There were some minor limitations on using the simulator especially in terms of memory and cpu constraints that would possibly only arise on a device.

TestFlight was used to distribute the internal iOS releases to the team members in the various geographical locations as its features made it easy to distribute ad-hoc builds and monitor beta testing usage. Another feature of TestFlight is the ability to collect and automatically symbolicate crash reports remotely. As testing was done in various sites this allowed developers access crash reports without having to physically connect to the device. TestFlight also has a checkpoint facility which means you can track what testers are doing when testing the application on their device, and if certain features of the application are been tested that are of interest to you.

Jira was used as a means of tracking issues that were raised by the customer and also developers could raise issues that required feedback from the customer. Of course, the use of Instant Messaging tools was common as developers coordinated individually to tackle specific problems.

Using these latest technologies we went some way to overcoming the traditional problem of communication in distributed development and specifically for a mobile application product.

6   Application Release and Analytics

6.1   Checklist for App Release to Store

The process for releasing a mobile app does not end when the software has been built, deployed and tested. Following the internal QA process that rigorously tests and verifies the app for release suitability, there follows a set of tasks that are required to get the app uploaded to the store for customers to download. This we found especially intricate for Apples iTunes store.

A checklist or 'application management guide' was designed to drive the app from internal VnV certification right through to release on the app store. The idea of this checklist is to bring all stakeholders involved in releasing the app into the same process and to provide a sequential set of steps to make the process a success. The main stakeholders involved in releasing our apps include:

- Product / Marketing Manager
- Technical Lead / Developer
- VnV / QA
- Dev Ops / Deployment

Depending on the application or project, one of the players above will drive the process from start to finish. This is important so that all of the information required by Apple is included in the submission and that the timelines for releasing the app are met. The checklist covers scenarios for new apps, upgrades, re-submissions, etc. A summary of the items in the checklist are captured below:

- Determine Application Name/ID/Description
- Decide on Payment/Subscription Model
- Create and collect mandatory assets (submission document, screen shots, icons)
- Create skeleton app in iTunes Connect, including in-app purchases
- Create App ID, provisioning certs, etc. in Developer Centre
- Add project/app to local build server
- VnV test all aspects of app
- Perform UAT and get customer approval
- Complete any backend changes required
- Validate pre-release version of app pointing at production backend
- Upload all assets to iTunes for approval

6.2   Application Analytics

In an attempt to understand how end users interact with our mobile applications, certain apps have been developed with Flurry Analytics [33] integrated. This free tool allows us to look beyond the downloading of the app and gain insight into how customers actually use the app. From this tool you can learn about who is using your app, when and where they use it and ultimately tailor your product roadmap to increase engagement, retention and profits.

7 Conclusion and Next Steps

The momentum and pervasiveness of smart device applications is clear. The requirement for a proven and yet flexible framework to guide the creation of these apps is apparent. From our experiences the Agile methodology is a good fit. Some of the tools may be different when developing and testing mobile applications but the principles remain the same: customer engagement, effective requirements gathering, quality build setup, short work sprints, regular releases, comprehensive test plans.

The complex matrix of mobile manufacturers, platforms, OS versions, etc. makes testing an ongoing struggle. A combination of simulators, real devices and somewhat expensive remote access tools can help combat the need to test as broadly as possible while the challenges of launching your app in an app store can be alleviated by incorporating that activity into your Agile process.

An evolving area that the EFM team would like to further explore is that of mobile app security testing. In the era of BYOD (Bring Your Own Device) the challenges of data confidentiality and integrity, user authentication and authorisation are increasingly relevant and important.

8 Literature

[1] International Telecommunications Union; "The World in 2011. ICT Facts and Figures"; 2011;

[2] Dediu, H; "When will smartphones reach saturation in the US?"; 2012; Asymco.com

[3] The World Bank; "Information and Communications for Development – Maximising Mobile", 2012;

[4] Ayvazian, B; "LTE TDD Operator Business Case & Adoption Forecast"; 2011; Prepared by Heavy Reading for China Mobile, Alcatel-Lucent, Ericsonn, Nokia Siemans, Qualcomm

[5] Spataru, A; "Agile Development Methods for Mobile Applications"; 2010; University of Edinburgh

[6] http://www.tssg.org/

[7] Dowling P; "Successfully transitioning a research project to a commercial spin-out using an Agile software process"; Euro SPI 2012 Industrial Proceedings; ISBN 978-87-7398-154-2

[8] Cleary F, Dowling P, Kehoe C; "An Agile software process successfully imlemented within an expanding research and commercial organisation"; Euro SPI 2008 Industrial Proceedings; ISBN 978-87-7398-150-4

[9] PERIMETER project website {http://www.ict-PERIMETER.eu/}

[10] Redmine {http://www.redmine.org/}

[11] Hudson {http://hudson-ci.org/}

[12] Jenkins {http://jenkins-ci.org/}

[13] Subversion {http://subversion.apache.org/}

[14] Git {http://git-scm.com/}

[15] EMMA {http://emma.sourceforge.net/

[16] Xcode {https://developer.apple.com/technologies/tools/}

[17] Cobertura {http://cobertura.sourceforge.net/}

[18] Frank {http://testingwithfrank.com/}

[19] Cucumber {http://cukes.info/}

[20] TestFlight {https://testflightapp.com/]

[21] DropBox {https://www.dropbox.com/}

[22] Zhang D, Adipat B; "Challenges, Methodologies and Issues in the Usability Testing of Mobile Applications"; International Journal of Human-Computer Interaction, Vol. 18, Iss. 3, 2005

[23] Díaz & Hilterscheid; Testing Experience; "Mobile App testing"; 2012; ISSN 1866-5705

[24] Instruments {https://developer.apple.com/technologies/tools/}

[25] Android Developer Bridge {http://developer.android.com/tools/help/adb.html}

[26] JMeter {http://jmeter.apache.org/}

[27] uTest | eBook ; "The Essential Guide to Mobile App Testing"; 2012

[28] Device Anywhere {http://www.keynotedeviceanywhere.com/}

[29] Perfecto Mobile {http://www.perfectomobile.com/}

[30] Paasivaara M, Lassenuius C; "Could Global Software Development Benefit from Agile Methods?"; IEEE International Conference on Global Software Engineering; 2006; 0-7695-2663-2/06

[31] Dorairaj S, Noble J, Malik P; "Distributed Agile Software Development: Towards a Grounded Theory"; New Zealand Computer Science Research Student Conference; 2011

[32] GoToMeeting {http://www.gotomeeting.ie/fec/}

[33] Flurry Analytics {http://www.flurry.com/index.html}

Author CVs

Phelim Dowling

Phelim qualified with a Bachelors degree in Computer Aided Manufacturing from WIT and has 16 years experience in the IT industry working in a variety of roles and companies. Starting his career with Intel he rose to the position of technical/project lead at their R&D facility in Oregon. Here he was responsible for coordinating and executing the development, testing and deployment of numerous mission critical products into Intel's manufacturing sites worldwide.

On moving back to Waterford he spent time as a Test Engineer with Waterford Technologies. Following this he worked as a consultant project manager specialising in IT project delivery in the highly regulated pharmaceutical industry. In 2006 Phelim joined the TSSG Verification and Validation team and has facilitated on projects such as iServe, Muzu, FeedHenry and more recently the EU FP7 funded OpenLab initiative.

Kevin McGrath

Kevin McGrath is a Bachelor of Business Studies graduate of Waterford Institute of Technology. He joined East Digifone (later to become o2 Ireland) as an Operations Analyst shortly after the Company was awarded the second Gsm licence for Ireland. With the introduction of a new billing system Kevin performed the role of Billing Coordinator and was responsible for the ongoing testing of new products launched by the company. In 2000 Kevin joined Openet Telecom as a Senior Test Engineer. In this role he developed and executed both manual and automated test cases initially at the company's development centre in Dublin and when required at various customer sites around the world.

Kevin joined the commercialisation team within Waterford Institute of Technologies TSSG in 2006 as a Verification and Validation engineer. He has worked on several Enterprise Ireland projects including FeedHenry SWAGGER, COIN, SSGS, EGP. He has also worked on Billing4Rent an EU funded project which provided a SaaS interface to Intec's Single.eView billing product.

Gemma Power

Gemma Power holds a first class honours degree in the BSc in Computing - Information Technology from Waterford Institute of Technology (W.I.T), a National Diploma in Electronic Engineering and a National Certificate in Electronic Engineering.

Gemma has extensive industry experience, both as a Functional Debug Technician in Apple Computer Cork and as a Hardware Service Technician at Campus Services in W.I.T. She also has seven years of experience as a researcher. She joined TSSG in September 2007 and worked as the project manager for the EU funded MORE project which was in the area of Wireless Sensor Networks & Validation and Test. Gemma is currently working on the SOCIETIES project where her primary focus is the testbed set-up, configuration and federation, in addition to the code integration and testing, proof of concept and demonstration aspects of the project.

Assessing partially outsourced processes – Lessons learned from TIPA assessments

Stéphane Cortina, Alain Renault, Michel Picard

Centre de Recherche Public Henri Tudor
29, avenue John F. Kennedy,
L-1855 Luxembourg
{ stephane.cortina ; alain.renault ; michel.picard }@tudor.lu

Abstract

This paper presents three process assessments that took place in Luxembourg in 2012. It depicts the TIPA® process assessment framework, and reports on its usage in three different partially outsourced contexts: a telecommunication service provider, a European institution, and a bank. Finally, this paper presents the lessons learned from these experimentations and some improvement perspectives.

Keywords

1 Introduction

"*Measurement is the first step that leads to control and eventually to improvement. If you can't measure something, you can't understand it. If you can't understand it, you can't control it. If you can't control it, you can't improve it.*" This quotation from Dr. H. James Harrington is particularly true when talking about processes. Indeed, by assessing the capability of a process you can objectively understand if this process achieves its outcomes, and moreover, you can control to what extent it reaches its purpose. But what happens when you can only assess a part of what you should measure? When performing a process assessment, what happens if the assessed organization does not own the whole process? What happens if several stakeholders take part in the game? Does it lead to just a partial understanding of the process capability? In other words, does it imply a limited process improvement? That is what we intend to determine in this paper by comparing three TIPA assessment projects that took place in Luxembourg in 2012.

2 TIPA: The Tudor's IT Process Assessment

Several years ago, the Public Research Centre Henri Tudor in Luxembourg launched the AIDA project, an innovative initiative whose goal was to develop an ISO-based methodology for performing IT service management process assessments [9][11]. AIDA preceded the development of TIPA [8][12], a generic framework used to measure the capability of processes in a repeatable way, by combining a standard description of processes contained in a Process Assessment Model (PAM), with an assessment method, based on the ISO/IEC 15504 standard (Figure 1). To ensure the objectivity of its results, the TIPA process assessment method strictly complies with all the requirements defined in the part 2 of the ISO/IEC 15504 standard [1].
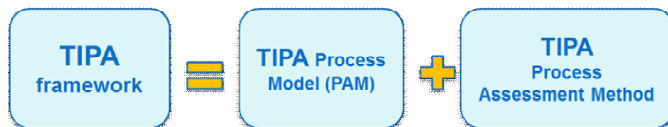
Figure 1 - The TIPA generic framework

ITIL[®] is a series of books that describe a comprehensive set of process-based methods for managing and delivering IT services [3][4][5][6][7]. This set of good practices is coming from the industry and is considered as the de-facto standard in IT Service Management (ITSM).

TIPA for ITIL is an 'instantiation' of the generic TIPA framework to the context of ITSM [12]. TIPA for ITIL combines the TIPA process assessment method (based on ISO/IEC 15504) with ITIL to objectively evaluate the capability of the ITIL processes (Figure 2).

Figure 2 - TIPA for ITIL

TIPA for ITIL is a recognized framework that has now been used for several years to assess the capability of ITSM processes within companies all around the world [10][14][15]. In 2012, 60 TIPA assessments have been officially reported, and 77 TIPA Assessors and Lead Assessors have been trained[1].

---

[1] Source: ITpreneurs

3 Experimentations

In 2012, the Public Research Centre Henri Tudor decided to update the TIPA PAM for ITIL in order to align with the latest version of ITIL published in 2011 (Figure 3). Once updated, three experimentations were run to confront the new PAM with the reality encountered by companies operating in Luxembourg.

In each case, the TIPA Lead Assessor provided advices about the roles that should be included in the sample of interviewees and about the processes that should be assessed. But for experimentations 1 and 2 the sponsor imposed the scope of the assessment (influenced by the problems encountered by the organization and the limited resources available for the assessment), and particularly, he decided that the interviewees could not come from outside the assessed entity (i.e. being an internal or external service provider).

ITIL 2011

Service Lifecycle

Service Strategy
ITSSM   IT Service Strategy Management
SPM   Service Portfolio Management
ITSFM   IT Service Financial Management
DEMM   Demand Management
BRM   Business Relationship Management

Service Design
SDC   Service Design Coordination
SCM   Service Catalogue Management
SLM   Service Level Management
AM   Availability Management
CAPM   Capacity Management
ITSCM   IT Service Continuity Management
ISM   Information Security Management
SUPM   Supplier Management

Service Transition
TPS   Transition Planning and Support
CHGM   Change Management
SACM   Service Asset and Configuration Management
RDM   Release and Deployment Management
SVT   Service Validation and Testing
CHGE   Change Evaluation
KM   Knowledge Management

Service Operation
EVTM   Event Management
INCM   Incident Management
REQF   Request Fulfilment
PRBM   Problem Management
ACCM   Access Management

Continual Service Improvement
SSI   Seven-Step Improvement

Figure 3 - Processes covered by the TIPA PAM for ITIL
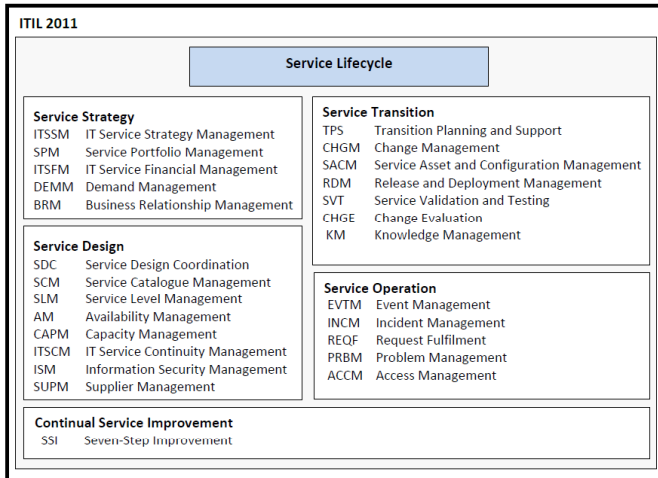
3.1   Experimentation 1: telecommunication service operator

Telecommunication service operator (assessed organization)

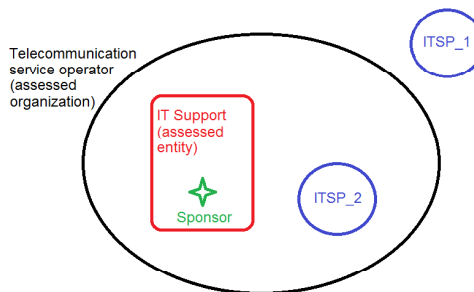ITSP_1

IT Support (assessed entity)

Sponsor

ITSP_2

Figure 4 - First experimentation context

During this first TIPA experimentation, initiated by a telecommunication service operator in Luxembourg, four ITIL processes were assessed. The purpose of this assessment was to determine precisely the capability level reached by these four processes, in order to identify the existing good practices that could be shared between the teams. This TIPA assessment also aimed at proposing an action plan to improve these processes.

The telecommunication service operator has outsourced a small part of its activities (mainly those related to software development) to an external IT service provider (ITSP_1). Some others parts of the activities are entrusted to another entity within the organization (ITSP_2). The sample of interviewees was composed of 13 employees from the telecommunication service operator, mainly from the IT Support team and without any representative from the (internal or external) service providers (Figure 4).

During this project, an assessment team composed of three certified TIPA Assessor and Lead Assessor assessed the Incident Management, Change Management, Release and Deployment Management, and Service Asset and Configuration Management processes with a target capability level of 2.

This first experimentation allowed stressing the good team spirit existing within the different IT Support teams from the assessed entity. Moreover, this assessment underlined that these teams have a good knowledge of the business activities and of how the IT supports these activities. However, the different interviews enabled to discover some weaknesses related to the external and internal stakeholders operating within the scope of the four assessed processes. Here is an example of these findings:

- Finding 1: *"The incident resolutions that are implemented by external stakeholders are taking too much time. Following these implementations by the external stakeholder (ITSP_1), there is sometimes the apparition of new bugs, or some regressions in the functionalities and in the service level "*

- Finding 2: *"In certain particular domain, there is a stakeholder-dependence (both internal ITSP_2 and external ITSP_1) which could lead to concerns, in terms of quality of service provided, competencies and reactivity."*

- Finding 3: *"The lack of formalism/professionalism in the relationship with the internal and external stakeholders."*

- Finding 4: *"Some changes, implemented by internal service provider (ITSP_2), are not planned in coordination with the business and are sometimes deployed without any consultation of the IT teams."*

- Finding 5: *"There is a risk that some pieces of source code, developed by external stakeholders, are deployed without any control and without any communication to the internal IT teams."*

The context of this TIPA assessment was complex, with multiple stakeholders (both internal and external to the assessed entity) taking part to different kinds of activities (incident resolution, change implementation, change deployment). The improvement recommendations following this experimentation were however limited by the scope of the assessed entity. But the different weaknesses noticed during the interviews were highlighted even if beyond the scope of the assessed entity.
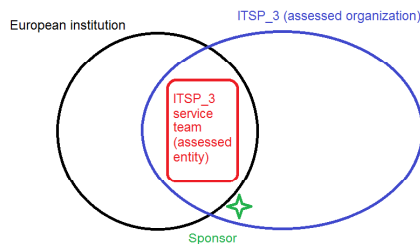
3.2   Experimentation 2: European institution

Figure 5 - Second experimentation context

In this second experimentation, the ITSM practices of an IT service provider that used to operate within a European institution in Luxembourg were assessed (Figure 5). This European institution used to outsource a great part of its activity to several IT service providers and one of them (ITSP_3) decided to organize an assessment of its processes' capability, as part of a global improvement initiative. For ITSP_3, which initiated this TIPA assessment project, the purpose was two-fold. Firstly, to demonstrate its professionalism and its efficiency, and secondly to ensure that the right organization is deployed and supported by the right tools and the right processes in order to deliver the service at the expected level of quality. Even if the interviews took place in the European institution premises the sample of interviewees was composed of six employees from the service team of ITSP_3 without any representative from the European institution itself. For this assessment, two certified TIPA Assessors and Lead Assessor composed the assessment team. This assessment team assessed (through 9 interviews) the Incident Management and the Request Fulfilment processes with a target capability level of 3, and the Event Management process with a target level of 2.

This second TIPA experimentation demonstrated that the service team was an experienced and well skilled support team, composed of motivated, service minded and committed people with a high level of seniority. The ITSP_3 team members have a good understanding of the objectives and activities related to the services they offer, even if there are some misunderstandings about ITIL concepts and terminology. However, the European institution structure is relatively complex and particularly the roles and responsibilities of the different service desks (within the European institution) were sometimes unclear to the ITSP_3 team members. Here is an example of the typical results coming from the process assessment performed:

- Finding 6: "*The ITSP_3 team is not the owner of the ticketing tool and adaptations to support the ITSP_3 requirements are difficult to address and implement (depend on an external team)*"

- Finding 7: "*In general, tools are owned by the European institution and their modifications are subject to approval outside the boundaries of ITSP_3 responsibilities*"

- Finding 8: "*The relationship is complex with the external groups that provide support. Frequently, tickets are logged and filtered by other groups and in many occasions mishandled and/or misrouted.*"

- Finding 9: "*There is a lack of visibility of the overall Request Fulfilment process as some of the activities are carried out of the scope of the ITSP_3 team.*"

- Finding 10: "*The priority of service requests is handled out of the boundaries of the ITSP_3 team and there is no clear perception of the handling of the internal priority.*"

- Finding 11: "*There is a clear contractual limitation that allows ITSP_3 to create only priority 3 tickets even if they have a higher level of priority.*"

The context of this project was not common. Indeed, the assessed organization was in this case an IT service provider that wanted to demonstrate to its client, a European institution, the professionalism of its practices and the capability level reached by its processes. But, due to the contractual context some of the improvement recommendations were clearly out of the scope of ITSP_3 (like those related to the modifications of the tools for which they do not have the ownership). For ITSP_3, the only way to take into account these improvement recommendations is by transforming them into "soft" suggestions that will be passed to the European institution.
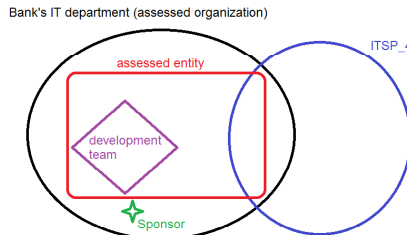
3.3   Experimentation 3: bank

Figure 6 - Third experimentation context

This third TIPA assessment was initiated by a well-known bank in Luxembourg. The bank had outsourced its ITSM activities into a dedicated service provider (ITSP_4) some years ago and the IT department of the bank wanted to know where they stand with regard to their ITSM practices shared between its internal development team, and the external service provider (Figure 6).

The purpose of this assessment was to determine the capability levels of some ITSM processes and to propose an action plan for improvement and alignment with the ITSP_4's practices. Particularly, the first objective was to identify improvement opportunities for the Incident, Problem and Change Management processes. The second one was to understand how capacity management activities were already managed in order to suggest a structured approach for the implementation of a Capacity Management process.

For that, an assessment team composed of four certified TIPA Assessors and Lead Assessors performed 23 interviews over five days in order to assess the Change, Incident and Problem Management processes with a capability level 3 targeted, and the Capacity Management process with a capability level 1 targeted. The sample of interviewees was composed of employees from the development team of the IT department and three members from the IT service provider (ITSP_4). For confidentiality purpose, and according to the TIPA assessment method, these persons have been interviewed separately.ITSP_4 was involved in the assessment through the provision of three staff members to have a full picture of the Incident, Problem and Change Management processes. However, the purpose of the assessment was not to analyse the ITSP_4 processes or to look after improvement actions on ITSP_4 side but to focus on supporting the improvement on the bank IT department's side. Here is an example of the findings resulting from this process assessment:

- Finding 12: *"The changes, realized at the infrastructure level, can be done without any communication from ITSP_4"*

- Finding 13: *"In a general way, there is a confusion between change management and release management and there is a lack of collaboration between the development team, the business and ITSP_4."*

- Finding 14: *"The incidents are "a priori" managed by ITSP_4 (who is supposed to play the role of first and second level of support). However in practice ITSP_4 only assure the first level of support which forces the 'development team' of the IT department of the bank to assure the greatest part of the work"*

- Finding 15: *"There is a lack of visibility on the process definition within the bank, and a lack of ownership of processes outsourced to ITSP_4 that lead to problems of responsibility"*

This experimentation depicts a classical situation where the assessed organization has a partial view on its process, and where it does not really know how its service provider is doing things. In that case, the improvement actions recommended by the assessment team in order to improve the capability of the bank's processes will be hard to implement, due to the ownership of ITSP_4 on some of the processes. This kind of findings suggests a reconfiguration of the relationship between the bank and its IT service provider.

4 Lessons learned

The three experimentations conducted in Luxembourg during the year 2012 and described in the previous section allowed to enhance and validate the latest version of the TIPA PAM for ITIL. On the purely process assessment perspective, these three assessments also permitted to highlight the following lessons learned:

- **The presence of (internal or external) stakeholders can impact the assessment results and the perceived capability of a process**

  The rating of some of the ITSM practices can be influenced when one or more stakeholders are taking part to the implementation of a process. Indeed, during the first experimentation the interviewees described a situation 'out of control' due to the presence of stakeholders (Findings 1, 4, 5). This, combined with the fact that no representative from the stakeholders was interviewed could be perceived as a poor level of maturity of the practices which could influence (negatively) the results of the assessment. The assessors should always try to interview stakeholders' representatives, in order to clarify their vision of the situation.

- **Assessors need to have the good level of visibility on all process activities including the ones managed by other parties (external or not)**

  It is crucial to interview representatives from each stakeholder playing a role in the assessed processes. Thus, it allows the assessors to analyse a full picture of the situation, to evaluate objectively each practice, and to draw meaningful process profiles. Indeed, as seen in the second experimentation, not interviewing any member of the European institution lead to lack of visibility on certain practices (Findings 6, 9, 10). In case of limited visibility, the assessors should always draw the sponsor's attention on how to interpret the results of the process assessment.

- **Performing a process assessment can enable to characterize the relationship with the stakeholders**

  After having performed an assessment, even if no representative from the stakeholders was interviewed (as in the first two experimentations), it is common to have collected valuable information about the relationship between parties, be it information that will be used to determine the capability level of the assessed processes (Findings 8, 11), or information that will be used to describe the current state of this relationship (Findings 2, 3). Even if the stakeholders' view is considered as outside the scope of the assessment, the assessors should always pay attention to the state of the stakeholders' relationships.

- **Performing a process assessment can impact the relationship with the stakeholders**

  A process assessment usually leads to a process improvement plan. When defining that plan, the assessor should be aware that its content can have an impact on the relationship with the stakeholders. During the second experimentation, one of the findings (Finding 7) depicted the ownership of a tool outside the scope of the assessed organization. It engendered an improvement recommendation aiming to clarify and document the specific requirements of a tool and promote negotiations to complete the tools and/or configurations of the tools. In this case, this recommendation can have impacts (in terms of workload, cost, and time) for the customer (European institution) of the assessed organization (ITSP_3). Another example oc-

curred during the third experimentation, where a finding (Finding 15) described a lack of visibility and ownership of a process. This kind of finding can lead to improvement recommendation such as: "Define clear roles and responsibilities" or "Appoint a process owner", which are not neutral and can heavily impact the relationship between the assessed organization and its stakeholders.

- **The presence of stakeholders in the sample of interviewees can make assessment-based improvements easier to deploy**

  During the third experimentation, the fact that representatives of the main stakeholder were involved in the assessment had positive impacts on the improvement plan. Indeed, some of the findings (Findings 12, 13, 14) engendered improvement recommendations impacting both the assessed organization and its stakeholder. In that kind of situation, the assessors could and should make some 'broader' recommendations, applicable to both parties, such as how to work together, how to communicate, to finally improve the quality of their relationship.

- **The value of a process assessment can vary depending on the selected entity to be assessed**

  ITIL often assumes that processes are mainly managing assets of infrastructure. However some of them may have a different scope and implications depending on the assessed entity, particularly if you are moving from service operation to software development. It was the case during the third experimentation, when the assessed organization was the 'development team' of the IT department of the bank. Its involvement in ITSM processes such as incident management or capacity management was quite limited and, in this context the assessor should be careful to the representativeness of the capability level reached by these processes.

- **Without involvement of all the process stakeholders, the effectiveness of assessment-based improvements can be limited**

  Whatever the involvement of some of the stakeholders during the assessment, the TIPA assessors should involve all of them during the implementation of improvements. Indeed, this would maximize the Return on Investment (ROI) of improvement investments. In the second experimentation, if the European institution does not want to improve the ticketing tool, some improvement actions that positively impact most of the process activities will be missing. In the third experimentation, you can optimize as much as you want the part of the process executed by the 'software development' department, if you do not act on the first level of support (executed by ITSP_4), the improvement effect will be limited from a global process perspective.

5 Conclusion

This paper describes three different TIPA process assessment projects that took place in Luxembourg in 2012. These assessments enabled to validate the new version of the TIPA PAM for ITIL. But, as depicted in this paper, these experimentations also allowed highlighting some lessons learned regarding the assessment of partially outsourced processes and mainly related to:

- the impacts of the assessment of processes that are partially outsourced on the stakeholders' relationship,
- the impacts of such situations on the proposed improvement plan,
- the composition of the sample of interviewees.

The ISO/IEC 15504-2 standard [1] is limiting the perimeter of an assessment to an organisational unit that needs to be clearly identified prior to the assessment. Moreover the guidance coming ISO/IEC 15504-3 [2] states that: *"The selection of assessees should be representative of the Organizational Unit being assessed. If the participants are representative of the Organizational Unit then the assessment results are more likely to provide an accurate view of the process capability"*.

But for the experimentations presented in this paper, the assessed organizational units, (i.e. the assessed entities) did not encompass the whole set of stakeholders. In such contexts, it is important to keep in mind that the presence of representatives of all stakeholders in the sample of interviewees is crucial for both assessment result and organizational changes following the assessment. When defining the scope of the assessment, the lead assessor should urge the assessment sponsor to meet all key actors of the process (particularly in case of process activities owned by other parties). This would help the assessors to have a better view on the process execution and to better understand the strengths and weaknesses of the assessed processes.

In case of absence of some stakeholders' representative in the sample of interviewees, the assessors should take into account their limited visibility, particularly, they should be very careful when rating the practices, when interpreting the results of a process assessment, and when proposing an improvement plan. Moreover, the assessors should, in this case, warn the sponsor of the limits of the assessment (in terms of accuracy of the results and levers of improvement).

Despite the requirements and recommendations set by ISO/IEC 15504 these experimentations show that the borders of an organisational unit can be too narrow for the implementation of a process. There is a need to go further and give more guidance to enable assessment of processes implemented across several organisational units.

The sample composition should of course ensure that assessees are representative of the (main) organiational unit being assessed, but it should also first ensure that they are representative of the processes themselves even if parts of them are outsourced.

Existing assessment frameworks should maybe be reviewed to take that issue into account. We will improve the TIPA framework [8] by the provision of recommendations aiming at enhancing the representativeness of the sample of interviewees, particularly when several stakeholders intervene in the assessed process, or when this process spreads over several organizational units.

Finally, in order to reinforce the confidence in the results when assessing partially outsourced processes one might consider including in the scope of such an assessment some of the Supplier Management practices as described in ITIL, or some of the process enablers described in the Process and Enterprise Maturity Model (PEMM) [16] to get a better understanding on how the stakeholder relationship is managed. But again it requires the support of the assessment sponsor.

These are some of the perspectives we plan to further experiment in 2013.

6 Literature

1. ISO, ISO/IEC 15504-2: Information technology - Process assessment - Part 2: Performing an assessment, 2003

2. ISO, ISO/IEC 15504-3: Information technology - Process assessment - Part 3: Guidance on performing an assessment, 2004

3. IT Infrastructure Library – Service Strategy, The Stationery Office Edition, ISBN 9780113313044, 2011

4. IT Infrastructure Library – Service Design, The Stationery Office Edition, ISBN 9780113313051, 2011

5. IT Infrastructure Library – Service Transition, The Stationery Office Edition, ISBN 9780113313068, 2011

6. IT Infrastructure Library – Service Operation, The Stationery Office Edition, ISBN 9780113313075, 2011

7. IT Infrastructure Library – Continual Service Improvement, The Stationery Office Edition, ISBN 9780113313082, 2011

8. ITSM Process Assessment Supporting ITIL®, Van Haren Publishing, ISBN 978-90-8753-564-3, 2009

9. Barafort B., Di Renzo B., Lejeune V., Simon J-M, ITIL Based Service Management measurement and ISO/IEC 15504 process assessment: a win – win opportunity, SPICE'2005, Klagenfurt, Austria, 2005

10. Hilbert R., Renault A., Assessing IT Service Management Processes with AIDA – Experience Feedback, EuroSPI'2007, Potsdam, Germany, 2007

11. Barafort B., Di Renzo B., Assessment and improvement integrated approach: combined use of the ISO/IEC 15504 (SPICE) and the Information Technology Infrastructure Library (ITIL), Proceedings of the National Conference SPIRAL'2004, Luxembourg, 2004

12. St-Jean M., TIPA to keep ITIL going and going, EuroSPI'2009, Alcala, Spain, 2009

13. Barafort B., Renault A., TIPA: 7 years experience with SPICE for IT Service Management, EuroSPI'2011, Roskilde University, Denmark, 2011

14. Tremblay S., Session 308: Process Maturity Assessment Done Right, itSMF USA - Fusion 12, Dallas, USA, 2012

15. www.tipaonline.org

16. Hammer M. (2007). The Process Audit. Harvard Business Review, April, pp. 111–124.

7 Author CVs

Stéphane Cortina

Stéphane works at the Public Research Centre Henri Tudor since 2007. This R&D Engineer is involved on research projects related to normalization, IT service management, business continuity management, and process assessment and improvement. He holds a Master degree in Software Engineering from the University Joseph Fourier of Grenoble. He is also a certified ITIL Expert, an ISO/IEC 15504 Assessor, a TIPA certified Lead Assessor, member of Luxembourg chapter of itSMF, and member of ILNAS network (The Luxembourg Institute for Normalization, Accreditation, Product Safety and Quality).

Alain Renault

Alain is Senior R&D Engineer with Centre de Recherche Public Henri Tudor in Luxembourg. He returned to research after spending eight years as a software engineer and IT support. Since 1998 he is working on process improvement projects, firstly focusing on software development activities and then on IT Service Management. Alain is an ISO/IEC 15504 Assessor, a TIPA certified Lead Assessor, Vice-President of the Luxembourg chapter of itSMF. As member of ILNAS network (the Luxembourg National Body), he is a member of the ISO working groups on Process Assessment and IT Service Management. He is now in charge of the TIPA® R&D project.

Michel Picard

Since 2004, Michel works at the Public Research Centre Henri Tudor in Luxemburg as R&D engineer for the "Business Process & Service Engineering" Unit (PROSERV). He works on projects related to the modelling and assessment of the Operational Risk Management (Basel II). Michel also works on projects related to normalization, IT service management and processes assessment and improvement. He obtained his Master degree in Computer Science from the Facultés Universitaires Notre-Dame de la Paix (FUNDP) in Namur (Belgium). He is also a certified ITIL Expert, an ISO/IEC 15504 Assessor, a certified TIPA Lead Assessor, member of Luxembourg chapter of itSMF, and member of ILNAS network (The Luxembourg Institute for Normalization, Accreditation, Product Safety and Quality).

Will the software development project be a success? The supplier perspective

Jaana Nevalainen[1], Paula Savolainen[1,2], Jarmo J. Ahonen[1,2]

[1] School of computing, University of Eastern Finland, Kuopio Campus
{jaana.nevalainen, paula.m.savolainen, jarmo.ahonen}@uef.fi
[2] School of Information Systems, Technology and Management, Australian School of Business, The University of New South Wales, Australia

Abstract

The purpose of this paper is to present our findings what are the elements which make software suppliers feel that the starting software development project will be a success. Our results indicate that the supplier's representatives have a mutual understanding about elements, which has to be in order before software development project start. The most important element that makes a project succeed is a clear definition of the goals of the project. Other important elements are project team and resources. According to our results, the suppliers' project managers had no mutual opinion of the most important elements. The project manager's initial impression is important because it has a great impact on the project manager's motivation. Good motivation makes it easier for the project manager to motivate the project team and motivation has been shown to be important for project success.

Keywords

Software development project, project success, project success element, supplier, motivation, project manager

1 Introduction

Although software has been developed through projects since the 1960s and countless studies have been conducted to understand why software project fails, there is no generic answer to question how to prevent software project failures [1-5]. Despite time, effort, and money, which have been used during the project, the customer may be unsatisfied with the project results. At the worst cases, when software projects have run over schedule and budget, and furthermore, it is understood there are too many problems in the projects, the projects are cancelled. This is done despite of money and other resources spent in the projects [6], [1]. For instance, Verner and Abdullah [7] studied strategic development project which was estimated to take 18 months, but ultimately took six years to complete. Project led a customer to £265 million losses. Moreover, the supplier's financial loss was £318 million. It was found that both the customer and the supplier made mistakes. The main problems indentified were in the areas of project complexity, contract, requirements, planning and control, execution and team assembly [7].

In addition to studies on software project failures, there are studies on how to prevent software project failure and how to ensure software project success (e.g., [8-11]). In these studies, many guidelines and frameworks have been suggested and different success factor lists generated (e.g., [12-15]). The main methods of studies on frameworks and success factor have been the use of questionnaires or comparison of other researchers' results on success factor lists.

There are few problems related to suggested frameworks and success factor lists in practice. One of them is related to heterogeneous respondents of questionnaires. Different people with different roles have answered questionnaires. Respondents have been, for example, project managers, team members or senior managers in both customer and supplier organizations. Therefore, results do not reflect how certain practitioners perceive project success or project success factors.

Another problem is that different project types and different perspectives are not identified and kept separate. Project types such as software development projects, which are conducted in-house or by supplier firm to an external paying customer (outsourced projects) and software product development projects are studied as they would be similar: having similar success criteria and success factors [10], [12], [14], [16], [17]. However, success criteria and success factors e.g. for in-house projects and projects conducted by suppliers to external customers (outsourced projects) seem intuitively different.

In this article, we concentrate on only one type of software development projects. We focus on projects which are conducted by supplier firms i.e. the main business of supplier firms is to provide software to customers. In these situations the customer is paying for the output of the project to the supplier firm. Moreover, we study software development projects success. Furthermore, we concentrate on project managers of supplier firms, who are responsible for project execution.

In this paper, we present our findings from analysis of answers given by participants of a seminar. Particularly, we are interested in what are the characteristics of a software development project which get a supplier's project manager to perceive that the starting project will be successful. Because the project manager is a central actor and responsible for leading the project to expected results s/he should be convinced of project success and therefore motivate other team members. When the project manager and other team members are confident and motivated, the team will strive for successful project outcome [3], [18-19].

We begin with a background, clarifying concepts project success criteria and project success factors, which have to be understood when discussing project success. We also clarify importance of perspective while discussing project success or failure. In section 3, we present our research methodology and, in the section 4, we present analysis of our data. Finally, in section 5, we present a summary of our study.

2 Background

When we are discussing project success, there are two concepts, which have to be understood. One is project success criteria and another is project success factors. Turner provides following definitions [20, p.47]:

1. *Success criteria:* The dependent variables by which we will judge
   the successful outcome of the project.
2. *Success factors:* The independent variables which will influence
   the successful achievement of the success criteria.

A simple example may clarify the difference between success criteria and success factors. If a success criterion is that the project has to be finished on time, the success factor should assist in achieving the criterion. In this case, the project success factor can be that timetable is planned and it is realistic.

Studies on project success criteria highlight usually three aspects: the project has to be finished within the budget and on time, and meet user's requirements and specifications [8], [20-21]. This set has been expanded to include other criteria, for example, client satisfaction and stakeholder's benefits [8], [22-24]. However, if the project does not meet these criteria, it does not automatically mean that the project is doomed to be a failure [16]. The project can be delayed and have exceeded the budget, but the project can considered as a success in the end [4], [13], [23].

Project success factors have been studied extensively, e.g. ([13-14, [16], [26-27]). Probably the most well-known researchers on this topic are Pinto, Slevin, and Prescot [28]. They presented a Top ten list of success factors in project implementation projects [26], and moreover, studied how success factors importance changes over the project life cycle phases [27]. One recently published literature review on critical success factors affecting software development projects was made by Nasir and Sahibuddin [12]. The main success factors were clear and frozen requirements, realistic estimations of schedule and budget, and a competent project manager. However, the data was collected without considering project type: it is not clear whether the projects were in-house projects or projects conducted by suppliers to external customers (outsourced projects). Moreover, it is not clear whose perspective was studied, i.e. customer's, who utilize software, or supplier's, who produce software. Nevertheless, the critical success factors they found are general, and at least clear and frozen requirements are difficult to obtain before project start.

In addition to importance of understanding concepts project success criteria and factors, it is important to understand whose perspective project success is discussed [29], and moreover, who is discussing project success. Different people perceive project success differently [5], [20], [30-32] and success depends on what the respondent expected of the project [20], [30-32]. Therefore, software development project success criteria should be different whether project success is discussed from supplier's or customer's perspective.

For each project, project success criteria should be defined, agreed, and communicated before project start [20]. Without defined success criteria it is not possible to evaluate whether the project has been successful or not [33]. Because success means different things to different parties, different project success criteria should be defined for both the customer and the supplier before project start.

After defining project success criteria, it is possible to identify project success factors [9]. However, although project success factors are identified, it does not guarantee project success, but discarding these factors can lead a project a failure [31].

We have clarified concepts project success criteria and project success factors. However, our focus is not to directly study neither project success criteria nor success factors. Instead, we are concerned on project manager's observations on upcoming project and especially supplier's project manager's perception of the starting project. We are interested in the impression of the project managers: do they perceive project successful and if they do, what are the existing factors or elements of the project which make the project manager perceive project successful?

3 Research methodology

This study is exploratory in nature. The method employed was data collection through a straightforward questionnaire to a limited target group. The first step was to define the actual theme or aim of the questions intended to elicit the answers. The second step was to devise the actual questionnaire and the third was to select a suitable target group and the time and place for the data collection. Finally, the data was collected, and the fifth step was the analysis, which is presented in the next section.
The definition of the actual aim of the questionnaire was discussed by a group of researchers. The aim of the research was to examine those elements which enable of project manager to form an impression about the potential success or failure of a project. The aim was not to find a list success factor or criteria, but the reasons why the project manager gets the initial impression of the potential success of the project.
The questionnaire was based on the idea that an experienced project manager will constitute his/her preliminary impression of the project while getting familiar with the project. The time in which we are interested in is the period that starts when the project manager has got the information that he/she will get a new project and ends when he/she has made himself/herself briefly familiar with the project.
That often includes preliminary reading of the most important documentation and discussions with salespersons and/or other persons who are more familiar with the project. That period may include discussions and meetings with relevant parties, but it does not extend to the moment when the project manager has acquired a complete understanding of the project and briefed the project team to the project.
In other words, we are most interested in the impression that the project managers gets from the period before he/she has been able to get a fairly comprehensive understanding of the project, but after he/she has had a brief look at the project. That time is important because it has an impact on the motivation of the project manager, and it is often difficult to change one's initial impression later on
There was a set of background question in order to facilitate better understanding of answers. The set covered the following issues:

- age
- gender
- education
- position
- number of years in current position
- name of the current employer
- the role of the employer (customer, supplier, education, other)

The venue for the questionnaire was a seminar hosted by the Finnish Software Measurement Association (FISMA ry), where all participants were somehow involved in software development. First the participants were provided an outline the aim of the questionnaire: it was stressed that the aim was not to find out general success factors but those elements which enable to form an impression of the potential success of the project. Moreover, the participants were asked to concentrate on the time before project start, not the whole project life-cycle phases. The actual question was to following one: Which are the elements which make you to think that the project will be a success?
There were 25 participants in the seminar, nine of them female. Of those participants 16 were from supplier organizations which develop software for various customers, seven the title of project manager. Five participants were from educational organizations (teaching staff), and therefore they could not be considered to be representatives of supplier or customer organizations. One of the participants was from a customer organization. The organizations and the titles of the respondents are shown in tables 1 and 2, respectively.

Table 1. Respondent's by organization

Organisation	Number
Consulting office	1
Eduational organization	5
Customer organization representative	1
Supplier organization	16
Research organization	2
Total	**25**

Table 2. The title of respondent's

Title	Number
Account manager	1
Business manager	1
IT Management consultant	1
Management	1
Development manager	1
Consultant	1
Lead enterprise architect	1
Lecturer	5
Business manager	1
Project support specialist	1
Project manager	7
Client manager	1
Engineering manager	1
Sv Solution architect	1
Executive manager	1
Total	**25**

The respondents' answers were then classified into larger groups that were considered to be related to each other or to refer the same element. This classification and grouping was done by one researcher and checked by two researchers. Special attention was paid to including all the answers in the analysis.

Some respondents included several elements in the same sentence; such sentences were broken up in order to be analyzed. Some respondents did not remember to pay attention to the start-up phase of the project – some of them wrote about elements not related to the start-up phase.

The analysis is presented in more detail in the next section.

4 Results

The answers included 167 different expressions, or elements, which were grouped into 18 different categories. Even though elements were able to place under a few categories, we put single element once under one category only. The categorization was done by one researcher and checked by two researchers.

The results of the categorization are presented in Table 3. The column entry on respondents refers to how many respondents provided at least one element that could be included in specific category. We divided respondents into three categories in order to observe whether there are differences between respondents. The supplier column refers to how many respondents from a supplier organization provided an answer for that category. The column other refers respondents from other organizations, and the column PM provides the number of project managers' responses.

Table 3. Responses according to category

Categories	Respondents	Supplier	Other	PM
Goals	14	10	4	3
Project team	13	6	7	2
Resources	11	7	4	3
Project plan	9	5	4	2
Project manager	9	6	3	1
Communication	8	4	4	2
Management	8	7	1	2
Schedule	7	3	4	1
Co-operation	6	4	2	0
Responsibility	6	5	1	1
Customer	5	4	1	1
Risks	5	3	2	0
Scope	5	3	2	1
Change management	4	0	4	0
Quality	4	1	3	0
Contract	3	3	0	2
Effort	3	2	1	2
Project board	2	0	2	0

A brief description of categories:

- The highest frequency in the answers was goals. The supplier organizations' representatives paid much attention to the definition of the goals, and three answerers were PMs. The supplier representative stated that the project must clearly have clearly defined goals and objectives. Other answers were similar: interest groups must commit to the project goals, and the goals must be realistic. Some answers included phrases like mutual understanding and realistic goals. Three PMs stressed the importance of realistic objectives.
- The project team was considered also a very important element. The answers included various phrasings on team skills and professionalism. Motivation and business domain knowledge were also mentioned.
- The respondent has a mutual opinion that resources has to be sufficiently and realistic. Resources means other than human resource.
- The respondent were of the view that the project plan should be well done and realistic. One answer stated that the plan should include scope, schedule, workers, customer and supplier responsibilities, risks, assignment and delivery content.
- The importance of the project manager was agreed upon. Answers relating project manager contained that PM is accomplished, has some experience, and manage processes. Only one PM gave answer concerning project manager: Project manager gets or acquires information needed in order to have the project finished, and knows the project's interdependencies on the customer's other operations.
- Two elements regarding communication were highlighted: It has to be planned and is continuous; and communication among parties concerned has to be effective. This latter element related to the execution phase. Before the project start, it is important to plan how to communicate, but the execution phase reveals the quality of actual communication. Only one supplier representative stated that has to be planned before kick-off meeting. Two PMs emphasized the importance that of communication is working with in the project and between interest groups; all parties consider were aware their own role, responsibility and output level on schedule.
- Managers stated that every project needs to managed and monitored, management should be systematic and professional and management commitment to the project is necessary.
- The responses felt that project schedule needs to be realistic.
- For effective co-operation, all parties concern had to know one another. This element is concern more execution phase because the effectiveness cannot be checked before the start of the project. Project managers had no comments on co-operation.

- The respondent's opinions were that project responsibilities have to be clearly defined. In addition, the ownership of the project should be clear.
- The recording to respondents, each customer need invest enough people and recourses and the customer should be committed and active.
- The bulk of the answers were related to the management of risks. The PMs had no statement regarding risks.
- It was felt that the project scope has to be explicit.
- The answers were mainly related to change management and its role. PM, however, had a comment on change management.
- There was an insistence on clear definitions of quality and quality management processes being in place. The PM, however, had no comments concerning quality.
- A part of responses felt that there should be a consensus on the contract. The project managers insisted on contract not having to be redone after once is the project starts.
- Effort estimates, stated, it should be realistic.
- It was opined that the project board should be active and project board members should have well defined roles and responsibilities. The PMs had no response related to this.

Our results propose, in general, that different people have different opinions on the elements that make a project succeed. This is supported by earlier studies (e.g., [15], [20], [30-31], but our results show that this difference may be surprisingly clear. Supplier firms rated highly different elements compared to respondents from other organizations. The respondents from the supplier firms rated goals, resources and management very high. Therefore process improvement efforts that pay most of the attention to those elements may be likely to provide tangible benefits. The project managers have, however, their own view on elements which they consider important from the viewpoint of project success. Another interesting result was that the project managers did not have a single or even a few elements that make them to get the impression that the project will succeed.

5 Discussion

In this article, we have reported results from our study on the elements which enable the supplier's project manager to get an idea of the success of a new project. The study was conducted by presenting a fairly open questionnaire in a seminar. Although the study was exploratory, the results are relevant for outsourced software development performed by supplier firms and process improvement in general.

The results suggest that the importance of different aspects and factors in this field depends on the perspective. This is in accordance with previous research (e.g., [differently [15], [30], [32]). The elements which were considered important in terms of software development project success are different between the supplier respondents than to the other respondents.

From the supplier's perspective, it is important to be able to set clear and well-defined goals for the project. That will make it easier to manage the project and achieve a result that is profitable and ensure the customer satisfaction. A satisfied customer is important to supplier firms, as the results in [6] imply. According to our study, the other representatives seem to value the selection of the project team over the definition of the project goals. A possible reason for this could be that the project team is the part of supplier firm the customer is contact in, and the customer has seldom direct access to the supplier's resources. Hence, the project team is an important aspect in outsourced software development.

The study shows differences between the opinions of the supplier representatives and the other representatives. Generalization based on these results, however, is limited in some ways. The respondents were limited in number. Several respondents did not follow the guidelines given, and for that reason, the answers are not limited to the period at the beginning of the project. Despite these limitations, the study emphasizes the differences that are present in any outsourcing situation.

In order to have software process improvement efforts succeed in the case of outsourced software development, the difference of perspectives should be taken into account. Without that, the motivation for the successful completion of the project and impact of project result are likely to be less than desired.

6   References

1. Cerpa, N., Verner, J.M.: Why Did Your Project Fail? Communications of the ACM, 52, 130-134 (2009)
2. Charette, R.N.: Why Software Fails. In: IEEE spectrum 42, 42-49 (2005)
3. Lindberg, K.R.: Software developer perceptions about software project failure: a case study. The Journal of Systems and Software. 49, 177–192 (1999)
4. Savolainen, P.: Why Do Software Development Projects Fail? Emphasising the Supplier's Perspective and the Project Start-Up. Doctoral dissertation, Jyväskylä Studies in Computing, 136, Jyväskylä University Printing House, Jyväskylä. (2011)
5. Pinto, J.K., Mantel, S.J., Jr. .: The Causes of Project Failure. IEEE Transactions on Engineering management. 37, 269–276 (1990)
6. Savolainen, P., Ahonen, J.J., Richardson, I.: Software development project success and failure from the supplier's perspective: A systematic literature review. 30, 458–469 (2012)
7. Verner, J.M., Abdullah, L.M.: Exploratory case study research: Outsourced project failure. Information and Software Technology. 54, 866–886 (2012)
8. Wateridge, J.: IT projects: a basis for success. International Journal of Project Management. 13, 169–172 (1995)
9. Wateridge, J.: How can IS/IT projects be measured for success?. International Journal of Project Management. 16, 59–63 (1998)
10. Hyväri, I.: Success of projects in Different Organizational Conditions. Project Management Journal. 37, 31-41 (2006)
11. Munns, A.K., Bjeirmi, B.F.: The role of project management in achieving project success. International Journal of Project Management. 14, 81–87 (1996)
12. Nasir, M.H.N., Sahibuddin, S.: Critical success factors for software projects: A comparative study. Scientific Research and Essays 6, 2174–2186 (2011)
13. Belassi, W., Tukel, O.I.: A new framework for determining critical success/failure factors in projects. International Journal of Project Management. 14, 141–151 (1996)
14. Dvir, D., Lipovetsky, S., Shenhar, A., Tishler, A.: In search of project classification: a non-universal approach to project success factors. Research Policy 27, 915–935 (1998)
15. Shenhar, A.J., Tishler, A., Dvir, D., Lipovetsky, S., Lechler, T.: Refining the search for project success factors: a multivariate, typological approach. R&D Management. 32, 111–126 (2002)
16. Wateridge, J.F: Delivering Successful IS/IT Projects: Eight Key Elements from Success Criteria to Review via appropriate management, methodologies and teams. Doctoral dissertion, Henley Management College, Brunel University (1996)
17. Andersen, E.S., Birchall, D., Jessen, S.A., Money, A.H.: Exploring project success. Baltic Journal of Management. 1, 127–147 (2006)
18. Gällstedt, M.: Working conditions in projects: perceptions of stress and motivation among project team members and project managers. International Journal of Project Management. 21, 449–455 (2003)
19. Belout, A., Gauvreau, C.: Factors influencing project success: the impact of human resource management. International Journal of Project Management. 22, 1–11 (2004)
20. Turner, J.R.: The handbook of project-based management. 3rd edition. The McGraw-Hill Companies, USA (1999)
21. Müller, R., Turner, R.: The Influence of Project Managers on Project Success Criteria and Project Success by Type of Project. European Management Journal 25, 298–309 (2007)
22. Ika, L.A.: Project Success as a Topic in Project Management Journals. Project Management Journal. 40, 6–19 (2009)
23. Atkinson, R.: Project management: cost, time and quality, two best guesses and a phenomenon, its time to accept other success criteria. International Journal of Project Management. 17, 337–342 (1999)
24. Shenhar, A.J., Dvir, D., Levy, O., Maltz, A.C.: Project Success: A Multidimensional Strategic Concept. Long Range Planning 34, 699–725 (2001)
25. Jugdev, K., Müller, R.: A Retrospective Look at Our Evolving Understanding of Project Success. Project Management Journal. 36, 19–31 (2005)
26. Pinto, J.K., Slevin, D.P.: Critical Factors in Successful Project Implementation. IEEE Transactions on Engineering Management. 1, 22–27 (1987)
27. Pinto, J.K., Prescott, J.E.: Variations in Critical Success Factors Over the Stages in the Project Life Cycle. Journal of Management. 14, 5–18 (1988)

28. Müller, R., Jugdev, K.: Critical Success Factors in Projects: Pinto, Slevin, and Prescott – the elucidation of project success. International Journal of Managing Projects in Business. 5, 757–775 (2012)
29. Ahonen, J.J., Savolainen, P.: Software engineering projects may fail before they are started: Postmortem analysis of five cancelled projects. The Journal of Systems and Software. 83, 2175–2187 (2010)
30. Lim, C.S., Mohamed, M.Z.: Criteria of project success: an exploratory re-examination. International Journal of Project Management. 17, 243–248 (1999)
31. Thomas, G., Fernández, W.: Success in IT projects: A matter of definition?. International Journal of Project Management. 26, 733–742 (2008)
32. De Wit, A.: Measurement of project success. International journal of project management. 6, 164–170 (1988)
33. Kappelman, A.L., McKeeman, R., Zhang, L.: Early Warning Signs of it Project Failure: The Dominant Dozen. International Systems Management. 23, 31–36 (2006)

7   Authors' CVs

Jaana Nevalainen

Jaana Nevalainen is a Researcher at the School of Computing at the University of Eastern Finland. Her research interests include software development project success factors from supplier perspective. She is starting her PhD research from this area.

Paula Savolainen

Paula Savolainen is a Project Manager at the School of Computing at the University of Eastern Finland. She received her DSc (Econ.) from the University of Jyväskylä in 2011. She has over 12 years experience in the software industry as a Developer, Systems Analyst, and Project Manager. She has worked as an IT coordinator in the Pohjois-Savo region, Finland, being responsible for general development of the ICT industry and the public sector, and has coordinated projects in Pohjois-Savon liitto, Savonia University of Applied Sciences, the University of Kuopio, and the University of Eastern Finland. Her current research concentrates on project success and the project start-up phase in supplier firms. Her articles have been published in Journal of Systems and Software, International Journal of Project Management, and in various book series (e.g. LNCS and LNBIP).

Jarmo J. Ahonen

Jarmo J. Ahonen is a Full Professor at the School of Computing at the University of Eastern Finland. Before joining University of Kuopio, which was later merged with University of Joensuu to create University of Eastern Finland, he worked in various management positions in both research and educational organizations. Before that he worked in software maintenance and development. His main fields of research are software project management, software quality, software engineering methodologies and software process improvement. He holds a PhD in Computer Science from the University of Joensuu.

High Maturity Levels: Achieving CMMI ML5 in a Consultancy Company

Lucas Grossi
Junior Consultant at AXPE Consulting
lgcgrossi@gmail.com

Jose A. Calvo-Manzano, Tomas San Feliu
Universidad Politécnica de Madrid
{joseantonio.calvomanzano, tomas.sanfeliu}@upm.es

Abstract

Most organizations that use the CMMI stop their process improvement journey at Maturity Level 3 or less. The discussion about CMMI high maturity levels has always been controversial, going most of the times through the consideration of return of investment, provider selection and interpretation issues. Achieving Level 5 is not an easy task; it derives a lot of steps during the way. This article will show an example of the implementation of Level 5 in a consultancy company in two constellations: Development and Service. It depicts the example in steps, to help the understanding of the whole process.

Keywords

CMMI-DEV, CMMI-SVC, High Maturity, Consultancy.

1 Introduction

The discussion about CMMI high maturity levels has always been controversial, going most of the times through the consideration of return of investment, provider selection and interpretation issues. Quotes like "it is too expensive", "I don't see benefits" and "My client don't ask for level 5" are usual. Over 90 percent of organizations submitting appraisal results to SEI are Maturity Level 3 or lower [1]. What could be discussed is why so many companies stay on Level 3 and why they should improve to Level 5. According to Campo [2], Level 5 can be the most cost effective of all the maturity levels.

However, achieving the Level 5 is not an easy task. Much of the hard work it takes to get there, and stay there, and the many experiences gained along the way may be forgotten in the afterglow of success. Many of the most difficult obstacles on the road to high maturity are related to measurement and analysis, since it is needed a well-defined and well-implemented measurement collection and analysis process.

The objective of this paper is to show the benefits of high maturity levels and how to achieve it in an IT company, based on the author experience in coordinating the implementation of CMMI ML-5 (Maturity Level 5) in two constellations (Service and Development) in a consultancy company. The paper will depict the way to achieve a high maturity level in steps, with examples of problems faced during it.

2 High Maturity Levels of CMMI

High maturity controversy dates back to the CMM years. When the CMM was released, few prototypes of high maturity organizations existed. The most oft-cited example at the time was the Space Shuttle Onboard Software project at IBM-Houston [2].

The number of CMM high maturity projects increased after the 1999 publication of a memo from the Office of the Undersecretary of Defence set an expectation of CMM Maturity Level 3 for software development contractors of ACAT I programs. If CMM Maturity Level 3 was an expectation, then CMM Maturity Level 5 was seen as a key discriminator for winning contracts. This pattern continued with the release of CMMI. If CMM Maturity Level 5 was a discriminator, then CMMI Maturity Level 5 was seen as an even greater discriminator [2].

To be in a high maturity level means that an organization:

- understands why they are doing what they are doing.

- knows "what to do" when problems are encountered (don't overreact to special causes – concentrate on finding common causes).

- defines their error-proof processes to allow for human fallibility.

- converts "blame" into "opportunity" (avoid using fear as a motivator).

- balances "empowerment" and "ownership" with "control".

- measures and predicts how much further they have to go to achieve their goals.

While statistical thinking and an understanding of variation are intrinsic to the definition of Levels 4 and 5, other factors that have been empirically observed —such as capturing product knowledge and addressing the human issues associated with process improvement and change management— are also crucial to continual improvement [3].

3 Benefits of CMMI 5

Attaining CMMI Level-5 does not guarantee that all process performance issues have been addressed. Nevertheless, you probably have a better understanding than ever before of how much room you have in your organization for improvement. Attaining Level-5 means that it has been officially rec-

ognized that you have the processes, tools, skills, and other resources and infrastructure in place that are necessary to properly collect, analyse and address these opportunities for improvement.

Level-5 is not the end of the journey, but rather the end of the first step and the beginning of the rest of the expedition.

One organization has probably invested millions or even tens of millions of dollars learning how to learn, change and improve. It has also hopefully built up a stable and proven process improvement infrastructure filled with process documentation, process group activities and process training materials. Now that the company have got the people in Engineering, Project Management and maybe even Configuration Management and Quality Assurance under process control, it can turn their process improvement catapults towards remaining organizational bastions of less-than-ideal processes like Business Development and Business Operations.

Nevertheless, achieving CMMI 5 also brings some benefits. This level of maturity is much more strategic focus, and this focus is built around establishing and managing against quality and process performance objectives that are aligned with business objectives. The processes establish a system of continuous evaluation and maintenance of business objectives, and the associated quality and process performance objectives. Progress against those objectives is analysed, and process improvements are identified based on their contribution towards achieving the objectives. Causal analysis and resolution techniques are used in support of these activities. On this level of maturity, the company is much more strategic focus.

According to Kumar [4], Infosys obtained in 5-year period 73% Improvement in delivered quality, 39% improvement in on-time delivery, 34% improvement in effort estimation accuracy and 4% improvement in overall defect removal effectiveness. Another study from Bang and Schneider [5] showed that LG CNS obtained in a 2 year period a decrease of 4.74% in the re-delivery ratio and 1.76% increase in the CSR delivery ratio.

According to Goldenson et. al. [6], the types of benefits that a company can get with CMMI 5 are related to:

- Process Adherence: work product completion improvement, reduction in cost of poor quality.

- Cost: decrement in the average cost to fix a defect, reduction in unit software costs, decrement in defect finding and fixing costs, increment in cost estimations accuracy.

- Schedule: reduction in turn around time, increment on the percentage of milestones met, decrement in the average number of days late, reduction in schedule variance.

- Productivity: improvement in productivity, improvement in reuse of software.

- Quality: reduction of software defects, reduction of defects rate, increment of focus on quality by developers, improvement in defect removal.

- Costumer Satisfaction: increment in award fees, increment in costumer satisfaction rating average.

- Return on Investment: increment of ROI, reduction of project cost.

4 Steps for Achieving Level 5: An Example in a Consultancy Company

Between the challenges to achieve the high maturity it is possible to mention: lack of understanding of model requirement, absence/ineffective management support, difficulty in communicating the benefits of high maturity to the organization, difficulty in implementing high maturity practices effectively therefore not maximizing business benefits and resistance to adoption and sustenance of culture change.

In the next sections, it will be discussed 4 steps to achieve the CMMI ML-5 in a consultancy company, with real situations faced during the implementation of this maturity level for two constellations, service and development.

4.1 Analysis of Objectives

Figure 1 shows the steps to follow during the analysis of objectives:
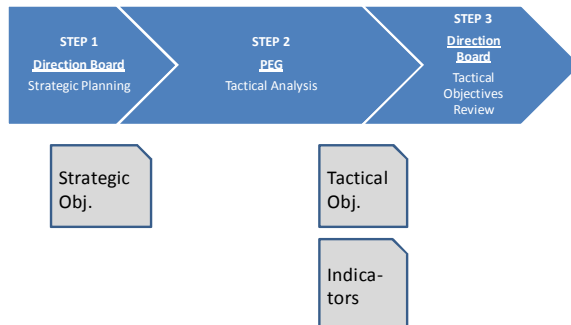
Figure 1: Analysis of objectives flow.

During the Step 1, the vision of the Director Board and its perspective on the business objectives and relevant stakeholders is unavoidable. This involvement is essential to set the objectives priorities. A widespread misconception is to delegate this activity to intermediate levels of the company, including to the process improvement team, that don't have the necessary vision to successfully undertake this activity, making a CMMI goal in itself.

Typical formulations of strategic business objectives include:

- In 2014 our revenue levels will be 50% above the current billing.

- In 2013 our share of Chinese market penetration will be 35% of the same.

- In 2014 our gross margin will be 30% higher than today.

- In 2014 the satisfaction of our customers will be 80% higher than today.

In the Step 2, the process improvement team will proceed to the hierarchical decomposition of the targets in indicators associated with the projects contribution and activities / processes relevant in those. The tactical vision will help to weigh the contribution to the strategic objectives of each of the activities and projects performed. The tactical objectives may be related to intermediate value of projects, final values of those and supplier activities.

Typical formulations of strategic tactical objectives include:

- Delivery deviation on intermediate milestones will be less than 3% of the length of them.

- Deviation of estimated cost vs. real cost lower than 5%.

- No Quality costs lower than 5%.

- Defects found by the client vs. defect found internally lower than 5%.

- Support cost during the 6 months after the deployment lower than 3% of the project cost.

The process improvement team will proceed to the analysis of data availability and validity, for each of the indicators defined. If these data did not exist, it will be formalized the gathering, recording and analysis of them, to proceed to the future analysis of the distribution of them. In the case that these data exist, the process improvement team will proceed to the analysis of distribution of each of the indicators, to determine their stability and proceed to the definition of quantitative objectives derived, this time considering the respective variations of them. The operations to be followed are:

- Analyse the mean and the standard deviation of the distribution.

- Consider the special causes of variation: mean $\pm 3\sigma$.

- Analyse the demographic data adjacent to each of them.

- Remove from the distribution if appropriate.

- Repeat the process until the sample doesn't have points out of control. This action should not result in a distribution that involves deletion of data higher than the one that would result from considering a distribution with values between: mean ± 2σ, with 80% of the total data.

- The resulting distribution is discussed whether to define achievable targets for improvement. If so, it is set the list of objectives with that scenario of improvement.

Not all defined objectives are applicable to all the projects. Some situations may involve decisions and prioritization scenarios, which may be affected: by organization strategic objectives which may prioritize ones respect to others; by project requirements which may be precedent for that project; by additional requirements developed by some stakeholder. Recall that the objectives set out in this operation, respond to the ability of the projects to deliver results on those margins. If it would be needed to significantly improve these specific objectives, it will be necessary to put in operation improvement initiatives to give that result.

To allow the comparison between data of different projects it will be needed to normalize each comparison indicators, such that the resulting ratios allow their horizontal analysis. The denominator, which usually best fits to the effective data normalization, is the indicator of project size or activities in which the indicator is relevant. The metrics used comprise number of pages of a document, number of function points, number of use case points, LOC, etc.

Finally in Step 3, the tactical objectives and the indicators derived are reviewed by the direction board, to prove the validity of the same. In this step, it could be needed a new iteration, to align the objectives with the needs developed by the direction board.

4.2 Baseline Consolidation

Not all projects are good candidates for baseline. It is necessary to select processes and attributes, which have attributes to help us to understand the ability to achieve the business objectives. This selection process should be following a systematic process using criteria. Typical criteria that can use for the selection of a process or sub-process include: the process or sub-processes is related to key business objectives or is an important predictor of performances. Once the process or sub process have been selected, is helpful to establish indicators that provide insight into the quantitative objectives.

It is especially relevant the concept of sub-process, "One set of activities in which the allocation of special causes of variation can be uniquely associated with demographic events or conditions internal to the same" [1].

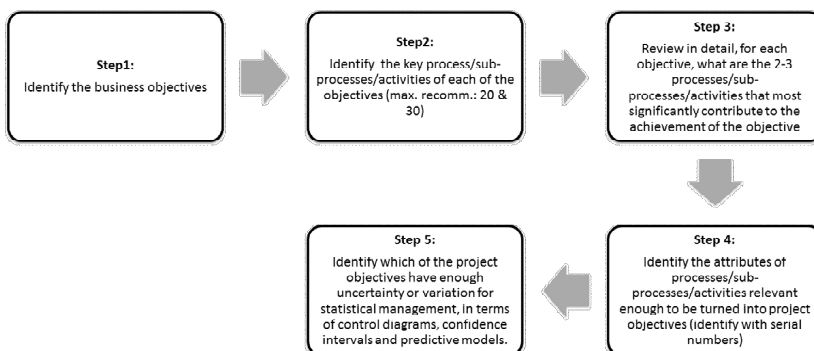Figure 2 shows the steps to be followed during the baseline consolidation.

Figure 2: Baseline consolidation flow.

During the analysis of historical data it should be clear that not all date are useful, since many of them may come from defined processes different from the actual ones, or be deficient in context information to enable its classification. To avoid useless work, before continuing with any data analysis it is essential to validate the quality and usefulness of the available data. For this, it will be needed to: validate the data production frequency, confirm that there are no meaningless values in the data sample (e.g. negative values for the number of defects), tabulate the enumerative variables (variables that include characters such as experience of people who could be weighed as enumerative variable between 1 and 7), select the variables that would be analysed and verify the data for consistency, with many cross-checks as possible.

Once the data are validated, it should be loaded in a statistical analysis tool (e.g. Minitab) to check in the control charts the presence of events that violate the criteria of being above or below mean ±3σ, verifying if these values correspond to special causes of variation or if they are inherent to the process. After that, analyse the possibility that the sample can contain two or more distributions, where it will be necessary to dispose of demographic variables that allow the performance of an ANOVA analysis, that segregate the potential distributions, from the corresponding qualifier agents (this operation will result in baselines as many distributions can be found in the sample).

If any of the control limits derived from the addition or subtraction of 3 times the standard deviation on average result in a meaningless value in the variable considered (negative number of defects, etc.), the cut value must be forced to the corresponding significant threshold.

When the baseline is considered reliable and the values are understood in terms of its physical meaning, the project will be saved in the statistical analysis tool, recording the mean and the resulting control limits, for its use on different tools (Excel, proprietary database, etc.). Then, continue using the determined baseline until relevant indications of the need to update it appear, as: the impact of improvement initiatives, resulting in new baselines; review period of the baselines; justification for inclusion of "false alarms", resulting in process internal values.

But what happens if the company doesn't have the sufficient data or if the data that are available don't have the disaggregation level or are no reliable? Has the company to wait 3 years the access to the high maturity levels? The answer is no, since it is possible to use the Delphi method to capture those data that doesn't have relevant historical information. The operative to follow is to:

- Formalize the hypothesis that these data correspond to the stable distribution of the corresponding process.

- Complete the measurement program of the organization.

- Measure a significant sample.

- Perform a hypothesis test, comparing the distribution derived from the Delphi data, with the data actually measured.

During the intermediate process, it should be noted the uniqueness of the data in use, not overestimating the decisions arising from them.

4.3   Predictive Models Development

To develop the predictive models we have to consider some points:

- In essence they should have statistical, probabilistic root or be related to simulations.

- They should predict intermediate o final outputs of the projects.

- They should use as input controllable factors, associated to the sub-processes, for the predictions.

- They should model the variation of the input factors, to predict the output variation range.

- They should allow the analysis of scenarios "what-if" during the planning, the replanning that could be derived and the correlation of the problems.

- They should connect activities in early phases of projects, with the activities of the subsequent phases of the lifecycle.

- They should allow the parameters adjustment, throughout the project, to analyse its impact on success.

Some examples of expected use of the predictive models are:

- In the consolidation of the defined project process.

- In the project management, to analyse the expected results and adjust the process.

- In evaluating solutions to correct situations where the objectives of the project will not be satisfied, to continue with the defined process.

- In OPM (Organizational …) and CAR (Causal Analysis and Resolution), to evaluate proposals, seeking opportunities for improvement, etc.

- To analyse the expected economical impact from a set of improvements.

According to the type of predictive and output variables considered, different statistical methods could be selected. Table 1 shows these situations.

	Y Continuous	Discrete
Discrete (X)	ANOVA & MANOVA & Dummy Variable Regression	Chi-Square & Logit
Continuous (X)	Correlation & Regression	Logistic Regression

Table 1: Statistical method vs. predictive/output variable.

Table 2 shows what could be predicted using a specific controllable factor for each statistical method defined in Table 1.

ANOVA, Dummy Variable Regression

Using these controllable factors...	To predict this outcome!
Type of Reviews Conducted; Type of Design Method; Language Chosen; Types of Testing	Delivered Defect Density
High-Medium-Low Domain Experience; Architecture Layer; Feature; Team; Lifecycle model; Primary communication method	Productivity
Estimation method employed; Estimator; Type of Project; High-Medium-Low Staff Turnover; High-Medium-Low Complexity; Customer; Product	Cost and Schedule Variance
Team; Product; High-Medium-Low Maturity of Platform; Maturity or Capability Level of Process; Decision-making level in organization; Release	Cycle Time or Time-to-Market
Iterations on Req'ts; Yes/No Prototype; Method of Req'ts Elicitation; Yes/No Beta Test; Yes/No On-Time; High-Medium-Low Customer Relationship	Customer Satisfaction (as a percentile result)

Chi-Square, Logistic Regression

Using these controllable factors...	To predict this outcome!
Programming Language; High-Medium-Low Schedule compression; Req'ts method; Design method; Coding method; Peer Review method	Types of Defects
Predicted Types of Defects; High-Medium-Low Schedule compression; Types of Features Implemented; Parts of Architecture Modified	Types of Testing Most Needed
Architecture Layers or components to be modified; Type of Product; Development Environment chosen; Types of Features	Types of Skills Needed
Types of Customer engagements; Type of Customer; Product involved; Culture; Region	Results of Multiple Choice Customer Surveys
Product; Lifecycle Model Chosen; High-Medium-Low Schedule compression; Previous High Risk Categories	Risk Categories of Highest Concern

Regression

Using these controllable factors...	To predict this outcome!
Req'ts Volatility; Design and Code Complexity; Test Coverage; Escaped Defect Rates	Delivered Defect Density
Staff Turnover %; Years of Domain Experience; Employee Morale Survey %; Volume of Interruptions or Task Switching	Productivity
Availability of Test Equipment %; Req'ts Volatility; Complexity; Staff Turnover Rates	Cost and Schedule Variance
Individual task durations in hrs; Staff availability %, Percentage of specs undefined, Defect arrival rates during inspections or testing	Cycle Time or Time-to-Market
Resolution time of customer inquiries; Resolution time of customer fixes; Percent of features delivered on-time; Face time per week	Customer Satisfaction (as a percentile result)

Logistic Regression

Using these controllable factors...	To predict this outcome!
Inspection Preparation Rates; Inspection Review Rates; Test Case Coverage %; Staff Turnover Rates; Previous Escape Defect Rates	Types of Defects
Escape Defect Rates; Predicted Defect Density entering test; Available Test Staff Hours; Test Equipment or Test Software Availability	Types of Testing Most Needed
Defect Rates in the Field; Defect rates in previous release or product; Turnover Rates; Complexity of Issues Expected or Actual	Types of Skills Needed
Time (in Hours) spent with Customers; Defect rates of products or releases; Response times	Results of Multiple Choice Customer Surveys
Defect densities during inspections and test; Time to execute tasks normalized to work product size	Risk Categories of Highest Concern

Table 2: Predictive outcomes for different types of controllable factors.

When developing the predictive models, one should take into account three important barriers, as lack of convincing results that propitiate predictions, lack of connection of outputs with the sub-processes,; and insufficient knowledge of the domain and the processes.

A lack of convincing results that propitiate predictions is due by the lack of alignment of activities with business objectives. A lack of connection of outputs with the sub-processes may prevent to discover the relationship between a change in the sub-process and the final output. And insufficient knowledge of the domain and the processes can difficult the identification of predictive variables.

During the achievement of CMMI ML5 in the consultancy company, some predictive models were developed. In the development constellation it was created a defect injection/detection model with the objectives to predict the defect that are expected to be detected in the following phases and to predict the defects that will escape to the client. The hypothesis used in this model is that the defects injection follows a stable distribution, defined from the historical data (alternatively using the Delphi method).

To increment this model, is was added a no-quality cost model, with the objective of predicting the defects fixing effort founded during the project lifecycle, including the maintenance costs in the transition phase. This no-quality cost model will be nested to the injection/detection model, complementing it with the effort distribution for the defects fixing depending on the stage at which they were detected and the phase in which they were injected, determined from the historical data (alternatively, using the Delphi method).

This model was divided in five phases: analysis, design, implementation, testing, client testing and warranty. The defects are injected just in the first three phases and could be removed in any phase using a peer review. Figure 3 shows the model.
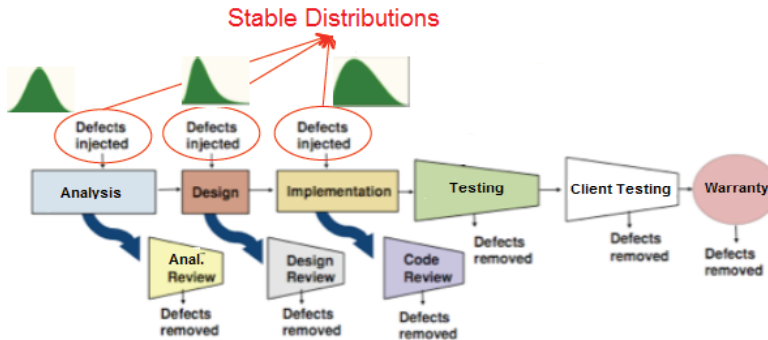
Figure 3: Defect injection/detection model.

The model also has input and output data, as showed below:

- Input data:
    - Density of injected defects: number of injected defects / the project size (in hours).
    - Percentage of defects removed by phase: in each phase the defects removed could be related to the same phase or previous phases, and this variable indicates this percentage. For example, in the implementation phase 10 defects were removed in the peer review, where 5 defects could come from the implementation phase, 3 defects from the design phase and 2 defects from the analysis phase, and this variable will indicate 50%, 30% and 20% respectively.
    - Removing efficiency by phase: number of defects removed in the phase / (defects escaped from the previous phase + defects injected in the phase).
    - Defect correction effort by phase: effort in hours to fix each defect in the phase.
- Output data:
    - Injected defects by phase: (density of injected defects * size of projects) + escaped defects from previous phase.
    - Defects removed by phase: injected defects by phase * detection efficiency by phase.
    - Escaped defects by phase: injected defects in the phase – removed defects in the phase.
    - Total effort by phase: sum of resolution effort of removed defects in each phase, where the resolution effort is calculated as – defects removed in the phase * percentage of defects removed in the phase.

For each development phase, a distribution was defined for each input data following the analysis of historical data. During this analysis, it was discovered that the development projects of the company could be divided in two groups, big (size > 1300 hours) and small (size < 1300 hours) projects. Based on this, the distribution of the input variables will be different depending on the size of the project. These two models (that work together) offer the possibility to compare the results (expected number of defects and the expected defects fixing effort) with the defined project objectives.

The models can be used:

- In planning to consolidate the project's defined process, being able to choose between taking or not the defect removing activities and being able to select the technique associated with such activities, which would result in different baselines and, consequently, different outcomes, which will be compared with project objectives.
- During project monitoring and control, from the actual results of the phase, analyse the degree of confidence to achieve the project objectives.

- During project monitoring and control, for decision making of the impact of corrective actions defined on project objectives.

- In the decision on improvement actions, to analyse its impact on the projects.

4.4   High Maturity Levels Impact in the Organization Agents

In the Maturity Levels 2 and 3 the development and implementation of processes are the important things to do: having project plans, managing against project plans, and identifying process improvements. Note, however, that the Maturity Levels 4 and 5 focus is on quality and process performance objectives derived from business objectives. Flowing the quality and project performance objectives down to programs, and using quality and process performance objectives as the basis for process improvement activity, are what sets the stage for the greater return on investment that may be realized from Maturity Levels 2 and 3. A business can only be successful if its programs are successful. At Maturity Levels 4 and 5, the entire organization becomes enlisted in helping the business achieve its objectives. Programs have to manage against those objectives, report status to higher-level management regularly, and take actions when the objectives are not being achieved. Programs in turn may establish their own quality and process performance objectives, based on achievement of award fees or other significant results [2].

In the high maturity levels, at the management level, the progress reports start to be oriented to the analysis of the likelihood of achieving the objectives, its evolution and the impact of corrective actions at that level of confidence. The project managers use the predictive models to consolidate the process defined and to control the probability to achieve the objective; and use the baseline for the statistical control of critical sub-processes.

The quality and improvement teams use the predictive models to analyse the impact of the improvements, and should review the consistent use of the predictive models to consolidate the project's defined process and to analyse the opportunity of meeting the objectives defined. Moreover, the consistent use of the process efficiency baselines process for the process statistical control should be reviewed.

5 Conclusions

What an organization gets out of CMMI-based process deployment and appraisals is a function of what the organization puts into it. Organizations that focus on maturity level ratings and CMMI minimal compliance are unlikely to derive benefits from their investment. Organizations that use the high maturity principles to deploy meaningful process improvement aligned with business objectives are organizations that are much more likely to reap greater return from their investment.

Maturity Levels 2 through 5 all offer benefits. Maturity Levels 2 and 3 help preventing disasters and gain control in the way work is performed in an organization. There is no denigrating in the improvements an organization can realize from implementing Maturity Level 2 and 3 processes.

However, Maturity Levels 2 and 3 are not focused on quality and process performance objectives as the driver of process improvement activity, and therefore set a lower ceiling on the benefits of CMMI-based process improvement. Using Maturity Level 4 and 5 processes to manage against quality and process performance objectives create a grass roots movement within an organization to meet business objectives. An organization where all individuals recognize their role and responsibility for business success is an organization that is more likely to achieve success.

Besides, the usage of the four steps showed in this work (analysis of objectives, baseline consolidation, predictive models development, high maturity levels impact in the organization agents) help the company during this process of achieving the high maturity levels.

6 References

1. CMMI Product Team: CMMI for SCAMPI Class A Appraisal Results 2012 Mid-Year Update. SEI, Carnegie Mellon University, (2012).

2. Campo, M.: Why CMMI Maturity Level 5?. Crosstalk Magazine, (2012).

3. Paulk, M. C; Goldenson, D.; White, D. M.: The 1999 Survey of High Maturity Organizations. Special Report CMU/SEI-2000-SR-002, (2000).

4. Kumar, S..: Tangible business benefits through higher process maturity – Infosys Case Study. SEPG 2007 North America Conference, (2007).

5. Bang, S.; Schneider, H.: Real-World Examples from a Successful Maturity Level 5 Organization. Case Study: The Korean Supreme Court Project, SEPG 2011 North America Conference, (2011).

6. Goldenson, D. R.; Gibson, D. L.; Ferguson, R. W.: Why Make the Switch? Evidence about the Benefits of CMMI, SEPG 2004 North America Conference, (2004).

Author CVs

Lucas Grossi

He holds a PhD in Computer Science by the Universidad Politécnica de Madrid and actually works as a junior consultant at AXPE Consulting, responsible for process improvement and the quality department. As process group leader for AXPE Consulting, Lucas coordinates the certifications in CMMI ML-5 in the Service and Development constellations, and also the TMMI ML-3, both in 2012.

Jose A. Calvo-Manzano

He holds a PhD in Computer Science. He is an assistant professor in the Facultad de Informática at the Universidad Politécnica de Madrid. He is teaching in the area of Software Engineering, specifically in the area of software process management and improvement. He has participated in more than 20 research projects (European, Spanish Public Administration as well as with private enterprises). He is author of more than 50 international papers. He is author of books related to software process improvement and software engineering topics also. He is the Head of the "Cátedra for Software Process Improvement for Spain and Latin American Region", where he has been a member of the team that has translated CMMI-DEV v1.2 and v1.3 to Spanish. He also holds the ITIL® v2 and v3 Foundation, and CMDB Certification

Tomas San Feliu

He holds a PhD in Computer Science. He is an assistant professor in the Facultad de Informática at the Universidad Politécnica de Madrid. He is teaching in the area of Software Engineering, specifically in the area of software process management and improvement. He has participated in more than 20 research projects. He is author of more than 50 international papers. He is author of books related to software process improvement and software engineering topics also. He has been a member of the team that has translated CMMI-DEV v1.2 and v1.3 to Spanish.

An ISO/IEC 15504 conformant Enterprise SPICE based security process capability model

Stéphane Rimas Kalpokas[1], Antanas Mitasiunas[1]

Vilnius University[1], 3 Universiteto Street, Vilnius, LT-01315, Lithuania
rimas.kalpokas@gmail.com, antanas.mitasiunas@maf.vu.lt Cortina, Alain Renault, Michel Picard

Abstract

In the context of modern information systems, security became one of the most critical quality attributes. The purpose of this paper is to address the problem of quality of information security. An approach to solve this problem is based on the main assumption that security is a process oriented activity. According to this approach, product quality can be achieved by the means of process quality – process capability. Introduced here, SPICE conformant security process capability maturity model is based on process capability modeling elaborated by world-wide software engineering community during the last 25 years, namely ISO/IEC 15504 that defines the capability dimension and the requirements for process definition and domain independent integrated model for enterprise-wide assessment and improvement Enterprise SPICE.

Keywords

Security, information security, capability maturity model, SPICE, Enterprise SPICE.This paper presents three process assessments that took place in Luxembourg in 2012. It depicts the TIPA® process assessment framework, and reports on its usage in three different partially out-sourced contexts: a telecommunication service provider, a European institution, and a bank. Finally, this paper presents the lessons learned from these experimentations and some improvement perspectives.

1 Introduction

Some three decades ago, software developers started to seek for established and confirmed procedures and solutions to cope with software crisis that was caused by recurrent exceeding of project cost and schedule as well as failure of functionality and quality. Inspired by traditional engineers, software engineering community has developed standards and models such as ISO/IEC 15504 and CMMI that have been used by numerous software organizations around the world for guiding tremendous improvements in their ability to improve productivity and quality. The concept of software process capability, which expresses process predictability, became an efficient working tool for process and product quality management.

The results of software engineering in terms of software process are generalized to any process capability assessment and improvement. In their turn, other "soft" engineers, e.g. innovation, follow pioneering way of software engineers. Software engineering being an extremely creative activity has been able to express it in process oriented terms. Developed and validated enhanced innovation and technology transfer process capability maturity model [1, 14] is another successful confirmation of the possibility to express in process oriented terms such creative activity as innovation.

The purpose of this paper is to provide a new approach for capability maturity modeling and to develop ISO/IEC 15504 conformant security process capability maturity model Security Process Capability Model as a core element of the approach proposed.

The state of the art in process capability maturity modeling and security process modeling is provided in the Sections 2 and 3. The Sections 4 and 5 contain authors' contribution to process capability modeling and security process modeling. The last Section concludes paper results achieved and provides future work to be done to complete the solution of the problem addressed.

The main idea for the modeling approach taken in this paper and the construction of a new primary process area is based on a related work done in [13].

2 Motivation and Capability Modeling

Security process capability model, introduced here, is based on process capability maturity modeling elaborated by a world-wide software engineering community. Software engineering community has considerably contributed to the state of the art of process modeling. The numerous attempts to solve the software crisis applying technological and methodological approaches were not successful. Consequently software engineers turned to the software development organizational issues aiming to keep software projects within planned scope, schedule and resources.

This approach is based on the assumption that product quality can be achieved by the means of process quality – process capability. High process capability cannot be established at once during the launch of activity. Process capability can be improved applying iterative procedure of process capability assessment and improvement.

Process capability is related to process predictability. Organizational maturity expresses the way organization activities are performed. The idea of maturity expresses the improvement path of organization activities to achieve better results. Process capability concept enables to measure the state of performance of organization's activities and to plan individual steps for processes capability improvement.

The research in this area is based on ideas originated from capability maturity models (CMM) developed since 1987 by Software Engineering Institute (SEI) of Carnegie Melon University. These models have evolved into CMMI version 1.3 [2-4] known as CMMI for Development, CMMI for Acquisition and CMMI for Services.

In parallel, the international community has developed an international standard for process assessment ISO/IEC 15504: Process assessment framework, also known as project SPICE (Software Process Improvement and Capability dEtermination) initiated by the Ministry of Defence of UK in 1991 [8, 9].

ISO/IEC 15504 represents the third generation of process capability maturity models that refer to an external process reference model. The process capability assessment framework is defined in the normative part of ISO/IEC 15504-2.

In this context, an approach taken by ISO/IEC 15504 [8, 9] referring to the external process reference model is particularly important. It enables to extend model's application area outside the software engineering. External process reference model must satisfy requirements of process definition in terms of process purpose and outcomes.

Third main source in process capability maturity arena is iCMM v2.0 (integrated Capability Maturity Model), leading to the issues of model integration and architecture representation, developed by US Federal Aviation Administration in 2001. It influenced a lot the current state of CMMs area [6] and is along the same lines as ISO/IEC 15504 (SPICE) and CMMI models. Based on external process reference model approach, the convergence of SPICE and iCMM models is possible and, in fact, it is completed as Enterprise SPICE initiative, i.e. the model FAA iCMM plays the role of baseline in the development of SPICE based Enterprise Process Reference Model and Process Assessment Model. Enterprise SPICE model consists of Process Reference Model supplemented by Process Assessment Model. Enterprise SPICE has been developed by a joint effort of more than one hundred experts representing 31 countries from all continents. Enterprise SPICE is the most challenging process capability assessment and improvement initiative for the last several years. The first stage of Enterprise SPICE [5] project is completed and the draft of the future standard is publicly available.

Hundreds of various generic and specific organizational maturity models have been developed. These models mainly provide the characteristics of maturity levels. However, very few of them provide a decomposition of activity modeled as a collection of processes defined in minimal terms, namely, a process name, a process purpose and the process outcomes.

3 Security Process Modeling Related Work

Security is a quality attribute of a system that is often implied because of the technical difficulty to prove or demonstrate otherwise. Because of that, security engineering aspect of system development often starts in the requirement specification as a sort of a set of preemptive technical and non-technical measures that are felt to increase security and ends with the implementation of said measures. When actual security issues occur, however, the measures are improved individually and often the systematic causes of the vulnerabilities exploited are overlooked. To avoid such situation a way to continuously monitor and improve security measures is needed. To achieve this, dedicated security-focused processes must be defined and institutionalized. Given the process-based view of security, for its quality to increase process performance should be continuously evaluated (assessed) and improved. Objective assessment is deemed impossible without a reference model, which, in this case, is a security process reference model. Once the security process reference model is developed, process capability determination and improvement, and therefore, systematic increasing security is a much easier task. Moreover, the need for systematic control, assessment and improvement of security is growing with the development of various complex information systems, cloud computing being the primary example. Cloud computing relies on trust between cloud service provider and consumer. It is believed that trust must be based on something provable, i.e. a certification. Certifications of various important aspects of cloud computing providers are foreseen to be required and some of them are already in development [16]. Certifications usually have a reference, to which the system evaluated can be qualitatively compared. Security certification, i.e., can be based on process capability evaluation using the Security Process Capability Model presented in this paper.

Enterprise SPICE can be seen as a universal tool for modeling various process-oriented activities that comprise an organization's information processing system (the term "information processing system" is understood as including every element of an organization that is producing, transferring or using information that its processes manipulate, including hardware, software, people and infrastructure). For the case of the model being universal and domain-independent, its main process areas cannot include application-specific processes. On the other hand, specific quality attributes such as security and safety are very important in every information-processing system. Therefore, foreseeing that security and safety might not be unique in this respect, "Special Applications" area was conceived introducing an additional process covering the specific area, namely *SAP.1. Safety and Security* [5, 7]. It defines Application Practices as goals to be achieved by implementing process areas in a way of in-

tentionally applying security and safety to base practices without naming them specifically. The knowledge of the concrete methodology of how these application practices are performed is implied rather than specified and therefore strongly relies on implementer. Therefore we can state that Enterprise SPICE defines security and safety as attributes applied to existing processes but not a process-based activity. The same, with some reservations, can also be said about the safety extensions to the ISO/IEC 15504 (Part 10) [10], +SAFE safety extensions to the CMMI-DEV [15] and the work done on security extensions to the ISO/IEC 15504 [12].

US Federal Aviation Administration viewed the iCMM as being insufficient in providing a framework for assessing and improving safety and security of a system has created and published Safety and Security Extensions for Integrated Capability Maturity Models [7]. It must be noted, that the evolutionary close relationship between the iCMM and Enterprise SPICE allows, with minor corrections, application of these extensions to the Enterprise SPICE. These extensions provide the relationship between Application Practices and specific Base Practices and additional implementation guidance.

The Systems Security Engineering Capability Maturity Model (ISO/IEC-21827, also SSE-CMM) is a process based representation of systems security engineering activity [10]. The security aspect of a system is central to this model and all the main process areas directly relate to this quality attribute throughout all the life cycle stages of a system. However, it is a standalone model, focusing on a single quality attribute and providing its own project and organizational process areas. And as such it is not ISO/IEC-15504 conformant and therefore does not easily relate to Enterprise SPICE context of capability process modeling. On the other hand, it provides a security engineering body of knowledge in a process-oriented way that can be reused in derivative works.

4   New Method of Capability Maturity Modeling

The goal that we had in developing the Security Process Capability Model is to build a framework that describes security as a process-based activity and is sufficiently detailed as a tool for any organization that wishes to assess and increase the capability of the security quality attribute of its processes.

The idea is to base Security Process Capability Model on Enterprise SPICE as much as possible because of its wide acceptance as a framework to process capability maturity assessment.

The supplementation of Enterprise SPICE with application area specific knowledge transforms it from a domain-independent model to domain-dependent model. While focusing on information security specifically potentially narrows its applicability, the model does not enforce any processes that would limit the set of organizations to Information Technology or related domains aside from having an information system, the security of which is the main focus of the model.

The method takes domain-specific knowledge codified in a published standard, in this case, the SSE-CMM, and extracts domain-specific processes. The organizational and support processes are seen as generic and are replaced with corresponding processes areas from Enterprise SPICE. In doing so, we enable the same framework of generic practices to apply to a newly-constructed Primary process category. In other words, as the Organizational and Support categories enable the capability dimension in Enterprise SPICE, they enable that in any newly-constructed Primary process category as long as it is SPICE (ISO/IEC 15504) conformant. Security Process Capability Model Primary Process category, in this case, is based on the eleven Security Process Areas described in the SSE-CMM [11].

5   Security Process Capability Model

Security process assessment and improvement is completely based on security capability model as a core tool for quality management. An idea is to build a new SPICE conformant Security capability model called Security Process Capability Model as an external Process Assessment Model according to requirements [8] using Enterprise SPICE capability model, that refers to the capability framework

defined in the normative part ISO/IEC 15504-2. The Process Reference Model of Security Process Capability Model consists of Primary, Organizational and Support process categories.

Organizational and Support process categories are reused from [5]. Primary process category is composed of three subcategories: Risk management process subcategory, Engineering process subcategory and Security process subcategory (see Figure 1 below). The Risk management and Security process subcategories are based on corresponding Process Areas from SSE-CMM. They contain the processes that represent the security domain-specific knowledge. The Engineering process subcategory comprises processes that are based on the Engineering process subcategory from Enterprise SPICE that functionally covers the remaining Security Process Areas from SSE-CMM.

According to ISO/IEC 15504-2, requirements for Process Reference Model process description must be done in minimal terms of process purpose and outcomes that are achieved as a result of process successful implementation. In addition to PRM Process assessment Model of Security Process Capability Model contains a set of indicators that explicitly addresses the purpose and outcomes, as defined in the PRM, and that demonstrate the achievement of the process attributes within. Description of Security Process Capability Model processes that belong to Primary process category, excluding the Engineering Process subcategory that is reused from Enterprise SPICE, is provided in Table 1 below.
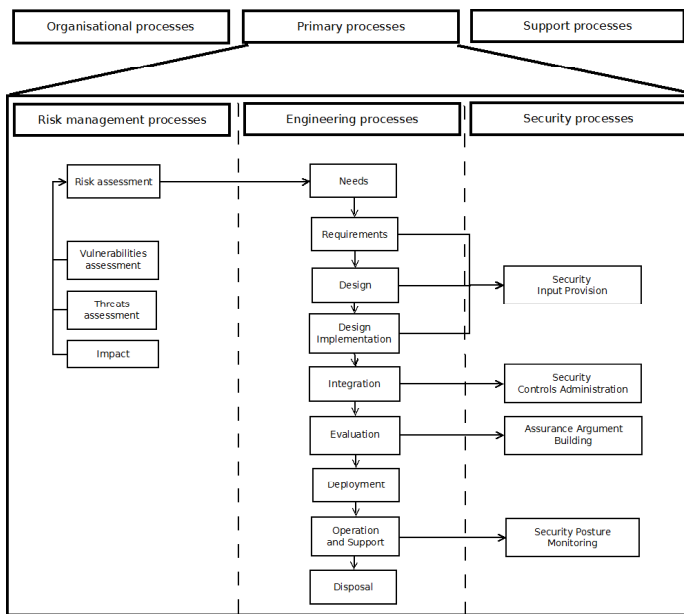

Fig. 1. Security Process Capability Model Primary process category and process relationships.

Table 2. Security Process Capability Model Primary process category excerpt.

PRIM.RISK.1. Vulnerability Assessment	
Purpose	**Outcomes**
To identify and characterize system security vulnerabilities	1) Vulnerability analysis method is selected; 2) Vulnerabilities are identified; 3) Vulnerability data is gathered; 4) System vulnerability is synthesized; 5) Vulnerabilities and their characteristics are monitored.
Base Practices	
PRIM.RISK.1.BP.1: Select vulnerability analysis method. [Outcome: 1]	

PRIM.RISK.1.BP.2: Identify vulnerabilities. [Outcome: 2]

PRIM.RISK.1.BP.3: Gather vulnerability data. [Outcome: 3]

PRIM.RISK.1.BP.4: Synthesize system vulnerability. [Outcome: 4]

PRIM.RISK.1.BP.5: Monitor vulnerabilities and their characteristics. [Outcome: 5]

PRIM.RISK.2. Threat Assessment

Purpose	Outcomes
To identify security threats and their properties and characteristics	1) Natural threats are identified; 2) Man-made threats are identified; 3) Threat units of measure are identified; 4) Threat agent capability is assessed; 5) Threat likelihood is assessed; 6) Threats and their characteristics are monitored.

Base Practices

PRIM.RISK.2.BP.1: Identify natural threats. [Outcome: 1]

PRIM.RISK.2.BP.2: Identify man-made threats. [Outcome: 2]

PRIM.RISK.2.BP.3: Identify threat units of measure. [Outcome: 3]

PRIM.RISK.2.BP.4: Assess threat agent capability. [Outcome: 4]

PRIM.RISK.2.BP.5: Assess threat likelihood. [Outcome: 5]

PRIM.RISK.2.BP.6: Monitor threats and their characteristics. [Outcome: 6]

PRIM.RISK.3. Impact Assessment

Purpose	Outcomes
To identify impacts that are of concern with respect to the system and to assess the likelihood of the impacts occurring	1) Capabilities are prioritized; 2) System assets are identified; 3) Impact metric(s) are selected; 4) Metric relationship is identified; 5) Impacts are identified and characterized; 6) Impacts are monitored.

Base Practices

PRIM.RISK.3.BP.1: Prioritize capabilities. [Outcome: 1]

PRIM.RISK.3.BP.2: Identify system assets. [Outcome: 2]

PRIM.RISK.3.BP.3: Select impact metric(s). [Outcome: 3]

PRIM.RISK.3.BP.4: Identify metric relationship. [Outcome: 4]

PRIM.RISK.3.BP.5: Identify and characterize impacts. [Outcome: 5]

PRIM.RISK.3.BP.6: Monitor impacts. [Outcome: 6]

PRIM.RISK.4. Security Risk Assessment

Purpose	Outcomes
To identify, analyze and evaluate the security risks involved with relying on a system in a defined environment	1) Risk analysis method is selected; 2) Exposures are identified; 3) Exposure risk is assessed; 4) Total uncertainty is assessed; 5) Risks are prioritized; 6) Risks and their characteristics are monitored.

Base Practices

PRIM.RISK.4.BP.1: Select risk analysis method. [Outcome: 1]

PRIM.RISK.4.BP.2: Exposure identification. [Outcome: 2]

PRIM.RISK.4.BP.3: Assess exposure risk. [Outcome: 3]

PRIM.RISK.4.BP.4: Assess total uncertainty. [Outcome: 4]

PRIM.RISK.4.BP.5: Prioritize risks. [Outcome: 5]

PRIM.RISK.4.BP.6: Monitor risks and their characteristics. [Outcome: 6]

PRIM.SEC.1. Security Input Provision

Purpose	Outcomes
To provide system architects, designers, implementers, or users with the security infor-	1) Security input needs are understood; 2) Security constraints and considerations are determined; 3) Security alternatives are identified; 4) Security of engineering alternatives are identified;

mation they need	5) Security engineering guidance is provided; 6) Operational security guidance is provided.

Base Practices

PRIM.SEC.1.BP.1: Understand Security Input Needs. [Outcome: 1]

PRIM.SEC.1.BP.2: Determine Security Constraints and Considerations. [Outcome: 2]

PRIM.SEC.1.BP.3: Identify Security Alternatives. [Outcome: 3]

PRIM.SEC.1.BP.4: Analyze Security of Engineering Alternatives . [Outcome: 4]

PRIM.SEC.1.BP.5: Provide Security Engineering Guidance. [Outcome: 5]

PRIM.SEC.1.BP.6: Provide Operational Security Guidance. [Outcome: 6]

PRIM.SEC.2. Security Controls Administration

Purpose	Outcomes
To ensure that the intended security for the system that was integrated into the system design is in fact achieved by the resultant system in its operational state	1) Security responsibilities are established; 2) Security configuration is managed; 3) Security awareness, training, and education programs are managed; 4) Security services and control mechanisms are managed.

Base Practices

PRIM.SEC.2.BP.1: Establish security responsibilities. [Outcome: 1]

PRIM.SEC.2.BP.2: Manage security configuration. [Outcome: 2]

PRIM.SEC.2.BP.3: Manage security awareness, training, and education programs. [Outcome: 3]

PRIM.SEC.2.BP.4: Manage security services and control mechanisms. [Outcome: 4]

PRIM.SEC.3. Assurance Argument Building

Purpose	Outcomes
To clearly convey that the customer's security needs are met	1) Assurance objectives are identified; 2) Assurance strategy is defined; 3) Security measures are defined; 4) Assurance evidence is controlled; 5) Evidence is analyzed; 6) Assurance argument is provided.

Base Practices

PRIM.SEC.3.BP.1: Identify assurance objectives. [Outcome: 1]

PRIM.SEC.3.BP.2: Define assurance strategy. [Outcome: 2]

PRIM.SEC.3.BP.3: Define security measures. [Outcome: 3]

PRIM.SEC.3.BP.4: Control assurance evidence. [Outcome: 4]

PRIM.SEC.3.BP.5: Analyze evidence. [Outcome: 5]

PRIM.SEC.3.BP.6: Provide assurance argument. [Outcome: 6]

PRIM.SEC.4. Security Posture Monitoring

Purpose	Outcomes
To ensure that all breaches of, attempted breaches of, or mistakes that could potentially lead to a breach of security are identified and reported	1) Event records are analyzed; 2) Changes are monitored; 3) Security incidents are identified; 4) Security safeguards are monitored; 5) Security posture is reviewed; 6) Security incident response is managed; 7) Security monitoring artifacts are protected;

Base Practices

PRIM.SEC.4.BP.1: Analyze event records. [Outcome: 1]

PRIM.SEC.4.BP.2: Monitor changes. [Outcome: 2]

PRIM.SEC.4.BP.3: Identify security incidents. [Outcome: 3]

PRIM.SEC.4.BP.4: Monitor security safeguards. [Outcome: 4]

PRIM.SEC.4.BP.5: Review security posture. [Outcome: 5]

PRIM.SEC.4.BP.6: Manage security incident response. [Outcome: 6]

PRIM.SEC.4.BP.7: Protect security monitoring artifacts. [Outcome: 7]

Conclusions and Future Work

The paper provides the following new results in process capability modeling and security process capability assessment and improvement:

1) A method for SPICE conformant process capability maturity modeling based on ISO/IEC 15504 capability framework and Enterprise SPICE domain independent external process model is proposed;

2) Based on the proposed methodology, a SPICE conformant Process Assessment Model of Security process capability maturity model called Security Process Capability Model is developed;

Following remaining future work should be done: validation of application sensitive process capability modeling approach versus application area implementation by referencing to base practices of domain independent process model; development of an approach to assessment and improvement of organization's security process based on the process reference model presented in this paper; evaluation of process oriented knowledge capability modeling as an application area implemented by base practices versus an approach of creation of a standalone, external model, by developing a new primary process category and reusing organizational and support process categories from Enterprise SPICE.

References

1. Boronowsky, M., Woronowicz, T., Mitasiunas, A. BONITA – Improve Transfer from Universities for Regional Development. The Proceedings of the 3rd ISPIM Innovation Symposium held in Quebec City, Canada - 12-15 December 2010 ISBN 978-952-265-004-7
2. CMMI-ACQ, 2010. CMMI for Acquisition, Version 1.3. Software Engineering Institute
3. CMMI-DEV, 2010. CMMI for Development, Version 1.3. Software Engineering Institute
4. CMMI-SVC, 2010. CMMI for Services, Version 1.3. Software Engineering Institute
5. Enterprise SPICE An Integrated Model for Enterprise-wide Assessment and Improvement. Technical Report - Issue 1. The Enterprise SPICE Project Team, September 2010, 184 psl., http://www.enterprisespice.com/page/publication-1
6. Ibrahim, L., Bradford, B., Cole, D., LaBruyere, L., Leinneweber, H., Piszczek, D., Reed, N., Rymond, M., Smith, D., Virga, M., Wells, C. FAA-iCMM, 2001. The Federal Aviation Administration Integrated Capability Maturity Model for Enterprise-wide Improvement. U.S. Federal Aviation Administration.
7. Ibrahim, L., Jarzombek, J., Ashford, M., Bate, R., Croll, P., Horn, M., LaBruyere, L., Wells, C., 2004. Safety and Security Extensions for Integrated Capability Maturity Models. U.S. Federal Aviation Administration.
8. ISO/IEC 15504-2, 2003. Information Technology – Process Assessment – Part 2: Performing an Assessment. International Standards Organization
9. ISO/IEC 15504-5, 2006. Information Technology – Process Assessment – Part 5: An Exemplar Process Assessment Model. International Standards Organization
10. ISO/IEC 15504-10, 2011. Information Technology – Process Assessment – Part 10: Safety Extension. Technical specification. International Standards Organization
11. ISO/IEC 21827, 2008. Information Technology – Security Techniques – Systems Security Engineering – Capability Maturity Model (SSE-CMM). Second Edition. International Standards Organization
12. Mesquida, A. L., Mas, A., Amengual, E. An ISO/IEC 15504 Security Extension. Software Process Improvement and Capability Determination. Series: Communication in Computer and Information Science (ISSN 1865-0929), Vol. 155. ISBN 9783642212321 p. 64-72
13. Mitašiūnas, A., Novickis, L. Enterprise SPICE based education capability maturity model. Workshops on business informatics research. Proceedings. Series: Lecture Notes in Business Information Processing (ISSN 1865-1348), Vol. 106. ISBN 9783642292309 p. 102-116
14. Novickis, L., Lesovskis, A., Mitasiunas, A. Technology Transfer Model and Web-based Solution for Transport Logistics Service Providers. Proceedings od the European Computing Conference (ECC'11), Paris, France, April 28-30, 2011.
15. +SAFE, 2007. A Safety Extension to CMMI-DEV, version 1.2. Software Engineering Institute
16. Sunyaev, A., Schneider, S. Cloud Services Certification. February 2013. Communications of the ACM. Volume 56, issue 2, p. 33-36

An approach to manage the concept phase of ISO 26262

Masao Ito
Nil Software Corp.
2-17-7, Kinuta, Setagaya, 1570073, Japan
nil@nil.co.jp

Koichi Kishida
Software Research Associates
4-1, Kioi-cho, Chiyoda, 1028578, Japan
k2@sra.co.jp

Abstract

ISO 26262[1] is a functional safety standard in the automobile field and it requires distinct safety requirements. Usually it is not easy to divide requirements into safety parts and non-safety parts because they are closely related to each other. That is, we have to perform two activities, functional development and functional safety activity, simultaneously. Other difficulty is a term *item*. From the definition the *item* is a "system (1.129) or array of systems to implement a function at the vehicle level". In concept phase we apply hazard analysis to an *item*, not system. The system definition comes after item definition and hazard analysis and risk assessment. So, it is hard to use the conventional methods (e.g. FMEA, FTA).

To support this situation we propose a method and a tool. Our method is an extension of KAOS (Knowledge Acquisition in autOmated Specification) and we also use the GSN (Goal Structuring Notation) and Scenario-Situation Matrix (SSM). We aim to support the scope of part 3 of ISO 26262. But we believe this approach is generic and can be used in a wide range of fields not limited to the automobile field.

Keyword

ISO 26262, functional safety, requirements, KAOS, GSN, DSPO, Hazard Analysis

1 Introduction

There are many methods to check a system from the viewpoint of safety. But it is hard to re-design a system after checking it. The functional safety standard for automobile, ISO 26262 requires the Hazard analysis and Risk assessment before system design.

First we indicate the characteristics of ISO 26262 and issues of it (chapter 2), then we present our approach to fit into the standard (chapter 3) and we compare them to the related works (chapter 4). Finally we conclude them (chapter 5).

It is important to clarify the way of work in concept phase, because we check or assess this activities based on the better understanding of them.

2 Characteristics of ISO 26262 in early phase

We focus on the early phase of the process that ISO 26262 requires, that is the concept phase. We consider the two characteristic points in this phase. One is the special term *item* in this standard. The second is the safety case. It is defined in the management of functional safety (part 2) and used whole lifecycle (2-2.4.6). So, we have to consider it also in concept phase.
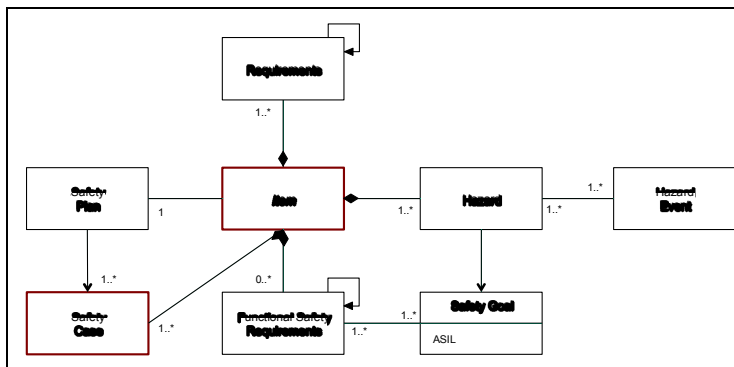

Figure 1 meta-model of elements in concept phase

In figure 1 we depict the keyword and their relationships in concept phase as a meta-model. In this paper we mainly focus on the keywords: the node *item* and safety case.

2.1   The term *Item*

The definition of the term *item* is "system (1-1.129) or array of systems to implement a function at the vehicle level, to which ISO 26262 is applied" (1-1.69). From this definition we may think an *item* is a system, but in early phase of a new project there is no clear definition of system. And in the guideline (part 10 of [1]) the type of relationship between the *item* and system is a realization (figure 3, p.4)[1]. At first, the *item* is only a kind of concept, that is not a concrete one, and become a system as its development proceeds.

---

[1] The guideline (part 10) was released eight month later after other parts are issued, so we may think that new definition is appropriate currently.

The concept phase starts from the *item* definition, and we initiate the safety lifecycle and do the Hazard Analysis and Risk Assessment (HARA). In this activity we identify hazards and classify hazardous events and describe the safety goal with the Automotive Safety Integrity Level (ASIL). Safety goal is the top-level safety requirements and we will derive functional safety requirements from it. In this process it is the item that we are hard to deal with because the conventional hazard analysis methods are premised on the well-described system. So, we have to answer the questions showing below:

- How can we describe and define an *item*?
- What method can we use to perform HARA with *item* definition?

2.2   Safety Case

There is another activity that we have to perform in concept phase. The part 3 of ISO 26262 defines the activities for management through the safety lifecycle. We focus on the next request: "This requirement shall be complied with for items that have at least one safety goal with an ASIL (A), B, C or D: a safety case shall be developed in accordance with the safety plan" (2-6.4.6.1). Safety goals are defined in the concept phase, so we have to start indicating evidences and arguments showing that a safety goal is achieved.

So we have a question showing below:

- How can we correlate the safety goal with the safety case?

3   Our Approach

To answer the previous questions that we present in the chapter 2, we propose a method as a solution. This approach consists of three techniques: KAOS, GSN and Scenario-Situation Matrix (SSM). Figure 2 shows a general view of our approach and the relations of those three techniques.
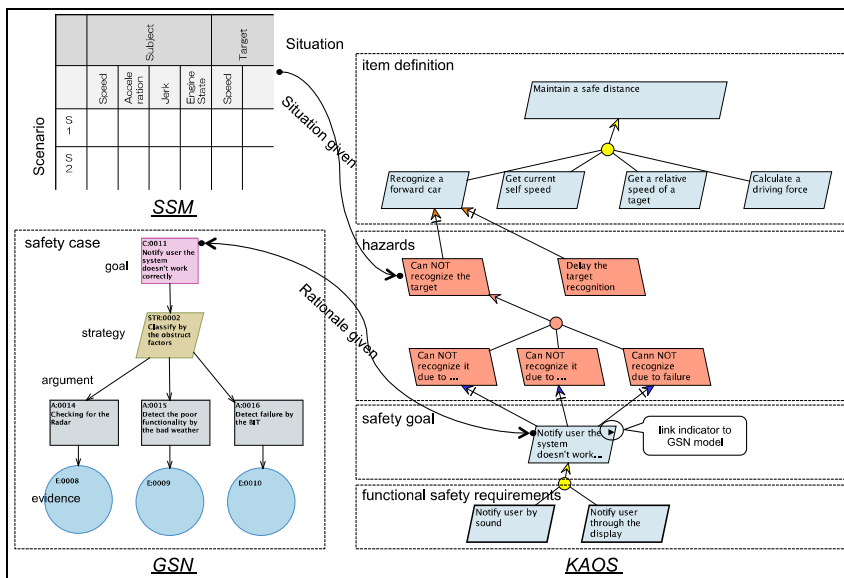

Figure 2 General view of our approach

First we introduce each technique (KAOS, GSN, SSM) briefly and then present our method and the process that indicates how we use those techniques.

3.1 KAOS

KAOS method (e.g. [2-4]) is a typical one in the Goal-Oriented Requirement Engineering (GORE) field. It has six models: goal model, obstacle model, object model, agent model, operation model and behaviour model. The key model in them is the goal model as a hub with other models including elements of other models in it (for example, we can use *agent* node or *operation* node within goal model). The top goal is refined with AND/OR relationship between a goal and sub-goals, and finally we can get the requirements. The structure is tree like, but strictly speaking, an acyclic directed graph.

There are several nodes. I'll just show some of them for the explanation purpose.

- *Goal* node: desired functional or qualitative behaviour

- *Requirement* node: low level goal (i.e. a terminal goal of a goal graph)

- *Obstacle* node: condition whose satisfaction may prevent some goals

- *Softgoal*: goal that do not have a clear-cut criterion for their satisfaction

3.2 GSN

The Goal Structuring Notation (GSN) (e.g.[5-7]) is "a graphical argument notation which can be used to document explicitly the elements and structure of an argument and the argument's relationship to evidence". Usually GSN is used to describe the safety case or assurance case (that covers the wider range than safety case does, e.g. security). It has several types of nodes. The main nodes are *goal* (requirement or claim), *argument* (connector between *goal* and *evidence*, originally used the same node as the *goal*) and *evidence* (solution). And there are the *context* node, *assumption* node and *justification* node to complement the main nodes.

3.3 Scenario-Situation Matrix (SSM)

We also introduce the Scenario-Situation Matrix (SSM). Situation is a combination of all things that exist on driving. The category of situation is like 'subject car', 'target car', 'perimeter (e.g. pedestrian)', 'road type', 'road condition', 'regulation', 'environment' and 'driver'. This category is again divided into elements. In case of the 'subject car' the elements are, for example, 'speed', 'acceleration', 'jerk', 'engine state' and so on.

The Scenario is the sequence of events with the defined situation. For example, under rainy weather we drive the car around 100Km/h on the highway. This includes the environment ('rainy weather'), 100km/h ('subject car') and road type ('on the highway'). By using this matrix we can create the complicated situation and prevent the oversight of situations.

We use this SSM in two places. First is the hazard analysis to find hazards in the hazard analysis phase and second is the calculation of the ASIL in the risk assessment phase.

3.4 Process

In order to develop or consolidate requirements we first use the goal model of KAOS. By using AND/OR refinements we can get the base of the *item* definition. On the way to refinements we might

notice the obstacles to achieve a goal[2]. In safety point of view an *obstacle* node is a candidate of hazard. We may analyse hazard in more formal way. In [8], we propose the Dependability Software Process Optimization (DSPO) method, which is a hazard analysis method and based on HAZOP[9]. The basic idea is very simple. If we choose the guideword "NOT", we try to negate the *goal* node. If, for example, a goal is "Recognize a forward car", we then think what if we cannot recognize a forward car (i.e. 'NOT' is the on of guidewords in HAZOP). With other guidewords we may be able to find other *obstacle* nodes.

Since we confine our discussion to the automobile domain in this paper, the SSM is specialized for the operation modes of the car, like driving, parking and so on. Usually situations are very complicated but we can cover them with just choosing elements from each category (that is, categories are orthogonal). We can find the *obstacle* node more easily with SSM, and each *obstacle* node is a candidate of the hazard. Then we consider a hazard events occurred as the realization of a hazard.

Now we can calculate the ASIL. There is no special technique, but SSM is very useful for deciding the elements of ASIL (i.e. Severity, Exposure and Controllability), the SSM and the goal model give us the relation between the consolidated situation and *obstacle* (hazard).

The *goal* node that solves the problem of the *obstacle* node expresses solutions to handle a hazard, and for recording the rationale of them we can use the GSN notation. In GSN the *argument* node and the *evidence* node assure the validity of the solution.

Table 1 shows the relation between ISO 26262 (part 3) and our approach.

Table 1 ISO 26262 (part 3) and our approach

ISO 26262 Requirements and recommendations			Our approach
3–5	Item definition	3-5.4.1	The functional and non-functional requirements of the item as well as the dependencies between the item and its environment shall be made available.	The *requirement* node that comes from the *goal* or *softgoal* nodes can be used to express the functional or non-functional requirements of the *item*
		3-5.4.2	The boundary of the item, its interfaces, and the assumptions concerning its interaction with other items and elements, shall be defined …	The boundary can be written by UML and we can link to it in the goal model. But it's the out of our scope in this paper.
3–6	Initiation of safety lifecycle	3-6.4.1	Determination of the development category	N/A (just decide new development or modification)
		3-6.4.2	Impact analysis and possible tailored safety lifecycle, in the case of modification	The impact analysis can be done through our goal model.
3–7	Hazard analysis and Risk assessment	3-7.4.1	Initiation of the hazard analysis and risk assessment	Distinguish the element of the *item* whether it has the internal safety mechanism or not.
		3-7.4.2	Situation analysis and hazard Identification	Use the SSM and DSPO
		3-7.4.3	Classification of hazardous events	N/A (just follow the standard)
		3-7.4.4	Determination of ASIL and safety goals	Use table 4 of the part 3. Safety goals corresponds to the solutions which are backed by evidence and arguments
		3-7.4.5	Verification	Correctness can be verified through the arguments of GSN.
3–8	Functional safety concept	3-8.4.2	Derivation of functional safety requirements	The *goal* nodes of KAOS that solves the problem of *obstacle* node are the base of functional safety requirements
		3-8.4.3	Allocation of functional safety requirements	If the architecture is writing in UML, we can allocate each requirement to classes or parts
		3-8.4.4	Validation criteria	In GSN we can present the controllability or effectiveness (4-9.4.3.2)
		3-8.4.5	Verification of the functional safety concept	It is supported by arguments of GSN

---

[2] In this context obstacle is not a stone or something on the road. The obstacle obstructs an achievement of goal in KAOS method. And it belongs to one of the obstacle categories: hazard, threat, dissatisfaction, misinformation, inaccuracy and unusability. In this paper we focus on the obstacle of hazard category.

4 Tool

In this chapter we present a tool Nirvana, which supports our approach. Nirvana is a platform with plugin mechanism. There are many types of plugins that support several methods such as UML, KAOS or GSN. In our approach the linking between models is essential and Nirvana provides this feature. The data could be managed in an integrated fashion through a data base management system (DBMS). There are several import links showing below:

- Goal model to other models of KAOS (e.g. *object* model, *obstacle* model)
- Goal model to diagrams of UML (e.g. class diagram, finite state machine diagram)
- Goal model to GSN (it assures that the solution to a hazard is valid)

To describe the boundary of *item*, we can use the class diagram of UML instead of object model of KAOS.
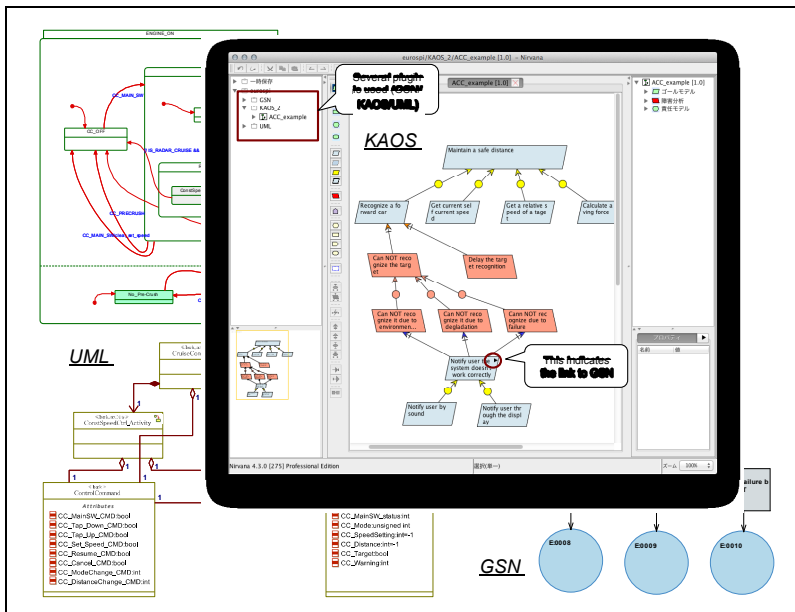

Figure 3 A tool Nirvana with plugins and other connecting tools

5 Related Works

MAENAD (Model-based Analysis & Engineering of Novel Architectures for Dependable Electric Vehicles) [10] is one of the large European research projects. The origin of this project is the EAST-EEA (Electronics Architecture and Software Technologies - Embedded Electronic Architecture) [11], and it was inherited by the ATESST (Advancing Traffic Efficiency and Safety through Software Technology) and ATESST2 project [12]. Those projects support the development of embedded system of automobile. There are two important characteristics. First it covers the whole lifecycle and supports the notation based on UML/SysML for various description including safety requirements, which is called EAST-

ADL (Architecture Description Language) [13, 14]. Secondly, it supports the environment and vehicle level description with feature and variants. Namely it supports the product line engineering.

The MANAD supports the full lifecycle development of the automobile, and our proposal refers only to the concept phase. So, if we focus on our scope, we can find that there are some differences between them. In the hazard analysis, the MANAD uses the use case and scenario. On the other hand we use the SSM to cover the every possible situation. And we use GSN to describe the safety case that helps us to structure the arguments. Certainly the EAST-ADL of the MANAD clearly defines the meta-model of safety case, but our approach can seamlessly connect between *item* definition and safety case.

As for combination of notations, use the notation of KAOS and GSN. In [15], authors extend the KAOS method in order to describe arguments in the goal model. In this approach we may only have to prepare a single diagram. Our approach uses several models to satisfy the various intentions. ISO 26262 require the various expressions in each lifecycle phase, which is the reason why we provide the several models and use them in accordance with the standard.

6 *Conclusion*

We propose a method for concept phase of ISO 26262. In this phase the process starts from definition of *item* that is an abstraction of systems. And the end of this phase we have to describe the functional safety requirements with normal functional and non-functional requirements simultaneously. In order to conform to this request, we use the three notations: KAOS, GSN and SSM. In KAOS normal requirements are expressed *goal* nodes or *softgoal* nodes in the goal model. Hazards are expressed as *obstacle* nodes, and safety goals are denoted as *goal* nodes with the *solution* refinement link. The validity of the relationships between a hazard and a safety goal is expressed in a GSN diagram. SSM has two roles. First we use exhaustive hazard analysis, and then we use it for the base of ASIL calculation. To connect those three expressions we use the linkage mechanism in our supporting tool.

Literature

[1] ISO, "ISO 26262. Road vehicles - Functional safety -," ed: ISO, 2011.

[2] A. Dardenne, S. Fickas, and A. V. Lamsweerde, "Goal-directed concept acquisition in requirements elicitation," *Proceedings of the Sixth International Workshop on Software Specification and Design,* vol. 97403, pp. 14-21, 1991.

[3] A. v. Lamsweerde, *Requirements engineering: from system goals to UML models to software specifications*: John Wiley & Sons Ltd., 2009.

[4] A. van Lamsweerde and E. Letier, "Integrating obstacles in goal-driven requirements engineering," 1998, pp. 53-62.

[5] T. Kelly and J. McDermid, "Safety Case Construction and Reuse Using Patterns," 1997, p. 55.

[6] Origin Consulting Limited. (2011). *GSN COMMUNITY STANDARD VERSION 1.* Available: http://www.goalstructuringnotation.info/documents/GSN_Standard.pdf

[7] J. Spriggs, *GSN-The Goal Structuring Notation: A Structured Approach to Presenting Arguments*: Springer Verlag, 2012.

[8] M. Ito, "Explorative Hazard Analysis Process in Requirement Phase," in *Euro SPI 2010*, Grenoble Institute of Technology, France, 2010, pp. pp. 7.25-7.31.

[9] CEI/IEC, "Hazard and operability studies (HAZOP studies) - Application guide, CEI/IEC 61882:2001," ed: IEC, 2001.

[10] MAENAD Consortium. Available: http://www.maenad.eu/

[11] EAST-EEA, "Embedded Electronic Architecture, Version 1.02," ed: EAST-EEA, 2004.

[12] ATESST2 Consortium. Available: http://www.atesst.org/

[13] D. Chen, H. Lönn, F. Törner, and H. Blom. (2007). *EAST-ADL assessment and update suggestions.* Available: http://www.atesst.org/home/liblocal/docs/ATESST_Deliverable_D2.2.2_V1.1.pdf

[14] A. Sandberg, D. Chen, H. Lönn, R. Johansson, L. Feng, M. Törngren, S. Torchiaro, R. Tavakoli-Kolagari, and A. Abele, "Model-Based Safety Engineering of Interdependent Functions in Automotive Vehicles Using EAST-ADL2," in *Computer Safety, Reliability, and Security*. vol. 6351, E. Schoitsch, Ed., ed: Springer Berlin Heidelberg, 2010, pp. 332-346.

[15] J. Brunel and J. Cazin, "Formal Verification of a Safety Argumentation and Application to a Complex UAV System," in *Computer Safety, Reliability, and Security*. vol. 7613, F. Ortmeier and P. Daniel, Eds., ed: Springer Berlin Heidelberg, 2012, pp. 307-318.

Author CVs

Masao Ito

Mr Masao Ito is CEO of Nil Software Corp, a Tokyo-based software tool development company. He has long career in IPSE (Integrated Process Support Environment) construction and consulting. He is a board member of SEA (Software Engineers Association of Japan) and a core member of SEA-SPIN. He is also serving as CEO of a venture company VCAD Solutions, Co., LTD.

Kouichi Kishida

Studied Astronomy at the University of Tokyo.
Started programming carrier in early 1960s.
Created an independent software house SRA in 1967, now serving as the technical advisor of the company.
Worked as an active volunteer to organize a number of professional societies in Japan such as Software Engineers Association (SEA), Japan SPI Consortium (JASPIC), etc.
Heavily involved in the international software engineering activity. Served as PC Co-chair of ICSE-1987. Gave keynote at ICSE-1988 and EuroSPI-2006. Keeps organizing annual international technical symposia/workshops in China, Korea and India since late 1980s.
Received ACM/SIGSOFT Distinguished Service Award in 2001.

Learner Control Charts: A Structured Way for Finding Common Causes with Decision Trees

Banu Deniz GUNEL, Onur ERDOGAN,
Ahmet DIKICI, Kadriye OZBAS-CAGLAYAN, Murat Kahraman GUNGOR

TUBITAK-BILGEM-YTE[1]
Quality and Process Management Unit
Ankara / TURKEY
deniz.gunel@tubitak.gov.tr , onur.erdogan@tubitak.gov.tr
ahmet.dikici@tubitak.gov.tr, kadriye.caglayan@tubitak.gov.tr, murat.gungor@tubitak.gov.tr

Abstract

Since 1920s, statistical process control (SPC) techniques have been widely used to ensure quality for both manufacturing processes and products themselves. SPC is also very useful to conduct quantitative monitoring and controlling for software processes, since it renders a better improvement and management strategies. In software processes, it is significant to determine process variations as accurate as possible. Here comes in the control charts where upper and lower control limits are defined for deciding the stable process. When an expected variation has occurred within the natural flow of the process, data points will fall inside these limits and these points of variations are called "common causes". However, when a data point falls outside of these limits it means there is an unexpected, extraordinary situation has occurred during the process enactment which is called "a special cause". A special cause can change a process in an uncontrolled and unpredictable way. When a special cause generates an unwanted variation in the process, then it is said that "there is a noise in the process" [1] which needs to be eliminated. On the other hand, a common cause can change a process in a natural way; which belongs to process itself.

In this study, we mainly focused on common cause variations, which occur during ordinary and expected interactions between the components of the process. In control charts, common cause variations normally appear in stable and consistent patterns. However, in some situations, analyzing common causes and sub-grouping data is a crucial step for finding more accurate and effective control charts, where noises can be more easily captured. To carry out these calculations, we select defect density metric for the system tests. Here, we introduce a new approach of generating control charts by analysing the common cause variations with the support of decision tree-learning algorithm. This learning algorithm helps us to define more accurate common causes of variations appeared in the process. This learning model searches for the best common causes in order to classify the defect density values of the system tests executed in different projects in our research institute. As a result of our study, we observed that our new control charts have narrowed upper and lower control limit values. Further in our study, we discussed the improvement opportunities and findings in our system tests and software development lifecycle approach with the support of decision trees and control charts.

---

[1] The Scientific and Technological Research Council of Turkey (TUBITAK) - Informatics and Information Security Research Center (BILGEM) - Software Technologies Research Institute (YTE)

Keywords

Statistical Process Control, Common Cause of Variation, Decision Trees, Control Charts, Quantitative Management, Defect Density, ISO 15504, CMMI-DEV, Iterative Incremental Development

1 Introduction

In recent years, due to its highly satisfactory performance results, rendering cost effectiveness and increase in productivity and product quality, thus reaping a high return on investment [2], the use of Statistical Process Control (SPC) is becoming more widespread. SPC is a method used to keep a process under control statistically to ensure quality [3]. The main idea here is to monitor causes of variations in a process and predict their progression within defined upper and lower control limits. SPC enables the process plans to be made more easily, more achievable and realistic and/or meet deadlines more strictly. It gives opportunity to behave proactively in order to increase customer satisfaction. Moreover, SPC can also be used for observing the results of software improvements since it provides an inner sight to the process enactment with its quantitative management perspective.

With the support of SPC, it is possible to monitor variations of the process and statistically control the quality attributes. Causes of variations in a process are classified as either common causes or special causes of variations. Common causes result from the normal and expected interactions among the process components such as techniques, staff, tools, etc. These variations are predictable and belong to the process naturally. Especially in complex processes, one of the main process improvement strategies is to analyse common causes of variations which then will allow us to explore sub processes [3]. Special cause variations, on the other hand, are the changes in the process, occurred unexpectedly or uncontrollably. They are surprising variations that are not natural to the process caused by factors such as disagreements in a team, unexpected computer crashes or deviations from standard practice, etc. In the existence of special causes, it is not possible to have a stable process.

Process improvement reference models like CMMI [4] and SPICE (ISO/IEC 15504-4) [5] recommend [6] SPC techniques to achieve a continuously improving, stable and mature software engineering process. In SPC, control charts are the most preferred technique [7–9] used for monitoring process activities statistically. The biggest advantage of using control charts in software engineering processes is that they facilitate; first distinguishing common and special cause variations with the defined detection rules and then, measure the behavior of process over. Moreover, they offer a comprehensive structure that helps defining boundaries of upper and lower limits and a mechanism that renders early detection of problems. Control charts are commonly preferred in verification processes such as reviews and testing [3], [7] since they directly help to improve product quality.

However, as there are so many diverse types of experience and examples using SPC and control charts, their application can be difficult to understand or to put them into practice effectively [8], [10], [11]. Especially analysing and detecting common causes of variations is a difficult subject [12] in software processes. These variations usually have some expected reasons and generally analysed by expert judgment techniques [3], [13]. In this study, we suggest to use a more structured method for finding common causes of variations with the support of decision tree-learning model. Briefly explaining, the decision tree-learning is a data mining approach, whose goal is to create a model that predicts data classification using given condition variables of a decision. In our case, the aim is to find and utilize an effective decision tree-learning model in order to predict the defect density data classification using given process variables as conditions. We share our experience related to this matter in the following sections.

In the next section, we introduce our institute, TUBITAK-BILGEM-YTE, and the Software Development Life Cycle (SDLC) approach used in recent projects of our institute in common. We also give background information about SPC and decision tree-learning models. Later, we define the problems faced during detecting common causes of variations and describe our new method that we proposed as a solution. Finally, we will summarize our experience we gained and present a general outcome of this work briefly.

2 Information Background

2.1   Statistical Process Control (SPC) and XMR Control Charts

In SPC, the most widely used tool for monitoring and controlling the process is control charts. There are different types of control charts in statistics [3], [7], [11], [13]. In software processes, generally the data points [14] are not as frequent as in manufacturing processes. We usually plot and evaluate each data point individually. For that reason, the most preferred control charts in the software engineering domain that produce this result are the Individual (X) and Moving Range (MR) control charts.

XMR control charts consist of two key features; centerline and sigma value. Centerline is the mean value of the data points and sigma is the standard deviation of these data points. In X charts, the data points are the individual values on the chart, while in MR charts, the data is obtained by calculating the absolute difference between two successive individual values. In these control charts, generally upper and lower control limits are calculated with 3 sigma values from the centerline [13], which covers 99% of all the complete data. Because of these beneficial aspects, we preferred XMR control charts to monitor progress of system tests process. Due to limited scope of this work, we included only the individual (X) charts in the paper.

2.2   Decision Tree-Learning

A decision tree is a tree like graph, is used for reaching a goal, where leaf nodes are the classifications of the goal, internal nodes tests an attribute and branches corresponds to attribute value. It is commonly used in order to generate rules from dataset based on non-parametric, data-driven modelling by dividing the instances recursively into homogeneous subsets [15].

The tree structure is generally used for the prediction using historical data as training dataset. Because of the ease of use and understanding, it is the charming supervised learning method among others i.e. artificial neural networks, Bayesian networks [16]. There are several algorithms for constructing decision trees such as C4.5, ID3 and CART [17]. The main ideas behind implementation procedure of these decision trees are similar but there are many differences pertaining to several aspects, such as the tree structure, the splitting criteria, and the pruning method [18].

Generally attribute selection measures come out as a heuristic approach for identifying the splitting criterion, which "best" divides a dataset into individual homogenous classes whose class labels are known. Choosing the splitting criterion among the independent variables that are considered the most relevant to dependent variable is the key success factor of decision trees. The most popular splitting criteria methods [19] are information gain and gain ratio in decision tree learning algorithms. In this study, gain ratio measurement is chosen as a splitting criterion.

A decision tree starts with a root node, to which all the instances of the training set are assigned. If there are more than two classes in the root node according to spitting criterion that means some samples meet the initial criteria for decisions need to be made, partitioning may be done and sub groups are formed. This node is turned into the test node and the most significant division criterion is searched among attributes. The newly occurred nodes can be a candidate for both terminal node where classification is reached or test node where branching will be continued. The process is recursively repeated until the terminal nodes are reached. Once the decision nodes are extracted, then one or more decision rules can be derived that describe the relationships between inputs and targets. This derivation, also called as rule induction, is the learning stage of decision trees. At this stage, the model is trained by using the training dataset for constructing the tree structure. Then, the test data classification is predicted with the given attributes whose class labels are unknown.

2.3   Software Development Approach

TUBITAK-BILGEM-YTE is a research and development institute providing e-government software

solutions with software engineering expertise. It is a middle scaled institute with 150 employees, 80 of whom are software engineers. During the software development activities, CMMI-DEV Version 1.3 [4] practices are followed by the organization with additional support of agile practices, such as iterative and incremental development (IID) [20], and test driven development approach.

All software development projects subject to this study use IID life cycle in order to develop and update new features of the software products. In this agile development approach, every iteration adds a bunch of new features and enhances current features in the software. After this, the software is delivered to the end user immediately. From this perspective, it supports end user collaboration strongly. With the IID approach, chance of getting early feedback from end users increases due to incremental deliveries. Therefore focusing on monitoring closely and improving the quality of each iteration becomes inevitable and a crucial step in order to serve quality products to end users. For that reason, both system test and peer review processes have been monitored and improved statistically with the support of XMR control charts in our institute.

2.3.1 Defect Management and Software Versioning

In each phase of software development lifecycle, it is critical to detect and correct defects before they are delivered in the software product. In our institute, defects are found through intense testing activities. Each defect found in the software product is then recorded into our defect management system as an issue. For each issue, a separate *Reported Version* of the product is identified, which later serves as a base for measuring the defect density.

In IID approach, after the completion of an iteration adding new features and fixing defects, a "test baseline" is created. If any defect is found during the tests, they are reported in the related test baseline. Then all the reported defects in the test baseline are fixed in the related version of the product from which another test baseline is created. These steps are repeated until we achieved system test exiting criteria. Through this process, we achieve a systematic increase in the stability of the product. After removing all the detected defects, a "production baseline" is created from the latest test baseline. In the last step, the now ready production baseline is deployed into the production environment from where end users can start to use the product in their operational scenarios. After that, if they find a problem in the system, they use our defect management tool to report the problem. The problem is then analysed by the development team. If they see that the defect is a blocking or a critical one; they immediately fix that defect and prepare a production patch. This step is repeated as many times as required until a new test baseline is created.

2.3.2 Defect Density

Defect density is the number of defects found in the production per thousand lines of source code. It is mainly used to give an insight about how stable a software product is. Due to the IID approach mentioned in previous section, the calculation of the defect density is slightly different from the original definition. Generally, the defect density can be calculated from either the test or the production baselines. For measuring the product's progress in the lifecycle, however, only the defect number retrieved from the production baseline is used. In our approach, in the calculation of the defect density, defects reported in the production patches are added to the related production baseline itself, since no test is run between a patch and the production baseline. In Table 1, example values used for calculating the defect density are given. In this example, the defect density of the *Production Baseline 1.0* can be calculated as (5+2) / 310, or more simply 0,023.

Table 1. Example Values Used in Calculation of Defect Density

Baseline Name	# of Found Defects	KLOC
Test Baseline 1.1	95	205
Test Baseline 1.2	30	255
Production Baseline 1.0	5	300
Production Baseline 1.1 (Patch)	2	310

Therefore, the defect density of kth production iteration is calculated as with the formula given below:

$$Defect\ Density_k = \frac{\sum_{i=1}^{n} Defects\ Found\ in\ Production\ Baseline_{k,i}}{(LOC_{km}\ /\ 1000)}$$

3 Problem Definition

In this study, we used the data fed by the four different projects carried out by our institute. Some common attributes of these projects are as follows:

- Java technologies are used,

- Web-based products are delivered,

- Project lengths are 2 to 4 years with 20–30 people working including developers, team leaders and project managers,

- The same software development processes are used.

As a first step, with the defect density values obtained from the mentioned projects, to analyse the initial situation, an individual control chart is generated. For this, 48 data points are plotted in the control chart as shown in Figure 1. In the given control chart, although the variations are within the upper and lower control limits, it is obvious that there are some problems hidden in the process. The variation range of the defect density here is too large, resulting in the negative lower control limit, which is mathematically impossible. Besides, we can also see some common causes in the variations of the process. For example, the data between 11th and 23th points distinctly vary from the others.
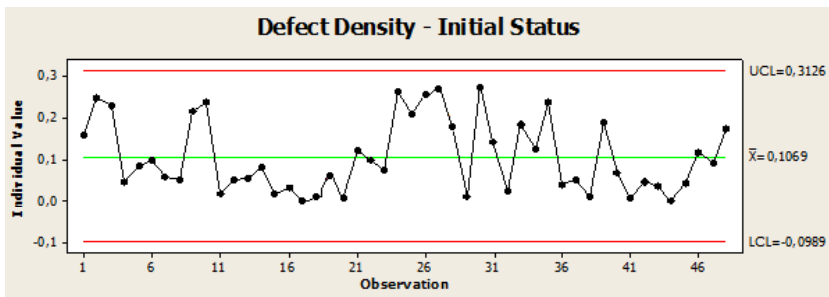

Figure 1. Individual Control Chart Obtained From the Whole Data

With the motivation of finding common causes and narrowing the upper-lower control limits range, some candidate process attributes are determined to put in the decision tree learning model. These process attributes are explained in Table 2.

Table 2. Candidate Common Causes

Attribute	Explanation	Discussion
Number of Defects Found in Test Baselines	It is the number of reported defects found in the test baseline before the related production baseline is created.	If the number of reported defects in a test baseline increases, number of defects found in the production baseline may decrease.
Duration Between Two Production Baselines	It is the time-period between two successive production baselines separated with test iterations.	If the duration of the iteration is too short or too long, it indicates that there might be some defects injected into the production or some defects are being escaped.

Number of Test Iterations	It is the number of the test base-lines created between two successive production baselines.	If the number of test cycles are increasing, it indicates that there might be some defects injected into the production or some defects are being escaped.
Number of Production Patches	It is number of production base-lines created between two successive production baselines.	If the number of production patches increases, the number of defects found in the production baselines may also increase.

4 Learner Control Charts as a Solution

As a summary, based on the heuristic approach we applied, we first select independent variables which were shown in Table 2. Then, we carry out an outlier analysis with the defect density, which is the dependent variable, and the independent variables, which were mentioned earlier as candidate common causes. We remove the values defined as outlier and begin to run the decision tree-learning system which finds significant rules of common cause variations in the model. After that, we separate the data into groups with these rules. Lastly, we publish final control charts with these data groups. In the following sections, the details of this application will be explained.

4.1   Preparation of Data

In order to find common causes within the system tests process, we first define a couple of measurement values. In the decision tree-learning model, these values are called "independent variables". Then we classify the defect density, which is the dependent variable in our case, values in groups with respect to their magnitudes. Since we had only 48 data points in the data set, considering their way of distribution, we decided to classify the data into two sets.

In this analysis, the data distribution was right-skewed, because of this the data set is divided into two sub-groups based on the median value of 0,0829. In this categorization, the first range is between 0 and 0,0829 and is called "*Low Defect Density*". The second range falls between 0,0830 and 0,279 and is called "*High Defect Density*". These categorized defect density values are used to construct the decision tree. Meanwhile, with the independent variables listed in Table 2, an outlier analysis is applied, assuming that any outlier data may indicate a noise or an inconsistent datum [21]. According to this, we removed five data points from the data set. Hence, we obtained all the independent and dependent variables with their classifications and they are ready to be used for preparing tree-learning model.

4.2   Eliminating Common Cause Candidates with Decision Tree-Learning Model

In this study, we have used the best single attribute selection test to create a top-down constructed tree by applying C4.5 algorithm [22] for decision tree-learning model. We used the gain ratio within the C4.5 algorithm as the main criterion to split the data into more homogeneous sub-groups. The summary of the decision tree-learning model is given in Table 3 which main characteristics of the decision tree are presented.

By using the "5 fold cross validation" method, we generated candidate decision trees. In these cross validation tests, 19 data out of 22, which belong to the "*Low Defect Density*" class, are correctly predicted, so the class recall is calculated as 86,36 %. Besides, for the "*High Defect Density*" class, 17 out of 21 data are correctly predicted by the model, which makes the class recall as 83,61 % with a general standard deviation of 9,23%.

Table 3. Decision Tree Attributes

| Learning Algorithm	C4.5
Attribute Selection Criterion	Gain ratio
Inputs	• Number of Defects Found in Test Baselines • Duration Between Two Production Baselines • Number of Test Iterations • Number of Production Patches
Output	Defect Density (DD)
Minimal Gain	Defined to produce a partition = 0,01
Maximal Depth	The maximum tree depth = 10
Validation Method	5-fold Cross Validation (stratified sampling)
Minimal Size for Split	The minimal size of a node in order to allow a partition = 5
Minimal Leaf Size	The minimal size of all leaves = 5
Confidence	Used for the pessimistic error calculation of pruning = 0,20
Number of Pre-pruning Alternatives	The number of alternative nodes tried when pre-pruning would prevent a split = 5
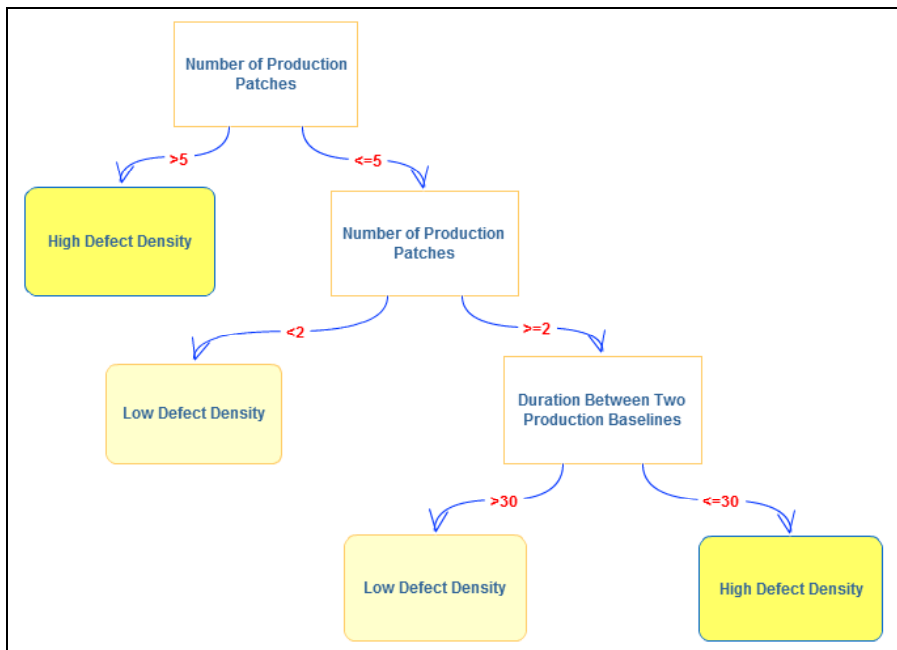
Figure 2. Decision Tree of Defect Density Classification

The final tree model constructed according to the parameters that make the model optimal is shown in Figure 2. Under these circumstances, the "Number of Production Patches" is the most significant attribute that affect the defect density classification. The second important attribute is the "Duration Between Two Production Baselines". The other attributes do not contribute to form homogenous sub-sets with the gain ratio. From this decision tree some rules are deduced for the data classification.

Rules obtained from Figure 2, are listed below for two different classification of defect density.

Low Defect Density Class Rules

- Number of Production Patches should be <u>less</u> than 2 OR

- Number of Production Patches should be between 2 and 5 AND Duration Between Two Production Baselines should be <u>more</u> than 30 days.

High Defect Density Class Rules

- Number of Production Patches should be <u>greater</u> than 5 OR

- Number of Production Patches should be between 2 and 5 AND Duration Between Two Production Baselines should be <u>less</u> than 30 days.

4.3   Final Process Performance Baselines

In the previous section, an efficient decision tree prediction model was found in order to predict defect density classifications. Next, we use the rules of common causes found with the prediction model to group the data in a way that will produce new control charts. We find that this approach, comparing to other approaches [3], [10], is more suitable to define the common cause variations in the process in a more structural way.
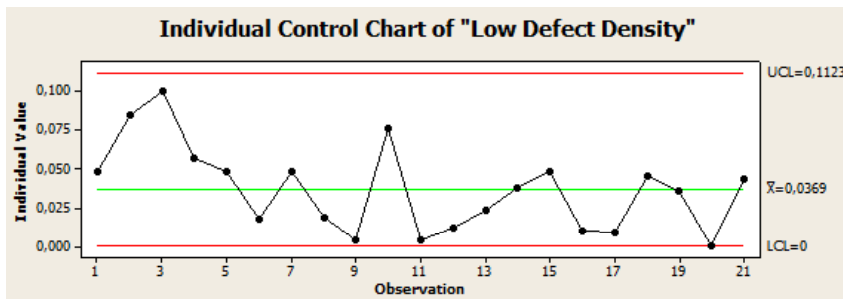

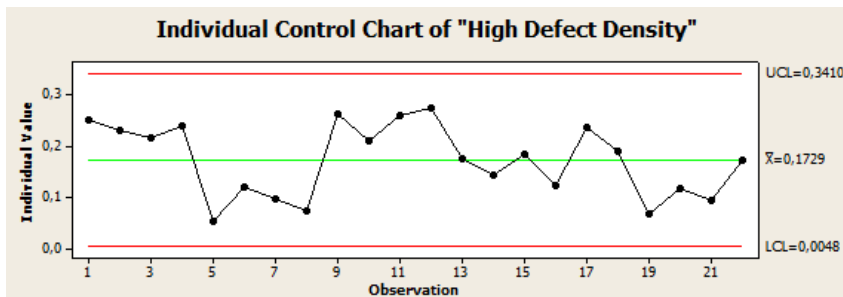Figure 3. Low Defect Density Individual Control Chart[2]

Figure 4. High Defect Density Individual Control Chart

Here, it should be emphasized that, the data in the final defect density control charts is grouped according to the rule sets derived from the decision trees resulting from common cause variations, but not directly from the median value of the defect density measurements. Always referring to the rules

---

[2] The value of the lower control limit, which is calculated as -0,0386, is shifted to 0; since it is logically impossible to have a negative value for defect density metric.

emerge from the natural flow of the process execution, we can evaluate the process performance much more effectively with the newly generated control charts.

At the end of the calculations based on the decision tree-learning model, we obtain a narrowed range of values in the final control charts. By comparing the control chart obtained at the beginning (see Figure 1) with the Low Defect Density Individual Control Chart (see Figure 3), which is produced by filtering the whole data with the "*Low Defect Density*" rule set, we can see that, the range value is reduced by 19%. On the other hand, the range of the High Defect Density Individual Control Chart, which is produced by filtering the whole data with the "*High Defect Density*" rule set, is reduced by 64% comparing to the initial status.

5 Summary and Discussion

In our study, we supported control charts with learning algorithms of decision trees to capture variations of the common causes appear in the process. This approach resulted in a significant improvement of the control charts narrowing the limits by at least 19%. As a future work, we will continue this by adding cumulative test efforts in the decision trees as a new common cause candidate.

As a result of analysis conducted, we detected some issues to be improved during the system tests process. The existence of a significant increase in the defect density value in the case when the development iteration was shorter than a month was the first issue we encountered. After some discussions and meetings with the project teams, we decided that the reason behind this situation is the shortened period of preparation time of the iteration which ended up waiving on some of the unit test implementations of new features. The second issue we came across was that as the number of production patches exceeds five, the value of the defect density significantly increases. As a future work, we want to find solutions for both problems of the kept-short iterations and exceeding patch numbers in the process.

6 References

[1] Donald J. Wheeler, *Understanding Variation: The Key to Managing Chaos*. SPC PRESS (Statistical Process Control Press), 2000, p. 174.

[2] D. L. Gibson, D. R. Goldenson, and K. Kost, "Performance Results of CMMI ® -Based Process Improvement," no. August, 2006.

[3] W. A. Florac, A. D. Carleton, and J. R. Barnard, "Statistical Process Control : Analyzing a Space Shuttle Onboard Software Process," *IEEE Software*, vol. 0740–7459, no. 00, pp. 97–106, 2000.

[4] M. B. Chrissis, M. Conrad, and S. Shrum, *CMMI for Development Guidelines for Process Integration and Product Improvement*, Third. 2013.

[5] S. Q. Journal, "SPICE : Software Process Capability dEtermination," vol. 224, pp. 209–224, 1993.

[6] F. J. Pino, M. T. Baldassarre, M. Piattini, and G. Visaggio, "Harmonizing maturity levels from CMMI-DEV and ISO / IEC 15504," no. September 2009, pp. 279–296, 2010.

[7] P. Jalote, S. Member, and A. Saxena, "Optimum Control Limits for Employing Statistical Process Control in Software Process," vol. 28, no. 12, pp. 1126–1134, 2002.

[8] K. U. Sargut and O. Demirörs, "Utilization of statistical process control (SPC) in emergent software organizations: Pitfalls and suggestions," *Software Quality Journal*, vol. 14, no. 2, pp. 135–157, Jun. 2006.

[9]   M. A. Lantzy, "Application of Statistical Process Control to the Software Process," *ACM*, vol. 0–89791–48, no. 92, pp. 113–119, 1992.

[10]   A. Tarhan and O. Demirörs, "Investigating Suitability of Software Process and Metrics," pp. 88–99, 2006.

[11]   Q. Wang, N. Jiang, L. Gou, X. Liu, M. Li, and Y. Wang, "BSR : A Statistic-based Approach for Establishing and Refining Software Process Performance Baseline," pp. 585–594, 2006.

[12]   W. A. Florac and A. D. Carleton, *Measuring the Software Process*. SEI Series in Software Engineering, 1999.

[13]   W. A. Florac and A. D. Carleton, *Measuring the Software Process*. SEI Series in Software Engineering, 1999.

[14]   D. Card, "Statistical process control for software?," *IEEE Software*, vol. 11, no. 3, pp. 95–97, 1994.

[15]   L. Minne, S. Eslami, N. de Keizer, E. de Jonge, S. E. de Rooij, and A. Abu-Hanna, "Statistical process control for validating a classification tree model for predicting mortality--a novel approach towards temporal validation.," *Journal of biomedical informatics*, vol. 45, no. 1, pp. 37–44, Feb. 2012.

[16]   T.-D. Do, "Towards Simple , Easy To Understand, An Interactive Decision Tree Algorithm," Cantho City, Vietnam.

[17]   V. Podgorelec, P. Kokol, B. Stiglic, and I. Rozman, "Decision trees: an overview and their use in medicine.," *Journal of medical systems*, vol. 26, no. 5, pp. 445–63, Oct. 2002.

[18]   J. H. Cho and P. U. Kurup, "Decision tree approach for classification and dimensionality reduction of electronic nose data," *Sensors and Actuators B: Chemical*, vol. 160, no. 1, pp. 542–548, Dec. 2011.

[19]   F. Harris Jr. and W. Williamsburg, "Information Gain Versus Gain Ratio: A Study of Split Method Biases," 2001.

[20]   C. Larman and V. R. Basili, "Iterative and Incremental Development : A Brief History," no. June, pp. 47–56, 2003.

[21]   I. Ben-gal, "Outlier Detection," in *Data Mining and Knowledge Discovery Handbook: A Complete Guide for Practitioners and Researchers*, Kluver Academic Publishers, 2005.

[22]   J. R. Quinlan, "Learning Decision Tree Classifiers," vol. 28, no. 1, pp. 71–72, 1996.

7 Author CVs

Banu Deniz GUNEL

Deniz works as a Software Process Improvements Engineer for two years at TUBITAK-BILGEM-YTE Quality and Process Management Unit. Before that, she worked in the same research institute as a software engineer for three years. She had her Master of Science degree in Middle East Technical University in Computer Engineering Department. Her main research topics are statistical process control in software processes and early project size and effort estimation models. She received her Bachelor's Degree in Hacettepe University, Computer Engineering Department.

Onur ERDOGAN

Onur works as a Measurement and Analysis Specialist for 2 years at TUBITAK-BILGEM-YTE Quality and Process Management Unit. He received his Bachelor's Degree in Hacettepe University, Department of Statistics. He completed his Master of Science at Middle East Technical University, Informatics Institute, Medical Informatics programs studying data mining in biomedical domain. He still continues his Ph.D. education on the same subject.

Ahmet DIKICI

Ahmet works as Software Development Unit Manager at TUBITAK-BILGEM-YTE. He received a B.S. in Computer Engineering (2002) and an M.S. in Software Engineering (2005) from Middle East Technical University. At present, he is a Ph.D. student in Information Systems at Informatics Institute, Middle East Technical University. His main research interests cover software process improvement, software quality and process modelling. He has 10 years of experience in software engineering as a software engineer, project team leader, deputy project manager and project manager.

Kadriye OZBAS-CAGLAYAN

Kadriye works as Project Monitoring and Evaluation Office Manager at TUBITAK-BILGEM-YTE. She received her B.S. in Computer Engineering and Information Sciences (1997) from Bilkent University and an M.S. in Computer Engineering (1999) from Baskent University. At present, she is a Ph.D. student in Computer Engineering, Middle East Technical University. Her main research interests cover software process improvement, software quality and software repository mining. She has 15 years of experience in software engineering as a software engineer and project manager.

Murat Kahraman GUNGOR

Murat is the main director of TUBITAK-BILGEM-YTE. He also worked as director of Information Technologies Institute at TUBITAK-BILGEM for a year. He has 14 years of experience in software engineering as a software developer and project manager. Between 2006 and 2011 he worked as project manager at Software Technologies Research Institute. He received his Ph.D. and M.S. in CS (2006, 2001 respectively) from Syracuse University, USA. His main research interests cover quality of large software systems, structural metrics and analysis tools.

Understanding Sprint Velocity Fluctuations for Improved Project Plans with Scrum: A Case Study

Filipe Albero Pomar, Jose A. Calvo-Manzano, Edgar Caballero
Facultad de Informática, Universidad Politécnica de Madrid, Campus de Montegancedo s/n,
28660 Boadilla del Monte, Madrid, Spain
filipe.albero.pomar@alumnos.upm.es, jacalvo@fi.upm.es, edgar.caballero@gmail.com

Abstract

Starting from the documentation of high Sprint Velocity fluctuations in a Scrum project, this paper presents a thorough approach to identify the sources of issues arising in the context of Scrum implementation. Given that Scrum provides guidance on identifying process issues, but not their root causes, various approaches are explored. This is of great relevance because Scrum defines project schedules relying heavily on Sprint Velocity and because it is the most widely used Agile methodology. The findings provide a new approach to evaluate such fluctuations and establish a more realistic project assessment than what is currently defined by Scrum. In this respect, this paper contributes to improve the understanding of the software development process using this Agile framework.

Keywords

Scrum, Sprint Velocity, Agile Planning, Agile Methodologies

1 Introduction

In today's competitive and fast-paced world companies are under pressure to adapt to an ever changing environment [Kot12]. The software development industry is not an exception. What started as a technique to develop industrial products faster has now become part of the IT sector under the name of Agile Software Development [TN86, Ver11]. Among many methodologies that promise increased agility, Scrum [SS11] is the most widely used globally [Ver11].

Scrum has been successfully used in a wide variety of industries to create software projects [Pic10]. As opposed to traditional development models, such as Waterfall, it defines the creation and delivery of software in small increments with little upfront planning. Hence, it allows companies to deliver value to their clients earlier and to adapt to changes faster. Although it exploits the benefits of Agile methodologies, it does present a key drawback. In fact, Scrum does not provide detailed guidance on how to identify the root cause of process issues when they occur [Sch04, SS11].

This paper provides a case study on the implementation of Scrum in a software development project of a non-governmental organization. Analysing the role of Scrum within the project, it investigates the root cause of high fluctuations in the amount of work done, known as the Velocity, in each short development cycle, called Sprint. This issue is of paramount importance because project scheduling is derived from the aforementioned metric [Coh09, Gri12]. In this respect, this paper contributes to improve the understanding of the development process within this Agile framework.

The paper is structured as follows. Section 2 characterizes the software development settings where high Sprint Velocities fluctuations are observed. Section 3 provides a structured approach for evaluating the possible forces at play causing the observed fluctuations. Finally, Section 4 presents conclusions.

2 Context

2.1 The Organization

THE COMPANY (fictitious name for confidentiality reason) is a small non-governmental organization based in the United Kingdom with a staff of 44 employees. One of the services it provides to the local community is a volunteering web-based search engine. Since it was first released in the year 2000, it has become the leading website for finding volunteering opportunities in the country. In 2012, the search engine had over 1 million registered users and a monthly average of 186K unique visitors and 3.2 million page views. The first version of the software was not designed to cope with this traffic; response times were slow, new functionalities were needed, but the old codebase was difficult to adapt. To solve these problems, a complete rewrite of the system was commissioned.

The new version of the website was written in-house and passed onto a newly formed team of developers to deliver it to production. The new team was formed in April 2011 and given full responsibility for bug fixing and creation of new minor features. In December of that year the system went live, completely replacing the legacy codebase. For the following six months, the team focused on bug fixing and creating new administrative subsystems not present in the original software.

2.2 Scrum Introduction

Scrum is an empirical process control model founded on three pillars: transparency, inspection and adaptation [SB02, Sch04]. It states that software must be created in small increments that deliver

business value to the customer. Instead of big upfront designs and exhaustive bureaucratic plans, it fosters the creation of a slim project plan that is revised and augmented as the project develops. Each software increment is created in a time-boxed period called Sprint that usually lasts two to four weeks [Ken12].

The steps for developing software with Scrum start with the creation of a prioritized list of requirements called Product Backlog [Pic10]. Just before each Sprint, the team gathers in a Sprint Planning Meeting in which it estimates requirements from the Product Backlog and decides what can be implemented. Throughout the Sprint, the Daily Scrum meeting is carried out to identify issues and to communicate what each individual is working on. High visibility tools (i.e. Sprint Backlog, Sprint Burndown) are used to communicate progress to all parties. To close the short development cycle, the team carries out two meetings: Sprint Review, where developers showcase work that has just been completed, and a Sprint Retrospective, to identify process improvements. The final outcome of the Sprint is a potentially shippable software increment [Coh06].

Release Plans are created using the two project measures of effort and capacity. Effort is measured in points and it is an abstract representation of requirements' sizes. Capacity, also known as Velocity, is the rate of progress at which the team completes requirements; it is calculated at the end of each Sprint by summing up all points of completed requirements. By using the team's actual Velocity it is possible to predict how much of the Product Backlog can be consumed in each future Sprint and to derive a Release Plan [Coh06, Gri12, SB02]. Trends in Velocity can also signal problems; Schwaber and Beedle identified patterns in Sprint velocities that indicate problems such as slipped release dates and the lack of proper velocity tracking [SB02].

Scrum identifies three main roles that work together daily throughout the project [SS11]. The Product Owner represents the business, provides a project vision, and generates requirements and their priorities. The Scrum Master, often a developer, ensures the correct implementation of Scrum. Lastly, the Development Team is composed of a cross-functional group of software developers.

2.3   Scrum Implementation

The project implemented Scrum as prescribed. All team members had prior experience with Agile methodologies, but only one knew Scrum in depth and had used it professionally. The framework was taught by the most experienced developer and achieved high buy-in from both developers and managers. The project data set available refers to the usage of Scrum from June 2011 to June 2012.

Throughout the 20 Sprints analyzed in this paper the team composition remains mostly unchanged. There are 3 senior developers, 1 front-end specialist and at the 20th iteration a junior developer is added to the team. With the exception of the front-end specialist, all members develop all application layers and create tests for each functionality. As defined by Schwaber and Beedle [SB02], the Development Team and the Product Owner are collocated and work together daily throughout the project. Requirements are written in the form of user stories [Coh04]. Estimations are done using a consensus-based technique called Planning Poker [Coh06] to quantify size (in points) of either new features or change requests. At the end of each two-week Sprints, all sizes of work items done are added together to what is known as Velocity, a widely used Agile metric [Ver11]. To the team, its Definition of Done [SS11] means the functionality is coded, verified by automated tests, approved by the Product Owner and stored under version control.

2.4   The Project

The project comprises one main application and eight web-based subsystems that support its operation. When considered as a whole, the codebase sizes up to over 50 KLOC, which is equivalent to a mid-sized system.

Scrum identifies some high visibility tools to communicate progress from the Development Team to the rest of the organization [Coh09, Gri12]. Throughout the project both the Sprint Backlog

and the Sprint Burndown are maintained. The Sprint Backlog is a whiteboard where requirements are placed at the leftmost part of the board and are physically moved to the right to indicate the progress towards completion. The aforementioned whiteboard is divided into four sections, starting from the far left: Not Started, Started, Signoff Pending, and Done. Next to the Sprint Backlog, a Sprint Burndown chart shows the amount of work still left to be done in the current Sprint. The team works from one Sprint to another without planning a wider horizon with a release planning. In fact, this business decision complies with Scrum as the Sprint Planning Meeting is a mandatory activity whereas Release Planning is not [SS11]. For this project the lack of release planning is never a real issue, as new functionalities are regularly delivered and deadlines met.

Nevertheless, going forward with the project, it becomes evident that better scheduling allows the creation of more reliable plans and setting stakeholders' expectations. In order to predict what can be accomplished, Scrum identifies historical Sprint Velocity as the most reliable forecast of future outcomes [Coh06]. In that direction there are two techniques for predicting future performance and both are based on past accrued velocities. First, the velocity of each Sprint can be plotted to identify any trend that could signal problems [SB02]. Second, it is possible to calculate a confidence interval to understand the probability of future velocities and employ it in the creation of a reliable project schedule [Gri12].

Figure 1 shows the Development Team's velocity per Sprint. In the first three Sprints the team is getting used to the Planning Poker and Scrum - therefore, over optimism leads to somewhat inflated estimations. Turning to the analysis of abnormal Sprints, Sprint 1 marks the use of Scrum. In this case, the team is already acquainted with the codebase through ad hoc tasks given to them. Sprints 15, 18 and 19 have zero velocity due to work on a new functionality based on new technologies that turn out to be problematic for production use. During Sprint 16, the team is engaged in the recruitment of a replacement for a senior developer. Finally, a previous employee familiar with the system is contracted.
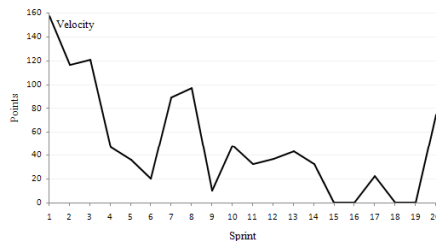
Figure 1: Historical Sprint Velocity

The descriptive statistics of the historical velocities show a median velocity of 36.5, standard deviation of 45.7 and mean of 49.2 points. Furthermore, there is 90% likelihood that in future Sprints the actual accrued velocity will fall between 20 and 75 points. The plot of the historical Sprint Velocity presents a strong downward trend that could signal problems. These prevent the creation of a reliable schedule for the project and, in fact, suggest that there is a process issue in play [Coh06].

The team follows Scrum as prescribed and uses the tools it provides to gauge the project's progress. However, the preliminary analysis signals a potential problem related to the implementation of the methodology. In this respect, the Scrum framework does not describe a procedure to identify the root cause of the problem. Based on the analysis of the project data, the next sections give a description of the troubleshooting efforts carried out to uncover the causes of the velocity fluctuations. This allows deriving some general considerations on problems detection, which may lead to the creation of better project schedules in Scrum.

## 3 Sprint Velocity Fluctuations Study

The absence of clear guidelines on identifying root-causes of Sprint Velocity fluctuations compels

Scrum practitioners to employ exploratory troubleshooting efforts at their discretion. This section describes a structured approach for evaluating the possible forces at play causing the observed fluctuations.

3.1   Commitments Are Not Fulfilled

The fulfilment of commitments by the team solidifies trust between developers and business people [Sch04]. As suggested by Cohn [Coh06], the Development Team decides what can be accomplished in a Sprint following the Consensus Driven approach. According to this technique, the team takes into consideration their availability and task complexities to decide what they can commit to. Even though the Development Team has free choice on deciding what they can commit to, they consistently fail to fulfil those commitments.

In order to gauge the amount of commitments not fulfilled, it is possible to resort to a traditional (non-Agile) project management metric called Schedule Performance Indicator (SPI) [Gri12]. The SPI is calculated as the ratio of Earned Value on Planned Value. In the case of Scrum, it can be described as the total points completed at the end of a Sprint over the total points the team committed to in the Sprint Planning Meeting. The descriptive statistics of the SPI series present a median of 64.17% of commitments fulfilled and 90% likelihood that commitments will be honoured by 37% to 83% in a future Sprint. When evaluating SPI, a useful complementary metric is Team Availability (TA), which is the percentage of developers available in the Sprint. Figure 2 shows the two aforementioned metrics in use. On Sprint 1 the team delivers 82% of the work it committed to while having 92% of the team present for the entire Sprint. It is important to note that either metric could potentially go beyond 100%, even if neither case ever occurred.
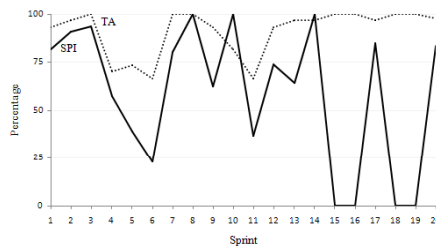
Figure 2: Schedule Performance Indicator (SPI) and Team Availability (TA)

The graphical analysis shows that commitments are only met three times out of twenty iterations. It could be argued that the Development Team does not finish its work because its focus is deviated from the Sprint into other tasks that either have no assigned points or are not value-adding activities (i.e. unscheduled meetings). Although these scenarios happen sporadically they are not the norm; time for meetings is always taken in consideration when committing to work during the Sprint Planning Meeting. For these reasons, SPI alone does not explain the high Sprint Velocity fluctuations, but provides input for further investigation.

3.2   Correlation of Team Availability and Commitment Fulfillment

While unaccounted leave (i.e. due to sickness) may lead to the systemic missing of commitments, SPI and TA present correlation coefficient of 0.12. In this respect, team availability does not provide a comprehensive explanation of the team poor commitment fulfilment.

3.3   High Work In Progress (WIP)

Work in Progress may be an important factor in the analysis. Features that are too big to be finished in an iteration can lead to high Work in Progress (WIP). High WIP can cause at least two big problems [Coh06, Gri12]. First, it leads to high context switching which is known to decrease developers' performance. Second, it means that many tasks are started, but not completed at the end of the Sprint. By not meeting the team's Definition of Done, those tasks are not added to the Sprint's velocity [Coh06, Kni07]. To curb WIP, each developer avoids working in more than one story at the time.

In addition, larger tasks (those longer than a day's work) can slip from one Sprint to another, which results in high WIP and low delivery. To tackle this problem, larger tasks are broken down into smaller ones that can be completed in one work day and only some uncommon tasks are allowed to be at most three days. Furthermore, to create good quality estimations the Development Team follows [Coh06] and ensures that estimations are made relative to each other and created against a baseline of sample user stories. Hence, WIP is kept to a minimum and is not the culprit of high Sprint Velocity fluctuations.

3.4   Team Dynamics and Rework

Communication between developers and Product Owner is frequent and honest. This and the clear Definition of Done leads to small amount of rework and allows the team to solve any doubts during estimation sessions; in other words, the team is confident that they know all that is needed when estimating. Team coherence is never affected by individuals different cultural backgrounds, as warns Cohn [Coh09], because of candid face-to-face communication. Although there could be politics in play pushing the team to over-commit, that is never the case.

3.5   Hidden Complexity

Ken Schwaber noted in his books that complexity in software projects is influenced by requirements, technology and people [SB02, Sch04]. When these factors interact, complexity rises and project control becomes increasingly challenging. In order to acknowledge complexity, projects can be categorized as Simple, Complicated, Complex, or Chaos depending on technology certainty and requirements agreement. The rationale behind these categories helps identify the software development process that better adapts to each project. Simple projects can be controlled through any methodology, including Waterfall; Complicated or Complex projects benefit most from empirical processes such as Agile; Chaos projects are highly unstable and cannot be properly controlled. Figure 3 [Nik13] (original from [Sta99] and presented in the context of software development by [SB02]) presents a visual representation of project complexities interrelations.
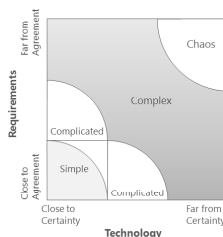

Figure 3: Project Complexity

The project analyzed in this paper can be classified as Complex when taken as a whole because not all technologies involved are mastered by developers and some requirements are far from

certain. If instead each subsystem is considered as a different project, different complexity classifications emerge. There is one Simple project, three Complicated projects, and five Complex projects. By aggregating subsystems in the aforementioned categories, it is possible to calculate confidence intervals at 90% for their SPIs. The picture that forms is as follows (some activities are included in the whole sample, but excluded from the detailed project samples due to their specific nature that does not allow an objective association to a specific subsystem):

- Simple: the team will meet their commitments by 81% to 100%

- Complicated: the team will meet their commitments by 74% to 100%

- Complex: the team will meet their commitments by 20% to 100%

As it can be seen from the confidence intervals, the complexity of the modules directly influences how the Development Team meets their commitments. The Simple subsystem contains much more stable requirements and defined set of technologies, which allows for greater predictability of work. Whereas Complicated or Complex subsystems present both technological and requirement novelties that surface after commitments are made and code is being developed.

This consideration allows explaining the root cause for the high Sprint Velocity fluctuations on a cause-effect basis. Each project's subsystem has different levels of complexity with respect to requirements and technologies. The Development Team does not account for these important differences in the Sprint Planning Meeting because it follows Scrum as prescribed that is creating estimations for the project as a single unit. Commitments are made, but are often missed because requirements change and some technologies used are new to the Development Team. Hence, the accrued Sprint Velocities present great variations over time. This is of relevance for both the specific project and the Agile community because Scrum uses a single measurement of historical Sprint Velocity to derive a project schedule.

Generally, the interpretation of the results of Scrum implementation greatly benefit from a more thorough analysis than that prescribed by the framework itself. In this sense, by detecting the specific impact of complexity in the components of one project we managed to clarify the root cause of what is observed to be large Sprint Velocity fluctuations.

4 Conclusions

This paper investigates possible reasons for high Sprint Velocity fluctuations in a Scrum project. Given that Scrum provides guidance on identifying process issues, but not their root causes, different approaches are explored. It is shown that Sprint Velocity fluctuations are caused by the team missing their commitments which in turn depends on the unaccounted complexities of different subsystems. This distortion occurs because Scrum is implemented as prescribed, that is, considering Sprint Velocity for the project as a single unit. Future research could formalize a Scrum extension to account for projects that have heterogeneous levels of requirements agreement and technologies certainty.

These findings can aid Scrum practitioners in creating more reliable schedules based on historical data. The use of historical Sprint Velocity along with the acknowledgement of the project's SPI and confidence intervals for different system's modules provide a more realistic project analysis than what currently is defined in the Scrum framework. In this respect, this paper contributes to improve the understanding of the software development process using this Agile framework.

5 Literature

[Coh04] Cohn, M.: User Stories Applied. Addison-Wesley Professional, Boston (2004)
[Coh06] Cohn, M.: Agile Estimating and Planning. Prentice Hall, Upper Saddle River (2006)
[Coh09] Cohn, M.: Succeeding with Agile Software Development Using Scrum, Addison-Wesley (2009)
[Ken12] Kenneth, S.: Essential Scrum. Addison-Wesley, Boston (2012)
[Kni07] Kniberg, H.: Scrum and XP from the Trenches. C4Media, United States of America (2007)
[Kot12] Kotter, J.: Accelerate!. Harvard Business Review (2012)
[Nik13] Malik, N., All Effective Enterprise Architects are Agile:
http://blogs.msdn.com/b/nickmalik/archive/2013/01/15/all-effective-enterprise-architects-are-agile.aspx
[Gri12] Griffiths, M. PMI-Exam Prep. RMC Publications, Unites States of America (2012)
[Pic10] Pichler, R.: Agile Product Management with Scrum. Addison-Wesley, Boston (2010)
[SB02] Schwaber, K., Beedle,M.: Agile Software Development with Scrum. Prentice Hall, Upper Saddle River (2002)
[Sch04] Schwaber, K.: Agile Project Management With Scrum. Microsoft Press, Redmond (2004)
[SS11] Schwaber, K., Sutherland, J.: http://www.scrum.org/Scrum-Guides (2011)
[Sta99] Stacey, R.: Strategic Management and Oganisational Dynamics The Challenge of Complexity, Harlow England (1999)
[TN86] Takeuchi, H., Nonaka, I.: The New Development Game. Harvard Business Review (1986)
[Ver11] VersionOne, State of Agile Survey:
http://www.versionone.com/pdf/2011_State_of_Agile_Development_Survey_Results.pdf (2011)

6 Author CVs

Filipe Albero Pomar

He received a double MSc in Software Engineering from the Universidad Politécnica de Madrid and the Libera Università di Bolzano. Filipe is a Certified Scrum Master and PMI-Agile Certified Practitioner and has put his Agile experience in practice at successfully guiding the implementation of Scrum in cross-cultural settings. Earlier in his career he earned many technology-related certifications and worked as software developer. His research interests are primarily in empirical software engineering with direct industry applicability. He can be contacted through professional networking sites and at filipe.albero.pomar@alumnos.upm.es

Jose A. Calvo-Manzano

He holds a PhD in Computer Science. He is assistant professor in the Facultad de Informática at Universidad Politécnica de Madrid. He is teaching in the area of Software Engineering, specifically in the domain of software process management and improvement. He has participated in more than 20 research projects (European, Spanish Public Administration as well as private enterprises). He is author of more than 50 international papers. He is author of books related to software process improvement and software engineering topics also. He is the head of the "Research Chair in Software Process Improvement for Spain and Latin American Region", where he has been a member of the team that has translated CMMI-DEV v1.2 and v1.3 to Spanish. He also holds the ITIL® v2 and v3 Foundation, and CMDB Certification.

Edgar Caballero

He is a PhD student in the "Facultad de Informática" at the Universidad Politécnica de Madrid. His research line is about "Quality Assurance Applied to Agile Methods" and he has recently been working about improvement quality in Scrum. He leads an agile software development team which investigates and implements agile methods. He currently works as a Director of IT Applications in UNIR (Universidad Internacional de la Rioja). Since early 2012, he is member of the Sectoral of Information Technologies and Communications of the Council of Rectors of Spanish Universities (CRUE- TIC Sectoral).

Toward Systematic Approach for Objective Based Software Process Tailoring

Tisanai Krisanathamakul[1], Chayakorn Piyabunditkul[1,2,3]
Apinporn Methawachananont[1], Horst Lichter[3]

[1]Software Engineering Laboratory (SWE)
National Electronics and Computer Technology Center (NECTEC), Pathumthani, Thailand
[2]The Sirindhorn International Thai-German Graduate School of Engineering (TGGS), Bangkok, Thailand
[3]RWTH Aachen University, Aachen, Germany
chayakorn.piyabunditkul@nectec.or.th

Abstract

In the paper, we propose a systematic process tailoring framework that is based on the business goals of companies as objective based likes define here, Goal Based Software Process Tailoring (*GB-SPT*). To be successful in implementing a Software Process Improvement (*SPI*) program, the process tailoring mechanism used plays a very crucial role; however, tailoring is often done by inexperienced practitioner. A software process tailoring framework should help practitioner to appropriately decide what to tailor based on a scoring matrix which is calculated to reflect the project characteristics. The objectives of the project are defined using quality attribute and a scoring matrix. By adopting the concept of Value Based Software Engineering (*VBSE*), the scoring matrix is define here, "Goal Based Scoring Matrix (*GBSM*)", also provides the information of contribution score gain per workload (*E*).

Keywords

Systematic Process Tailoring Framework, Software Process Improvement (*SPI*), Value Based Software Engineering (*VBSE*), contribution score (*CS*), Goal Based Scoring Matrix (*GBSM*), Goal based software process tailoring (*GB-SPT*), Software Project Life Cycle Process (*SPLCP*).

1   Introduction

Software process tailoring is crucial because organizations have limited resources as stated in publications like [1], or [2]. Hence, software process models that were designed to cover all possible tasks, roles, artifacts etc. should be modified for a better resource utilization based on the project characteristics [4]. Although software process tailoring is important and difficult, it is often done by inexperienced practitioners. One important question is how to establish an appropriate tailored process based on defined project characteristics, even if an organization has a software process tailoring guideline. Several approaches propose process tailoring frameworks to cope with this tailoring problem (e.g. [6], [7], [8], [9], and [10]).

Taking this situation into account, we intend to develop a process tailoring framework that supports practitioners to tailor software processes in a systematic way based on existing tailoring ideas [13], [14], [15], [16], [17], [18], and [19]. Besides that, the appropriate tailoring level is directly related to optimize resource utilization. Therefore, the main objective is to develop a tailoring framework that is value-driven [5]. The values of the Critical Success Stakeholders (*CSSs*) are characterized through a set of quality attribute. The contribution of software development activities to the quality attributes are determined and used as an indicator to judge the appropriate tailoring level for certain software project characteristics.

2   Methodology

Our methodology starts by analyzing the goals of the *CSSs* in order to identify the quality attributes that influence the behavior of the software process. This is based on the concept of Value Based Software Engineering (*VBSE*) [3] where the rational decision making always relies on the realized *CSSs* value. The *VBSE* theory is addressed and applied in the managerial aspects of software engineering, plus considerations involved in the personal, cultural, and economic values [11]. *VBSE* integrates the stakeholder's value propositions into the system's definition, design, development, deployment, and evolution. These processes are critical to the cost and associated value of system's success which is assessed by quantitative and qualitative sources. *VBSE* is unavoidably involved with software and information system product and process technology, and their interaction with human values [12].

From our point of view, the *CSSs* in software development can be classified in three basic groups; organizational management team, software development team and customer. The organizational management team is responsible to monitor the overall performance of software processes and resource utilization. The development team focuses on the working environment issues while the customer obviously is interested in the quality of the developed software product. Table 1 shows some examples for major interests of these *CSS* groups. These interests influence the way we operate the software development process.

Table 1. An example of the Quality Attribute following *CSSs*

Critical Success Stakeholders (*CSSs*)	Quality Attribute
Organizational Management team	Resource Utilization, Knowledge Base, Quality Assurance
Software Development team	Project Calendar, Project Risk Management
Customer	Defect

From the value realized by *CSSs* in terms of quality attribute, we can categorize the quality attribute that affects the software processes as in Table 2. Each quality attribute has its own impact on consideration to conduct software processes in either encourage or discourage way. By setting the negative/positive impact to the quality attribute, we can use it as a project characteristic for the tailoring process.

Table 2. An example of the Impact in Quality Attribute based on Tailoring Framework

No.	Quality Attribute	Impact (Negative/Positive)
Q_1	Project Calendar	*Negative impact (-)*: Within the rushed project (time critical), the tailoring degree likely to be set for the most crucial activities only. *Positive impact (+)*: With enough time, the project may include some more activities for a better project quality.
Q_2	Quality Assurance	*Negative impact (-)*: The unusual project needs extra care in some of software processes especially in the area of project management and quality assurance. *Positive impact (+)*: The routine project that normally repeats its past activity may require far less monitoring than the unusual project one.
Q_3	Resource Utilization	*Negative impact (-)*: With limited resources, project deems to perform only top priority software processes. *Positive impact (+)*: More profound range of processes to be included into project according to the available resources.
Q_4	Project Risk Management	*Negative impact (-)*: For unusual and inexperienced project, development team needs more support from extra software process to cope with the uncertainty and risk management. *Positive impact (+)*: The nature of project that team member already gains enough background capabilities will less likely to be worried about lacking of risk preventive.
Q_5	Defect	*Negative impact (-)*: The unusual project needs more attention to the testing related software processes. *Positive impact (+)*: The prototype project can tolerate more on error, so it could perform less aggressive testing related software process.
Q_6	Knowledge Base	*Negative impact (-)*: The best practice project is required particular attention in the overall software process. This due to the fact that organization tries to develop the best practices as reference purpose. *Positive impact (+)*: Contrast to the best practice development project, some extreme software processes can be omitted for simplicity.

As described in Table 3, we also set up three levels of process characteristic that determine how important the process is in the context of quality attribute. In "Mandatory" characteristic, the process contributes major role in this quality attribute and it is definitely be included into the project.

In the case of "Alternative" characteristic, even these process attributes are quite important but there are various choices to be selected from several groups of process path. These processes are grouped from its objective similarity. In the event of one path is selected; the other paths can be omitted from the project. Normally these alternative process attributes are designed to respond various scenarios following different environment.

For "Optional" characteristic, these process characteristics are important only for a certain case. Therefore, these optional characteristics are main target to be tailored for the project.

Table 3. A definition of the level of Importance on Process Characteristic

Level of Importance	Description
Mandatory	Process is critical for quality attribute and have to be included into the project
Alternative	Process is important however it can implement in numbers of way that match up with the quality attribute
Optional	Process may be omitted in the case that irrelevant to the quality attribute

Furthermore, the weighted factors are added to the "Optional" process characteristics for reflecting the level of intensity related to the quality attribute. In our work, three organizational intensity levels are proposed as Preferable (P_o), Neutral (N_o) and Less-relevant (L_o) according to the contribution level of process characteristic (see Table 5).

In our software process tailoring framework, we also use the collection of spent workload (E) data (see Table 8) which means the man hour estimation to perform software processes. Based on contribution score and workload data, we use them to calculate the Goal Based Scoring Matrix ($GBSM$) as a reference for judging the appropriate tailoring level of the project characteristic.

Moreover, there are two definitions of threshold values, the contribution score (CS) which is organizational threshold value and the $GBSM$ threshold level value (see Table 13) which is contribution score per workload (CS/E) (see Table 9). Both of the values are based on software industrial or organizational benchmarking.

Normally, the process characteristic (important level, intensity level and workload) are subjects to be committed from responsible group of organization which may establish by a steering committee group like Software Engineering Process Group ($SEPG$) to consider its appropriation before using in the tailoring framework.

3   The proposed of Goal Based Software Process Tailoring (*GB-SPT*) Framework

The typical problem that occurs during the software process tailoring is how to decide the appropriate tailoring level. Therefore our proposed framework aims to help practitioner to design the process tailoring mechanism in systematic way. Furthermore, organization should have the process tailoring approach that reflects to their valued realization. In our *GB-SPT* framework, the cost/benefit values are calculated in term of contribution score based on process and project characteristics which are related to the *CS* and *GBSM* for performing these characteristics. The *CS* and *GBSM* will be used as a guidance to decide the appropriate tailoring level based on the process characteristic and project characteristic from organizational process and project process respectively. These number are due to the fact that we should include high value of the contribution score from the process characteristics into our project while may omit the less contribution score by considering the organizational threshold value.

The proposed *GB-SPT* framework consists of two parts; organizational process and project process. By considering the flow of the process framework as we illustrate in Fig. 1, and we can summarize the stage of tasks as follow:

- Case of organizational process:

Step 1. Organization specifies the quality attribute that associates with the value realized by *CSSs* in which can be stated in organizational policy and business goal.

Step 2. Organization sets up process characteristics and stores as it part of an Organizational Standard Software Process (*OSSP*). The process characteristics include important level (M_o-mandatory/A_o-alternative/O_o-optional) and the detailed of optional as an organizational intensity level (P_o-preferable/N_o-neutral/L_o-less-relevant).

Step 3. Organization sets up the organizational weighted distribution (WD_o) for responding a process tailoring to the quality attribute.

Step 4. Organization identifies the workload (E) value to perform process characteristic which is specified as an organizational benchmark which refers to man-hour (*M-H*) of a person to perform a strenuous task over time based on organizational definition.

Step 5. Organization specifies the organizational threshold value for the contribution score (*CS*) and the Goal Based Scoring Matrix (*GBSM*) as guidance to decide the appropriate tailoring level. The level of threshold can derive from assessment model in which organization uses as a reference model in software process improvement program.

- Case of project process:

Step 6. According to the project goal based on quality attributes, the project leader specifies the project characteristic as the project weighted distribution (WD_p) in term of a project intensity level (P_p-preferable/N_p-neutral/L_p-less-relevant).

Step 7. From step 6 then, using the project weighted distribution (WD_p) and spent workload (E) data from part of organization to calculate the *GBSM*.

Step 8. From the contribution score, project leader can decide the appropriate tailoring level by comparing the project values with the *CS* and the *GBSM* (*CS/E*) threshold values which is set up from the organizational benchmarking value based on the reference assessment model and finally,

Step 9. The result will be software project life cycle process (*SPLCP*) that responds to the tailored process depend on quality attributes in which are derive from project goals and environments.
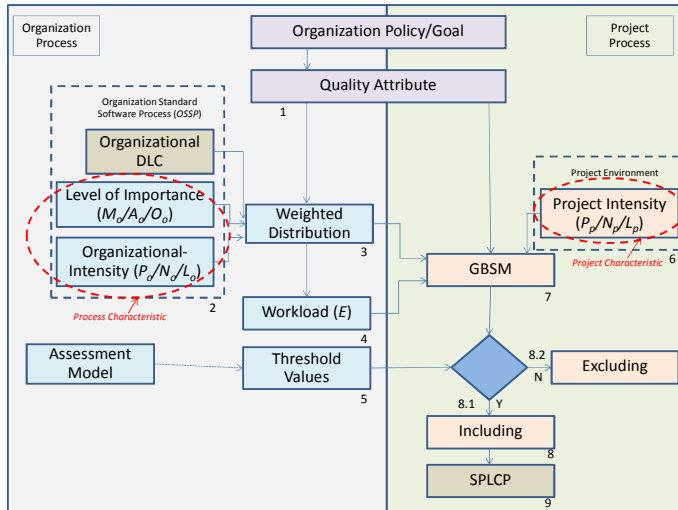

Fig. 1. Goal Based Software Process Tailoring Framework

Table 4 shows an example of process characteristic which is identified quality attributes in term of weighted factor based on process characteristic and intensity level. The organizational weighted factors (W_o) are set as a property of the process attribute that reflect to established quality attribute. The value of weighted factor is determined according to the process attribute. The Mandatory (M_o) group and Alternative (A_o) group are set at "1.0". While "Mandatory" indicates this process must be included, "Alternative" process will only be selected one from the selecting group.

For the "Optional (O_o)" group, the W_o of process characteristic is set according to its intensity level; preferable (P_o) level is set to "0.75", neutral (N_o) level is set to "0.50" and less-relevant level (L_o) is set to "0.25" as shown in Table 5.

Table 4. An example of the P_x and the Organizational Weighted Factor (W_o)

The Organizational Process Characteristic: P_x			
Quality Attribute	**Level of Importance ($M_o/A_o/O_o$)**	**Level of Intensity ($P_o/N_o/L_o$)**	**Organizational Weighted Factor (W_o)**
Q_1-Project Calendar	M_o	-	1.00
Q_2-Quality Assurance	A_o	-	1.00
Q_3-Resource Utilization	O_o	P_o	0.75
Q_4-Project Risk Management	O_o	N_o	0.5
Q_5-Defect	M_o	-	1.00
Q_6-Knowledge Base	O_o	L_o	0.25

Table 5. A definition of the Organizational Process Characteristic (P_x)
with the Organizational Weighted Factor (W_o)

The Organizational Process Characteristic: P_x		
Level of Im-portance **(Mo/Ao/Oo)**	**Level of Intensity** **(Po/No/Lo)**	**Organizational Weighted Factor** **(W_0)**
Mo- Mandatory		1.00
Ao- Alternative		1.00
Oo- Optional	*Po*- Preferable	0.75
	No- Neutral	0.50
	L_o- Less-relevant	0.25

From the organizational weighted factor (W_o), we set each process characteristic in response to quality attribute, the weighted distributions which are calculated within the group of "Optional" characteristic. Because we would like to keep the existence of "Mandatory" and "Alternative" group then we will justify the tailoring level only for the "Optional" group.

The weighted distribution of the process characteristic "P_x" responded to the quality attribute "Q_i" in the form of the organizational weighted distribution (WD_o) which is normalization as $\frac{wf}{\sum_{i=2}^{n} wf}$ where "n" is the number of quality attribute in the "Optional" group. Table 6 shows the result of the organizational weighted distribution (WD_o) for process characteristic "P_x" as illustrated based on objectives/goals based scoring model [5] as an example in table 6 that is normalized from an example in table 4.

Table 6. An example of the P_x and the Oraginzational Weighted Distribution (WD_o)

The Organizational Process Characteristic: P_x	
Quality Attribute	**Organizational Weighted Distribution** **(WD_o)**
Q_1-Project Calendar	1.00
Q_2-Quality Assurance	1.00
Q_3-Resource Utilization	0.50
Q_4-Project Risk Management	0.33
Q_5-Defect	1.00
Q_6-Knowledge Base	0.17

Furthermore, the justification of tailoring level reflects to the project characteristic which calculated by using the organizational weighted distribution (WD_o) as shown in Table 6. Definitely, the project information is present "project goal" in term of "quality attribute" value which is provided by a project leader. Moreover, organization will be prepare reference judgments described in Table 2 for informing project leader to decide based on the project goal as "project characteristic" in term of the project weighted distribution (WD_p) which means how important quality attributes are in different project environment.

The WD_p is determined through the important levels which are preferable (P_p), neutral (N_p) and less-relevant (L_p) as illustrated in Table 7. The project leader has to specify each quality attribute with one of three intensity levels ($P_p/N_p/L_p$), and then process tailoring framework will transform those important levels into weighted factor by calculating the contribution score. The transformed weighted factors are set as "0.75" for preferable, "0.50" for neutral and "0.25" for less-relevant.

Table 7. A definition of the the WD_p values According to the Level of Intensity

Project X	Project Weight Distribution (WD_P)					
	Q_1	Q_2	Q_3	Q_4	Q_5	Q_6
Level of Inten-sity **($P_p/N_p/L_p$)**	0.75/ 0.50/ 0.25	0.75/ 0.50/ 0.25	0.75/ 0.50/ 0.25	0.75/ 0.50/ 0.25	0.75/ 0.50/ 0.25	0.75/ 0.50/ 0.25

The contribution score (*CS*) is the production between organizational characteristic and project characteristic (*CS*=*WD*$_o$*WD*$_p$). The mean value of these contribution scores ($\frac{\sum_{n=1}^{n} CSi}{n}$), where "*n*" is the number of quality attributes (here in our example "*n* = 6"), represents the process contribution to the quality attribute that realized as objectives/goals in project environment.

In General, the process characteristics are yielding a high contribution score reflect its importance to the organizational and project characteristic. In this case, the contribution scores with the lower value than the defined threshold value are targeted to be tailored out. In term of comparing, the process contribution score with organizational threshold value, we can systematically judge the appropriate tailoring level rely on the organizational benchmarking value.

However, the appropriate tailoring level is not justified only with the contribution score (*CS*) value which is higher than threshold value but also take the economical point of view into account. To abide this idea, the workload (*E*) data that indicate the effort to conduct the process characteristics are collected in term of man-hour. This workload data are periodically collected from the observation in their development environment. This will ensure that the information reflects closely to the practical use in a time appropriate.

The workload is consists of three levels according to how much man-hour needed to perform process characteristics as in Table 8. The workload weighted factor (*WD*$_e$) three levels are set in the same concept of intensity level as 0.75 for level 3 (hard), 0.50 for level 2 (normal) and 0.25 for level 1 (light).

Table 8. A definition of the Degree of Workload and the Workload Weighted Factor (*WD*$_e$)

Degree of Workload	Workload Value (*M-H*)	Workload Weighted Factor (*WD*$_e$)
Level 3 (Hard)	$E >= 20$	0.75
Level 2 (Normal)	$10 =< E < 20$	0.50
Level 1 (Light)	$0= < E < 10$	0.25

The *GBSM* that used for judging the appropriate tailoring level is the result from the contribution score divided by workload (*CS/E*) as illustrated in Table 9. The value of *E* here is the weighted factor that be transformed from the workload level as described in Table 8.

This *GBSM* shows the contribution level of each process characteristics responded to the project goal as a contribution score gain per workload. The comparison between scoring matrix and organizational threshold value can be used as guidance for project leader to decides the appropriate tailoring level (*CS/E* should be more or equal to threshold value).

Table 9. A definition of the Goal Based Scoring Matrix (*GBSM*) concept

Contribution Score per Workload	Goal Based Scoring					
	Process A	Process B	ProcessC	Process D	…	Process N
	CSp_A/Ep_A	CSp_B/Ep_B	CSp_C/Ep_C	CSp_D/Ep_D	…	CSp_N/Ep_N

The final software project life cycle process (*SPLCP*) will be revise from tailoring procedure by using *GBSM* which is based on process and project characteristics that are categorized as "Mandatory" and "Alternative". In summary, all of the "Mandatory" processes will be included into the project while the "Alternative" will select only one from the alternative groups. Lastly, "Optional" is selected from contribution score and scoring matrix values.

4 Simulating Goal Based Software Process Tailoring (*GB-SPT*) Framework and Example

We demonstrate our proposed for *GB-SPT* framework, the process and project characteristics are selected to show the idea of using quality attribute as objectives/goals for tailoring software process to

fit the project environment. Fig. 2 illustrates an example of the "Project Planning" process in our environment which is composes of four procedures which are respectively defining as create project definition, establish work break down structure (*WBS*), calculate project estimation, then review and commit the project confirmation. Regarding from this simulated example, agile is selected among three alternatives of software development lifecycle (*SDLC*) model which are selected from Waterfall, Adaptive and Agile SDLC model.

In this simulation, we shown an example of the alternatives and the process characteristic of "project planning" from *OSSP* as well as the scenario that quality attributes are set as "PNLLPN" in case Y and "PNLLNN" in case Z as shown in Fig. 3.
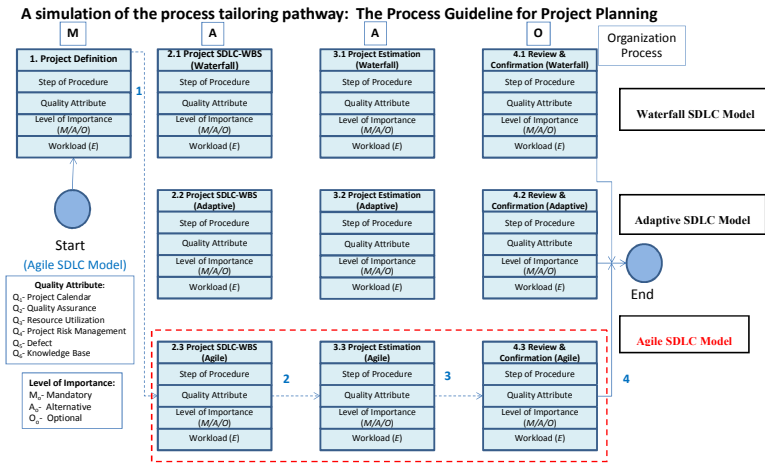

Fig. 2. An example of the Process Tailoring Simulation in "Project Planning"
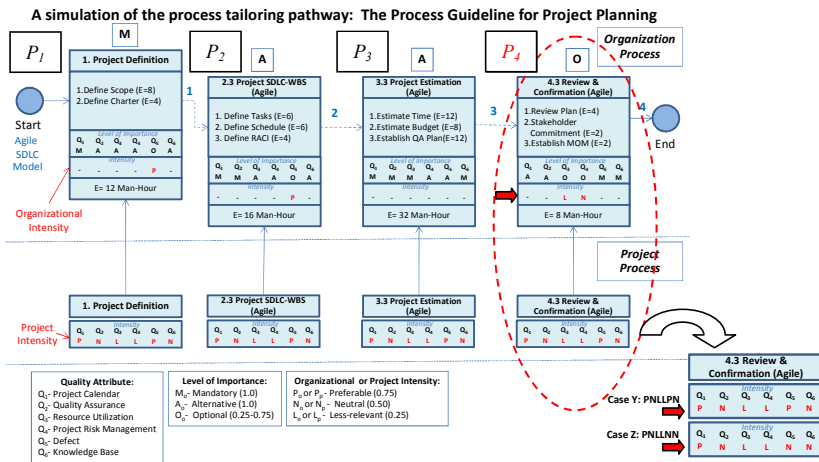by selecting "Agile" pathway from *OSSP*

Fig. 3. An example of the simulation for Case Y and Case Z in "Agile" pathway

From Fig. 3, we capture an example of "4.3 Review & Confirmation (Agile)" as "P_4" process to describe a detail of setting the process characteristic and the organizational weighted factor (W_o) as shown in Table 10. Then, table 11 shows the result of the organizational weighted distribution (WD_o) according to the W_o.

Table 10. An example of the P_4 process and the organizational weighted factor (WD_o)

Process: P_4-Review & Confirmation (Agile)			
Quality Attribute	**Process Attribute**	**Intensity Level**	**Organizational Weighted Factor (W_O)**
Q_1-Project Calendar	A	-	1.00
Q_2-Quality Assurance	A	-	1.00
Q_3-Resource Utilization	O	L	0.25
Q_4-Project Risk Management	O	N	0.50
Q_5-Defect	M	-	1.00
Q_6-Knowledge Base	M	-	1.00

Table 11. An example of the WD_o of P_4 process according to W_o from Table 10

Process: P_4-Review & Confirmation (Agile)		
Quality Attribute	**Organizational Weighted Factor (W_O)**	**Organizational Weighted Distribution (WD_o)**
Q_1-Project Calendar	1.00	-
Q_2-Quality Assurance	1.00	-
Q_3-Resource Utilization	0.25	0.25/(0.25+0.50) = 0.33
Q_4-Project Risk Management	0.50	0.50/(0.25+0.50) = 0.67
Q_5-Defect	1.00	-
Q_6-Knowledge Base	1.00	-

According to the steps we described in section 3, the result of contribution score (CS) as display in Table 14. This is based on this project scenario case Y (PNLLPN) as shown in Table 12. From our preliminary setup of organizational threshold value as 0.45 as display in Table 13, this CS result shows that the tailoring framework will not recommend any processes to be left out from this project scenario.

Table 12. An example of the Project Intensity Value of the case Y and the case Z

Case Y: Project Intensity Value					
Q_1	Q_2	Q_3	Q_4	Q_5	Q_6
P	N	L	L	P	N
0.75	0.50	0.25	0.25	0.75	0.50

Case Z: Project Intensity Value					
Q_1	Q_2	Q_3	Q_4	Q_5	Q_6
P	N	L	L	N	N
0.75	0.50	0.25	0.25	0.50	0.50

Table 13. An example of the Organizational threshold value

Organizational threshold	Value
Contribution Score (CS)	0.45
Contribution Score/Workload (CS/E)	0.90

As stated earlier, the workload (*E*) are observed and set up in terms of man-hour and transformed into threes level of the workload weighted factor (*WD*$_e$)as shown in Table 8. The *GBSM*, which takes the workload (*E*) into account, is calculated as (*CS/E*) and the results are displayed as in Table 15. For *GBSM* threshold value that was set at 0.90 which formulate from 0.45 divide by 0.50 which is a mid-weighted factor point of "Level of workload", then, the process *P*$_3$ is recommended to be tailored out from this project scenario. Then finally, we left the process *P*$_1$, *P2* and *P4* after our *GB-SPT* framework for the case Y.

Table 14. The Simulation of Contribution Score (*CS*): Project Planning (*P*$_{1-4}$) in the case Y (PNLLPN scenario) based on *CS* threshold = 0.45

Contribution Score (CS)	Goal Based scoring			
	Process 1	Process 2	Process 3	Process 4
	0.58	0.58	0.58	0.49

Table 15. The Simulation of Goal Based Scoring Matrix (*GBSM*): Project Planning (*P*$_{1-4}$) in the case Y (PNLLPN scenario) based on *CS/E* threshold = 0.90

Quality Contributions per Workload	Goal Based scoring of Project Planning (*P*$_{1-4}$)			
	P$_1$	*P*$_2$	*P*$_3$	*P*$_4$
CS	0.58	0.58	0.58	0.49
E	0.5	0.5	0.75	0.25
CS/E	1.17	1.17	0.78	1.94

From Table 14 and 15 above, we also demonstrate the different result from dissimilar project scenarios based on a project intensity value as the scenario case Z which is set (PNLLNN) as shown in table 12, the result of the contribution score (*CS*) and the *GBSM* are shown in Table 16, and Table 17.

Table 16. The Simulation of Contribution Score (*CS*): Project Planning (*P*$_{1-4}$) in the case Z (PNLLNN scenario) based on *CS* threshold = 0.45

Contribution Score (CS)	Goal Based scoring			
	Process 1	Process 2	Process 3	Process 4
	0.46	0.46	0.46	0.42

Table 17. The Simulation of Goal Based Scoring Matrix (*GBSM*):: Project Planning (*P*$_{1-4}$) in the case Z (PNLLNN scenario) based on *CS/E* threshold = 0.90

Quality Contributions per Workload	Goal Based scoring of Project Planning (*P*$_{1-4}$)			
	P$_1$	*P*$_2$	*P*$_3$	*P*$_4$
CS	0.46	0.46	0.46	0.42
E	12	16	32	8
CS/E	0.92	0.92	0.61	1.67

With the same organizational threshold level as *CS* value, now in this project scenario (PNLLNN) the process P_4 is also recommended to be tailored out from this project. For *GBSM* threshold value that was set at 0.90, the process P_3 are also recommend to be tailored out from this project scenario. Again, finally, we left the process P_1 and P_2 after our *GB-SPT* framework for the case Z.

Simulations here show how we can use quality attribute for process tailoring mechanism. By setting the quality attribute that fit to the project environment and also the appropriate organizational threshold value, then our proposed framework can recommend the tailoring level in systematic way. The result of objectives/goals based scoring matrix gives a set of process that contributes most value to the project goals. Then we can use it as an initial set of process tailoring framework for further necessary adjustment.

5 Conclusions and Future works

This research proposes the idea of developing the systematic process tailoring framework as *GB-SPT* that respond to the organizational value realization point of view as well as the project environment. The judgment of tailoring level is based on the quality attribute which is established by the organizational Critical Success Stakeholders (*CSSs*). The quality attribute is distributed to calculate the contribution score (*CS*) for process and project characteristics. The underlying idea of Value Based Software Engineering (*VBSE*) is to conduct a process and project value which is used to calculate the contribution score gain per workload ratio in terms of quality attribute. The project is systematically determined by considering the *GBSM*.

To enhance the advantage of using this *GB-SPT* framework, we plan to add indicator which can reflect to the industrial standard like Capability Maturity Model Integration (*CMMI*). This enhancement will ensure that not only does the appropriate tailoring level which is consider the contribution score or quality gain per workload but it also supports the need of industrial standard.

6 References

1. M. Sivashankar, A.M.Kalpana, Dr. A. Ebenezer Jeyakuma: A Framework Approach Using CMMI for SPI to Indian SME's. In: 2010 International Conference on Innovative Computing Technologies (ICICT) (2010)
2. M.P.Thapliyal, Pratibha Dwivedi; Software Process Improvement in Small and Medium, Software Organizations of India. In: International Journal of Computer Applications (0975 – 8887), Vol. 7– No.12, pp. 37–39 (October 2010)
3. S. Biffl, A. Aurum, B. Boehm, H. Erdogmus, P. Grunbacher (eds.): Value-Based Software Engineering. Springer (ISBN 978-3-540-29263-0) (2006)
4. N. du Preeza, D. Luttersb and H. Nieberdinga: Tailoring the development process according to the context of the project. In: CIRP Journal of Manufacturing Science and Technology, Vol.1 Issue 3, pp. 191-198 (2009)
5. Tisanai Krisanathamakul, Chayakorn Piyabunditkul, Apinporn Methawachananon, Pawarat Nontasil: SPI through Process Tailoring Framework with Objective-Based Scoring Metric. In: EURO SPI 2012, pp. 11.23-11.31 (June 2012)
6. Victor R. Basili and H.Dieter Rombach: Tailoring the software process to project goals and environments. ACM, pp. 345-357 (1987)
7. Julio A. Hurtado Alegria, Maria Cecilia Bastarrica, Alcides Quispe, Sergio F. Ochoa: An MDE Approach to Software Process Tailoring. In: ICSSP 2011, Waikiki, Honolulu, HI, USA, ACM (May 21-22, 2011)
8. Mark P. Ginsberg and Lauren H. Quinn: Process Tailoring and the software maturity model. In: CMU/SEI-94-TR-024, Carnegie Mellon University, Pittsburgh, pp. 1–53 (November, 1955)
9. N. du Preeza, D. Luttersb and H. Nieberdinga: Tailoring the development process according to the context of the project. In: CIRP Journal of Manufacturing Science and Technology, Vol.1 Issue 3, pp. 191-198 (2009)
10. Pettersson F., Ivarsson M., and Gorschek T.: A Practitioner's Guide to Light Weight Soft-ware Process Assessment and Improvement Planning. In: Journal of Systems and Software, vol. 81(6), pp. 972-995 (2008)
11. Barry Boehm, Dan Port, and Kevin Sullivan: Value Based Software Engineering. In: Workshop on New Visions for Software Design and Productivity (December 13-14, 2001)
12. Barry Boehm, Apurva Jain: An Initial Theory of Value-Based Software Engineering. In: USC-CSE-2005-502 (February 2005)
13. Budlong, F., Szulewski, P., Ganska, R.: Process tailoring for software project plans. In: Technical Report, Software Technology Support Center of the U.S. Air Force (1996)

14. Basili, V., Rombach, D.: Tailoring the software process to project goals and environments. In: Communications of the ACM, pp. 345-357 (1987)
15. Nanda, V.: On tailoring an organizational standard software development process for specific projects. In: Proceedings of the 11th International Conference on Software Quality, pp. 1-13 (2001)
16. Oscar Pedreira, Mario Piattini, Miguel R. Luaces and Nieves R. Brisaboa: A Systematic Review of Software Process Tailoring. In: ACM SIGSOFT Software Engineering Notes, Volume 32 Issue 3, pp. 1-6 (May 2007)
17. Bustard, D.W., Keenan, F.: Strategies for systems analysis: Groundwork for process tailoring. In: Proceedings of the 12th IEEE International Conference and Workshops on the Engineering of Computer-Based Systems (ECBS05), pp. 357- 362, IEEE Press (2005)
18. Julio A. Hurtado Alegría, María Cecilia Bastarrica, Alcides Quispe, and Sergio F. Ochoa: An MDE approach to software process tailoring. In: ICSSP '11 Proceedings of the 2011 International Conference on Software and Systems Process pp. 43-52 (2011)
19. Peng and Balasubramaniam Ramesh: Software Process Tailoring: An Empirical Investigation. In: Journal of Management Information Systems, Volume 24, Number 2/Fall 2007, pp. 293-328 (2007)

Knowledge Management to Support using Multi-model Environments in Software Process Improvement

Muñoz Mirna, Mejia Jezreel, Muñoz Edrisi
Centro de Investigación en Matemáticas, Av. Universidad no 222, 98068 Zacatecas, México
{mirna.munoz, jmejia, emunoz}@cimat.mx

Abstract

Organizational software process improvement offers a key opportunity for organizations to become more efficient. As consequence the software industry, among others, is more interested in it. However, one of the most common issues identified when an organization tries to implement a software process improvement initiative is the difficulty that they must face in the selection of the reference model and its adaptation to their current scenario. Moreover, selecting the wrong reference model according to the way the organization works becomes a trigger to increase resistance to change. In this context, effective integration of models and standards means a tool that improve information sharing and communication that can play a crucial element for the implementation of multi-model environments to be used as reference model in software process improvements. In this sense, knowledge management technologies have proved to be highly promising for supporting this integration. In this work, an ontological framework is presented as the technology for information and knowledge models sharing. The paper includes the development of the model process ontology that is considered a key element toward providing the right knowledge management in multi-model environment according to organization business goals needs to implement software process improvements.

Keywords

Software process improvement, knowledge management, software supporting tools, multi-model environment, ontologies, folksonomies

1 Introduction

Nowadays, software stands for a main building block in developing the activities of many companies and organizations since it creates added value to products and services. As a result of the increasing importance of software, new challenges and demands on software development, operation and maintenance appear [1]. Then, the software industry is becoming an important factor in the core of the economy around the world. In this context, it is well known that the quality of software products is largely dependent on the processes that are used to create it [2], therefore, the process improvement allows organizations to create strategic advantages with respect to its competitors [3].

However, although many organizations are motivated to improve their software processes, very few know how to do so in a proper way. One of the problems of introducing process improvement in organizations is because they are under mandatory or market pressure to use more than one improvement model [4]. According to Lisa Marino and John Morley [5], three out of every five large organizations are already facing the challenges of using multiple models to meet the organizational business goals.

As a solution, organizations worldwide are adopting several international standards and models, in an effort to improve their processes [6]. However, the difficulty of implementing successfully process improvement using multi-model environments as reference is well known. Two issues associate with this difficulty are: a) select what models or process areas make sense for each organization [7] and 2) collect and process huge amounts of data that must be managed in those models and standards.

Therefore, the goal of this paper is to present a knowledge management framework that supports the establishment of multi-model environments as reference to implement software process improvements.

This paper is structured as follows: section two presents the background; section three presents the knowledge management framework; section fourth describes the development of the knowledge management framework; section five presents the knowledge management framework usability and; finally, section six shows the conclusions.

2 Background

The method showed in this paper was developed as an evolution of the method proposed by a methodology for a gradual and continuous software process improvement, focusing on minimizing change resistance called MIGME-RRC (its Spanish acronym) [8] to analyze external best practices. After analyzing the method, our research team identified the next findings: 1) the method needs to identify business goals and 2) the method needs support to integrate different models and standards. Based on the findings two actions were performed as follows:

2.1   Updated Multi-model Method

Key activities in order to formalize tacit knowledge and asses the process performance are included based on "identifying best practices method" and "assess the organizational performance" from MIGME-RRC methodology.

Using a top-down approach to establish the multi-model environment, the updated method is composed of 5 steps: 1) Identify organizational business goals; 2) Identify internal best practices, 3) Assess organizational current practices performance, 4) Priority process areas 5) Analyze external best practices and 6) Identify dependences. The updated method is showed in Figure 1.

As Figure 1 shows, the updated method starts formalizing and understanding the business goal. Then, the organizational knowledge is analyzed identifying the internal practices that are carried out in the organization. After, the organizational internal practices performance is assessed. Next, depending on the internal

practices performance, process areas are prioritized to address the improvement effort. Then, external best practices are analyzed to select those practices that will be included in the multi-model environment. Finally, the analysis of dependences among best practices is done and the multi-model environment is integrated.

By this way, the resulted method helps organizations in the use of a multi-model environment as reference model, based on the needs of organizational business goals, the current organizational knowledge and having into account the dependences between the current practices and the external practices to be implemented. By this way organizations are able to choose those external practices that best fit the way they work and make more efficient its processes.
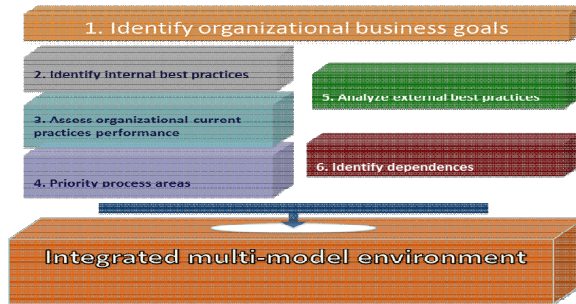
Figure 1. Multi-model updated method

2.2 Fidings After Appliying the Updated Muti-model Method

The first four steps of the updated multi-model method have been applied to a set of SMEs of software development from Zacatecas, México. After applying these phases, there were achieved the target to address the improvement effort. However, there were highlighted some needs in order to optimize the use of the method too.

Next, there are listed the identified needs:

- The analysis and formalization of tacit information takes a lot of time because the interviews generate too much information, so it difficult its analysis.
- The mapping among models and standards implies huge amounts of information
- The time to provide a multi-model environment takes more than tree moths because the amounts of information that must be analyzed.

In this context, it is important to remember that SMEs features such as, limited customers with high dependency, projects with short delivery time, limited staff with many activities, lack of processes culture, and lack of resources. Then, the three findings above mentioned difficult the application of the updated multi-model method because of the time that represent to applied it.

This reason highlights the need to provide support to the updated multi-model method in order to make it more effective to implement multi-model improvements in SMEs. Even if the method is applied to large organization the time of performing it can be significantly reduced.

Therefore, after performing a systematic review focused on identify expected requirements in support tools for software process improvement in SME's [10], there were focused the use of ontologies and folksonomies as a solution to manage huge amount of information in the improvement domain.

Besides, we believe that providing a knowledge management framework should meet the optimization of the updated multi-model method requirements as follows: a) representing the resource model, which

comprises the different economic, human, informatics and material devices; b) representing the procedure model, which stands the strategy for carrying out a process; c) defining the sequence of activities for the transformation or development of software and the associated products; and d) providing the right knowledge management in multi-model environment according to organization business goals needs.

Next sections, define the Knowledge management framework proposed by this research work.

3 Knowledge Management Framework

Ontologies are emerging as a key solution to knowledge sharing in a cooperative business environment, since they can express knowledge and knowledge relationships with clear semantics. Then, ontologies play an important role in forthcoming information-management solutions to improve the integration of information in processes domain.

This work applies semantic technologies for the representation of standard and models processes to establish a multi-model environment. Therefore, the updated multi-model method proposes to use an ontology as support that should be general enough to be applied to any software process organization. Additionally, this ontology can be used as a straightforward guideline for standardizing processes.

To achieve this, the knowledge management framework proposed in this research work comprises the development and application of one folksonomy and two ontologies, as follows:

- *Tacit Process Folksonomy:* it is in charge of facilitating the modeling and formalization of tacit knowledge extracted from the organizational processes and process users.
- *Process Model Ontology:* it is in charge of refining processes previously modelled standardizing them (e.g. the project planning process currently performed in an organization). It is also in charge of simplifying the processes posed by the different software processes improvement technologies (e.g. project planning processes from the CMMI or ISO-15504).
- *Ontology for Software Process Improvement Technologies:* this ontology is in charge of integrating the different technologies focused on helping organizations to improve their processes. The aim of this ontology is to identify the common and minimum software improvement components found within the technologies. Those components are directly related to different process best practices, which are considered key elements in software process improvements.

By this way, once the process has been captured and standardized, the proposed knowledge management framework stands for an opportunity to suggest the use of best practices which best fit and cover the essentials issues of an organization in order to have a more efficient process improvement.

4 Development of the Knowledge Management Framework

As mentioned in last section, the knowledge management framework proposed in this research work has been divided in three phases; due to process is the core of this research work, as first phase, this research work is presenting the development of the process model ontology. This ontology aims to have a common terminology of processes. Then, the ontological development and application cycle methodology presented in [11] has been applied in order to develop the process model ontology. Following sections describe the phases proposed by PDCA methodology to develop the process model ontology.

4.1 PDCA Methodology

To develop a support to integrate different models and standards this work uses the Plan, Do, Check (Study) and Act (by its acronym PDCA) cycle methodology [11]. PDCA methodology allows us to design an ontological model that supports the multi-model updated method. As Figure 2 shows, PDCA proposes a cycle of four steps for problem solving that includes: planning (definition of a problem and a hypothesis

about possible causes and solutions), doing (implementing), checking (evaluating the results), and action (back to plan if the results are unsatisfactory or standardization if the results are satisfactory) [11].
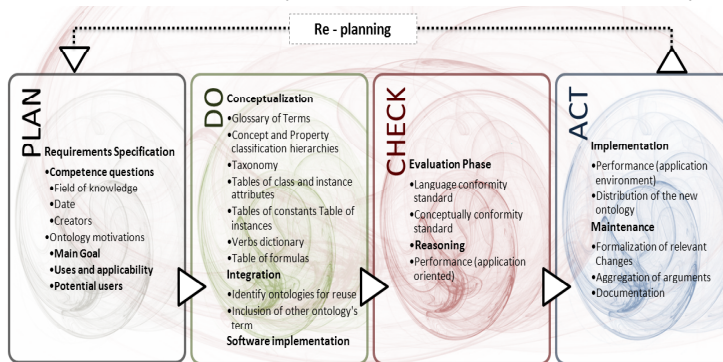

Figure 2. PDCA methodology [11]

The PDCA cycle emphasizes the prevention of error recurrence by establishing standards and the ongoing modification of those standards [11].

The PDCA cycle is applicable to all types of organizations and to all groups and levels in an organization, and some important issues attained are well known as it: (1) provides a framework for the application of improvement methods and tools; (2) allows project plans to adapt as learning occurs, (3) provides a simple way for people to empower themselves to take action that leads to useful results in the pragmatic tradition of learning, and (4) facilitates the use of teamwork to make improvements [11].

4.2 Plan Phase

As first step from the plan phase, the scope of the process model ontology has been defined and formalized. Specifically the domain lies in the structuration of different processes that are found in software development organizations. Then, it is necessary to capture the essentials of a wide range of process activities related to software development and maintains and the associated products. Those functions have been semantically modelled using the necessary components for describing a process.

As second step in this phase, it is necessary to define the domain to be modelled based on adequate sources of knowledge. The sources of knowledge are encompassed by international software process improvement technologies such as Capability Maturity Model and Integration for Development (CMMI-DEV) [12]; Team Software Process (TSP) [13]; Project Management Body of Knowledge (PMBOK)[14]; ISO/IEC15504 Information technology – Process assessment [15]; ISO 9001:2000- Quality Management System [16] and ISO/IEC 12207-2008 [17]. Even more, the use of another standard of production process, specifically ANSY/ISA- standards, has been used allowing a better model characterization. Besides, the web ontology language (OWL) has been used for modelling the ontology (following the language of previous works [11]).

It is important to highlight that for this process model ontology a software process is a set of activities, methods, practices, and transformations which people use to develop and maintain software and the associated products [18]. In a more specific way, the software process should establish both the management and technical environment in which tools; methods and people are applied in order to perform the task of developing software.

Therefore, to talk about software process implies: 1) identify roles and specific activities; 2) establishes metrics and 3) provides input and output criteria for each step related to the software development and it related products [19].

4.3   Do phase

A glossary of terms has been developed containing and identifying the usable and potential terms, as well as, their right meaning for the process domain. Then, the main components of the process domain are gathered from the sources of knowledge and listed on the glossary of terms. Table 1 shows a part of the entire table which consist of 34 key potential terms we have identified for process domain.

Table 1. Concepts related to the software process domain

Action	The state of doing. Minor processing activity (lowest level of processing activity)
Activity	Measurable amount of work performed to convert inputs into outputs. It is composed by a set of actions.
Availability	Perform its designated or required function. It is the aggregate of the resource's accessibility, reliability, maintainability, serviceability, and security.
Change	A variation, deviation, or modification in the current state.
Continuous Improvement	The action of improve continuously the efficiency of a process
Customer Satisfaction	The degree of satisfaction provided by the goods or services of a company as measured by the number of repeat customers.
Customer	An individual or organization that receives or consumes products (goods or services) and has the ability to choose between different products and suppliers.
Development	An event constituting a new state in an individual process to meet specific objectives or requirements.
Environment	The sum total of all surroundings of a process which provide conditions and its development
Goal	An aim or desired result. A projected computation of affairs that a person or a system plans or intends to achieve.
Process	A major processing activity that usually results in a change in the input element in the initial state.

The aforementioned terms are formalized in a taxonomic manner considering the "is-a" property. As a result, the whole domain is organized in seven top terms or classes, as shown in Figure 3. Each top class unfolds their corresponding subclasses. It is important to mention that description logic (DL), automatically name to the root class as Thing (owl:Thing).

In order to identify the properties that relate the new classes of ontology contained in the taxonomy, the following verbs have been proposed: *to carry out, to define, to fulfill, to derive, to improve, to has and to provide* among others. Besides, the properties of the ontology have been identified and the data and the object type properties established the relationship among the classes of the taxonomy. A list of some of the object and data type properties is presented in Tables 2 and 3.

The restrictions and axioms build constraints between the properties are mentioned in Tables 2 and 3. Some examples of restrictions are quantifier restrictions, cardinality restrictions and hasValue restrictions. Finally, ontology model contains 67classes and 57 properties. The ontology has been edited in Protégé [20], which is a description logic reasoning system. Protégé is widely used open source ontology and knowledge based editor.

Figure 3. Process model ontology domain

Table 2. Example of object properties relating to software process classes

Property	Domain	Range
makes_up	Action	individualProcess
provided_by	CustomerSatisfaction	Customer
has_measurement	CustomerSatisfaction	Measurement
is_ruled_by	Activity	Policy

Table 3. Example of data properties relating to software process classes

Property	Domain	Range
finalTime	TimeInterval	Date time
InitialTime	TimeInterval	Date time
maxValue	TimeInterval	Float
referenceName	TimeInterval	String
unitofMeasure	TimeInterval	String

4.4 Check phase

In this phase, the language and the conceptuality have been checked with the support of experts in the domain to ensure that the ontology process model meets the user's requirements. The ontology validation has been supported by a short informatics application, which allows to navigate through the ontology via web and to make annotations about the structure of the classes and the properties between them. Basically the ontology has been checked by members of Software Engineering Doctors expert on software process improvements of CIMAT- Zacatecas Research Center. All of them involved in process, software process improvement and multi-model domain. Besides, to check the ontology use, reasoners such as RacerPro Reasoner from Protégé and Pallet Reasoner from Swoop were used to validate the consistency of the model resulting in a successful compilation (see figure 4).
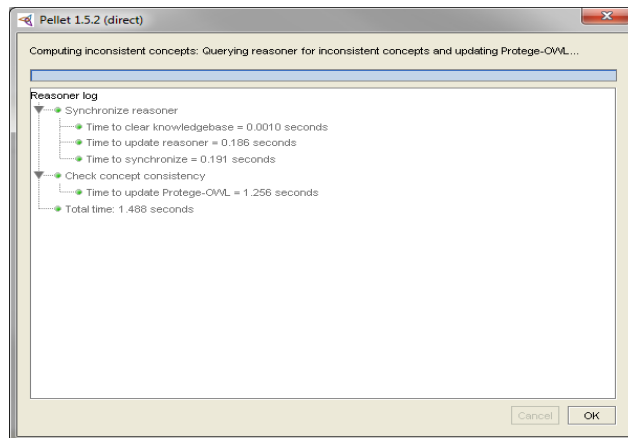
Figure 4. Successful reasoning of process ontology consistency

4.5 Act phase

This phase comprises all the actions over the model necessary to repair the defects and suggestions arisen from the check phase. Although additional actions have been defined, the number of actions has decreased as a result of the model debugging, which eliminated unnecessary or unused restrictions. Finally, all the changes made in the ontological model, must be recorded in the documentation of the project. In addition, all the changes related to the use of the current model must be monitored, checked and recorded. Again, all the formal changes and the aggregation of arguments must be recorded in the documentation of the project.

5 Knowledge Management Framework Usability

Software process improvement provides a way to create strategic advantages to organizations in order to be more efficient and therefore competitive. However, the implementation of software process improvements has not have the expected results mainly because the difficulty that an organization faces when adapting the selected reference model to their current scenario [21] [22].

 Then, the use of multi-model environments as reference to implement software process improvements has been raised. However, using multi-model environments includes to collect and process huge amounts of data that most of the time is difficult to manage for an organization.

This research work is useful because it provides a support to achieve the selection of correct models or process areas practices that make sense for an organization [7]. So it allows to establish a multi-model environment to be used as a reference accorded to organization business goals needs when implementing a software process improvement.

In this context, a key aspect is to be able to storage "processes" in a right way, respect to this, our research work emphasize on achieving the storage of organizational processes in a standardized way, so they can be managed and served as a base to address the improvement effort having a solid based that allows: 1) to address the improvement effort toward the achievement of organizational business goals; 2) to provide a multi-model that contains those practices that best fit the way the organization works; and 3) to get a software process improvement focused on the organization need instead of reference models or standards.

6 Conclusions

Nowadays, not all software improvement implementations have the expected results. One of the main problems is the hard work that the implementation of a specific models and standards represent for organizations because the amount of internal and external knowledge that must be managed in this kind of initiatives.

In this way, this research work presents a solution to this problem providing: 1) a method that helps organization to establish multi-model environments; and 2) a knowledge framework that allows to optimize and standardize the management of information in software process improvement.

By this way it is possible to provide organizations with best practices through multi-model environments accorded to their business goals and that best fit with the way they work. Therefore, current organizations processes become more efficient respect to the achievement of the organizational business goals.

7 Literature

1. Münch, J., Armbrust, O., Kowalczyk, M., & Soto, M. Software Process Definition and Management. Berlin/Heidelberg: Springer. (2012).

2. Williams T. How do organizations learn lessons from Projects—And do they?. IEEE Transactions on Engineering Management, 55(2). 248-266. (2008).

3. Soto-Acosta P., Martínez-Conesa I., & Colomo-Palacios R.. An empirical analysis of the relationship between IT training sources and IT value. Information Systems Management, 27(3), 274-283. (2010).

4. Garcia S. "Process Improvement At the Edges", Software Engineering Institute (SEI) Library. Carnegie Mellon University. 1-22. Presentation Retrieved June 2009 from http://www.sei.cmu.edu/library/assets/20081218webinar.pdf. (2007).

5. Marino L & Morley J. Process Improvement in a Multi-model Environment Builds Resilient Organizations. NEWS AT SEI, Software Engineering Institute (SEI). Retrieved December 2009 from http://www.sei.cmu.edu/library/abstracts/news-at-sei/02feature200804.cfm. (2009).

6. Andelfinger U., Heijstek A. & Kirwan P. A Unified Process Improvement Approach for Multi-Model Improvement Environments. NEWS AT SEI, Software Engineering Institute (SEI). Retrieved December 2008 from http://www.sei.cmu.edu/library/abstracts/news-at-sei/feature1200604.cfm. (2009).

7. Forrester E. & Wemyss G. CMMI and Other Models and Standards. CMMI Version 1.3 and Beyond. 81-108. Retrieved December 2012, from http://www.sei.cmu.edu/. (2011).

8. Calvo-Manzano J. A., Cuevas G., Gómez G., Mejia J., Muñoz M., & San Feliu T. Methodology for process improvement through basic components and focusing on the resistance to change. Journal of Software Evolution and Process, 24 (5), 511-523. (2012).

9. Muñoz M., Mejia J., Calvo-Manzano J. A., San Feliu T. & Alor G. Advantages of using a multi-model environment in software process improvement, The Electronics, Robotics and Automotive Mechanics Conference (CERMA). 2011 IEEE (pp. 397-402), Cuernavaca, Morelos, DOI 10.1109/CERMA.2011.85. (2011).

10. Muñoz Mirna, Mejia Jezreel, De Amescua Antonio, Calvo-Manzano Jose A., Cuevas Gonzalo, San Feliu Tomás, Expected requirements in support tools for software process improvement in SMEs, CERMA 2012- Electronics, Robotics and Automotive Mechanics Conference, pp 135-141, ISBN 978-0-7695-4878-4. (2012).

11. Muñoz, E., Espuña, A., & Puigjaner, L. Knowledge Management Tool for Integrated Decision-making in Industry: Technology for ontology development aimed at integrated decision support systems in process industries. LAP LAMBERT Academic Publishing. (2012).

12. SEI CMMI Production Team, S.: CMMI for Development v1.3. Software Engineering Institute, January, (2011).

13. Humphrey W. Introduction to the team software process. Massachusetts: Addison-Wesley. (2006).

14. IEEE Computer Society. IEEE guide adoption of PMI standard A guide to the project management body of knowledge; IEEE Std 1490-2003 (Revision of IEEE Std 1490-1998). 0_1-216. doi: 10.1109/IEEESTD.2004.94565. (2004).

15. International Organization for Standardization. ISO/IEC 15504: 2004 Information Technology – Process Assessment, Part 1 to Part 5. (2004).

16. AENOR. Sistemas de gestión de calidad (ISO 9001:2000), UNE-EN ISO 9001, Norma Española. (2000).

17. International Organization for Standardization. Systems and Software Engineering - Software Life Cycle Processes, IEEE STD 12207-2008," Ieeexplorer, pp. cl-138. (2008).

18. Abran, A., Moore, J.W., Bourque, P., Dupuis, R., Tripp, L.L.: Guide to the Software Engineering Body of Knowledge (SWEBOK). IEEE. (2004).

19. Cuevas, G., De Amescua, A., San Feliu, T., Arcilla, M., Cerrada, J.A., Calvo-Manzano, J.A., García, M. Gestión del Proceso Software. Universitaria Ramon Areces, pp. 406-447, (2002).

20. Horridge, M., Jupp, S., Moulton, G., Rector, A., Stevens, R., Wroe, C. A practical guide to building owl ontologies using protege 4 and co-ode tools, Tech. rep., The University Of Manchester, United Kingdom. (2007).

21. Potter N. & Sakry M. Developing a Plan. In Addison-Wesley (Ed.), Making Process Improvement Work. pp. 1-49. (2006).

22. Morgan P. Process Improvement- Is it a lottery? Methods & Tools, practical knowledge for the software developer, tester and project manager, 15(1), 3-12, Retrieved December 2007, from http://www.methodsandtools.com/PDF/mt200701.pdf, ISSN 1661-402X. (2007).

8 Author CVs

Muñoz Mirna

She holds a PhD in Computer Science. She is currently a researcher in Computer Science at the Research Center in Mathematics (CIMAT), Zacatecas, México. She is a member of member of "Research Chair in Software Process Improvement for Spain and Latin American Region". Her research field is Software Engineering, including both technology (methods, techniques, tools and formalisms) and management. Her current research interests are: how to implement software process improvements focused on minimizing change resistance and the use of multi-model environments and software tools in software process improvements, software process improvement models and standards, project management, change management, knowledge management, software quality, team software process ad IT security. She has published several technical papers on project management, multi-model environment and process improvement. She has been a member of the team that translated CMMI-DEV v1.2 and v1.3 to Spanish.

Mejia Jezreel

He holds a PhD in Computer Science. He is currently a researcher in Software engineering at the Research Center in Mathematics, Zacatecas, México. He is a member of "Research Chair in Software Process Improvement for Spain and Latin American Region". His research field is Software Engineering, including both technology (methods, techniques, tools and formalisms) and management. His current research interests are: software process improvement, multi-model environment, project management, acquisition and outsourcing process, agile methodologies, software process improvement models and standards, knowledge management software quality, team software process and IT security. He has published several technical papers on project management, software teams, software process improvement, acquisition and outsourcing processes. He has been a member of the team that translated CMMI-DEV v1.2 and v1.3 to Spanish.

Edrisi Muñoz

He holds a PhD in chemical process engineering by Universitat Politècnica de Catalunya (UPC) from Spain and a MSc. in industrial engineering by Instituto Tecnológico de Orizaba by from Mexico. His research aims at knowledge management by the development of ontologies and management frameworks for decision-making support in different areas. He is actually researcher in Computer Science at the research center in mathematics (CIMAT), Zacatecas, México and collaborates with the UPC as part of the research team, participating in different Mexican and European research projects.

Success Factors for Ideation based on Experiences in Automotive Industry

Martin Neumann[1], Andreas Riel[2]

[1] KSPG AG, Neuss, Germany
[2] EMIRAcle AISBL c/o Grenoble INP, G-SCOP UMR5272, Grenoble, France

Martin.Neumann@de.kspg.com, Andreas.Riel@grenoble-inp.fr

Abstract

The increasingly saturated markets require a constant innovation offensive of new products and new product generations. Especially in the automotive industry, the rapidly changing market conditions force both Original Equipment Manufacturers (OEM) and suppliers to innovate more and more quickly and radically. To cope with this growing demand for innovations, the structuring and management of idea generation and idea selection, so-called "ideation", become increasingly important for companies nowadays. Companies typically treat ideas in the Fuzzy Front-End (FFE) of New Product Development (NPD). This paper attempts to give a significant contribution to organise the FFE more systemically and systematically so that more successful ideas are generated, selected and finally transferred to the NPD. To meet this target, this work combines the results from NPD research with insights gained from interviews with industrial expert from R&D and innovation management to define key success factors for ideation. These factors provide the pillars for the further design of a generic ideation process model and—based on this universal approach—the derivation of company-specific ideation processes which take into account the organisation's specific context and innovation culture. Using the situation at the automotive supplier KSPG AG as a practical case study, these key success factors will be explained vividly.

Keywords

Fuzzy Front-End of Innovations, Idea Generation, Idea Management, Process Development, Innovation Management, Automotive Supplier Industry

1 Introduction

Nowadays, many companies are sure that their current innovation power is not enough to guarantee long-term market success because they fail to master the initial phase of their innovation activities in an optimal manner. So they stress the need that innovation management has to act more systemically and systematically to close gaps between their actual innovation creation and their innovation potential. In other words, innovation management has to find a way to organise the so-called Fuzzy Front-End (FFE), the pre-phase of the New Product Development (NPD) [1]. This very early phase is characterised by a lack of structure and therefore also a lack of means to assess and improve its performance [2].

However, companies need to improve the performance in a way that more successful ideas are generated, selected and finally transferred to the NPD [3,4]. We want to introduce the term "ideation" for this central task of innovation management. This paper focuses on idea generation and selection for innovations of products, services or business models with commercialisation potential on the market, which is denoted as "ideation" henceforth.

The overall target of the research underlying this paper is to propose a process-based approach which supports ideation—as the origin of innovation—systematically and methodically [5]. To do so, it is essential to understand ideation better and to formulate universal key factors that lead to success in ideation. Thus, our methodology to find the targeted generic ideation process model is based on a set of general key success factors identified from literature and expert interviews to guarantee its generic character.

Our previous research presents the necessary steps toward the deployment of a new ideation process to enhance the existing innovation management at the automotive supplier KSPG AG [5,6,7,8,9,10]. However, this paper highlights the key success factors for ideation which are essential on the way towards the derivation of an ideation stage-gate [11] process model that is typical for process-driven companies of Western countries [12].

Section 2 presents shortly the general framework of our research, such as the description of the objectives, research question and selected methodology of coming up with an ideation process. Details are already published in [5]. Section 3 focuses on the presentation of the fundamental results of the literature research and the expert interviews. Based on these theoretical principles and industry experiences the derived general key success factors of ideation—on which a generic process has to be based—will be described more detailed. Section 4 describes how these success factors found their way into the generic ideation reference process model and further on into the KSPG-specific ideation process. Finally, section 5 concludes this paper, and gives an outlook on the next steps in the authors' research.

2 General Framework for the Creation of an Ideation Process

2.1 Objectives and Research Question

Numerous works of the member teams of the associations CIRP, Design Society, EMIRAcle as well as EuroSPI on the design of products and services have demonstrated the importance of the upstream phase of the development process [13,14,15,16]. The fuzzy front-end has to be seen as the very beginning of all processes in a company: any design begins with the birth and the implementation of one or more incremental or radical ideas. This process aspect—especially in regard to structuring and managing ideas—represents an area that has not yet been researched exhaustively.

Up to now, several models—mainly in the field of NPD research—exist that try to solve this dilemma by finding a structure embedded in a defined process to explain the fuzzy front-end. The most obvious

characteristic of these models is that they assume the existence of an idea without explaining how this idea was born. Especially Khurana and Rosenthal emphasise the need for further research in this field [12]. Here is a clear gap in literature which we want to bridge with our research.

The objective of our research work is to find a structured and process-oriented approach that supports companies best to exploit promising opportunities in order to achieve the challenge to increase the quantity and quality of ideas that can be transformed into commercially successful innovations. Because one of the authors' practical experience in his function as innovation manager of the German automotive supplier KSPG, the main focus of this paper is to define key success factors of ideation which have to be taken into account when creating an ideation process model suitable for the process-oriented automotive supplier industry [5].

The assumption of our research work is that companies have to find ways to organise the earliest phases of their innovation management with a strong focus on leveraging ideation within and across their entire organisational structures. Taking into account this main issue, we can formulate our central research question as follows:

How is it possible to create a structured approach, which explains ideation as the core task of the FFE, and to implement this process in a company's environment such that it successfully facilitates innovation management in practice?

This general research question leads to the following sub-question: How can companies deal with the particularities of ideation to innovate more efficiently and effectively than they do today? A first step towards the creation of an ideation process is to explore best practices examples.

2.2   Methodology

Because this work has been carried out in a practical environment, a pragmatic worldview [17] dominates the research work. This philosophical idea influences the practice of research and shapes the research design. In the centre of this research work stands the solution to a practical problem. *How must an ideation process that works look like?* This urgent need for action explains the pragmatism [18].

Led by this general orientation, we choose a qualitative design for our research [18]. We liked to explore and understand the drivers towards an ideation process applicable to the automotive supplier industry. Therefore, our research methodology linked theoretical principles with industry experiences and happen in sequential – but interlinked – steps.

First of all, we put a focus on analysing best practice examples. We derived from them key success factors representing the main causes for success.

These key success factors will be derived from Innovation Management, Fuzzy-Front-End Research, and NPD Research. A literature review will cover the theory part, and expert interviews will provide new insights or approve aspects compared to the findings based on the secondary data. Based on these key success factors a generic reference process model will be created. This description of a general ideation process can be used as a basis for the configuration of a company-specific modification of the reference model and is designed for reuse [19,20]. Finally this process model was adapted to the specific corporate conditions at KSPG, a German automotive supplier. This adaptation of the ideation process model can be achieved by identifying company-specific needs for action to ensure the practical implementation.

In the further course of this paper, we want to focus on the results from the literature review and the expert interviews and we will present our findings concerning success factors for ideation.

3 Key Success Factors for Ideation

3.1 Success Factors Based on Literature Review

Success factor research of innovation can look back on nearly five decades of history and has established itself especially in business administration, where the success-centred view dominates the understanding of innovation management. Innovations in corporations have to increase profit in a sustainable manner. Based on this purpose, the question arises of identifying the drivers for innovation success [21].

So the success factor research, which represents an independent, empirically oriented approach, enjoys great popularity in literature. Not only literature of innovation management, also the literature of NPD deals with the topic of success factors, and the lines between these two research fields are often blurred. But the underlying subject of all these studies is exclusively success, which is difficult to define concretely due to its multi-dimensionality and multi-causality. However, numerous empirical studies are engaged in the central issue to find a universally valid concept that helps companies when they risk entering the market with an innovation [21].

Focusing even more on NPD literature, Ernst [22] gives a very impressive review of the empirical literature regarding success factors of NPD. He summarise the findings of 30 years of NPD research in a compact and structured way, by categorising the identified success factors according to Cooper and Kleinschmidt's [23] five elements for a company's overall new product performance: 1. NPD process (including customer integration), 2. organisation, 3. role and commitment of senior management, 4. culture and 5. strategy.

Another very insightful study in the field of NPD literature, which Ernst does not include in his literature review, comes from Zien and Buckler [24] who investigated twelve highly innovative companies in the United States, Europe and Japan. One very relevant aspect for our own research work is the fact that Zien and Buckler identified that leaders of continually innovative companies are aware of the fuzzy front-end of innovations and centre this innovation phase in the companies' activities [24]. This finding confirms the relevance of our own research work.

Zien and Buckler investigated seven key principles, which are universal but each of the researched companies has its own company-specific implementation practice. Also these factors are not only relevant for the three crucial stages of innovation (the fuzzy front-end, the NPD, and the commercialisation), they also influence sustainably the whole company's innovative capacity over time. These seven factors are shown in Table 1 [24].

This study reveals principles of highly innovative companies, which generates an environment where innovation and high productivity influence can prosper. This confirms the aspect that an innovation friendly corporate culture is the fundamental for a working front-end process including ideation [25].

Although there is a very large number of publications concerning success factors for the NPD (including the fuzzy front-end) or the whole innovation process, there is a lack of publications that explicitly refer to success factors for idea generation and idea selection [22]. Mostly idea generation is only mentioned as a success factor of the NPD without further description. The same applies to idea selection, which is implicitly included in the success factors of continuous commercial assessment of the NPD project.

However, based on our literature review we identified several relevant aspects that influence the success of an ideation process. From our point of view, these aspects are of practical importance and are actionable in a corporate environment.

In the further course, we like to see these aspects confirmed in the subsequent expert interviews as important, and we want to identify new issues that are really crucial in practice in the view of our interview partners.

Table 1: Seven Key Principles at work in Highly Innovative Companies [24, 25]

No.	Principle	Short Description
1	Sustain faith and treasure identity as an innovative company	Leaders demonstrating in every decision and action that innovation is important to their company
2	Be truly experimental in all functions, especially in the front-end	Encouraging purposeful evolution and encouraging employees to try new things
3	Structure "really real" relationships between marketing and technical people	Developing real relationships between marketing and technical people
4	Generate customer intimacy	Generating customer intimacy by encouraging their employees to interact closely with customers
5	Engage the whole organisation	Engaging the whole organisation in understanding that innovation is the fundamental way that the company brings value to its customers
6	Never forget the individual	Continuing to value the individual and set an environment that is conductive to high motivation
7	Tell and embody powerful and purposeful stories	Telling powerful stories that reinforce the principles and practices of innovation

3.2 Success Factors Based on Expert Interviews

Based on the findings from the literature review, we conducted qualitative interviews with experts in the field of ideation and innovation management to validate and complete our previous results from theory. The aim of these interviews was to survey external experts first, in order to explore current best practices. With the interviews of external experts we wanted to get a stimulus from outside the case study's company to assure learning from others. Internal expert interviews were part of the case study.

Therefore, the selection of the experts for the qualitative survey was done according to certain criteria, which were considered as important for both the research question and for the subsequent analysis of the data. The most important criterion was the professional expertise of the persons concerning ideation and innovation management.

Another broader selection criterion was to focus on best practice examples. Thus, we identified companies from the automotive industry (OEMs and suppliers) but also from other business sectors, like telecommunication equipment manufacturing, machinery and process technology, chemical manufacturing, and computer services. All these companies are internationally renowned for their innovation powers. Finally, we attained 13 expert interviews.

Concerning the sampling of the industry sectors and the respective companies which have come into consideration, we clustered them into three target groups. Finally, the design of the external expert interviews can be outlined as shown in Table 2.

Table 2: Survey Design of Expert Interviews

Target Group	Scope	Reason for sampling	Data collection procedures	Companies
1	German automotive OEMs	• German automotive industry is regarded as innovation leader in the industry • Access available to interview participants or secondary data	• Interviews • Analyses of various publications from relevant congresses	• OEM 1 • OEM 2 • OEM 3 • OEM 4
2	Successful German automotive suppliers (Tier 1 supplier)	• The case study's company belongs to this segment • Comparison is interesting and necessary	• Interviews • Analyses of various publications from relevant congresses	• Supplier 1 • Supplier 2 • Supplier 3
3	Worldwide recognised innovation leaders, non-sector-specific	• Inspiration from interdisciplinary perspectives on other industries	• Interviews • Analyses of various publications from relevant congresses	• Innovator 1 • Innovator 2 • Innovator 3 • Innovator 4 • Innovator 5 • Innovator 6

The identification of success factors of ideation processes represents our main focus within the expert interviews. In order to find factors which we can map to our reference process model of ideation, we first asked the interviewees the open-ended question: "According to your experience and/or considerations, what are key success factors of an ideation process?" This should encourage the interview partners to talk freely and to reveal more information voluntarily. In the second step, we confronted them with our list of possible aspects derived from the literature review.

The first question and answer session with our interviewees showed very clearly that top management commitment has a very high rank amongst the critical success factors of ideation processes. Without exception, all the interviewed experts agreed to the fact that top management commitment is indispensable to enforce innovation activities. Two experts declared that they had seen several ideas fail due to lack of a visible commitment of top management. Without the top management commitment, the resources for ideation activities in terms of time and budget will not be granted.

Another very important success factor identified in particular at German automotive OEMs is the fact that ideation needs to focus. A well-defined strategic orientation has to be visible for everyone who is involved in ideation. A cohesive strategy has to describe the future vision for the company's products and services. That strategy needs to state clearly the long-term perspective that all participants in the ideation process are in line with this future focus. This strategy-oriented approach needs open lines of communication that are regularly and consistently managed. Additionally, it is important that market changes and the evolution of the company's new products make it necessary to revise the strategy regularly. This flexible modification of strategy must not be underestimated. Otherwise there is the threat of losing differentiation attributes with respect to the competition. Creating and sustaining Unique Selling Propositions is a crucial objective of the interviewed OEMs, in particular because of their global market presence.

Ideation needs a good prerequisite. This includes, from our interviewees' points of view, diligence work in form of analysis of the market situation, a competitive environment, customer needs, technology trends, current and future legislation, etc. OEM 1 sees in this preparation the prerequisite to target and optimise idea generation.

All the involved interview partners agreed that a systematic and transparent pursuit of ideas is needed. "Ideas may not disappear without a trace", stated OEM 1. This leads to the assumption that especially the generation of ideas must involve a broad mass of employees and integrate external stakeholders as well. OEM 3 pointed out: "Creativity evolves from Networks."

However, networking and stakeholder integration needs clear structures including roles, responsibilities, mandates, reporting lines, etc. Leveraging interdisciplinary teams with strong leadership may influence the idea generation positively. For OEM 4 this aspect seems to be a promising factor. "Someone has to have the lead to pull ideas through", added OEM 2. A clear role allocation also leads to a successful ideation. These roles are: leaders of ideation activities, promoter of ideas and mentors of idea promoters. "But it is important", Innovator 4 pointed out, "that a common mental model exists between these roles and that they adhere to the same clear process model."

Our interviews also revealed the fact that to facilitate creativity, it is a vital success factor to balance between specific and well-defined problem solving activities, in form of guided ideation, and giving the employees their freedom of generating ideas without corporate specifications. This last point will enable ideas out of the box but requires special budget.

To solve the problem of budget allocation during ideation, OEM 3 and Innovator 6 see the need for a competition spirit among ideation teams/idea contributors and their generated ideas. There has to be competition for the budget, where only the best ideas receive the needed financial resources. This demands for entrepreneurship, because only if the ideas contributors think like entrepreneurs enforcing internal idea marketing, their ideas will obtain the recognition they deserve.

The experts agreed on the importance of idea communication as a bidirectional exchange:

1. From the idea contributor: integrate ideas into a story to gain attention and highlight customer value and marketing aspects, and

2. to the idea contributor: assure quick response times to idea submissions and guarantee a systematic and transparent follow-up of ideas.

Regarding idea selection, the interviewed experts emphasised again the need for practical indicators to monitor and select ideas. For OEM 1 and Innovator 6 a comprehensible decision-making is essential.

Asked about rewarding schemes as a success factor, the experts did not prioritise this aspect very highly. In their understanding, it is crucial to motivate employees, but not only by financial rewards. They stressed that rewards are also about recognition and being able to do satisfying work that challenges the mind and allows setting free their creativity.

Globally speaking, these expert interviews validate and enrich our previous key findings from literature review. Our discourse with experts in the fields of ideation and innovation management helped us to identify best practices and to derive success factors from them.

When we look at the multitude of identified success factors that we found in our literature review and expert interviews, we can observe that a clustering of the factors based on their frequencies is possible and helpful in the practice and business context. With regard to our ideation reference process model we grouped the success factors into prerequisite, generation, and selection aspects of ideation. This summary of the success factors represents the fundamental objectives of ideation and is a very good starting point for defining the stages of our ideation reference process model.

3.3 Best Practices for a Structured Ideation Process

Our extensive literature review in the principle subject areas of innovation management, fuzzy front-end research and NPD research, as well as our examination of best practice examples from the automotive industry and other worldwide recognised innovation leaders result in the identification of key success factors for an effective ideation process. These following six main success factors seem to us as actionable and promising for the creation of an ideation process model.

Success Factor No. 1: Ideation starts at the top management.

Even the best ideas need support and commitment from the management board. All the interviewed external experts agreed on this. Our interviews with internal experts at KSPG also revealed that this commitment is essential to push the development of new products. One of our experts stated: "Based on my experiences, it might be important that the top management provides a statement regarding the most important innovative topics, which they really want to be realised." Top management commitment can be seen as the prerequisite for establishing the basic conditions for innovative activities to be carried out and for employees to understand their responsibility and to be encouraged to think beyond the status quo.

Regular innovation board meetings helps to implement the objective of top management commitment, as top management is supposed to commit to innovation strategy and innovation priorities there. Ideation calls and timelines are directly derived from these outcomes, and have to be communicated in the entire organisation. So, the explicit call for ideation activities and the clear commitment to them by the top management is absolutely essential and must be clearly visible for all employees.

Success Factor No. 2: Ideation needs a clearly defined focus.

This was one of the major findings from our expert interviews. Idea contributors have to have a guidance in terms of which topics are considered relevant by top management. In this way, they can focus their efforts and meet the expectations of decision makers. Ideation topics shall be aligned with strategic company objectives.

Griffin and Page support this conclusion that the presence of a clear strategy has a positive influence on the success of new products [26]. Therefore the definition of the company's innovation focus that is aligned with the overall company's core mission and values is inevitable. This strategic focus helps reduce cost and time effort during the creation and realisation of ideas, and leverages the effectiveness of the ideation process [27,28].

A systematic analysis of the company's total situation and environment for the identification of areas of action increases additionally the effectiveness during the generation and selection of ideas.

Success Factor No. 3: Ideation happens in networks.

In various studies about successful innovation teams, the integration of stakeholders with diverse expertise and profession from different cultures during the front-end stages of the product development process represents a key factor for sustainable development [13,29]. This diversity is crucial to innovation and the development of new ideas [29].

The essential consideration here is that the integration of these stakeholder groups in the innovation management process is a key step for making innovation sustainable, as it allows taking into account the requirements and constraints imposed by the different actors of the product/system life cycle [30].

Involving external stakeholders in company-wide innovation management is also the core characteristic of Open Innovation strategies, which have originally been coined by Chesbrough in 2003 [31]. The move towards stakeholder-integration based ideation and innovation has become a must [7,8,9,10]. To be innovative in future, it is important that product and/or service ideas are not only the result of one core group of employees.

Success Factor No. 4: Ideation demands creativity.

The promotion of creativity and its integration in corporate processes enhances the quality and quantity of ideas. To encourage creativity, freedom and intense people management is necessary. In contrast, the efficiency of the invested funds can be only achieved by discipline and high emphasis on process management. Overall, the requirements of the market and the customer needs dominate the creative technical ideas of the developer [32].

The effective management of the early phase of innovation has to generate an efficient process that gives sufficient freedom for creative development of the employees. Also, this process needs to be flexible enough to react to changing market demands, which occurs through new customer needs or new technological possibilities [32].

Therefore, a company has to manage the ideation environment in a balanced mix of overall flexibility and guided focus [33]. The resulting area of conflict between creativity and resource efficiency provides the breeding ground for developing new product ideas [32].

Success Factor No. 5: Ideation needs entrepreneurship.

Ideas have to be promoted and communicated in the company to increase general awareness. By installing a growing spirit of competition between ideas to win the needed resources in form of time and money, idea contributors are forced to promote their ideas. Our research shows that competition between ideas and their marketing in the company raises the level of maturity and the quality of ideas.

Success Factor No. 6: Ideation requires organisational orientation.

Finally, target–oriented decision-making processes with transparent evaluation criteria enable the communication and conversion of ideas. Here it is very important that the right decision makers are identified and actively involved.

All these success factors contribute to the creation and reinforcement of an organisational culture of Open Innovation, facilitating the integration of numerous stakeholders in the ideation process. They are all logically linked with one another, which confirms that they make sense.

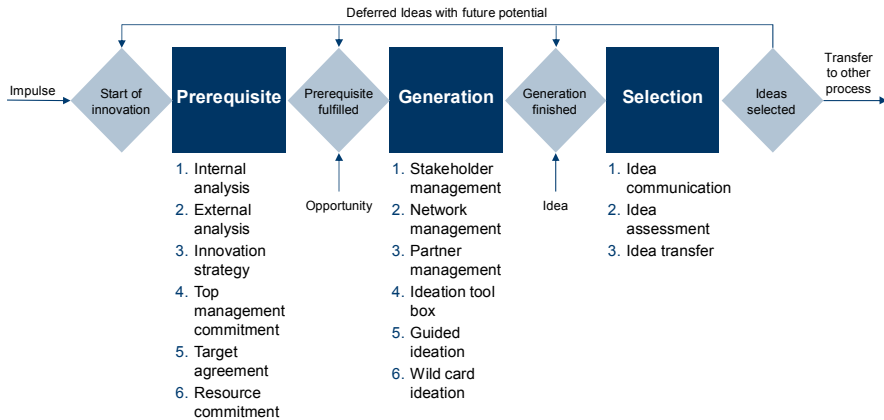# 4 The Ideation Process at the German automotive supplier KSPG

4.1 Ideation Reference Process Model

Our main objective for the creation of the ideation reference process model was to achieve a clear and simple mapping of the identified six key factors to stages and gates of the ideation process, in a way that each of these elements can also be implemented in any specific organisation. The advantage of leaving the model broad is that this allows any company to tune their existing ideation processes to the most effective elements instead of blindly copying the whole ideation reference process model.

For this purpose, we have created the process based on the three fundamental aspects of ideation, which we derived from our research findings:

1. *Prerequisite* covers all the activities expected from top management level to set the right frame for the whole ideation process, like internal and external analysis, definition of business unit innovation strategy, commitment of top level management visible for all employees, agreement on the ideation targets and priorities and commitment to available resources.

2. *Generation* is the active execution phase. Here all the ideation activities are fully devoted to facilitate the generation of ideas to the maximum. These activities include the creation of ideas in networks of internal and external actors, usage of creativity techniques out of the company-specific ideation tool box, guided ideation and the speciality of "Wild Card Ideation".

3. *Selection* consists the idea assessment, idea communication and the transfer of ideas to the subsequent NPD. This stage is dedicated to find and campaign the best ideas for the upcoming development process.

These three main elements correspond to the stages of our ideation reference model. Each phase has its specific ideation activities, which also occur from our research of best practices and represent in their core also the success factors that have to be implemented in order to fulfil the related gate requirements and to pass to the next stage. Taking all this into account during the creation of the process, we have finally designed the ideation reference process model as shown in Figure 1.

Figure 1. Ideation Reference Process Model

This ideation stage-gate reference process model capitalises on the strong and systematic involvement of internal and external stakeholders in ideation, and therefore follows intrinsically the modern paradigm of Open Innovation. Using this ideation stage-gate reference process model allows to derive a company-specific ideation process.

4.2 Derivation of the KSPG-specific Ideation Process

In order to validate our proposed ideation reference process model, a first case study has been carried out at the German automotive Tier 1 supplier KSPG AG. The global objective of this study is to derive a KSPG-specific ideation process from the ideation reference process model seen in Figure 1 and to initiate the deployment of this new process in the corporate environment.

The derivation of a KSPG-specific ideation process started with the identification of priority areas of action based on the analysis of the achievement levels of each key success factor of ideation (listed in section 3.3) in the currently existing innovation process in the company. This as-is analysis revealed that at KSPG AG initially ideation was supposed to be driven by an idea database accessible by all employees in the company's intranet, without any transparent overall organisational direction towards broad contribution to innovation [7]. As a result, only a few product engineers actually submitted ideas that were mainly related to improvements of existing products. The selection process was quite opaque, involving only a few high-ranking decision makers.

In a further step the organisation framework at KSPG AG was analysed intensely to determine elements that are necessary to achieve each stage and gate of the ideation reference process model. One important goal is to motivate the organisation at KSPG AG for the integration of internal and external stakeholders, e.g. customers, to leverage ideation. One essential step in this direction of a better collection of customer ideas at KSPG was the creation of permanent customer-related teams with team members from all KSPG sales divisions [7]. Since its introduction two years ago, this procedure of collecting and storing systematically customer insights becomes a major improvement for capitalise on customer ideas at KSPG.

Based on the key success factors of ideation, all the identified KSPG-specific action fields and the determining elements of KSPG's corporate organisation, the creation of a company-specific instantiation of the reference process model had been done. Particular care had to be taken to avoid that the new ideation process causes significant additional work effort for the involved actors in order to minimise resistance against change, and to render its introduction feasible in terms of the availability of decision makers. To this aim, synergetic effects from existing established processes in the company

have been maximised in the specification of each specific action and tool related to the stages and gates proposed by the reference process.

The result is the new KSPG AG C^{3}IP (Call & Commit, Connect & Create, Choose & Cancel Ideation Process) that completely complies with the company's process rules and notations, and precedes their ADP (Advanced Development Process). The C^{3}IP is a real instantiation of the proposed ideation reference process model containing KSPG-specific implementations of actions and tools. One effective measure is the creation of regular internal KSPG Ideation Meetings [5]. In the context of a large strategy project of the KSPG division Mechatronics, which started in late summer 2012 and ended in February 2013, the formation of a dedicated cross-divisional and cross-functional ideation team helped to generated more than 350 ideas in a very short period of time. In this team were members of all the Business Units, especially the responsible Product Development Managers, and Advanced Engineering, but also representatives from Sales and Controlling. To enrich the idea generation phase also externals were included. During this time two students who were actually working on their diploma thesis in the department of Advanced Engineering were assigned to look exclusively for new ideas during two days. This 'fresh eyes'-approach supplemented the ideation team very well with new perspectives. In addition, so-called Innovation Review Meetings with attention of the Business Unit Leaders enabled the further assessment and selection of these ideas that leads to the clear definition of a Top 10 list of future product development activities. The findings from the here presented research work helped to handle this approach very efficiently and effectively.

Because of its added value, the KSPG AG C^{3}IP has been strongly supported by top management stakeholders from the departments Research and Technology and Advanced Engineering, and shall be introduced at a large scale in 2013.

5 Conclusion and Outlook

One major milestone of the new idea generation process at KSPG is to present new ideas worthy to pass the money gate, the so called R&D strategic committee, at the end of each year. During the year 2012, the development of the process takes shape which really could fulfil this task. Up to now, the new idea generation process at KSPG promise to offer the following advantages:

- more focus
- more challenge, and
- more involvement.

In conclusion, the described conceptual thoughts concerning idea generation outlined in this paper illustrate the immense impact of the idea generation process for the whole innovation process. Due to the fact that the early phase of the innovation process represents a very recent field of research, we believe that our approach closes some gaps and represents a very good compromise to make dynamic ideation activities systematic while at the same time keeping up the high level of creativity that is necessary to let new ideas flourish.

The next essential step in our research is to validate and improve our generic ideation reference process model according to the results of the implementation of the company-specific process at KSPG. To complete the full picture, the efficiency our approach has to be evaluated. Therefore it is very important to derive from our ideation reference process model other company-related ideation processes. Because with the increasing amount of company-specific processes, more case studies are available providing usable experiences from practise and valuable lessons learned.

Another major aspect of our research work is the hypothesis that the integration of different experts in the process of ideation—more precisely the encouragement of diversity in the creation and assessment of ideas—must contribute significantly to increase the number, the quality and the relevance of received ideas. These subjects—stakeholder integration and open innovation—propose a wide range of further research and studies.

6 Literature

[1]   Smith P.G., Reinertsen, D.G. (1991). Developing Products in Half the Time. Van Nostrand Reinhold, New York 1991

[2]   Zhang Q., Doll W.J. (2001). The fuzzy front end and success of new product development: A casual model. In: European Journal of Innovation Management, Vol. 4, Issue 2, pp. 95-112

[3]   Khurana A., Rosenthal S.R. (1997). Integrating the Fuzzy Front End of New Product Development. In: Sloan Management Review, Vol. 38, No. 2, pp. 103-120

[4]   Herstatt C., Verworn B. (Eds) (2007). Management der frühen Innovationsphasen: Grundlagen – Methoden – Neue Ansätze. 2nd edition, Gabler Verlag, Wiesbaden 2007

[5]   Neumann M., Riel A., Ili S., Brissaud D. (2012). Towards an Ideation Process Applied to the Automotive Supplier Industry. In: Winkler D., O'Connor R., Messnarz R. (Eds.): Systems, Software and Services Process Improvement – 19th European Conference, EuroSPI 2012, Vienna, Austria, June 25-27, 2012, CCIS 301, Proceedings, pp. 229-240, Springer-Verlag, Berlin Heidelberg 2012

[6]   Neumann M., Riel A., Brissaud D. (2010). Innovation Management System for the Automotive Supplier Industry to Drive Idea Generation and Product Innovation. In: Proceedings of the 17th EuroSPI Conference, Vol. 17, pp. 7.13-7.24

[7]   Neumann M., Riel A., Brissaud D. (2011). IT-supported innovation management in the automotive supplier industry to drive idea generation and leverage innovation. In: Journal of Software Maintenance and Evolution: Research and Practice. doi: 10.1002/smr.578

[8]   Neumann M., Riel A., Brissaud D. (2011). Sustainable innovation management in the automotive indrusty. In: Int. J. Technology Intelligence and Planning, Vol. 7, No. 4, pp. 327-343

[9]   Neumann M., Riel A, Brissaud D. (2011). Improvement of Innovation Management through the Enlargement of Idea Sources. In: O'Connor, R., Pries-Heje, J., Messnarz, R. (Eds.): Systems, Software and Services Process Improvement – 18th European Conference, EuroSPI 2011, Roskilde, Denmark, June 27-29, 2011, CCIS 172, Proceedings, pp. 121-132, Springer-Verlag, Berlin Heidelberg 2011

[10] Neumann M., Riel A., Brissaud D. (2011). Improvement of Idea Generation by Capitalisation on Internal and External Stakeholders. In: Industrial Proceedings of the 18th EuroSPI Conference, Vol. 18, pp. 3.25-3.33

[11] Cooper R.G. (2011). Winning at New Products: Creating Value Through Innovation. 4th edition, Basic Books, New York, NY 2011

[12] Khurana A., Rosenthal S.R. (1998). Towards Holistic "Front Ends" in New Product Development. In: Journal of Product Innovation Management, Vol. 15, Iss. 2, pp. 57-74

[13] Riel A., Tichkiewitch S., Messnarz R. (2009). Integrated Engineering Skills for improving the System Competence Level. In: Software Process Improvement and Practice (SPIP). Wiley, Vol. 14, Nov/Dec 2009, pp. 325-335

[14] Riel A.; Tichkiewitch S.; Messnarz R. (2010). Qualification and Certification for the Competitive Edge in Integrated Design. In: CIRP Journal of Manufacturing Science and Technology. Special Issue on Competitive Design, Vol. 2, Iss. 4, pp. 279-289

[15] Tichkiewitch S., Veron M. (1998). Integration of Manufacturing Process in Design. In: Annals of the CIRP, Vol. 47, Iss. 1, pp. 99-102

[16] Tichkiewitch S., Brissaud D. (2000). Co-ordination between product and process definitions in a concurrent engineering environment. In: Annals of the CIRP, Vol. 49, Iss. 1, pp. 75-79

[17] Cherryholmes C.H. (1992). Notes on pragmatism and scientific realism. In: Educational Researcher, Vol. 21, No. 6, pp. 13-17

[18] Creswell J.W. (2009). Research design: Qualitative, quantitative, and mixed methods approaches, SAGE Publications Inc., 3rd edition, Thousand Oaks, California

[19] Meboldt M. (2008). Mentale und formale Modelbildung in der Produktentstehung – als Beitrag zum integrierten Produktentstehungs-Modell (iPeM). IPEK Forschungsbericht, Band 29, Institut für Produktentwicklung Universität Karlsruhe (TH), Karlsruhe

[20] Lassmann W. (Ed.) (2006). Wirtschaftsinformatik: Nachschlagwerk für Studium und Praxis, Gabler Verlag, Wiesbaden

[21] Hauschildt J., Salomo S. (2011). Innovationsmanagement, 5th edition, Vahlen, München

[22] Ernst H. (2002). Success factors of new product development: a review of the empirical literature. In: International Journal of Management Reviews, Vol. 4, Iss. 1, March 2002, pp. 1-40

[23] Cooper R.G., Kleinschmidt E.J. (1995). Benchmarking the firm's critical success factors in new product development. In: Journal of Product Innovation Management, Vol. 12, Iss. 5, pp. 374-391

[24] Zien K.A., Buckler S.A. (1997). From Experience Dreams to Market: Crafting a Culture of Innovation. In: Journal of Product Innovation Management, Vol. 14, Iss. 4, pp. 274-287

[25] Koen P.A., Ajamian G.M., Boyce S., Clamen A., Fisher E., Fountoulakis S., Johnson A., Puri P., Seibert R.A., Davidson J., D'Amoe R., Elkins C., Herald K., Incorvia M., Johnson A., Karol R., Seibert R., Slavejkov A, Wagner K. (2002). Fuzzy Front End: Effective Methods, Tools and Techniques. In: Belliveau P., Griffen A., Somermeyer, S. (Eds.): The PDMA ToolBook 1 for New Product Development, John Wiley & Sons. Inc., Hoboken, New Jersey, pp. 5-35

[26] Griffin A., Page A.L. (1996). PDMA Success Measurement Project: Recommended Measures for Product Development Success and Failure. In: Journal of Product Innovation Management, Vol. 13, Iss. 6, pp. 478-496

[27] Meyer C. (1997). Relentless Growth: How Silicon Valley Innovation Strategies Can Work in Your Business, The Free Press, New York

[28] Clausen T., Geschka H., Krug J. (2012). Innovationsstrategien. In: Ili S. (Ed.): Innovation Excellence – Wie Unternehmen ihre Innovationsfähigkeit systematisch steigern, Symposion Publishing GmbH, Düsseldorf, pp. 95-128

[29] Forbes Insights (2011). Global Diversity and Inclusion: Fostering Innovation Through a Diverse Workforce. Study, July 2011, retrieved from http://images.forbes.com/forbesinsights/StudyPDFs/Innovation_Through_Diversity.pdf last accessed on 16/04/2013

[30] Saunière J.-C., Temam O. (2010). Innovation and Performance - Where's your R&D at? PricewaterhouseCoopers Study

[31] Chesbrough H.W. (2003). Open Innovation: The New Imperative for Creating and Profiting from Technology. Harvard Business School Press: Boston

[32] Sandmeier, P., Jamali J. (2007). Eine praktische Strukturierungs-Guideline für das Management der frühen Innovationsphase. In: Herstatt C., Verworn B. (Eds.): Management der frühen Innovationsphasen: Grundlagen – Methoden – Neue Ansätze, 2nd edition, Gabler, Wiesbaden, pp. 339-355

[33] Nambisan S. (2002). Designing Virtual Customer Environments for New Product Development: Toward a Theory. In: The Academy of Management Review, Vol. 27, No. 3, pp. 392-413

7 Author CVs

Martin Neumann

Dr. Martin Neumann is Manager Innovation Services within the central research and technology department of KSPG AG and has been working since 2004 for KSPG AG and the Pierburg GmbH in the fields of market research and innovation management. He has professional work experience in marketing, consulting and market research, and he has been certified as European Innovation Manager.

Andreas Riel

Dr. Andreas Riel is Program Manager Innovation and Senior Management Consultant for the EMIRAcle association, based in Grenoble, France. He has 10 years of experience in automotive powertrain R&D and innovation management, with a focus on virtual and integrated methods for efficiency and quality assurance, as well as innovation transfer from academic research partners to industrial application. In academia, he has two years of experience on coordinating the VRL-KCIP Network of Excellence in innovative product and systems development, as well as more than three years of experience as the Program Manager of Innovation projects of the EMIRAcle research association. He has been certified by the ECQA as Innovation Manager and Trainer, Integrated Design Engineer and Trainer, and EU Researcher-Entrepreneur and Trainer.

Social Media Networking - A tool for Business Improvement and Certification of Professionals

Ricardo Colomo-Palacios[1], Richard Messnarz[2], Antonio Amescua Seco[1], Kerstin Siakas[3], Daniel Palosi[4], Darragh Coakley[5], Adrienne Clarke[2]

[1]U3CM, Spain, [rcolomo, amescua] @inf.uc3m.es
[2]ISCN GesmbH/LTD, Austria & Ireland, aclarke@iscn.com
[3] ATEI Thessaloniki, Greece, siaka@it.teithe.gr
[4] Dennis Gabor College, Hungary, palosi@gdf.hu
[5]Cork Institute of Technology, Ireland, darragh.coakley@cit.ie

Abstract

Social Media Networking represents an approach which can be used for business improvement and knowledge networking. In this paper we outline how the traditional knowledge management approach has changed to a networked approach of sharing knowledge. We also explain how social media are used as a business tool. SIMS (ECQA Certified Social Media Networker) is a new qualification which is available in 2013 and is meant to train the experts and industry people to join the community for using social media as business improvement and tool and as a means for knowledge networking. The SIMS project is supported by the European Commission (Programme Lifelong Learning - action Leonardo da Vinci-Transfer of Innovation); 2011-1-ES1_LEO05-35930.

Keywords

Knowledge Networks, Social Networks, Improvement, Business tool

1 Introduction

The emergence of knowledge societies has multiplied the extent to which both productivity innovations and social transformations rely on knowledge capital. Drucker argues that in the emerging electronic economy, knowledge is the primary resource for individuals and for the economy overall (Drucker, 1992). This author further stresses that "knowledge worker productivity is the biggest of the 21st century management challenges" (Drucker, 1999). Today, hardly any industry remains unaffected by the evolution of network-like relationships within and between firms. The openness and richness of networks are believed to foster a fertile environment for the creation of entirely new knowledge, while also accelerating the innovation rate. Not in vain, knowledge networking is a term used to signify a number of people, resources and relationships among them, who are assembled in order to accumulate and use knowledge primarily by means of knowledge creation and transfer processes, for the purpose of creating value (Seufert, Krogh, & Bach, 1999).

In this scenario, a recent study published in IEEE Transactions on Learning Technologies states that it is crucial to address what today's knowledge workers need (Chatti, Schroeder, & Jarke, 2012). In this work, authors define a good knowledge networker as one who has the ability to:

- Create, harness, nurture, sustain, and widen his/her external network to embrace new knowledge nodes.

- Identify connections, recognize patterns, and make sense between different knowledge nodes.

- Locate the knowledge node that can help achieving better results, in a specific learning context.

- Aggregate and remix.

- Cross boundaries, connect, and cooperate.

- Navigate and learn across multiple knowledge networks.

- Help other knowledge networkers build and extend their networks.

Knowledge is a key factor in the growth and development of the economy as a whole. Knowledge management has been a hot topic since late eighties. The old paradigm of knowledge management assumes that we store all knowledge and implement more and more sophisticated algorithms to search that knowledge and give advice. The new paradigm which we follow in expert and knowledge networks does not believe in that because the tacit knowledge is much larger than the stored one so that any decisions made on just the stored knowledge are questionable in an expert network. In this scenario, knowledge clusters are a new way to approach the problem.

On the other hand, the Social Web is represented by a class of web sites and applications in which user participation is the primary driver of value (Gruber, 2008). Web 2.0 is a term coined in 1999 by Darcy DiNucci to describe web sites that use technology beyond the static pages of earlier web sites. Online social networking tools are reinvigorating Knowledge Management by making it easier for employees to participate in knowledge creation and organizations have been able to help employees connect across disparate regions (O'Dell & Hubert, 2011). Organisations are implementing Enterprise Micro Blogging (EMB)-based solutions for creating a new channel for organisational and team communication. This is especially beneficial when information may be relevant for a whole group of users that is unknown in advance (Riemer, Richter, & Bohringer, 2010). Social Media Networks are useful as communication channel among organizations (Perrigot, Basset, & Cliquet, 2011). Furthermore, literature underlines the importance of social media for collective innovation communities and online innovation communities (Ahlqvist, Bäck, Heinonen, & Halonen, 2010; Di Gangi, Wasko, & Hooker, 2010). Finally, and taking into account that gender also plays a major role when it comes to using Web 2.0 technologies (Milovanović et al., 2012, Warren et al., 2012), social media tools must include ways to deal with these differences.

Given the importance of Social Media Networking and its new way of fostering communication, branding and marketing both from a personal and organizational perspective, this new job role has been

developed by ECQA in order to warrantee the availability of trained and certified professionals around Europe. But beyond the importance of this job role to ECQA community and EU as a whole, there's a potential use of social media networking skills as a support for the community and to the whole society that this paper wants to tackle. Moreover, and given the volatile nature of IT skills in general and social media skills in particular, there are a set of challenges that this paper also review in order to support the sustainability of the certificate in the future.

2   Social media as a business tool

Reality is costly, and in the age of the rising importance of internalizing negative externalities and the need for sustainable development, it will be more and more costly for individuals and organisations as well. According to Krzysztof Zanussi: "…there is not enough reality for everyone!" (EDEN conference Gdansk, 2009) a solution has to be found to be able to give the access to worldly goods for the next generations too.

Can virtuality, partly social media be the right instrument for that? European and World statistics confirm that. In 2012, 72% of EU27 households had access to the internet via a broadband connection, compared with 30% in 2006 (Eurostat, 2012), which is a quite conspicuous rise. "Portugal had the largest share of internet users who posted messages to social media (75% of internet users), and the Czech Republic (35%), France (40%) and Germany (42%) the lowest. The Netherlands (17%) and Hungary (16%) recorded a proportion of internet users who created websites and blogs in 2012 that was almost double the EU27 average." (Eurostat, 2012)

Can businesses recognize the opportunity in social media? Reaching the consumers through conventional marketing methods is fair costly; after Wanamaker and Leverhulme half the money spent on advertising is wasted; the trouble is we do not know which half. Social technology has developed in the recent years to a business tool, which lets organisations having an insight on a new way of consumer deliverance, but faster and at lower cost. "Moreover, in addition to engaging consumers directly through social media, companies are watching what consumers do and say to one another on social platforms, which provides unfiltered feedback and behavioral data (e.g., do people who "like" this movie also "like" that brand or vodka?)". (MGI, 2012)

The research of McKinsey Global Institute (MGI) reveals the organisational functions (see Fig. 1.), where social technologies can contribute to the value addition with some exact examples of industries like automotive industry.
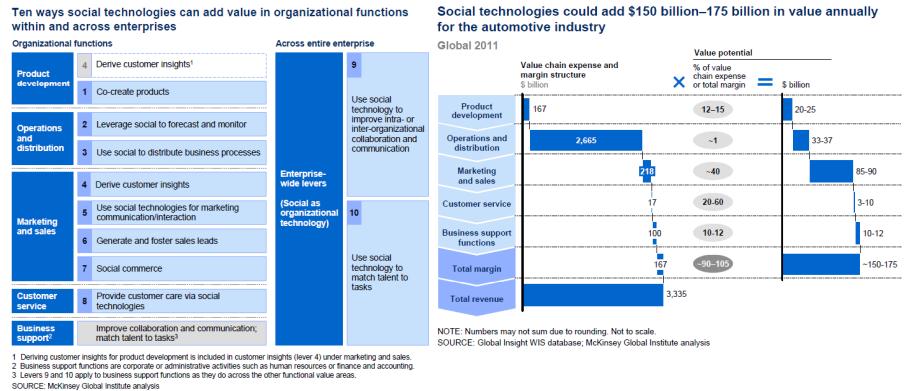

Figure 1. Value addition of social technologies (Source: MGI, 2012)

The Conference Board and the Rock Center for Corporate Governance at Stanford University con-

firmed also the importance of social media with the research done in the spring of 2012. (Larcker, Larcker, & Tayan, 2012)

The international survey of The Data Warehousing Institute (TDWI) carried out in February 2012 among business executives, business or data analysts, IT professionals, consultants "pointed out Social media data analysis can expand customer analytics by providing an unfettered, outside-looking-in view of an organization's brands, products, services, and competitors" (TDWI, 2012).

According to the answers, social media is not the main data source for monitoring customer analytics (see Fig. 2. where social media accounts for 31%) at the moment, but due to the expected further rise of the number of social media networkers, organisations have to concentrate more on social media platforms.
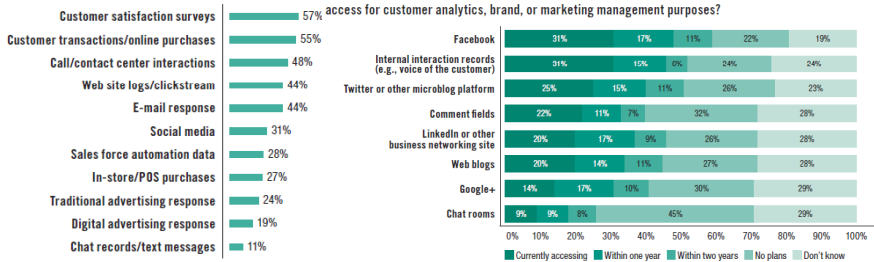
Figure 2. Role of social media at monitoring customer analytics (Source: Stodder, 2012)

To be able to add value to the organisation with the help of social media technologies in a structured way, a need for skilled professionals arises from the side of the industry. The EU funded project Social Media Networker Skills aims to satisfy this kind of demand within the framework of the EU Lifelong Learning Programme.

3 ECQA Social Media Networker

In order to support individuals and organisations in building competences in using social media for networking a new innovation transfer project entitled 'ECQA Certified Social Media Networker Skills (SIMS)' is currently developed within the frame of the European-wide accepted scheme of non-profit European Certification and Qualification Association (www.ecqa.org). The aim of this project is the transfer of the Social Media Networker Skills to the industry. The two year project started in October 2011 with funding from the EU Lifelong Learning Programme.

The project aims at developing a new skill set and a job role qualification study program, where competencies in social media networking are customised for the European industry into an online study program complemented with an on-line examination and certification training and certification schema for social media networker. A pilot training will take place in the participating organisations/member states (Austria, Greece, Hungary, Ireland and Spain) and the study programme will be refined and improved based on systematic feedback.

In the end of the funding period a Social Media Networker Job-Role Committee (JRC) will be created by a number of experts in the area for ensuring sustainability and exploitation. The JRC initiates, develops and frequently revises the training and certification schema. Since the JRC group provides the know-how and material needed for certification and training, it also has the right to decide upon policy related to the job-role in alignment to the overall strategy and processes of ECQA. By other words the Social Media Networker job-role is taken to the market through a self-funding system and its future development is decided by the JRC. The training material is frequently revised and updated for ensuring high quality outcomes, wide acceptance and diffusion in the industry.

As the number of people who are taking the social media networker certification grows the database of exercises and exam questions will grow. In order to make the system sustainable comprehensive and systematic updating, maintenance and quality assurance of the database are incorporated enabling wider acceptance and user satisfaction. The ultimate measure of success will be the widespread valorization and the sustainability of the project outcomes. The benefit will satisfy the requirements of all stakeholders starting from the individuals (trainers and trainees), the participating organisations and the funding bodies.

Emergent challenges for organisations regarding skills of their workforce in social media networking practices and the advantages organisations can gain when actively using contemporary social media platforms are summarised below:

- **Creation of an innovative culture** involves a learning process that builds on evaluation, reflection and development of the organisation toward response maturity for emerging challenges. The relationship between social attribution and technological possibilities are cornerstones for the learning process. Social media are excellent tools for this (Siakas et al., 2012)

- **Tapping of collective explicit and tacit knowledge and intelligence of users** (customers and consumers) by social media networks and thus reaching beyond the conventional boundaries of the organisation (Bullinger, 1999)

- **Open innovation** involve the process of ideation through ideas flowing out of the organisation for evaluation and flowing into the organisation as new offerings and new business models. Social media are (Chesbrough, 2003). Social media are important for 'reflection in practice' and launching of prototypes for user tests before a product or service is launched on the market (Siakas et at. 2012).

- **User participation for value creation** facilitated by social media is increasingly appreciated by organisations (Gruber, 2008).

- **Leverage of disparate assets** of people from different cultures, different disciplines and different organisations are facilitated by social media networking (Siakas et at. 2012).

- **Word-of mouth marketing** is obtained effectively through social media marketing and the Next Generation of Business Engagement (Evans and McKee, 2010)

Organisations have increasingly recognised the value of social media for creating added value and competitive advantage. The basis of the operations today in the knowledge based economy is the knowledge of the individuals facilitated through knowledge sharing and collaboration as well as understanding of user needs through open and collective innovation and user participation. Organisations have increasingly understood the potential for using social media networking in their everyday operations and the importance of a skilled workforce in social media practices.

4 Social Media and Knowledge Networking: A European Vision of Improvement Driven by Expert Clusters

Especially in SPI, we could observe many methods and concepts to come and claim that they represent the silver bullet to solve all problems. When we observe the last 20 years of SPI, it is actually not true that we ever found one single solution that solved all problems. It is rather so that all new concepts and methods ran through a hype cycle and were networked and integrated later into a combined use of methods and concepts based on industrial feedback and experience (Messnarz et. al. 2011).

Also the old paradigm of knowledge management assumes that we store all knowledge and implement more and more sophisticated algorithms to search that knowledge and give advice. The new paradigm which is followed in expert and knowledge networks does not believe in that because the tacit knowledge is much larger than the stored one so that any decisions made on just the stored knowledge are questionable in an expert network (Messnarz et. al. 2011).

Thus the strategy is to create social web based expert clusters around key industry topics. Key articles are stored around topics and experts can be connected from there. Instead of a query to a database

we use a query to experts clustering around key topics.

To establish this approach in Europe the leading industry and the researchers must learn how to act in a social space, how to collaborate and how to also protect interests at the same time.

The SIMS qualification is seen as such a base knowledge to be transported to the experts to form a basis for collaborating in social media based knowledge and expert clusters.
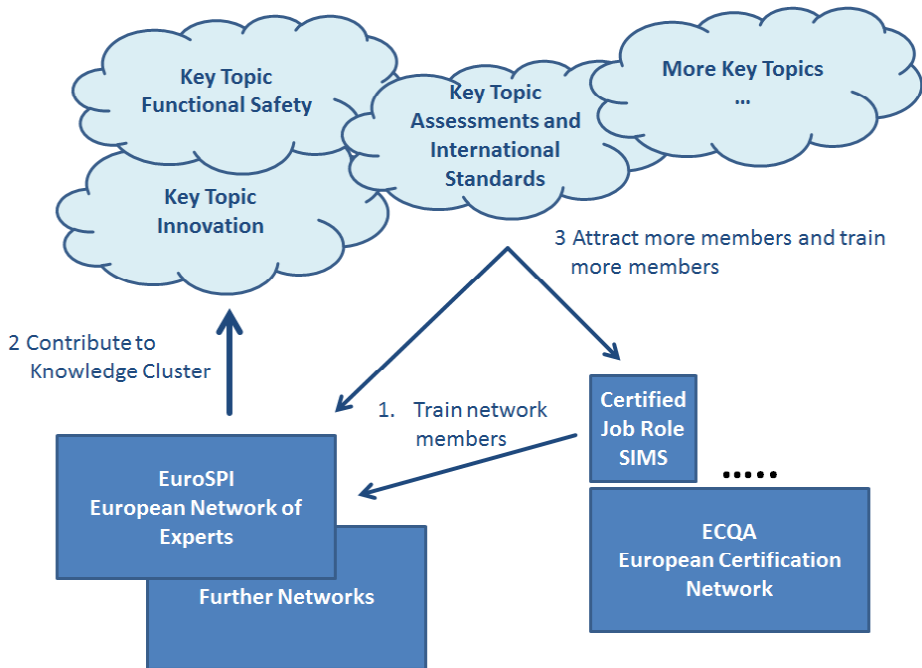

Figure 3. Example Integration of Skills like SIMS into a European Network (Source: Messnarz et.al, 2011)

Fig.3 shows the integration of SIMS into ECQA (Europea Certification and Qualification Association) and EuroSPI (European Systems, Software and Services Process Improvement and Innovation) in one holistic approach. ECQA offers to people a training and certification of social media skills. EuroSPI started to establish so called workshop communities for key topics in industry. The members of the workshop communities are trained and certified.

Then the workshop communities continue to use the knowledge to collaborate in a social knowledge area through which more experts can be attracted.

This then leads to a loop which continuously increases the number of experts involved, the workshops at EuroSPI and the ECQA pool of certified people.

This is only an example implementation of SIMS in a European context and could be applied in a similar setting in different industrial groups.

5 Europe Wide Certification

The SIMS project applies European standards assuring a Europe wide available certification.

ECQA (European Certification and Qualification Association) is a result of two EU LLP projects (EQN 2005 - 2007, and EU Cert 2007 - 2009) which set up a Europe wide standard for skills definitions, skills assessment, training performance, and certification. So far above 30 LLP projects applied ECQA, more than 60 training bodies from 18 countries support it, and we certified above 7000 managers in all European countries and even in USA, India, Australia, Russia, etc.

The ECQA certified social media networker implements all these quality standards of skill sets, standard modular structure of training material and standard Europe wide exams and certification.

You can find more information at ecqa.org, under certified job roles.

6 Conclusions and future work

SIMS training has been developed and is available in 2013. The SIMS skill card and training will be promoted Europe wide and at the EuroSPI 2013 event for building a first Europe wide expert base for collaborating in the social media knowledge approach.

The training is also available online and people can attend the training from home and their work place. The exams are based on a Europe wide standard exam questions pool and exam servers of the ECQA. Join the community and contact the project leader Ricardo Colomo-Palacios to be invited to the online courses.

Future Work will be focused on the development of training devoted to specific needs as the use of social media in product development, innovation management and co-creation environments.

7 Acknowledgements

8 Literature

Ahlqvist, T., Bäck, A., Heinonen, S., & Halonen, M. (2010). Road-mapping the societal transformation potential of

social media. *foresight*, *12*(5), 3–26.

Bullinger, H.J. (1999). Turbulent times require creative thinking: new European concepts in production manage-

ment, International. Journal Production Economics, No. 60-81, pp. 9-27

Chatti, M. A., Schroeder, U., & Jarke, M. (2012). LaaN: Convergence of Knowledge Management and Technolo-

gy-Enhanced Learning. *IEEE Transactions on Learning Technologies*, *5*(2), 177–189.

Chesbrough, H. (2003). Open Innovation, The New Imperative for Creating and Profiting from Technology, Har-

vard Business School Press, Boston, USA

Di Gangi, P. M., Wasko, M., & Hooker, R. (2010). Getting customers' ideas to work for you: Learning from Dell how to succeed with online user innovation communities. *MIS Quarterly Executive*, *9*(4), 213–228.

Drucker, P. F. (1992). *The Age of Discontinuity: Guidelines to Our Changing Society*. Transaction Publishers.

Drucker, P. F. (1999). Knowledge-worker productivity. *California management review*, *41*(2), 79–94.

Evans, D. and McKee, J. (2010), Social Media Marketing, the Next Generation of Business Engagement, Wiley Publishing, Inc., Indianapolis, USA

Eurostat newsrelease (2012). *Internet access and use in 2012,* Eurostat Press Office, 185/2012 - 18 December 2012

Gruber, T. (2008). Collective knowledge systems: Where the Social Web meets the Semantic Web. *Web Semantics: Science, Services and Agents on the World Wide Web*, *6*(1), 4–13.

Larcker, D., Larcker, S., Tayan, B. (2012). *What Do Corporate Directors and Senior Managers Know about Social Media?* Director Notes, No. DN-V4N20, October 2012, The Conference Board, USA, p. 6.

McKinsey Global Institute, MGI (2012). *The social economy: Unlocking value and productivity through social technologies.* McKinsey Global Institute, p.14, pp. 154-157.

Messnarz, R., Biro, M., Koinig, S., Reiner, M., Vajde-Horvat, & Ekert, D. (2012). The Future of SPI Knowledge and Networking in Europe –A vision. In Rory V. O'Connor, Jan Pries-Heje, Richard Messnarz (eds), Systems, Software and Services Process Improvement, Communications in Computer and Information Science, CCIS 172, pp. 268 – 278, Springer.

Milovanović, M., Minović, M., Štavljanin, V., Savković, M., & Starčević, D. (2012). Wiki as a corporate learning tool: case study for software development company. *Behaviour & Information Technology*, 31(8), 767-777.

O'Dell, C., & Hubert, C. (2011). *The New Edge in Knowledge: How Knowledge Management Is Changing the Way We Do Business*. John Wiley & Sons.

Perrigot, R., Basset, G., & Cliquet, G. (2011). Multi-channel communication: the case of Subway attracting new franchisees in France. *International Journal of Retail & Distribution Management*, *39*(6), 434–455.

Riemer, P. D. K., Richter, D. A., & Bohringer, D.-W.-I. M. (2010). Enterprise Microblogging. *Business & Information Systems Engineering*, *2*(6), 391–394.

Seufert, A., Krogh, G. von, & Bach, A. (1999). Towards knowledge networking. *Journal of Knowledge Management*, *3*(3), 180–190.

Siakas, K., Messnarz, R., Georgiadou, E. and Naaranoja, M. (2012), "Launching Innovation in the Market Requires Competences in Dissemination and Exploitation", in D. Winkler, R.V. O'Connor, R. Messnarz (eds), Systems*, Software and Services Process Improvement, Communications in Computer and Information Science,* Springer Verlag Berlin Heidelberg, Selected Key-note, pp. 241-252

Stodder, D. (2012). Customer Analytics in the age of social media. TDWI Research, Renton, Washington, USA.

Warren, J., Stoerger, S., & Kelley, K. (2012). Longitudinal gender and age bias in a prominent amateur new media community. *new media & society*, 14(1), 7-27.

9 Author CVs

Ricardo Colomo-Palacios

Ricardo Colomo-Palacios is an associate professor at the Computer Science Department of the Universidad Carlos III de Madrid. His research interests include applied research in information systems, software project management, people in software projects and social and semantic web. He received his PhD in computer science from the Universidad Politécnica of Madrid (2005). He also holds a MBA from the Instituto de Empresa (2002). He has been working as software engineer, project manager and software engineering consultant in several companies including Spanish IT leader INDRA. He is also an Editorial Board Member and Associate Editor for several international journals and conferences and Editor in Chief of International Journal of Human Capital and Information Technology Professionals.

Richard Messnarz

Dr Richard Messnarz has 23 years industry experience and is the chairman of EuroSPI, the coordinator of the EU Certified Innovation manager, the Vice president of the European Certification and Qualification Association, the moderator of task forces of 24 leading German companies, the editor of 7 books (IEEE, Wiley, Springer) on process improvement, and the director of ISCN since 1997. He is a principal ISO 15504 assessor working for firms like ZF Friedrichshafen AG, Robert BOSCH, Magna, Continental, T-Systems, etc.

Antonio Amescua Seco

Prof. Amescua is a full professor in the Computer Science Department at Carlos III University of Madrid. He has been working as software engineer in a public company (Iberia Airlines) and in a private company (Novotec Consultores) as a software engineering consultant. He founded Progresion, a spin-off company, in order to offer advanced Software Process Improvement Services. He received his PhD in computer science from the Universidad Politécnica of Madrid. His research interests include Software Process Improvement, Software Reuse and Software Project Management.

Kerstin Siakas

Kerstin Siakas is an associate professor at the department of Informatics of ATEI Thessaloniki Greece. Her research interests include Multidisciplinary Approaches of Systems Engineering, Knowledge and Quality Management. She has a particular interest in the social, cultural and political approaches and their effect on society, such as globalization, outsourcing, eLearning and eInclusion, as well as on the subsequent challenges for managers, educators and governments. She received her PhD in Software Quality Management from the London Metropolitan University (2002). She also has an extensive industrial experience in developing Information Systems on different levels in different countries.

Daniel Palosi

Daniel Palosi is an associate professor and head of institute at the Institute of Economics and Social Sciences of Dennis Gabor College in Hungary. His research interests include economic approaches of eLearning and eDucation, controlling, sustainable development and financial aspects of renewable energy projects. He received his PhD at the Doctoral (PhD) School for Management and Business Administration from the Kaposvár University in Hungary (2008). He gathered industrial experience in the field of controlling among others at Robert Bosch, and as a member of the Hungarian Rectors' Conference he has an overview and influence on the Hungarian Higher Education System.

Darragh Coakley

Darragh Coakley is a Multimedia developer/ researcher working for the DEIS Department of Education Development since September 2006, having previously worked in a Cork based graphic design consultancy (Dowling and Dowling). He has been involved with a number of European projects and was the project manager & coordinator of a recent transnational European project (ELM) aimed at developing the training and accreditation for the job role of a European e-learning manager. He holds a BA Honours Degree in Multimedia.

Adrienne Clarke

Adrienne Clarke studied Media in London and graduated as a Bachelor of Arts. She has more than 10 years' experience working in the Netherlands and in UK, followed by 12 years at ISCN. She has specialized in marketing and event management and organises the marketing and administration of the annual EuroSPI conferences. She has been working for ISCN as a EuroSPI coordinator, marketing manager and course manager since 2000. Adrienne Clarke will manage the dissemination of the project results through the ISCN dissemination networks such as EuroSPI and ECQA.

Understanding the Environment of Small and Medium Enterprises to Implement Software Processes Improvement

Mejia Jezreel[1], Muñoz Mirna[1], Muñoz Edrisi, Calvo-Manzano Jose A.[2],
[1] Centro de Investigación en Matemáticas
Avda. Universidad no. 222, 98068, Zacatecas, México
jmejia@cimat.mx, mamunoz@cimat.mx, emunoz@cimat.mx
[2] Facultad de Informática, Universidad Politécnica de Madrid
joseantonio.calvomanzano@upm.es

Abstract

It is well known that software product quality is largely dependent on the process that is used to create it. However, even when many organizations are motivated to implement software process initiatives (SPI), not all know how best to do, especially in Small and Medium Enterprises (SMEs) where due to its especial features, they have to be carefully in how to manage its resources to assure their market survival. As a result, implementing SPI initiatives in SMEs has become a great challenge, mainly because SMEs' work culture is immersed in a chaotic and changing environment. This paper presents a method for understanding the environment of SMEs in order to identify findings that allow to address a successful software process improvement in SMEs. Besides, an experience of applying the proposed method in 4 SMEs is presented.

Keywords

SMEs; Software Process Improvement; Business Goals; Goal Question Metrics.

1 Introduction

It is well known that software product quality is largely dependent on the process that is used to create it [4] consequently; the software industry is more and more concerned about software process improvement (SPI) [5]. In particular, Small and Medium Enterprises (SMEs) because in the past two decades this kind of organizations in the software development industry has emerged, grown and become strong [1,2,3,5,6]. Consequently, the SMEs are become more and more concern about software process improvement (SPI).

Unfortunately, the implementation of SPI in SMEs has been a path full of obstacles because they have very limited budgets to improve their software processes [2,4,5,7]. Besides, the SMEs' features such as high innovation and adoption, agile for change, limited customer with high dependency and projects with short delivery time limited staff, and lack of process culture and minimum trained related to processes [4-7] make difficult the SPI in SMEs. Moreover, SMEs' work culture, which has been identified as a key aspect in order to implement a successful software process improvement, is immersed in a chaotic and changing environment. Besides, SMEs face a great challenge: competitiveness problem, because they are organizations that handle limited economics resources and are unable to implement models and standards that ensures the quality and productivity in software product development.

In this context, although there have been developed standards and models focused on software process improvement, the implementation of these models and standards bring many obstacles on SMEs due to its organizational structure and special features. Because they are created for large companies, besides, these only indicate "what" without indicating "how to perform the activities contained in them". Besides, the SMEs' has a lack of processes culture and minimum trained related to software processes improvement.

Therefore, the goal of this paper is to present a method that allows SMEs: 1) to establish the organization goals aligned with the Organization's work culture and its organizational structure to identify findings for software processes improvement and 2) to focus the effort when developing a software process improvement initiative.

This paper is structured as follows: in section 2 the research background of the main issues including concepts such as: SMEs, models and standards, business goals and the Goal Question Metrics (GQM) is presented; in section 3 the proposed method is described; in section 4 the experience of applying the method is showed; finally, in section 5 the conclusions are covered.

2 Background

In this section the concepts in which the proposed method is based are described.

2.1 Small Entities (SMEs)

The European Commission [24] defines three levels of small to medium-sized enterprise as follow: a) Small to Medium: employ fewer than 250 persons, have an annual turnover not exceeding 50 million Euro, and an annual balance sheet total not exceeding 43 million Euro; Small: employs fewer than 50 persons, its annual turnover or annual balance sheet total does not exceed 10 million Euro, and Finally, Micro: employs fewer than 10 persons and its annual balance sheet total not exceeding 2 million Euro.

2.2 Models and Standards

Institutions such as the Software Engineering Institute (SEI), the Project Management Institute (PMI), the Institute of Electrical and Electronics Engineers (IEEE), and the International Organization for

Standardization (ISO) have been focused on the Software Processes Improvement [12].

The most widespread models and standards developed are: Capability Maturity Model and Integration for Development (CMMI-DEV) [17]; Team Software Process (TSP) [18]; Project Management Body of Knowledge (PMBOK)[19]; ISO/IEC15504 Information technology–Process assessment [16]; ISO 9001:2000-Quality Management System [21]; ISO/IEC 12207-2008 [22]. However, these models and standards only indicated "What to do" without showing "How to do" to perform the practices or activities contained in them. Besides, most of them were initially created to be implementing in large organizations. In this context, some organizations create their own software development process model, by tailoring models and standards such as CMMI and SPICE [17] [20]. However, most of SMEs do not adapt any standard or models [24] because, they perception about them, is related to high cost, documentation and bureaucracy.

2.3   Business Goals

It is well known that organizations should have their business goals as the main reference when they implement a software process improvement. But, what are business goals? How an organization can measure a business goal? Next, the concepts business goals and business indicator are briefly described.

- *Business goals* are goals or objectives established by senior management in order to ensure their continued existence and enhance its profitability, market and other factors that should be achieved to the organization's success [13-19, 23]. Some examples of business goals that an organization can establish may include: reducing developed cycle time, reduce the number of change request during an integration phase, increase the number of errors found after the second phase; increase customer satisfaction, reduce de number of project with any kind of deviation, and so on.

- *Business indicators* are words that help organizations to be specific about the measures they need to get information about business goals. Some examples of business indicators are licenses renewal rate, number of customers, quoted lead times, number of common processes, reductions in product development or service cost, management rules, time to accommodate design changes, ration of development time to product life [23].

2.4   Goal Question Metric

This section presents a brief description of Goal Question Metric (GQM) because the proposed method is based on it. GQM consist of 3 main stages [9]:

1. *Define the main goals of the organization.*
2. *Get the questions for each goal in order to know whether the desired objectives are met.*
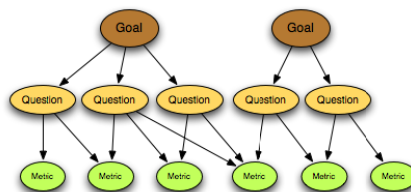3. *Define metrics to answer the questions in a quantitative manner.*

Figure 1. Goal Question Metric methodology

GQM is defined as a hierarchical tree (see Figure 1); the goals have the highest hierarchy. For each objective is defined a set of questions to break the goal in its main parts. Each question in turn has a

set of metrics in order to answer questions quantitatively; a metric can answer questions of different objectives.

GQM is a very useful for software process improvement due to its maturity and flexibility in any organizational environment [12-15]. GQM has been used to improve processes in companies like Ericsson [12] focusing on the process of change management and process engineering requirements, and Nokia [13] focusing on their development processes and software maintenance. Besides, GQM has been used in fields outside of software development [14-16].

3   Proposed Method

The goal of the proposed method is to identify findings in SMEs in order to improve and focus the organization's effort for a software process improvement initiative. In order to develop the proposed method the main stages of GQM (see section 2.4) have been taken into account. These steps have been combined with other activities. Figure 2 shows a general scheme of the proposed method.
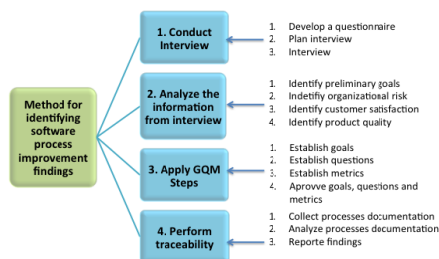

Figure 1. Proposed Method

3.1   Conduct Interviews

The purpose of this activity is to extract the tacit knowledge of the organization's senior manager (CEO). To achieve this, the senior manager who has structured and founded the organization is interviewed. The interview is semi-structured and recorded using questionnaires as a guide. Table 1 shows the subactivities for this activity.

Table 1. Subactivities of the activity *Conduct interviews*

Subactivities	Description
Develop a question-naire	Develop a questionnaire, which should have open questions and contain easily understood questions. So, it will allow to collect information about the organization' structure and actual work culture.
Plan interviews	Plan interviews as closely as possible according to the number of meetings agreed on with the senior manager.
Interviews	Interviews with senior manager.

3.2   Analyse the information from the interview

The purpose of this activity is to analyze the information gathered from previous interviews. Then the structure and word culture of the organization are established. Table 2 shows the subactivities for this activity.

Table 2. Subactivities of the activity: *Analyze the information from interview*

Subactivities	Description
Identify preliminary goals	Analyse data from interviews in order to identify preliminary goals. The organization must have a purpose and/or goals clearly stated and communicated to all stakeholders (everyone from the customer and staff), so they have a clear understanding of the goals. Then, they can make decisions that are consistent with the organization's overall goals. Organizations without accurate goals will be solving the wrong problems.
Identify organizational risk	Identify organizational risk in order to attempt to find out what may go wrong. If the senior manager claims there are no risks on the organization, then he has not performed proper risk management.
Identify customer satisfaction	Identify the expected customer' satisfaction. For example delivering innovative and quality products and solutions on time, within schedule and budget constraints, and with minimal risk.
Identify product quality	Identify the functions, activities, processes, work products, and/or services that are performed, produced and/or rendered, against established best practices, standards and procedures.

3.3   Apply GQM steps

The purpose of this activity is to analyze the information gathered from activity 2. "Analyze the information from interviews" and then perform the main steps of GQM. Table 3 shows the subactivities for this activity.

Table 3. Subactivities of the activity *Conduct interviews*

Subactivities	Description
Establish goals	A goal is defined for an object, for a variety of reasons, with respect to various quality models, from various points of view relative to a particular environment. Objects of measurement are products, processes, and resources.
Establish questions	A set of questions are identified and established to characterize the way to assess and achieve a specific goal.
Establish metrics	A set of data is associated with every question in order to answer it in a quantitative way.
Approve Goals, questions and metrics	Once the proposed goals, questions and metrics are established these must be approved by the organization's senior manager (CEO).

3.4   Perform traceability

The purpose of this activity is to find the metrics within organization's processes in order to find the data to accomplish the proposed metrics and answer its related questions. Table 4 shows the subactivities for this activity.

Table 4. Subactivities of the activity *Traceability*

Subactivities	Description
Collect processes documentation	Request and collect the organization's processes documentation.
Analyze processes documentation	Once the current processes are collected, they are analysed. The performance's data should be assessed in order to know how is the organization's performance in relation to the proposed metrics in the prior activity "Apply GQM steps" is covered.
Report findings	Make a list of findings. This list will help to focus the organization's software process improvement.

Once GQM activities are applied the goals, questions and its metrics are obtained. The results are used in order to perform traceability with organization' processes and to find the data that allows obtaining the values for proposed metrics.

4 Experience of applying the proposed method

This section presents the experience of implementing the proposed method in Software SMEs over a 6-months period. The SMEs included in this experience were primarily located in México. For security and confidentiality reasons of the SMEs involved in this experience, all the SMEs' names will be kept on secret during this case study analysis. Next, a briefly description of the involved SMEs are included:

SME1 is a Software SME established in February 2011. It was started as a joint venture of other SME related to telecommunications consultancy and a research center with financing provided by CONACyT (National Counsel of Science and Technology). This SME is specialized in web applications, the most used software is: PHP as programming language, YII application Framework and MySQL as database.

Actually, SME1 is composed of: a) 7 junior developers with less than a year of experience, b) 2 senior developers with an avg. of five years of experience, c) 1 project leader with over 20 years of experience in administrative roles and d) 1 technical leader with 10 years of experience. Besides, 5 team members are certified Scrum Masters and the whole team received internal training in Scrum and Extreme Programming (XP). The most important process in SME1 is "Software Development". In order to achieve this process, Scrum agile methodology defined by the ScrumAlliance.org [3] is performed.

SME2 is a Software and Consulting SME established in 2007. This SME develops software and consults for government agencies offering solutions usually of financial nature. SME2 has tripartite division units in the states of Monterrey, Guadalajara and Zacatecas. The main office (Zacatecas) has a team of twenty persons that supports other units. SME2 has achieved Moprosoft (Process Model for Software Industry) Level 1 certification in 2009. This has enabled the SME2 to be an organization skilled at assessing and improving their own processes both developing and maintaining of software and service product.

SME3 is a Software SMEs established in 2009. The main activity of this SME is the application of FPGA (Field Programmable Gate Array) and embedded systems development, providing comprehensive solutions designed to generate specific products and designs that suit to every customer. It also provides support services and training in the use and exploitation of new technologies.

SME3 has 8 employees covering the following business areas: Business Management, Production, Design, Embedded Software, Micro controllers, Design PCB (Printed Circuit Board), and Research and Innovation.

Actually, SME3 does not have implemented any quality model or standard. However, the management department has taken a general ITIL course, (only part of CAB (Change Advisory Board)) in order to achieve that each production area agrees and is aware with the changes in the projects.

SME4 is a Software and Consulting SME established in 2009. It provides software development services to a customer (Deloitte). Therefore, its main objective is to deliver on time all work requested by Deloitte. Its main development is correcting or improving its customer's systems. Actually, SME4 is composed of 7 employees trained in IBM Lotus Notes technology. At present, SME4, does not have implemented any quality model or standard.

4.1 Applying the proposed method

In this section the method activities are performed in order to prepare the 4 Software SMEs to implement software processes improvement.

4.1.1 Conduct interviews

In this activity, the interviews date and questionnaires for interviews were established. As a result, the senior manager who has structured and founded the organization for each SME was interviewed.

4.1.2 *Analyze the information from interview*s

Information gathered from previous interviews was analyzed. Then, preliminary goals, organizational risk, customer satisfaction and product quality are identified. As a result, the needs and preliminary goals for each SME are identified.

4.1.3 Apply GQM steps

Once the information from previous activities is obtained, GQM activities are applied in order to obtain the goals, questions and metrics in each SME (SME1, SME2, SME3 and SME4).

4.1.3.1 Establish Goals

The business goals defined by each SME are:

- SME1. G1: Establish the maintenance and support processes, in order to increase the customer satisfaction.
- SME1. G2: Ensure that the quality and reliability of a software product is supported through a maintenance process.
- SME2. G1: Analyze self-monitoring equipment in order to assess the use of human resources, from the point of view of the project manager.
- SME2. G2: Analyze the lifecycle of software development in order to assess compliance with the CMMI level 2 practices, from the point of view of the development project leader.
- SME3. G1: Establish a change management process to meet the goal of each change and increase the performance of all team members.
- SME3. G2: Establish a supplier management process, to improve the planning of activities throughout the project lifecycle.
- SME3. G3: Evaluate the quality of products in order to meet customer satisfaction.
- SME3. G4: Ensure the quality of delivered products to record historical data.
- SME4. G1: Deliver on time change requests.
- SME4. G2: Provide a mechanism for estimating time and correcting defects.
- SME4. G3: Have an estimation mechanism for creating new functionality and new projects.
- SME4. G4: Have a database of project information.

4.1.3.2 Establish question and metrics

According to the defined goals in the previous subactivity a set of questions and metrics are proposed related to the goals and question establishes in each SME. It is considered important to note that the question and its metrics are not described in this paper because the main goal is to show the identified goals in each SMEs and to inform the findings in order to prepare the SPI in SME.

Once the questions and metrics are established the goals with its questions and metrics were mapped. Figure 3 shows a mapped example.
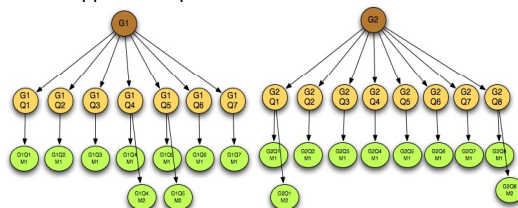

Figure 1. Established Goals, Question and Metrics

4.1.4 Traceability

Once GQM activities are applied to obtain the goals, questions and its metrics, the results were used in order to perform traceability with *organizations* processes and to find the data that allows to obtain the values for proposed metrics.

5 Report findings

This section shows the findings identified by performing the proposed method:

- SME1: In the current processes of this SME the technical support requests from customers are handled in an informal way. Then, this SME has problems in software maintenance because its main development process does not have a defined process related to support request. Because maintenance activities have been ignored in its agile processes under the argument that maintenance is just other iteration after the final release.

- SME2: the way recommended to address the identified goals for this SME is to map the MoPro-Soft practices that currently are being carried out in this SME with practices of CMMI Level 2 processes in order to identify the practices that are not being implemented according to CMMI level 2. Besides, this mapping must identify which metrics are essential, since in the current organizational practices there are not defined metric. Metrics identified should also be automated for collecting and analyzing them.

- SME3: this SME has not implemented any quality model or standard. Therefore, according to identified goals, it is necessary to establish a Configuration Management Process as one of the integral processes because is not defined within the organization. Moreover, there is a need to control the supplier management in order to control and monitor the dates of sending and receiving developed hardware designs. Besides, mechanisms for managing effective delivery of IT services (Information Technology) must be established. Also, the quality and requirements of all types of established products must be assessed.

- SME4: this SME has not implemented any quality model or standard. Therefore, according to identified goals, a mechanism to estimate time and defect correction must be established in order to achieve the deadline set by the client. In addition, time estimation mechanisms for the defects correction must be established. All this, should be supported by the development of a database of historical data for each project.

After analyzing the business goals, some common features have been identified. These features do not allow the implementation of software processes improvement in this SMEs:

- The processes standardized or documented do not reflect the SMEs actual work culture.

- The perception about models and standards quality do not add value to its business, because they think that the models and standards are developed for large companies.

- Processes or procedures related to quality control are not formalized.

- Most of the project management, planning and requirements change practices are not standardized within organization.

- The software development phases are not formalized (e.g. analyzing, designing, implementing and testing).

- Most of the software projects are driven by a short schedule for its delivering.

- The resources to training in software processes (e.g. CMMI, ISO) are very limited.

- Support and maintenance issues (process and/or procedure) are not addressed in the software development lifecycle.

- Customer and supplier management issues are not addressed in the SMEs.

- Metrics to measure the performance and quality of products and processes in the organization are not defined.

5.1   Recommendations for SPI in SMEs

Taking into account the findings from this experience, it is possible to propose recommendations for the implementation of Software Processes Improvement in SMEs. According to the experience the recommendation for both SMEs with establishing model and without establishing models are:

1. The implementation of software process improvement must be aligned with business goals. The business goals have two characteristics in the SMEs:

   - They are not defined and not established.

   - They are defined but are not aligned to the actual business goals in the SMEs.

   Therefore, the business goals should be established based on the structure and work culture of the SMEs in order to guide them toward successfully products, processes and resources improvement.

2. The implementation of software process improvement must be aligned with actual organization work culture because if the established processes do not reflect the organization actual work culture, the processes are perceived as too much documentation and the developers priority is focused only on coding.

3. The established of metrics to assess processes, products and resources must be aligned with actual organization work culture. Metrics definition must be aligned to business goals taking into consideration the actual organization work culture. These metrics should be simple to obtain real data that allow to focus the software process improvement.

6   Conclusions

The proposed method allows SMEs to establish and identify the organization goals and metrics aligned with the structure and actual work culture in the organization to identify findings for software processes improvement. By this way, it is possible to focus the effort of a software process improvement initiative in SMEs.

The experience showed in the section 4, confirms that the main reasons for not implementing software processes improvement in SMEs or adopting models and standards are that the business goals in SMEs are not defined and not established. Otherwise, these are defined but are not aligned to the real business goals in the SMEs. Moreover, Software Processes improvement initiatives do not take into account the SME actual work culture.

Finally, the experience got of implementing the proposed method is an evidence of the adequacy of the method to find findings in SMEs in order to improve software process.

With regard to future work, we are extending our study by identifying more SMEs in Mexico in order to gain more insight about how to prepare the implementation of Software Processes Improvement. This could help to obtain results and findings about the best way to implement SPI in SMEs.

7   Literature

1. Pino J.F., Garcia F., Piattini M.(2008). "Software process improvement in small and medium software enterprises: a systematic review". SQJournal, vol. 16, pp. 237-261.
2. Garcia I., Pacheco C. and Cruz D. (2010) "Adopting an RIA-based tool for supporting assessment, implementation and learning in software in software process improvement under the NMX-I-059/02-NYCE-2005 standard in small software enterprises". Eighth ACIS International Conference on Software Engineering Research. Management and Application.
3. Wymenga P., Spanikova V., Derbyshire J. and Baker A. (2011) "Are EU SMEs recovering", Annual Report on EU SMEs 2010/2011. Rotterdam. Cambridge.

4. Dyba T. (2005) "An empirical investigation on the key factors for success in software process improvement". IEEE Transactions on Software Engineering. Vol. 31.

5. Garcia I. and Pacheco C., (2009) "Toward Automated Support for Software Process Improvement Initiatives in Small and Medium Size Enterprises", R. Lee & N. Ishii (Eds.): Soft. Eng. Research. Manage. & Appli. 2009. SCI 253. pp. 51–58.

6. Instituto Pyme. (2012) Las PYMES en México- clasificación oficial de las Pymes de acuerdo a su tamaño.

7. Habra N., Alexandre S., Desharnais J., Laporte C.Y. and Renault A. (2008) "Initiating software process improvement in very small enterprises". Inf.Softw.Technol. vol. 50. Pp. 763-771.

8. David J. Anderson, Kanban. (2010) Successful Evolutionary Change for Your Technology Business.

9. Basil V., Caldera G., and Dieter Rombach H. (1994) The Goal Question Metric Approach.

10. Deemer P., Benefield G., Larman C., and Vodde B. (2009) The Scrum Primer.

11. VanHilst M, Pankaj K. G., and Lo C. (2005) Repository mining and Six Sigma for process improvement. 2005.

12. Berander P., Jönson P. (2006) A Goal Question Metric A Goal Question Metric Based Approach for Efficient Measurement Framework Definition.

13. Oinas A., Defining Goal-driven Fault Management Metrics in a Real World Environment: A Case-Study from Nokia.

14. Sarcia S. A. (2010) Is GQM+Strategies Really applicable as is to non-Software Development Domains?

15. Hussain A., Elaine Ferneley. (2008) Usability Metric for Mobile Application: A Goal Question Metric (GQM) Approach.

16. Osorio B., Irrazabal O., Garzás J. (2011) Toward improving Agile Mantema: measurement, control and evaluation of maintenance projects in SMEs.

17. SEI CMMI Production Team, S. (2011) CMMI for Development v1.3. Software Engineering Institute.

18. Humphrey W. (2006). Introduction to the team software process. Massachusetts: Addison-Wesley

19. IEEE Computer Society. (2004). IEEE guide adoption of PMI standard A guide to the project management body of knowledge; IEEE Std 1490-2003 (Revision of IEEE Std 1490-1998). 0_1-216. doi: 10.1109/IEEESTD.2004.94565

20. International Organization for Standardization. (2004). ISO/IEC 15504: 2004 Information Technology – Process Assessment, Part 1 to Part 5.

21. AENOR. (2000). Sistemas de gestión de calidad (ISO 9001:2000), UNE-EN ISO 9001, Norma Española

22. International Organization for Standardization. (2008). Systems and Software Engineering - Software Life Cycle Processes, IEEE STD 12207-2008," Ieeexplorer, pp. cl-138,

23. Goethert, W., Hayes, W., Experiences in Implementing Measurement Programs, Software Engineering Institute, Technical Note CMU/SEI-2001-TN-026. (2001)

24. Basri S. and O'Connor R. (2010) Understanding the Perception of Very Small Software. Companies towards the Adoption of Process Standards. EuroSPI 2010, CCIS 99, pp. 153–164.Author CVs

Author CV's

Mejia Jezreel

He holds a PhD in Computer Science. He is actually researcher in Computer Science at the research center in mathematics (CIMAT), Zacatecas, México. His research field is Software Engineering. He is a member of MPSEI (Mejora de Procesos Software en el Espacio Iberoamericano) research group. His main research field is process improvement, focusing in the topics of multimodel environment, project management, acquisition and outsourcing process, solicitation and supplier agreement development and agile methodologies. He has published several technical papers on acquisition process improvement, project management, TSPi, CMMI, muti-model environment. He has been a member of the team that has translated CMMI-DEV v1.2 and v1.3 to Spanish.

Muñoz Mirna

She holds a PhD in Computer Science. She is actually researcher in Computer Science at the research center in mathematics (CIMAT), Zacatecas, México. She is a member of MPSEI research group. Her research field is Software Engineering. Her current research interest consists to incorporate process improvement focusing on diminishing change resistance. She has published several technical papers on project management, multi-model environment and process improvement. She has been a member of the team that has translated CMMI-DEV v1.2 and v1.3 to Spanish.

Muñoz Edrisi

MSc. in industrial engineering by Instituto Tecnológico de Orizaba by from Mexico. Doctor of Philosophy in chemical process engineering by Universitat Politècnica de Catalunya (UPC) from Spain. His research aims at knowledge management by the development of ontologies and management frameworks for decision-making support in different areas. He is actually researcher in Computer Science at the research center in mathematics (CIMAT), Zacatecas, México and collaborates with the UPC as part of the research team, participating in different Mexican and European research projects.

Calvo-Manzano Jose A.

He holds a PhD in Computer Science. He is assistant professor in the Facultad de Informática at Universidad Politécnica de Madrid. He is teaching in the area of Software Engineering, specifically in the domain of software process management and improvement. He has participated in more than 20 research projects (European and Spanish Public Administration). He is author of more than 50 international papers. He is author of books related to software process improvement and software engineering topics also. He is a member of SOMEPRO research group, where his main research field is process improvement, focusing in the topics of requirements management, project management, process deployment, and solicitation and supplier agreement development. He is the head of the "Research Chair in Software Process Improvement for Spain and Latin American Region", where he has been a member of the team that has translated CMMI-DEV v1.2 and v1.3 to Spanish.

An industrial assessment for a multimodel framework

Xabier Larrucea, Izaskun Santamaria

TECNALIA. Parque tecnologico de Zamudio 202, 48170 Bizkaia, Spain
{Xabier.larrucea;Izaskun.santamaria}@tecnalia.com

Abstract

Software process improvements initiatives (SPI) are facing complex environments where stakeholders need to integrate different reference models in their organisations. There are several approaches to multimodel environments defining some steps or activities that should be carried out for implementing it efficiently inside organisations. This paper presents a report on the use of a multimodel framework integrating 3 quality reference models in 47 SPI initiatives. Basically this report is based on statistical data extracted from industrial assessments.

Keywords

Multimodel framework, quality models, process improvement

1   Introduction

Multimodels environments [5,6,7,8,9] are becoming a widely common situation for organisations such as [1, 2, 3, 10] where authors identify a framework basically based on mappings between process models at a high level. Multiple models, regulations, standards and laws coexist in the same environment generating and adding complexity to products developments, and this is the same case for safety environments. In a broad sense most of software process improvement (SPI) initiatives involve this kind of situations. The added value of SPI initiatives is clear not only from organisations' revenues and performances point of views, but also from their entire business.

Multimodels have been discussed in literature during these recent years such as in [10]. These works provide some valuable insights for people with a high knowledge on some specific standards, but they do not provide a roadmap on how to effectively apply to company settings especially in safety critical systems or even in small settings. In addition there is a scarce set of industrial experiences using multimodel approaches from a conscious point of view, and applying a defined framework.

SMEs are playing an important and relevant role in the global economy [14]. In fact *SMEs account for roughly two thirds (66%) of employment within European Union*, and it is very similar to Latin America settings where the 80-90% of companies are micro enterprises [14]. Therefore Europe and Latin America share a common situation where SMEs represent the backbone of the economy. There are several reference models used for assessing specific aspects of an organization. In the software industry context there is a set of quality models (e.g., Capability Maturity Model Integration –CMMI® for software development [16])) used for assessing quality of software developments. Despite its different Process Areas (PA) CMMI® is used for specific aspects in software developments organizations. There are other models such as [17] for improving their competitiveness in security. However both models are not aligned in their conception and organizations are not only worried about these two aspects, and usually SMEs require other support for improving business models.

Based on this situation this paper presents some results on the adoption of a multimodel framework [13] and a ser of quality reference models [15] from an industrial perspective including process areas from business aspects, process oriented model aspects and security aspects (if they apply). ITMark has been applied in several organizations with excellent results as we show in this paper. We use as a generic framework our multimodel framework [13] that can be applied in safety environments but also in other domains, and this paper shows this case.

Therefore the main objective of this paper is to provide industrial results on the following industry questions (IQ):

IQ1. Is it useful our previously defined multimodel framework [13] in an industry context?
IQ2. What are the assessment results for each reference models?
IQ3: Are the results from different reference models related in somehow?

This paper is structured as follows. First an overview of the ITMark method and a description of our multimodel framework are presented. Second some statistical data from 47 organisations based on our approach are reported. Third section includes a set of relevant tools. And finally a conclusion section ends this paper.

2   Research background

2.1   ITMark

In general in order to be competitive, SMEs must demonstrate not only benefits for the achievement of a relevant position in the market but also they need to improve different areas within the organization. Nevertheless different quality models used nowadays covering some important aspects most of them do not cover all areas of organizations and their business as whole. Organizations aiming to improve their business through quality cannot be only focused on improving a small part of their business. They need to provide a holistic view of the organization's areas as well as their business aspects. ITMark includes 3 quality reference models that are described as follow:

Table 1. CMMI® process areas involved in ITMark

CMMI PA Category	Maturity Level 2	Maturity Level 3
Process Management		OPF: Organisational Process Focus OPD: Organisational Process Definition OT: Organisational Training
Project Management	PP: Project Planning PMC: Project Monitoring and Control SAM: Supplier Agreement Management	IPM: Integrated Project Management RSKM Risk Management
Engineering	REQM: Requirements Management	RD: Requirements Development TS: Technical Solution PI: Product Integration VER: Verification VAL: Validation
Support	CM: Configuration Management PPQA:Process and Product Quality Assurance MA:Measurement and Analysis	DAR: Decision Analysis and Resolution

A. Business Processes Model

ITMark business processes model is based on Business Management and it has been divided in 10 specific categories:

- Market
- Management
- Products&Services
- Sales, Marketing and distribution
- Strategy and committee
- Financial analysis
- Customer's profile and analysis
- Investment factors
- Development and production
- Industry and macro environment

Each category is composed in its turn by 10 important elements. Therefore this model contains 100 elements for characterizing business aspects.

B. Software, Systems and Services Engineering model

Software and systems development process model is based on CMMI® [3] structured in several process areas as it is shown in table 1. This assessment method is mainly focused on maturity level 2 and 3 because these process areas cover the most relevant management and engineering practices.

C. Security Management model

Information security is not only related to protect our systems from external attacks, but also to protect from internal attacks mainly due to human errors, hardware and software fails, and disasters. There are several security attributes contributing to a holistic security model. However we have considered as main attributes: confidentiality, integrity and availability based on ISO [4] and [5]. In fact ITMark adapts these ISO requirements to SMEs (Small and Medium Enterprise) and it categorizes all these attributes in three levels:

- First level includes control attributes relevant to any organization. These elements are gathered as a set of logical and physical controls and procedures related to information classification and business evolution. The prevailing legislation is also another important concept to take into account, mainly those related to information technologies. This paper reports experiences from Spain and therefore it

was applied the Spanish legislation. However this model represents a general framework and it can be adapted to other regions and legislations.

• Second level is related to control deployment and it need to be justified through risks analysis. This is a basic activity for any information security management system. Security is sponsored by management issuing policies for the organization. Employees receive training on security based on their roles and profiles. Security attributes coming from the first level are improved and increased with new requirements upgrading their complexity.

• Third level is focused on risk management. At this level some practices related to secure software developments are added.

2.2 Multimodel Framework

Our approach is to use our previously defined framework [13] for assessing organization taking into account three different quality reference models. In this context we have instantiated each Quality Model of our multimodel framework by each reference model on ITMark. In addition we have included a new point of view on the right side of Fig.1 in order to represent the assessment point of view that it is going to be presented in this paper. The rest of elements such as "Safety Integrity" are not still tested in this industrial scenario. This framework defines the following three layers:

● **Realm of Reference technologies:** this layer is used by technology experts having a deep knowledge on standards, laws, reference models and so on. For example the research work presented in [10] refers this layer as the framework for supporting multi-models Harmonization. In our approach all these technologies, standards and models are represented by ITMark quality reference models.
● **Realm of Organisational processes:** this layer is mainly focused on the identification of ongoing processes inside an organization. All these processes can be explicitly defined or implicitly performed.
● **Realm of Evidences:** there are a wide set of different evidences in a company and most of them can be assessed different times and with respect to different models or regulations. This layer is used for assessing each organisation. The results of these assessments are described from statistical point of view in this paper.
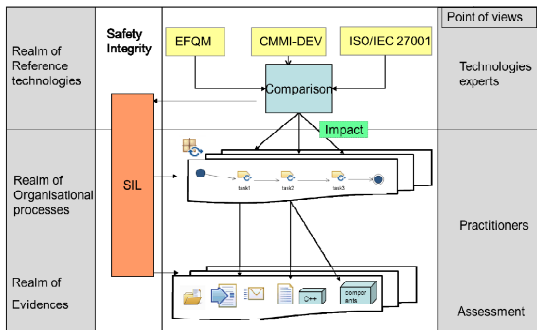

Fig. 1. Multimodel framework based on [13] and tailored for ITMark environment

The use of our framework implies the use of the harmonization process (Fig.2). This paper just presents the Multimodel results phase.

Fig. 2. Harmonization process [13]

3   *Statistical data from multimodel assessments*

ITMark has been applied in several organizations. This paper reports data from 47 assessments applied in 44 organisations in 7 different countries (Fig.3).
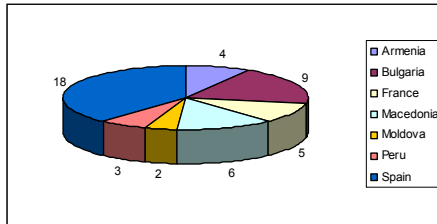

Fig. 3. Countries and number of assessments

All assessments scope includes business processes, information security and CMMI® level 2 process areas analysis.

A.   Business Category

Based on this experience the most critical category in this area is the "Strategy and committee" category. As shown in Fig.4 organisations fail on this category and it is not treated correctly.
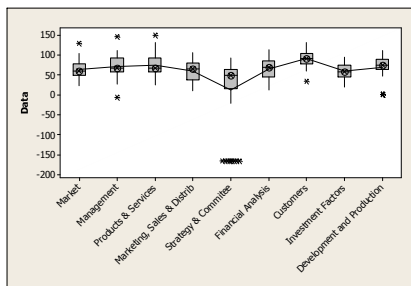
Fig. 4. Assessments results and business categories

From the analysis of these data filtered by categories and observing the correlations among them we obtained the following results (Fig. 5 (a), (b)):

- 20% (9 out of 47) do not have a business plan and they obtained as result -166.7
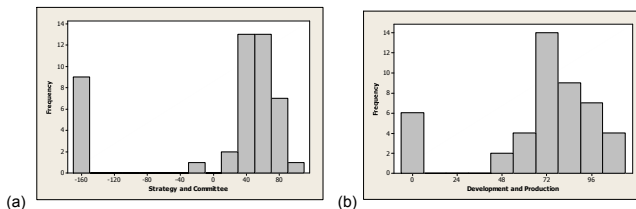- 13% (6 out of 47) do not have defined development nor production processes.

Fig. 5. (a) "Strategy and Committee" and (b) "Development and Production" categories histograms.

There is a strong relationship between organization's strategy and business plans definitions, and assessments results from these categories. This fact has been pointed out in correlation diagrams (Fig 6). In detail, the set of organizations do not having a business plan defined faced the following problems:

- Management strategy was not defined and therefore their mission and strategic objectives are blurred and/or unknown.

- Their resulting products are not aligned to market needs.

- Market positioning is unknown.

- Commercial activities are carried out without a strategy nor defined plan

- Benefits from products development are unknown. Project development is carried out without any formalism.


(a)   (b)
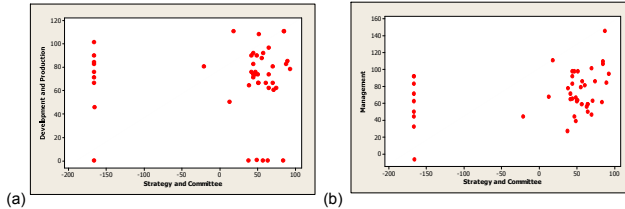
Fig. 6. Correlation diagrams related to "strategy and committee", "management" and "development and production"

Two critical improvement aspects arise from this situation:

- Strategy and business plan definition

- Development and production processes definition

As described in Fig. 7 well-defined processes definition contributes significantly to Products and Services development with a high level of quality. It is also possible to develop good products without defined processes but in this case all these organizations need to manage how employees develop products and depend on people's experience in order to achieve development projects objectives. In addition we can point out that there are some well defined processes but they are not very useful and effective. This situation reflects typically organizations where people define processes theoretically, from a bureaucratic point of view, in an exhaustive manner and poorly applicable to real projects.
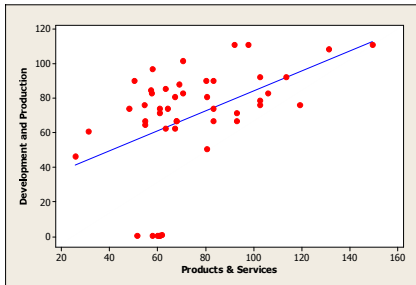
**Fig. 7.** Dependencies between "Products & Services" and "Development & Production"

B.   CMMI® model process areas

Development and production processes are assessed with respect to CMMI® practices and the results are shown in Fig.8. This graphic represents process areas dispersion found in during the assessments. A line connects all process areas medians. PPQA and MA are the process areas with the upmost spread during the assessments.
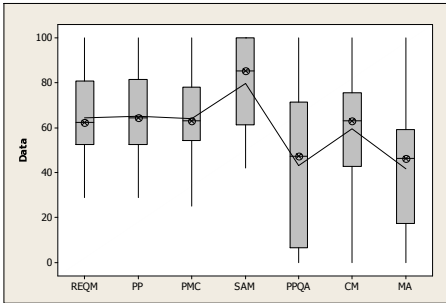
Fig. 8. Results from CMMI® model process areas assessments

More specifically and concerning PPQA Figure 8 shows that 23% (11 out of 47) of organizations do not have defined their main developments processes because it is considered as a waste of time. Nevertheless PPQA purpose is to provide a vision over the state of the processes and to maintain them up to date. As it is shown in Figure 6 developed products quality depends on their related processes applied by the organization.
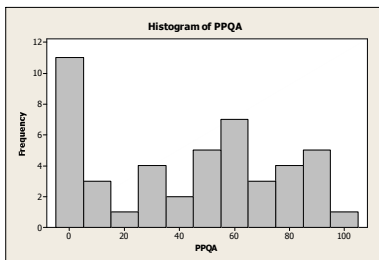
Fig. 9. PPQA probability plot and histogram

With respect to Measurement Analysis process area the main reason for this dispersion is that this process area requires setting up a measurement framework linking business objectives to metrics used within projects. In addition to these difficulties, data gathering, data analysis and an adequate usage of this information produced the results shown in Figure 9.

Nevertheless Figure 8 shows that MA process area contributes to activities performance related to "Strategy & Committee" category.

MA and PPQA areas share a common business objectives definition practice being the cornerstone for linking business side with development and production side.
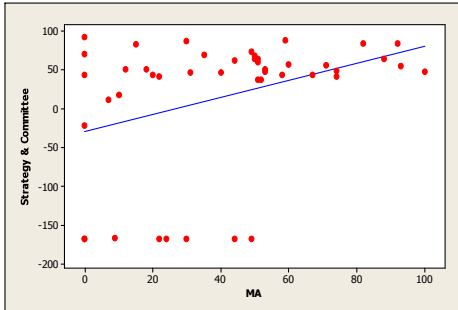
Fig. 10. MA and Strategy & Committee correlation-

C.   Information security

91% of the analyzed organizations have achieved level 1 and 4 % achieved level 2 on security information. In general security assessments demonstrate that security aspects are under considerations despite the scarce management involvement for security definition activities.

Information security management should be integrated with quality management system in order to derive standards, procedures and security controls from the management level. These elements will be spread within the organization.

4   Enabler tools

All these improvements initiatives require some reference tools for helping organizations to adopt an improvement discipline. This section suggests some supporting tools for each knowledge area: Apart from Minitab [20] used as tool for the analysis of statistical data, we suggest the following tools based on our experience:

•   Business area: we use EFQM [11] for analyzing business risks, investment factors, etc.

•   CMMI®: Eclipse Process Framework [21] is used as a tool for defining software processes. Due to its low cost it is recommended to be used for software process definition.

•   Information security: we have developed a questionnaire based tool for analyzing the related standards. It is an excel-based tool.

All these tools are not mandatory for improving adopting the multimodel framework. However they enable the analysis of the organizations and they provide a path for a continuous improvement in a holistic strategy.

5   Conclusions

All organizations require a degree of innovation in their business in order to overcome the global crisis. Some of them can innovate through quality improvement. Dr. Kao says in its seminal book about innovation "Innovation Nation" [19]:

"My own definition of innovation is both integrative and aspirational. I define it as the ability of individuals, companies, and entire nations to continuously create their desired future".

Our multimodel framework allows a certain degree of innovation combining the benefits of different models. In addition as stated in [18]: "it is impossible to determine if a person's products are innovative if they have never been seen, used, or evaluated" [18]. Therefore we have performed 47 assessments on multimodel environments. Based on the industrial question defined during the introduction we can argue that:

IQ1: We have applied our previously defined multimodel framework [13] in 47 industry contexts.

IQ2. We have provided a summary of the assessments from our 47 cases.

IQ3: The assessment results are shown in this paper where some correlations between aspects occur systematically.

Companies require new ways for assessing quality models and this multimodel framework tested in this paper integrates three relevant areas for SME: business, security, and software and systems areas.Nowadays organizations need not only to improve technical processes in order to obtain better business results but also business and information security aspects. All improvements initiatives must be aligned with business and strategic objectives, and this method allows this synchronization.

The resulting data analysis comes from 47 improvement initiatives deploying this approach. As result 89% of organizations achieved a good quality level. The cornerstones of this multimodel framework are:

- • Development of a clear business strategy and an appropriate business plan

- • Definition of development and production processes related to other business areas.

- • Development of a measurement framework and analysis supporting organizations and project management

- • Carrying out quality assurance activities.

- • Involvement of the management during decision making in technical and management processes

Currently we are still gathering more data from other ongoing deployments of this method. From a technical point of view we are developing a tool set providing support to the process definition and execution. This tool set allows metric definition on processes and deployment facilities within organizations.

6   References

1. Cater-Steel, Aileen; Tan, Wui-Gee; Toleman, Mark. "Challenge of adopting multiple process improvement frameworks". 2010 Second International Conference on Computer Engineering and Applications
2. Aliza Abdul Latif, Marina Md Din, Rozita Ismail "Challenges in adopting and integrating ITIL and CMMi in ICT Division of a Public Utility Company". 2010 Second International Conference on Computer Engineering and Applications
3. Ivan Rozman, Romana Vajde Horvat, Jozsef Gyorkos, Marjan Hericko "PROCESSUS – Integration of SEI CMM and ISO quality models". *Software Quality Journal* **6**, (1997) 37–63
4. Pat Kirwan, Jeannine Siviy John Morley, and Lisa Marino "Maximizing your Process Improvement ROI through Harmonization" http://goo.gl/1oog7
5. Jeannine Siviy, Pat Kirwan, Lisa Marino, and John Morley. "The Value of Harmonizing Multiple Improvement Technologies: A Process Improvement Professional's View" May 2008 http://goo.gl/PIBwT
6. Jeannine Siviy, Pat Kirwan, Lisa Marino, and John Morley. "Strategic Technology Selection and Classification in Multimodel Environments". http://goo.gl/WJXtV
7. Pat Kirwan, Jeannine Siviy, Lisa Marino, John Morley." Improvement Technology Classification and Composition in Multimodel Environments". http://goo.gl/iSaZR
8. Pat Kirwan, Jeannine Siviy, Lisa Marino, John Morley. "Process Architecture in a Multimodel Environment". http://goo.gl/l6a9cSa
9. Pat Kirwan, Jeannine Siviy, Lisa Marino, John Morley." Implementation Challenges in a Multimodel Environment". http://goo.gl/AcUIM
10. Maria Teresa Baldassarre, Danilo Caivano, Francisco J. Pino, Mario Piattini, Giuseppe Visaggio. "Harmonization of ISO/IEC 9001:2000 and CMMI-DEV: from a theoretical comparison to a real case application" Software Qual J DOI 10.1007/s11219-011-9154-7
11. EFQM http://www.efqm.org/en/
12. Paulk M. A History of the Capability Maturity Model for Software. http://goo.gl/q3CsK

13. Larrucea X., Santamaria I., Panaroni P. A Harmonized Multimodel Framework for Safety Environments. Systems, Software and Services Process Improvement. 19th European Conference, EuroSPI 2012, Vienna, Austria, June 25-27, 2012. Proceedings. DOI: 10.1007/978-3-642-31199-4_11.

14. Edit Lukacs."The economic role of SMEs in the world economy, especially in Europea" European Integration Studies, Miskolc, Volume 4. Number 1. (2005) pp. 3-12.

15. ITMark. European Software Institute. http://www.esi.es/index.php?op=15.1.2

16. Mary Beth Chrissis, Mike Konrad, and Sandy Shrum, "CMMI®: Guidelines for Process Integration and Product Improvement" (2nd Edition) The SEI Series in Software Engineering, 2006.

17. ISO/IEC 17799:2005 (27002) - Information technology — Security techniques — Code of practice for information security management, 2nd edition 2005-06-15

18. Frans Johansson. The Medici Effect. Harvard Business School Press. ISBN: 1-4221-0282-3

19. John Kao. "Innovation Nation: : How America Is Losing Its Innovation Edge, Why It Matters, and What We Can Do to Get It Back" Free Press 2007. ISBN: 13-978-1-4165-3262-2

20. Mintab www.minitab.com

21. Eclipse Process Framework. http://www.eclipse.org/epf

Towards measuring the impact of the SPI Manifesto: a systematic review

Mary Luz Sánchez Gordón[1], Ricardo Colomo-Palacios[1], Antonio Amescua Seco[1],

[1]Universidad Carlos III de Madrid, Spain,
mary_sanchezg@hotmail.com, rcolomo@inf.uc3m.es, amescua@inf.uc3m.es

Abstract

EuroSPI initiative (European System & Software Process Improvement & Innovation Service) is celebrating its twentieth anniversary in 2013. One of the main outputs of this initiative is the SPI Manifesto, a publication that gathers experience from various contributors in the area of process improvement. The aim of this paper is to analyze the impact of publications by means of a systematic literature review of the SPI Manifesto. This analysis was conducted using the most significant bibliographic databases. The result of the analysis reflects the growing influence of the initiative detected in the gradual appearance of scientific papers dedicated to the SPI Manifesto in research journals.

Keywords

SPI Manifesto, systematic literature review, software engineering, EuroSPI

1 Introduction

Today, firms have become IT usage-dependant organizations seeking to have increasingly efficient and innovative technological services and solutions (Lucio-Nieto, Colomo-Palacios, Soto-Acosta, Popa, & Amescua-Seco, 2012). In this scenario, software plays a crucial role in improving the productivity and efficiency of development activities and competitive strategies (Biro, Colomo-Palacios, & Messnarz, 2012). To satisfy the various requirements of a software process, companies have made a great effort in Software Process Improvement (SPI) (García et al., 2012). Thus, Software Process Improvement (SPI) is a systematic approach to increase the efficiency and effectiveness of a software development organization and to enhance software products (Unterkalmsteiner et al., 2012). The two most internationally used SPI models are Capability Maturity Model Integration (CMMI) and ISO/IEC 15504 (SPICE). These models define a process improvement approach that provide organizations with the essential elements to set process improvement goals, establish a point of reference for assessing current processes and support the improvement of their performance (Mesquida, Mas, Amengual, & Calvo-Manzano, 2012). However, although SPI initiatives have been around for many years, many companies are still experiencing SPI implementation problems (Niazi, 2012). One of the drawbacks of these models is their complexity (Kelemen, Kusters, & Trienekens, 2012; O'Leary & Richardson, 2012; Sulayman, Urquhart, Mendes, & Seidel, 2012).

The SPI manifesto (Pries-Heje & Johansen, 2010) describes the state-of-the-art of SPI. The SPI manifesto was edited and reviewed by a large group of international SPI experts back in 2009 to make sure that it would achieve broad acceptance. It is structured according to three Values and ten Principles as follows:

SPI must involve people actively and affect their daily activities

- Know the culture and focus on needs
- Motivate all people involved
- Base improvements on experience and measurements
- Create a learning organization

SPI is what you do to make your business successful

- Support the organization's visions and objectives
- Use dynamic and adaptable models as needed
- Apply risk management

SPI is inherently linked with change

- Manage the organizational change in your improvement effort
- Ensure all parties understand and agree on process
- Do not lose focus.

The impact of SPI in the literature in these three years is notable e.g. (Korsaa et al., 2012, 2013); however, to the best of the authors' knowledge, there is no formal literature review on the impact of SPI Manifesto in the research field. Thus, the aim of this paper is to review the literature in order to understand the implications of the SPI Manifesto with the objective to detect potential improvements and developments in the area.

2 Research Methodology

2.1 Motivation and Objectives

The literature presents a lack of studies on the applications of the SPI Manifesto. In fact, exploring previous research shows that a comprehensive systematic review does not exist on the topic. Therefore, this study will facilitate the understanding of the current status of research in different areas and address further investigation.

2.2 Research Method

In order to achieve an overview of the state of the art, the research was carried out following Kitchenham and Charters' (2007) guidelines on Systematic Literature Review (SLR). An SLR is a defined as a methodical way to synthesize existing work in a manner that is fair. An SLR is a means of identifying, evaluating and interpreting all available research relevant to a definite research question or topic area or phenomenon of interest. After reviewing the literature on SLR for similar research objectives, it can be identified that there is no previously published search on the topic.

This section presents each step followed to carry out this systematic review study, based on the guidelines provided by Kitchenham and Charters (2007).

2.3 Planning

The goal of the study is to achieve an overview of the current status of the SPI Manifesto in scientific literature. An SLR protocol was adapted to describe the plan for the review. The protocol includes research background, research questions, search strategy, study selection criteria and procedures, data extraction, and data synthesis strategies to ensure that the study is undertaken as planned and reduce the possibility of researcher bias. In this review protocol, the whole study timetable was not decided from the beginning, but rather the actual timetable of the study and results produced were recorded as the study progressed.

2.4 Research questions

The research question is twofold:

1.   What is the impact of the SPI Manifesto in the scientific literature? and

2.   What areas of research are more influenced by the SPI Manifesto?

The keyword used to find an answer to the research question was: SPI Manifesto.

The results expected at the end of the systematic review were, among other things, to discover what surveys exist as well as to identify the implications of the SPI Manifesto in scientific literature. We also expected to see what applied research had been carried out on the topic.

2.5 Search strategy and search process

The search strategy includes search resources and search process. Each one of them is detailed as follows:

2.5.1. Search resources.

This study was planned to find all the literature available about the SPI Manifesto. The list of sources that the systematic review was conducted with is:

•   ScienceDirect, on the subject of Computer Science,

•   Wiley InterScience, on the subject of Computer Science,

•   IEEE Digital Library (http://ieeexplore.ieee.org),

•   ACM Digital Library (http://portal.acm.org) -,

•   SpringerLink (http://link.springer.com), and

- As grey literature, Google Scholar was explored.

In SpringerLink, two important conference proceedings were found: Software Process Improvement and Capability Determination (SPICE) and Systems, and Software and Services Process Improvement (EuroSPI). These publications were taken into consideration because they are the most important specialized events.

2.5.2. Search process

The overall search process is depicted in Figure 1 and is explained in what follows. First, the search string was applied in December 2012, returning 44 papers (in total). Irrelevant and duplicate papers were removed and a set of 32 unique papers remained. After this, the industrial proceedings conferences of EuroSPI (from 2010 to 2012) were manually searched in January 2013. In these books, five relevant papers were found. All of them were included in the list of papers identified, which showed that the initial search did not discover these important sources.

Second, the papers were reviewed based on titles, abstracts, conclusion, references and keywords, and then were classified into three different types:

- Relevant papers: if the paper satisfies one of the two inclusion criteria (explained in what follows).

- Process assessment papers: if the paper is related to the SPI Manifesto and is relevant.

- Excluded papers: other papers, which are not relevant to the topic.

When there was doubt about the classification of a paper, it was included in the relevant group, leaving the possibility of discarding the paper during the next phase when the full texts of the papers were studied.

Third, each full article was retrieved and read to verify its inclusion or exclusion. The reason for exclusion or inclusion in this third phase was documented.

Fourth, in order to check the consistency of the inclusion/exclusion decisions, a test-retest approach and re-evaluation of a random sample of the primary studies was made.
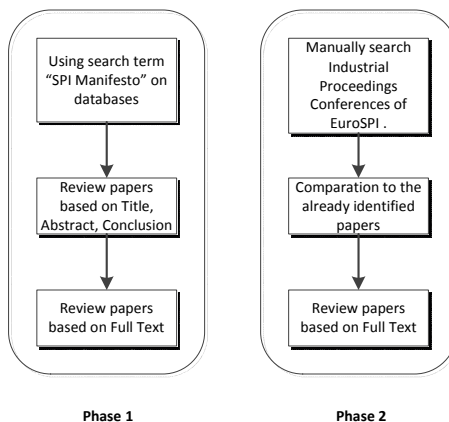

Phase 1   **Phase 2**

Figure 1. Search process description

Inclusion and exclusion criteria. A paper is kept in this mapping study if it satisfies one of the two criteria:

- The paper is explicitly related to the SPI Manifesto.

- The paper is relevant to software engineering research.

The papers were first reviewed based on titles, abstracts, and keywords and they were classified into three different types:

- Relevant papers: if the paper satisfies one of the two inclusion criteria.

- Process assessment papers: if the paper is related to the SPI Manifesto because it used as a reference.

- Excluded papers: other papers, which do not satisfy one of the two inclusion criteria.

The authors reviewed all 37 papers and put them into these different groups according to the previously mentioned criteria. This list was reviewed in order to check for inconsistencies. When there was disagreement about the classification of a paper, it was included in the relevant group, leaving the possibility to discard the paper during the next phase when the full papers were studied. The result of this stage was that 25 papers were classified as relevant.

There is a risk that some papers have been missed. Therefore, this study cannot guarantee completeness, but it can still be trusted to give a good overview of the relevant literature on the SPI Manifesto.

2.6. Data extraction

The data extracted from each paper was documented and kept in a reference manager. After identification of the papers, the following data was extracted:

- Source (journal or conference)

- Title

- Authors

- Publication year

- Classification according to topics in Table 3.

- Summary of the research, including which questions were solved

Based on the criteria for classifying papers, all the papers were reviewed, and the corresponding data was extracted.

To be able to analyze the 25 papers, there was a need to classify them in more ways than just according to the framework defined in Section 2. For this purpose, further criteria for classifying the papers were defined based on what information was available in the papers. When needed, the topics were updated or clarified during the classification process.

2.7 Results

The data required for analysis was extracted by exploring the full-text of each paper. Table 1 presents the results of the first phase and the source of the documents. Table 2 presents the results in phase 2.

Table 1. First phase results without filtering

ScienceDirect	0
Wiley InterScience	5
IEEE Digital Library	3
ACM Digital Library	1
SpringerLink	12
Google Scholar	24
Total	45
Total (without duplication)	32

Table 2. Second phase results

First phase results	32
EuroSPI Proceedings	5
Second phase excluded	12
Total	*25*

Table 3 includes the classification of papers with regards to the knowledge area.

Area	Topic	Relevant studies
Education	Training & Certifications	(Nevalainen & Schweigert, 2010), (Korsaa et al., 2010), (Korsaa et al., 2012), (Draghici, Draghici, Olariu, & Canda, 2012), (Martinez & Salviano, 2011)
	Dissemination	(McQuaid & Kasse, 2010), (Santana, Melo, Gusmão, & Goldman, 2012), (Kelemen, 2013)
Research	Provide evidence of the importance of the SPI Manifesto	(Biró, Korsaa, Nevalainen, Vohwinkel, & Schweigert, 2012), (Biró, 2012), (Unmüßig, 2011)
	Explain the importance of the current study	(Barreto & Rocha, 2010) , (Cuevas, Mejia, Munoz, & San Feliu, 2010), (Finnemann, Natorp, & Pries-Heje, 2012), (Siakas, Messnarz, Georgiadou, & Naaranoja, 2012), (Polgár & Biró, 2011)
	Validate the results of the study	(Salviano, 2011a), (Salviano, 2011b), (Kaynak & Karagoz, 2012), (Lopez & Garay, 2012)
	New contributions	(Kjær & Jørgensen, 2010), Gubaidullina, L., & Käkölä, T. (2010), (Koinig et al., 2011), (Peisl & Schmied, 2012),
	Generate further study proposals	(Buglione, 2011)

3 Topics influenced by the SPI Manifesto

3.1 Training

The first version of the SPI Manager Skills set and the training material was done originally by Sintef (Norway) in a European EQN project during 2005–2006. A completely new version of the SPI Manager Skills set has been under development in the EU Cert project since late 2008. Based on the ECQA idea of standardization by roles, skills and training, on one hand; and a vast amount of knowledge and experience in the software industry on the other hand, the SPI Manifesto and the SPI Manager training set a landmark in the ongoing improved professionalism of the industrialization of software development and service management (Nevalainen & Schweigert, 2010; Korsaa et al., 2010, 2012). Draghici et al. (2012) demonstrated how a qualification and certification program, like CertiBPM, developed and implemented with the support of the European Certification and Qualification Association (ECQA), can be used to better support the SPI efforts and strategies in Romanian companies. In these efforts the SPI Manifesto was present.

Martinez and Salviano (2011) presented a vision of process modeling and improvement with three types of models (Process Capability Profiles, Process Enactment Description and Process Performance Indicators). This view is then used as a basis for introducing Modeling driven Process Improvement, the worldview of PRO2PI Methodology (Process Modeling Profile to drive Process Improvement). They applied this method in an enterprise and, in its initial phase, it included training in software process improvement based on the SPI manifesto so that the participants could better understand it. On other hand, Salviano (2011b) presented a process improvement cycle that uses the SPI Manifesto and the PRO2PI Methodology to guide the process improvement.

3.2 Dissemination and recognition

McQuaid and Kasse (2010) presented the SPI Manifesto in a Software Quality Professional Magazine. Santana et al. (2012) wrote "*Melhoria de processo de software utilizando métodos ágeis e o modelo MPS.BR*", a book that describes, among other aspects, the SPI Manifesto.

Kelemen (2013) reported a review evolution of software process improvement in his PHD thesis and included the SPI Manifesto in it.

3.3 Provide evidence of the importance of the SPI Manifesto

Biró et al. (2012) presented a survey in which actual industrial surveys, whose results are analyzed and interpreted in general terms, link them to philosophical issues partly raised in recent initiatives like the SPI Manifesto, the ECQA PI Manager Certification and SEMAT. Furthermore, Biró (2012) presents a review of SPI from a historical perspective and includes the SPI Manifesto as an example of reconciliation between the "heavyweight" and agile ways to tackle the problem of software processes. Unmüßig (2011) points out that people are the key element for SPI, as is also drawn by the SPI Manifesto.

3.4 Explain the importance of the current study

Cuevas et al. (2010) analyze the results of the implementation of a project management software process improvement in an organization, using a methodology that allows using a multi-model environment as reference. In this paper the authors agreed with the SPI Manifesto about the need to use new processes defined through the improvement initiative.

Barreto and Rocha (2010) present the results of a survey that aimed to identify characteristics that can determine the similarity among software projects and also a measure to indicate the level of similarity among them, including some insights from the SPI Manifesto in its definition.

Finnemann, Natorp and Pries-Heje (2012) studied management support in IT project management and identified 16 categories of management support actions supporting their findings on the need for support of the SPI activities underlined in the SPI Manifesto.

Siakas et al. (2012) described the new certified Valorization Expert profession. This study considered the SPI Manifesto because it contains organizational success principles which directly support the dissemination and distribution of improvement ideas, human aspects and networked learning and growth.

Finally, Pólgar and Biró (2011) proposed the application of the usability methodology for SPI and included the SPI Manifesto among the methods to put it into practice.

3.5 Validate the results of the study

Salviano (2011a) proposed a modeling view of process and process improvement with three types of process models (Process Capability Profile, Process Enactment Description and Process Performance Indicator) and an example on a process improvement cycle. Furthermore, this author identified that the effort is consistent with the SPI Manifesto´s values and principles.

Kaynak and Karagoz (2012) described the lessons learned from the execution of the process improvement program in an IT solutions provider company. One of the key implementation practices was "Actively Involve People", and it was identified in accordance with the SPI Manifesto. This practice was implemented in order to overcome "Low motivation of the personnel".

Lopez and Garay (2012) uses an organizational model based on the *autopoiesis* theory to describe SPI as an enabler of the organizational change process and underlines that the SPI Manifesto suggests that standards and processes must fit an organization's real needs.

3.6 New contributions

Kjær and Jørgensen (2010) presented the Mjølner Company's SPI program. They used the three values of the SPI Manifesto and its ten principles as a basis for the evaluation of their current practices and as a guide for future improvements. They further reported the principles and values of the SPI Manifesto that have been implemented in Mjølner's software process.

Gubaidullina, L., & Käkölä, T. (2010) described a study in which the authors assess the extent to which the SPI Manifesto could be used to help providers to recover from situations in which CRM deployment projects are about to fail and to avoid similar problems proactively in future.

Koinig et al. (2011) reported a first mapping between SPI and Social Responsibility based on the SPI Manifesto and the ISO 26000 standard. They suggested that Social Responsibility concerns should be considered an integral part of SPI because interaction in both directions has been found.

Finally, Peisl and Schmied (2012) presented a conceptual framework to extend their previous research on innovation capability determination. They also proposed to adapt principles 3 and 8 of the SPI Manifesto considering some of the thoughts on transforming innovation.

3.7 Generate further study proposals

Buglione (2011) proposed Light Maturity Models (LMM), a feasible way to apply the 'maturity model' concept in a 'light' way for agile applications and/or organizations. In this work, the author suggested the possibility of building LMM by reinforcing the 12 principles in the Agile Manifesto with the SPI Manifesto.

4 Summary of results

The relevance of the SPI manifesto leads to consider two major areas: education (training and dissemination) and research (provide evidence of the importance of the SPI Manifesto, explain the importance of the current study, validate the results of the study, new contributions and generate further study proposals). Out of a total of 25 relevant studies, according to Figure 2, studies can be classified as training (21%), explain the importance of the current study (21%), validate the results of the study (17%), generate new contributions (17%), dissemination (12%), provide evidence of the importance of SPI Manifesto (8%) and generate further study proposals (4%).
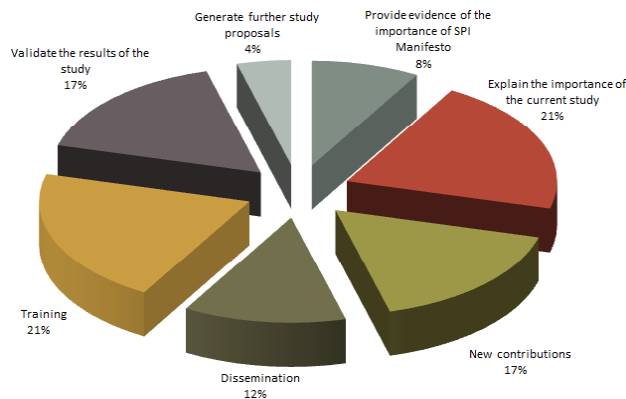

Figure 2: Distribution of papers in various topics.

Considering the results from various sources, according to Figure 3, the majority of the studies (72%) are distributed as follows: EuroSPI (40%) and other conferences (32%), other sources (28%), journals (20%), thesis (4%) and books (4%).
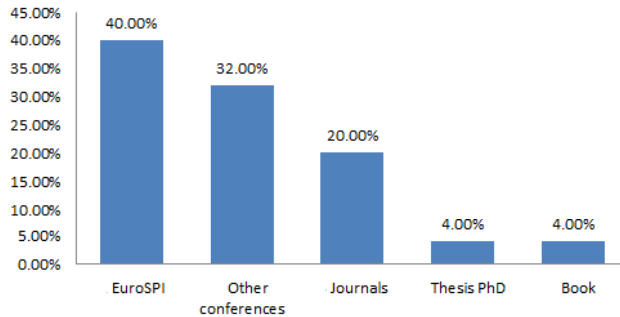
Figure 3: Distribution of research papers from various sources. Figure 3:

5 Conclusions, limitations and future work

This paper presents a systematic literature review of the SPI Manifesto. The rigor of the search process is one factor that distinguishes this review from traditional reviews. In order to perform an exhaustive search a variety of sources including, ACM, IEEE, Wiley InterScience and SpringerLink were inspected. On the other hand, manual searches were made for industrial conference proceedings of EuroSPI to complement results with works that are not listed in major databases.

The results show that, although the importance of the SPI Manifesto is increasing, this initiative is still deeply linked with the EuroSPI scenario and must find ways to expand its applicability in terms of publications outside this community and in terms of new applied research in industrial settings.

One limitation on this type of research is reliability. Therefore, three researchers were involved in this study. They discussed the reliability threats early in the design phase and agreed on the procedure, considering activities to mitigate the effect of one researcher's bias. Furthermore, due to the limited number of primary studies, evidence from all types of primary studies was considered.

In addition, given the exploratory nature of this paper, there are numerous lines of research that could be carried out based on the results obtained. One of them is to provide empirical evidence to support and validate the SPI Manifesto in order to expand its dissemination beyond the scientific field. Another line of research is to identify the relationships between the SPI Manifesto and the current software process improvement models.

6 Literature

Barreto, A. O. S., & Rocha, A. R. (2010). Analyzing the Similarity among Software Projects to Improve Software Project Monitoring Processes. In Quality of Information and Communications Technology (QUATIC), 2010 Seventh International Conference on The (pp. 441 –446). Presented at the Quality of Information and Communications Technology (QUATIC), 2010 Seventh International Conference on the. doi:10.1109/QUATIC.2010.79

Biró, M. (2012). Historical perspective on the state of software process improvement. Informatika, XIV. No. 2, 14–18.

Biro, M., Colomo-Palacios, R., & Messnarz, R. (2012). Editorial: European Systems and Software Process Improvement and Innovation (EuroSPI). IET Software, 6(5), 403–404. doi:10.1049/iet-sen.2012.0141

Biró, M., Korsaa, M., Nevalainen, R., Vohwinkel, D., & Schweigert, T. (2012). Agile maturity model -

Go back to the start of the cycle. In R. Messnarz, D. Ekert, M. Christiansen, J. Johansen, & S. Koinig (Eds.), Systems, Software and Services Process Improvement: 19th European Conference, EuroSPI 2012, Vienna, Austria, June 25-27, 2012, Industrial Proceedings (pp. 5.9–5.28). DELTA.

Buglione, L. (2011). Light maturity models (LMM): an Agile application. In Proceedings of the 12th International Conference on Product Focused Software Development and Process Improvement (pp. 57–61). New York, NY, USA: ACM. doi:10.1145/2181101.2181115

Card, D. N. (2004). Research directions in software process improvement. In Computer Software and Applications Conference, 2004. COMPSAC 2004. Proceedings of the 28th Annual International (p. 238 vol.1). Presented at the Computer Software and Applications Conference, 2004. COMPSAC 2004. Proceedings of the 28th Annual International. doi:10.1109/CMPSAC.2004.1342834

Cuevas, G., Mejia, J., Munoz, M., & San Feliu, T. (2010). Experiencia en la Mejora de Procesos de Gestión de Proyectos Utilizando un Entorno de Referencia Multimodelo. RISTI - Revista Ibérica de Sistemas e Tecnologias de Informação, (6), 87–100.

Draghici, A., Draghici, G., Olariu, C., & Canda, A. (2012). A Way to Support SPI Strategy through CertiBPM Training and Certification Program in Romania. In D. Winkler, R. V. O'Connor, & R. Messnarz (Eds.), Systems, Software and Services Process Improvement (pp. 276–287). Springer Berlin Heidelberg. Retrieved from http://link.springer.com/chapter/10.1007/978-3-642-31199-4_24

Finnemann, M., Natorp, L., & Pries-Heje, J. (2012). Designing Management Support for a Project. Proceedings / Information Systems Research In Scandinavia (IRIS), 2012.

García, I., Pacheco, C., Mendoza, E., Calvo-Manzano, J. A., Cuevas, G., & San Feliu, T. (2012). Managing the software process with a software process improvement tool in a small enterprise. Journal of Software: Evolution and Process, 24(5), 481–491. doi:10.1002/smr.504

Gubaidullina, L., & Käkölä, T. (2010). Using SPI Manifesto to Recover from CRM Deployment Project Failures and to Proactively Eliminate Similar Failures in Future: An Action Research Study in a Russian organization. International Research Workshop on IT Project Management 2010. Retrieved from http://aisel.aisnet.org/irwitpm2010/11

Kaynak, O., & Karagoz, A. (2012). Experience Report: Implementation of a Multi-Standard Compliant Process Improvement Program. In R. Messnarz, D. Ekert, M. Christiansen, J. Johansen, & S. Koinig (Eds.), Systems, Software and Services Process Improvement: 19th European Conference, EuroSPI 2012, Vienna, Austria, June 25-27, 2012, Industrial Proceedings (pp. 3.1 – 3.10). DELTA.

Kelemen, Z. (2013). Process based unification for multi-model software process improvement. Eindhoven University of Technology Library.

Kelemen, Z. D., Kusters, R., & Trienekens, J. (2012). Identifying criteria for multimodel software process improvement solutions – based on a review of current problems and initiatives. Journal of Software: Evolution and Process, 24(8), 895–909. doi:10.1002/smr.549

Kitchenham, B., & Charters, S. (2007). Guidelines for performing Systematic Literature Reviews in Software Engineering (version 2.3) (EBSE Technical Report EBSE-2007-01.). Keele University and Durham University. Retrieved from http://cdn.elsevier.com/promis_misc/525444systematicreviewsguide.pdf

Kjær, M. Y., & Jørgensen, J. B. (2010). Mjølner's Software Process Improvement: A Discussion and Strengthening Using the SPI Manifesto. In A. Riel, R. O'Connor, S. Tichkiewitch, & R. Messnarz (Eds.), Systems, Software and Services Process Improvement (pp. 222–232). Springer Berlin Heidelberg. Retrieved from http://link.springer.com/chapter/10.1007/978-3-642-15666-3_20

Koinig, S., Sicilia, M.-A., Messnarz, R., García-Barriocanal, E., Garre-Rubio, M., Siakas, K., & Clarke, A. (2011). Understanding the Relation of SPI and SR: A proposed Mapping of the SPI Manifesto to ISO 26000:2010. In R. O'Connor, J. Pries-Heje, & R. Messnarz (Eds.), Systems, Software and Services Process Improvement: 18th European Conference, EuroSPI 2011, Roskilde, Denmark, June 27-29, 2011, Industrial Proceedings (pp. 7.27 – 7.37). DELTA.

Korsaa, M., Biró, M., Messnarz, R., Johansen, J., Vohwinkel, D., Nevalainen, R., & Schweigert, T.

(2012). The SPI manifesto and the ECQA SPI manager certification scheme. Journal of Software: Evolution and Process, 24(5), 525–540. doi:10.1002/smr.502

Korsaa, M., Johansen, J., Schweigert, T., Vohwinkel, D., Messnarz, R., Nevalainen, R., & Biró, M. (2010). The people aspects in modern process improvement management approaches. In Systems, Software and Services Process Improvement: 17th European Conference, EuroSPI 2010, Grenoble Institute of Technology, France, Septiembre 1-3, 2010, Industrial Proceedings (p. n/a–n/a). Retrieved from http://onlinelibrary.wiley.com/doi/10.1002/smr.570/abstract

Korsaa, M., Biro, M., Messnarz, R., Johansen, J., Vohwinkel, D., Nevalainen, R., & Schweigert, T. (2012). The SPI manifesto and the ECQA SPI manager certification scheme. Journal of Software: Evolution and Process, 24(5), 525–540. doi:10.1002/smr.502

Korsaa, M., Johansen, J., Schweigert, T., Vohwinkel, D., Messnarz, R., Nevalainen, R., & Biro, M. (2013). The people aspects in modern process improvement management approaches. Journal of Software: Evolution and Process, 25(4), 381–391. doi:10.1002/smr.570

Lopez, J., & Garay, A. (2012). Software process improvement as an organizational self-production process. In Informatica (CLEI), 2012 XXXVIII Conferencia Latinoamericana En (pp. 1–10). Presented at the Informatica (CLEI), 2012 XXXVIII Conferencia Latinoamericana En, Medellin, Colombia. doi:10.1109/CLEI.2012.6427183

Lucio-Nieto, T., Colomo-Palacios, R., Soto-Acosta, P., Popa, S., & Amescua-Seco, A. (2012). Implementing an IT service information management framework: The case of COTEMAR. International Journal of Information Management, 32(6), 589–594. doi:10.1016/j.ijinfomgt.2012.08.004

Martinez, M., & Salviano, C. (2011). Método para Estabelecimento de Referências em Ciclos de Melhoria de Processo. In ANAIS DO SBQS 2011:X SIMPOSIO BRASILEIRO DE QUALIDADE DE SOFTWARE, 6 A10 DE JUNHO (pp. 167–182). Presented at the X Simpósio Brasileiro de Qualidade de Software - 2011 - Curitiba, PR, Curitiba, PR.

McQuaid, P., & Kasse, T. (2010). Software Process Improvement (SPI) Manifesto with Introduction. Software Quality Professional Magazine, Vol. 12 No. 3. Retrieved from https://secure.asq.org/perl/msg.pl?prvurl=http://rube.asq.org/software-quality/2010/06/software-quality/software-process-improvement-spi-manifesto.pdf

Mesquida, A. L., Mas, A., Amengual, E., & Calvo-Manzano, J. A. (2012). IT Service Management Process Improvement based on ISO/IEC 15504: A systematic review. Information and Software Technology, 54(3), 239–247. doi:10.1016/j.infsof.2011.11.002

Nevalainen, R., & Schweigert, T. (2010). A European scheme for software process improvement manager training and certification. Journal of Software Maintenance and Evolution: Research and Practice, 22(4), 269–277. doi:10.1002/spip.438

Niazi, M. (2012). An exploratory study of software process improvement implementation risks. Journal of Software: Evolution and Process, 24(8), 877–894. doi:10.1002/smr.543

O'Leary, P., & Richardson, I. (2012). Process reference model construction: implementing an evolutionary multi-method research approach. IET Software, 6(5), 423–430. doi:10.1049/iet-sen.2011.0195

Peisl, T., & Schmied, J. (2012). Innovating Innovation: A Conceptual Framework. In D. Winkler, R. V. O'Connor, & R. Messnarz (Eds.), Systems, Software and Services Process Improvement (pp. 217–228). Springer Berlin Heidelberg. Retrieved from http://link.springer.com/chapter/10.1007/978-3-642-31199-4_19

Polgár, P. B., & Biró, M. (2011). The Usability Approach in Software Process Improvement. In R. V. O'Connor, J. Pries-Heje, & R. Messnarz (Eds.), Systems, Software and Service Process Improvement (pp. 133–142). Springer Berlin Heidelberg. Retrieved from http://link.springer.com/chapter/10.1007/978-3-642-22206-1_12

Pries-Heje, J., & Johansen, J. (2010). SPI Manifesto. Retrieved from http://www.madebydelta.com/imported/images/DELTA_Web/documents/Ax/SPI_Manifesto_A.1.2.2010.pdf

Salviano, C. F. (2011a). A Modeling View of Process Improvement. In R. V. O'Connor, T. Rout, F. McCaffery, & A. Dorling (Eds.), Software Process Improvement and Capability Determination (pp. 16–27). Springer Berlin Heidelberg. Retrieved from http://link.springer.com/chapter/10.1007/978-3-642-21233-8_2

Salviano, C. F. (2011b). Process Improvement in an R&D&I Center Using Enterprise SPICE and SPICE for Research Models. In R. V. O'Connor, T. Rout, F. McCaffery, & A. Dorling (Eds.), Software Process Improvement and Capability Determination (pp. 204–207). Springer Berlin Heidelberg. Retrieved from http://link.springer.com/chapter/10.1007/978-3-642-21233-8_22

Santana, C., Melo, C., Gusmão, C., & Goldman, A. (2012). Melhoria de processo de software utilizando métodos ágeis e o modelo MPS.BR.

Siakas, K., Messnarz, R., Georgiadou, E., & Naaranoja, M. (2012). Launching Innovation in the Market Requires Competences in Dissemination and Exploitation. In D. Winkler, R. V. O'Connor, & R. Messnarz (Eds.), Systems, Software and Services Process Improvement (pp. 241–252). Springer Berlin Heidelberg. Retrieved from http://link.springer.com/chapter/10.1007/978-3-642-31199-4_21

Sulayman, M., Urquhart, C., Mendes, E., & Seidel, S. (2012). Software process improvement success factors for small and medium Web companies: A qualitative study. Information and Software Technology, 54(5), 479–500. doi:10.1016/j.infsof.2011.12.007

Unmüßig, A. (2011). The Human as one key element for holistic Software Process Improvement (SPI) - The Human being is the most interlinked element in SPI. In R. E. Messnarz, D. Ekert, M. Christiansen, J. Johansen, & S. Koinig (Eds.), Systems, Software and Services Process Improvement: 18th European Conference, EuroSPI 2011, Roskilde, Denmark, June 27-29, 2011, Industrial Proceedings (pp. 11.1 – 11.15). DELTA.

Unterkalmsteiner, M., Gorschek, T., Islam, A. K. M. M., Cheng, C. K., Permadi, R. B., & Feldt, R. (2012). Evaluation and Measurement of Software Process Improvement #x02014;A Systematic Literature Review. IEEE Transactions on Software Engineering, 38(2), 398–424. doi:10.1109/TSE.2011.26

7 Authors' CVs

Mary Luz Sánchez Gordón

Mary Luz Sánchez Gordón is a PhD student in the Computer Science Department at Universidad Carlos III de Madrid, Spain. She studied Computer Engineering at Universidad Central del Ecuador, Quito, Ecuador. She also got her master's in Education at this University. She has more than 10 years' experience in the software industry and five years in research and teaching in Ecuador. Her research interests are Software Process Improvement and Knowledge Management.

Ricardo Colomo-Palacios

Ricardo Colomo-Palacios is an associate professor at the Computer Science Department of the Universidad Carlos III de Madrid. His research interests include applied research in information systems, software project management, people in software projects and social and semantic web. He received his PhD in computer science from the Universidad Politécnica of Madrid (2005). He also holds an MBA from the Instituto de Empresa (2002). He has been working as a software engineer, project manager and software engineering consultant in several companies including INDRA, the Spanish IT leader. He is also an Editorial Board Member and Associate Editor for several international journals and conferences and Editor-in-Chief of the International Journal of Human Capital and Information Technology Professionals.

Antonio Amescua Seco

Antonio Amescua is a full professor in the Computer Science Department at Carlos III University of Madrid. He has been working as a software engineer in a public company (Iberia Airlines, now International Airlines Group) and in a private company (Novotec Consultores) as a software engineering consultant. He founded Progression, a spin-off company, in order to offer advanced Software Process Improvement services. He received his PhD in computer science from the Universidad Politécnica of Madrid. His research interests include Software Process Improvement, Software Reuse and Software Project Management.

Transition to a Product Line Infrastructure: Experience at Akari Software

Paraic Hegarty, Eoghan O'Leary, Danny Dineen, Akari Software, Ireland, phegarty@akarisoftware.com
David Bustard, Adrian Moore, University of Ulster, UK, dw.bustard@ulster.ac.uk

Abstract

Akari Software is a small software development company based in Cork, Ireland. It was created from its parent company, Digital Crew, in 2008 to take forward a family of software products supporting curriculum management in higher education. The work had started with one specific commission in 2006 but within four years, a small product suite had been developed, with ten organisations using customised variants of those products. In effect, a product family had been created but, unfortunately, this had occurred without the underlying structure and processes needed to ensure the efficient creation and maintenance of variants. This is a common issue faced by software companies as their product range expands. In 2010, Akari began making internal changes to develop a product range infrastructure that was suitable both for its existing products and its plans for future growth. Localisation was a particular concern, as the company wished to sell into other parts of Europe. The opportunity was also taken to switch to an agile model of software development; this had the effect of enhancing the production cycle but also facilitated a general programme of ongoing improvement. The purpose of this paper is to describe the product line transition that occurred, both to share the experience involved and to identify a number of general lessons learned. The work was supported by a FUSION Project, funded by InterTradeIreland, undertaken in collaboration with the University of Ulster.

1 Introduction

Software created through competitive tendering aims to meet the specific needs of one customer. Often, however, the development company will recognise an opportunity to market tailored variants to similar customers. If successful, a software product line (or family) is created [1, 2]. Unfortunately, there is a dilemma here for the development company: when should it commit to setting up the infrastructure necessary to support a product family efficiently? It is difficult to attempt such work at the outset, as the tendering process typically means that there is pressure to keep implementation costs to a minimum. Later, when seeking new business, costs again need to be kept down to attract potential customers, so there is little incentive to invest in infrastructure. As a result, the product family can grow without the underlying support that would facilitate the creation and maintenance of new variants.

The net indirect loss through inefficiency increases as the family is extended, as suggested in Figure 1 [3, 4]. The graph illustrates the basic economic argument involved: if there is investment in a product line infrastructure initially, the cost of each variant is less than that incurred with traditional development methods. With a small number of products, however, traditional methods may be cheaper but as the family size increases beyond a certain break-even point, the economic case for introducing a product line infrastructure becomes progressively harder to ignore.
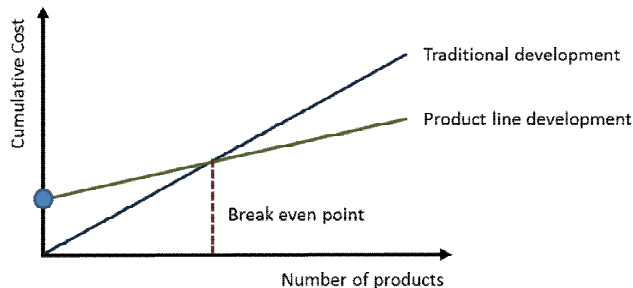
Figure 1: Relative costs of product line and traditional development (adapted from [3]).

The purpose of this paper is to describe the experience of one small company (12 staff), Akari Software, making the transition from traditional to product line development. The work involved both the creation of a product line infrastructure and the introduction of an internal improvement programme to support ongoing enhancement. The next section of the paper describes Akari, its curriculum management software and the internal challenges it recognised in 2010. A section then describes how Akari approached the transition to product line development. This required a number of internal changes, which were supported by a 12-month InterTradeIreland FUSION project [5], undertaken in collaboration with the University of Ulster. The paper concludes by bringing together a number of lessons learned from this transition and identifies work that is still in progress.

2 Initial Situation

Akari Software offers curriculum management software to higher education institutions to help them take an integrated approach to the design, delivery and ongoing evolution of curricula. The suite currently has four main elements:

- The core product is *Akari Document*, which helps build specifications for courses of study (programmes) and their individual components (modules), including the identification of module con-

tent, learning outcomes, reading lists, learning resources, delivery schedules, assessment schemes, module dependency and other relevant information.

- *Akari Publish* is a complementary tool that presents curricula information, either on a searchable public website or in printed form, for the benefit of staff, students, and other stakeholders. This is handled in a way that facilitates compliance with European Credit Transfer and Accumulation System (ECTS) requirements.

- *Akari Resource* helps with the management of human and capital resources used in the delivery of a curriculum, linking to HR, Payroll, and Facilities Management systems to determine delivery costs per student for both individual modules and full programmes.

- Finally, *Akari Review* (in development) will help manage the review of curricula in line with specific institution practices. This involves managing the workflow in approving changes or additions to a curriculum, to ensure compliance with defined procedures and standards.

As is evident from these descriptions, the elements of the suite are strongly interconnected but there is also substantial tailoring involved in producing an instance for a specific organisation. Examples include variations in the way that modules are defined, the procedures needed to make curricula changes, and the availability of local systems from which additional information can be extracted. The level of variation meant that initially it was easier to maintain separate systems for each customer institution. However, separate management became progressively less convenient as the number of customers increased, in line with the economic model illustrated in Figure 1.

Eventually, in 2010, Akari made the decision to restructure its software suite and associated development process. This acknowledged the need to improve the efficiency with which each new instance of the Akari suite could be created, deployed and maintained. In addition, as the company wished to extend its sales outside Ireland, the software had to be adjusted to accommodate a range of different European languages and other local variations. The next section describes the changes that were implemented.

3 Product Line Transition Strategy

It was recognised in 2008, when Akari Software was first created as a separate entity, that it would need to improve its infrastructure to meet its ambitious expansion plans. In principle, this could be handled through an incremental programme of change [6-8], managed in parallel with normal day-to-day activity. In practice, however, while such infrastructure work was clearly important, its priority always seemed to end up lower than that of immediate tasks, so little progress was made. It was decided, therefore, that a more formal initiative was needed and that, of the alternatives available, a FUSION project seemed most suitable [5]. Such a project would enable transitional changes to be analysed, implemented and evaluated separately from normal business, and then integrated when complete, as suggested in Figure 2.
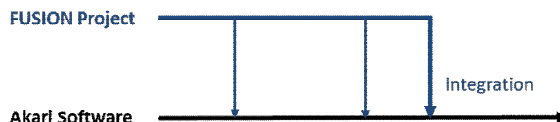

Figure 2: Product line transition scheme.

The funding obtained supported the appointment of a software developer for one year and consultancy input from academic staff at the University of Ulster. The final integration stage included the assimilation of the developer appointed to the project.

Preparing the FUSION proposal had the advantage of making explicit the specific product line changes required. However, preparing the proposal also made clear that a 12-month project with one developer would not be sufficient to complete the work. Some changes would have to be handled by

the company—both during the project and after it had finished. For example, it was more appropriate for Akari to take responsibility for the definition and establishment of the product configurations.

Clements and Northrup define a software product line as [1] "*a set of software-intensive systems sharing a common, managed set of features that satisfy the specific needs of a particular market segment, or mission and that are developed from a common set of core assets in a prescribed way*." The vision for Akari Software therefore, was to have an integrated collection of products, managed within a single repository—both to facilitate the ongoing maintenance of existing products and the creation and testing of new variants. The basic structure involved is summarised in Figure 3.
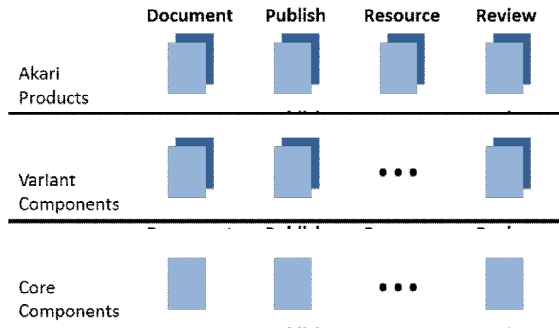
Figure 3: Akari product variant structure.

The top level of the diagram identifies the four Akari products in the suite, each of which exists in a range of variant forms. Each variant is made up of a set of components, some of which are part of the core platform and others that are specific to the variant. As an indication of scale, there are currently seven product variants, using hundreds of variant components on a platform of thousands of core components.

The FUSION project covered three specific pieces of work needed to help set up the desired product line structure:

- The refactoring of existing components to minimise language dependency. In particular, there was an immediate need to allow for languages other than English in the product interfaces. The initial target was a version supporting German.

- The introduction of test automation so that products could be re-evaluated quickly after changes were made. This was particular important, for example, for modifications made to the core components, as faults there were potentially wide-reaching.

- The introduction of deployment automation to facilitate the construction of variants for specific operating environments. It was expected that this would also include refactoring to introduce flexibility in the creation of new variants, based on selection from a range of defined options.

The work was tackled in the order shown, as this was expected to provide the best cost-benefit for the company. Another significant change tackled in parallel with the FUSION project was a move to agile development. The agile model was adopted to improve implementation practice, in general, but also had specific benefits for product line engineering. The sub-sections that follow discuss these initiatives in more detail.

3.1   Building an Integrated Product Repository

The first step in moving to a product line infrastructure is to integrate the product components within a configuration management system that allows individual products to be assembled as and when re-

quired. If variants have deviated significantly on separate development paths then there can be significant work required at this stage to factor out common elements so that variation is minimised. In the case of Akari, variations had been necessary to handle different databases, application servers and file systems in addition to customer-specific functionality.

Initially the Akari repository was built around Subversion but was later transferred to Git because of its distributed model and superior ability to manage branches and merges.

3.2   Language Localisation

The first piece of work tackled within the FUSION project was the adjustment of the product suite to handle different user interface languages. An important early decision was to make language selection both dynamic and easily accessible, so that a language switch could occur from any screen. Such flexibility was probably beyond what most users required but had benefits for testing and, most importantly, was impressive when demonstrating the suite to prospective customers. This was probably a factor, for example, in Akari later competing successfully for the development of a dual Irish-English curriculum management system at NUI Galway.

The immediate problem faced in refactoring the product suite to manage language dependency was the extent to which text was hardwired into the implementation. There were around a thousand individual instances of specific text being used directly in the code. Each section had to be modified and the suite retested thoroughly to ensure that no errors had been introduced. The approach taken was to put all text into a table that could be replaced quickly when a new language was selected.

The removal of direct text references was tedious and time-consuming but only covered part of the language dependency problem. Text was also used in tab labels so other language variants needed to be chosen carefully to avoid layout problems in the user interface. A more significant issue was that the product suite interface included fifty-four images with fixed language strings. Each had to be considered individually to decide if a specific image should be created for each language variant or the interface changed to avoid the variation. In practice, the interface was changed to remove the need for any language-specific images.

A more fundamental adjustment needed, however, was in the interface between the product suite and its operating environment. If, for example, the suite was running in an English-language environment but French had been selected for communication, then any prompts that would normally be issued by the environment, such as error messages, would have to be handled within the suite to avoid inconsistency. Bootstrap [9] was introduced to provide this level of control.

Overall, therefore, while language localisation seems straightforward in concept there were many messy time-consuming design and implementation details to handle, which meant that the work involved was significantly underestimated.

3.3   Test Automation

When the FUSION project began, variants of the Akari suite were mostly tested by hand, as is common with many web-based applications. Moving to a product line architecture required the introduction of test automation [10-12]. As a first step, however, it was necessary to produce a more precise definition of the Akari products and their variations. This turned out to be another slow time-consuming task but benefitted hand-testing immediately and had the effect of bringing out existing ambiguity and other deficiencies. Work on test automation was accelerated by the move to agile development as discussed in Section 3.5 below. This included the building of a suite of acceptance tests using CFSelenium [13] ,MXUnit [14], and Jasmine [15], with Jenkins, Bamboo and Ant as a unit test automation platform. Current work is focusing on load and compatibility/accessibility testing and the introduction of a test-driven approach to software coding.

3.4　Deployment Automation

The efficient management of variability across products is central to creating an effective product line infrastructure [16, 17]. The work on test automation identified that variability. This included standard issues such as variations in date and time formats and using local currency appropriate to the customer site. More specifically for Akari, however, there were variations in language and concepts across different educational institutions in relation to, for example, awards, credits and levels. The way in which choice was implemented had to be thought through carefully. Where possible, it seemed best to build the variability into the products and allow users to select their preferences. Like language selection (Section 3.2), however, this can involve substantial refactoring, with the costs that that entails. The move to agile adoption, as discussed in the next section, has clarified the range of desirable adjustments, which will be tackled as time and priorities permit. Work that has been completed so far includes support for groups of programmes (courses) and groups of modules (subjects); module teaching & learning strategies that vary by mode of delivery; the sharing of stages across multiple programmes; and the handling of multiple awards and awarding bodies per programme.

3.5　Agile Adoption

A recent paper looking specifically at the relationship between agile development and product line engineering [18], noted tentatively that although these approaches appeared to have conflicting values, there could be benefits overall. This seemed particularly true of organisations that were evolving and/or dealing with a volatile market, such as Akari. A survey of software development practices in Northern Ireland [19], which included companies taking a product line development, suggested that the benefits of an agile approach were so significant that there was little risk in its adoption for any organisation. Based on this evidence and the rate at which agile adoption was occurring [20], Akari made the decision to move immediately to agile development, in parallel with the product line transition. A 'total immersion' approach was used, using an external consultant as guide, and taking time out to train everyone in the company simultaneously. This helped build agreement on an implementation strategy, and the initial agile process was also designed during the training sessions.

The process that emerged was based on Scrum [21]. For support, it combined low-tech information radiators (sprint boards) with background technology (Greenhopper, with Jira for issue tracking). At each sprint retrospective, the process was reviewed and adjusted to suit the needs and wishes of the development team. Such changes ranged from the layout of information on the sprint boards and the definition of 'done', to more major adjustments, including greater emphasis on up-front design and a move to Scrumban [22] to introduce ideas from lean engineering.

One immediate benefit of agile adoption was in code productivity—though the greater volume of code produced also increased the number of errors handled. Efficiency was restored by separating out bug fixes from the development of new functionality. The current process enables the company to provide accurate effort, time and cost estimates to customers in response to change requests. This was achieved by taking feature requests through elaboration and planning, and then using a derived 3-month sprint efficiency measure (story points/man day) to estimate the effort. A more subtle additional benefit is that the broad approach to the introduction of agile thinking has led to the application of agile principles and processes in other functions of the business.

4　Conclusions

This paper has described one company's experience of moving to a product line infrastructure. Five aspects of the transition were considered individually: (i) building an integrated repository; (ii) language localisation; (iii) test automation; (vi) deployment automation; and (v) agile adoption. Some general lessons can be drawn from this work. In particular:

- *Adopt agile practices early in the process*: Despite reservations in the literature, Akari found immediate benefit in adopting an agile approach to product line development. The strategy of train-

ing everyone at the same time and having collective input to the design of the development process proved particularly effective—both in building cohesion and bringing the associated wordload to the surface. The gains suggest that earlier adoption would have been even better, as an agile structure is a general aid to organisational change.

- *Think ahead to avoid a steep transition*: The FUSION project helped kick-start the product line transition. It was useful for Akari to be aware of external funding opportunities so that such options could be taken into account when it became clear that a significant initiative was required. Ideally, however, it would have been better to avoid a step change by making smaller adjustments earlier. Again, this possibility seems more likely if agile development had been in use from the outset—with the agile approach there is an assumption of periodic refactoring, so problems are less likely to build up to a point where a major (risky and expensive) change is required.

- *Be flexible where practical* : Although there can be significant cost in setting up a product line infrastructure initially not all elements are expensive. In particular, isolating language dependency in software can be achieved at relatively low cost if taken into account at the outset. Effectively, this is just an example of good software engineering practice in anticipating where change may occur, and allowing for such future adjustment.

- *Maintain a single code base*: In essence, a product family starts with the second sale. Even if there is substantial variation involved, it seems important to keep the code together within the same repository and have the variation managed through configuration control. This is again just good development practice and little different from the management of successive versions of a product as it evolves.

Overall, although the work involved was significantly underestimated, and is still in progress, the transition to a product line infrastructure has been very successful for Akari Software. The adoption of the agile approach to development was a major factor in that success and seems a highly desirable first step in product-line implementation, in general.

The work at Akari has also illustrated the importance of product-line development to companies of all sizes, even though historically the emphasis has been on relatively large organizations that develop products of substantial size and variation.

Work on further advances in the field continues to be funded [23] though the product-line concept is now well established in software and systems engineering, as indicated by an associated international standard (ISO/IEC 26550) that is shortly to appear [24].

5 Acknowledgements

6 Literature

1. Clements, P., & Northrop, L. (2002). Software product lines. Addison-Wesley.

2. Pohl, K., Bockle, G., & Van Der Linden, F. (2005). Software product line engineering (Vol. 10). Springer.

3. Schmid, K., & Verlage, M. (2002). The economic impact of product line adoption and evolution. Software, IEEE, 19(4), 50-57.

4. Boehm, B., Brown, A. W., Madachy, R., & Yang, Y. (2004, August). A software product line life cycle cost estimation model. In Empirical Software Engineering, 2004. ISESE'04. Proceedings. 2004 International Symposium on (pp. 156-164). IEEE.

5. InterTradeIreland, *www.intertradeireland.com*.

6. van der Linden, F. J., Schmid, K., & Rommes, E. (2007). Software product lines in action: the best industrial practice in product line engineering. Springer.

7. Kircher, M., Schwanninger, C., & Groher, I. (2008). Transitioning to a software product family approach-challenges and best practices. In Software Product Line Conference, 2006 10th International (pp. 9-pp). IEEE.

8. Zhang, G., Shen, L., Peng, X., Xing, Z., & Zhao, W. (2011, September). Incremental and iterative reengineering towards Software Product Line: An industrial case study. In Software Maintenance (ICSM), 2011 27th IEEE International Conference on (pp. 418-427). IEEE.

9. Bootstrap, *http://twitter.github.com/bootstrap/*.

10. McGregor, J. (2001) Testing a Software Product Line. Software Engineering Institute. Paper 630. http://repository.cmu.edu/sei/630.

11. Pohl, K., & Metzger, A. (2006). Software product line testing. Communications of the ACM, 49(12), 78-81.

12. Lamancha, B. P., Usaola, M. P., & Velthius, M. P. (2009). Software Product Line Testing. A Systematic Review. ICSOFT (1), 23-30.

13. Selenium, *http://docs.seleniumhq.org/*.

14. MXUnit, *http://mxunit.org/*.

15. Jasmine, *http://pivotal.github.com/jasmine/*.

16. Metzger, A., & Pohl, K. (2007, May). Variability management in software product line engineering. In Companion to the proceedings of the 29th International Conference on Software Engineering (pp. 186-187). IEEE Computer Society.

17. Babar, M. A., Chen, L., & Shull, F. (2010). Managing variability in software product lines. Software, IEEE, 27(3), 89-91.

18. Hanssen, G. K. (2011). Agile software product line engineering: enabling factors. Software: Practice and Experience, 41(8), 883-897.

19. Bustard, D., Wilkie, F.G. & Greer, D. (2013, April). The Maturation of Agile Software Development Principles and Practice: Observations on Successive Industrial Studies in 2010 and 2012. In Engineering of Computer Based Systems (ECBS), 2013 IEEE 20th International Conference and Workshops on (to appear). IEEE.

20. West, D., Grant, T., Gerush, M., & D'Silva, D. (2010). Agile development: Mainstream adoption has changed agility. Forrester Research.

21. Schwaber, K. (2009). Agile project management with Scrum. Microsoft Press.

22. Ladas, C. (2009). Scrumban-essays on kanban systems for lean software development. Modus Cooperandi Press.

23. European Commission (2013). EU Research Projects, *http://cordis.europa.eu/projects/*.

24. ISO/IEC (2013). FDIS 26550. Software and Systems Engineering - Reference model for product line engineering and management.

7 Author CVs

Paraic Hegarty

Paraic Hegarty is CTO of Akari Software and fulfills the Scrum Product Owner role. His technical interests include ontologies, software development methodologies and the integration of business process mapping and software engineering. He was Fusion Consultant and later Company Supervisor for the Akari Software / University of Ulster Fusion project.

Eoghan O'Leary

Eoghan O'Leary is CEO of Akari Software. He has a particular interest in the application of technologies to boost business performance and is also the joint holder of a number of patents in the prevention of on-line fraud. He was Company Supervisor and later Chair of the Project Management Group for the Akari Software / University of Ulster Fusion project.

Danny Dineen

Danny Dinneen is a Software Developer and QA Engineer with Akari Software. He was the Fusion Graduate on the Akari Software / University of Ulster Fusion project and now has responsibility for the QA infrastructure and process in Akari.

David Bustard

David Bustard is a Professor of Computing Science at the University of Ulster. His main area of teaching and research is in Software Engineering and his current interests include agile software development, enterprise agility, systems analysis and self-managing systems. Most of this work is in collaboration with partners in the software industry.

Adrian Moore

Adrian Moore is a Senior Lecturer in Computing at the University of Ulster. He has broad interests across teaching, research and enterprise, particularly in web technologies and multimedia applications programming. He was the acadamic lead for the Akari Software / University of Ulster Fusion project.

A Project Management improvement programme according to ISO/IEC 29110 and PMBOK®

Antoni Lluís Mesquida and Antonia Mas

University of the Balearic Islands, Palma de Mallorca, Spain
{antoni.mesquida, antonia.mas}@uib.es

Abstract

This paper describes the results obtained and the lessons learned from the implementation of the ISO/IEC 29110 standard in four small software development companies, clustered in a joint process improvement programme. The main results of this initiative have been on the one hand, the definition of a standardized set of processes and procedures and, on the other, the development of a Process Asset Library to support project management good practices within the participant organizations. The PMBOK® Guide was used to complement the ISO/IEC 29110-5-1-2 standard, in order to provide detailed knowledge about project management inputs, outputs and best techniques.

Keywords

Software Process Improvement (SPI), Project Management, Very Small Entities (VSE), Process Asset, ISO/IEC 29110, PMBOK®.

1 Introduction

This paper presents the obtained results and the lessons learned from the implementation of the ISO/IEC 29110 standard in four VSEs. The initial aim of the project was to support a set of software organizations, both in the definition of the best practices that should follow their software development processes, and in the management of their projects.

In order to optimize resources and reduce the costs of consultancy, generally difficult to afford for this kind of companies, a programme involving four organizations was created. All these organizations belong to TurisTEC (Cluster of companies and institutions dedicated to the production and implementation of technological solutions for the Tourism Industry).

Initially, these companies raised no restriction on the adoption of one or other standard since the certification was not a short-term goal for them. Thus, given that the main interest of the companies was to improve internally, we should propose a set of best practices to be incorporated into their projects. From our experience in implementing process improvement programs in SMEs [1, 2, 3] and, given the nature of these organizations [4], we suggested to adopt the ISO/IEC 29110 international standard for software process improvement, since it is specific for VSEs and completely fitted their needs [5, 6].

Moreover, if they wish to opt for the certification in the future, they would be already aligned with a recognized international standard.

It was decided to begin working on the production processes, that is, the software implementation process activities. Since the companies participating in this programme had already showed their interest in process improvement and taken part in previous quality actions, they did already perform many of the tasks proposed by this process. Thus, regarding to implementation processes, our work was limited to complete/add some best practices, improve some assets and create some new more. However, none of these four companies had dedicated resources to management. Therefore, the main goal of our work supporting these organizations was focused on establishing the project management processes. For this reason, in this paper we only present the results related to the implementation of project management process.

In order to define the set of project management processes and assets that could be used by the organizations in all their projects, the following standards were used:

- ISO/IEC 12207:2008, which establishes a common framework for all the software life cycle processes. The process map of this standard is very wide and it may be too ambitious for this kind of organizations.

- ISO/IEC 29110-5-1-2:2011, specific standard for software development companies up to 25 employees that includes a subset of ISO/IEC 12207 processes adapted to the particular needs of these organizations. It is structured in only two processes, *Software Implementation* and *Project Management*.

- The Project Management Body of Knowledge (PMBOK[®]) Guide, specific for project management processes. It has been promoted and developed by the Project Management Institute (PMI[®]), a not-for-profit professional association which primary goal is to advance the practice, science and profession of project management.

The paper is organized as follows. Section 2 briefly describes the international standards used in this work, ISO/IEC 29110-5-1-2 and the PMBOK[®] Guide. Section 3 presents the existing relation between these two standards. Section 4 shows the Process Asset Library developed for supporting the deployment of ISO/IEC 29110-5-1-2 Project Management process. Section 5 discusses the lessons learned from its application in the companies. Finally, conclusions are summarized in Section 6.

2 Background

2.1 ISO/IEC 29110-5-1-2

The ISO/IEC 29110-5-1-2 standard [7] is aimed at providing a management and engineering guide which is applicable to the vast majority of small and very small entities that do not develop critical software. This standard defines two processes, *Software Implementation* and *Project Management*. The purpose of the *Software Implementation* process is the systematic performance of the analysis, design, construction, integration and tests activities for new or modified software products according to the specified requirements. The purpose of the *Project Management* process is to establish and carry out in a systematic way the tasks of the software implementation project, which allows complying with the project's objectives in the expected quality, time and cost.

The *Project Management* process has the four following activities:

- PM.1 Project Planning

- PM.2 Project Plan Execution

- PM.3 Project Assessment and Control

- PM.4 Project Closure

The ISO/IEC 29110-5-1-2 *Project Management* process is related with a subset of ten ISO/IEC 12207 [8] processes. Table 1 shows the relations between the objectives of the ISO/IEC 29110-5-1-2 *Project Management* process and these ten software life cycle processes.

Table 1: ISO/IEC 12207 processes related to ISO/IEC 29110-5-1-2 Project Management process

Objectives of the ISO/IEC 29110-5-1-2 Project Management process	ISO/IEC 12207 processes
PM.O1. The *Project Plan* for the execution of the project is developed according to the *Statement of Work* and reviewed and accepted by the Customer. The *Tasks* and *Resources* necessary to complete the work are sized and estimated.	Project Planning Process Measurement Process
PM.O2. Progress of the project is monitored against the *Project Plan* and recorded in the *Progress Status Record*. Corrections to remediate problems and deviations from the plan are taken when project targets are not achieved. Closure of the project is performed to get the Customer acceptance documented in the *Acceptance Record*.	Project Assessment and Control Process Measurement Process Software Acceptance Support Process Software Problem Resolution Process
PM.O3. The *Change Requests* are addressed through their reception and analysis. Changes to software requirements are evaluated for cost, schedule and technical impact.	Software Requirements Analysis Process
PM.O4. Review meetings with the Work Team and the Customer are held. Agreements are registered and tracked.	Software Review Process
PM.O5. *Risks* are identified as they develop and during the conduct of the project.	Risk Management Process Software Review Process
PM.O6. A software *Version Control Strategy* is developed. Items of *Software Configuration* are identified, defined and baselined. Modifications and releases of the items are controlled and made available to the Customer and Work Team. The storage, handling and delivery of the items are controlled.	Software Configuration Management Process
PM.O7. Software Quality Assurance is performed to provide assurance that work products and processes comply with the *Project Plan* and *Requirements Specification*.	Software Quality Assurance Process

The ISO/IEC 29110-5-1-2 standard has adapted these ten processes to the needs of small and very small entities to facilitate the introduction of project management best practices. More concretely, and as expected, this standard covers the activities related to the triple constraint: scope, time and cost. Due to it is a standard adapted to small organizations, change management and risk management are addressed from a more general perspective. However, it does consider neither the issues related to the organizational management (human resources and infrastructure) nor the aspects related to the providers (procurement and supply).

2.2 The PMBOK® Guide

The Project Management Body of Knowledge (PMBOK®) Guide [9] is a collection of recognised good practices that are widely applied by project management professionals and practitioners for the successful management of projects around the world. The PMBOK® Guide provides guidelines for managing projects within an organization and also promotes a common vocabulary within the project management profession for discussing, writing, and applying project management concepts.

The project management good practices in the PMBOK® Guide cover the entire project lifecycle, from proposal to delivery, final acceptance and closing. The standard defines 47 project management processes which are grouped into five categories known as Project Management Process Groups: Initiating, Planning, Executing, Monitoring and Controlling, and Closing. Moreover, the PMBOK® Guide recognises ten Knowledge Areas typical of almost all projects:

- Project integration management

- Project scope management

- Project time management

- Project cost management

- Project quality management

- Project human resources management

- Project communications management

- Project risk management

- Project procurement management

- Project stakeholder management

3 Relation between ISO/IEC 29110-5-1-2 and the PMBOK® Guide

The first step of our work consisted on finding all the existing relations between the tasks of the ISO/IEC 29110-5-1-2 *Project Management* process and the PMBOK® Knowledge Areas in order to increase the knowledge about these tasks and also, to facilitate the development of organizational process assets to be used in all the projects within the company. As a first conclusion, we observed that the information of the processes of the following PMBOK® management areas should be considered and used during the development of our project management process asset library:

- *Project Integration Management*. This area includes the processes and activities of management, including: formally constitute a project, plan it, monitor it, control it and direct it to closure. The Integration Management is a cross-sectional area and basic, whatever the level of management applied. Therefore, and as might be expected, the specific standard for small and very small entities includes a reference to such activities.

- *Project Scope Management*, *Project Time Management*, *Project Cost Management*. Nor is it surprising that ISO/IEC 29110-5-1-2 considers the processes related to the triple constraint: define the work required to complete the project, subdivide it into smaller components (project activities), develop the schedule and estimate both the resources and the cost needed to carry out the project.

The ISO/IEC 29110-5-1-2 standard includes the identification of the roles, responsibilities and skills required for the necessary profiles of the project team, and the selection of the most appropriate to each profile. In contrast, the standard does not cover other tasks of *Project Human Resource Management*, such as training and team member's performance control.

ISO/IEC 29110-5-1-2 considers the identification of risks that may affect the project and their control during the project execution, but in a very general way, without specifically mentioning the tasks related to the quantitative and qualitative risk analysis addressed by the PMBOK® Guide area *Project Risk Management*.

The project management good practices of the following PMBOK® Knowledge Areas are not considered by the ISO/IEC 29110-5-1-2 standard:

- *Project Quality Management*. While ISO/IEC 29110-5-1-2 considers as aspects of quality assurance the verification, validation and review of the products obtained, the PMBOK® *Project Quality Management* area refers to the processes and activities that establish responsibilities, objectives and quality policies for continuous improvement of the processes carried out throughout the project.

- *Project Communications Management*. The standard does include neither the identification of the stakeholders' information needs nor the definition of the communication plan among all project stakeholders.

- *Project Procurement Management*. The standard does not cover any of the processes related to the purchase or acquisition of products, services or results to be obtained out of the project team.

- *Project Stakeholder Management*. The standard does not address any aspect related to the identification of the persons or organizations affected by the project, or with the documentation of relevant information regarding their interests, involvement and impact on project success.

4 Process Asset Library to support Project Management in SMEs

From the input and output products proposed by the ISO/IEC 29110-5-1-2 *Project Management* process, and from the inputs, techniques and outputs proposed by the selected processes of the PMBOK® Guide, a Process Asset Library to support the performance of project management tasks has been developed. The developed assets are categorized into four groups of assets (that correspond to the four ISO/IEC 29110-5-1-2 *Project Management* process activities): Project Planning, Project Execution Plan, Project Assessment and Control and Project Closure.

4.1 Project Planning assets

Table 2 shows the developed assets that can be used during the planning phase of the project. For each asset, the table shows the ISO/IEC 291105-1-2 *Project Management* process tasks that use or produce that asset (second column) and the processes of the PMBOK® Guide which techniques can be used to expand the knowledge necessary to use the asset (third column).

Table 2: Project Planning assets

Asset	ISO/IEC 29110-5-1-2 Project Management Tasks	PMBOK® Processes
Project Charter	PM.1.1 Review the *Statement of Work*.	4.1 Develop Project Charter 5.1 Plan Scope Management
WBS (Work Breakdown Structure)	PM.1.3 Identify the specific *Tasks* to be performed in order to produce the *Deliverables* and their *Software Components* identified in the *Statement of Work*. Include *Tasks* in the SI process along with verification, validation and reviews with Customer and Work Team *Tasks* to assure the quality of work products. Identify the *Tasks* to perform the *Delivery Instructions*. Document the *Tasks*.	5.4 Create WBS 6.1 Plan Schedule Management
Schedule	PM.1.4 Establish the *Estimated Duration* to perform each task. PM.1.7 Assign estimated start and completion dates to each one of the *Tasks* in order to create the *Schedule of the Project Tasks* taking into account the assigned Resources, sequence and dependency of the *Tasks*. PM.1.8 Calculate and document the project *Estimated Effort and Cost*.	6.2 Define Activities 6.3 Sequence Activities 6.5 Estimate Activity Durations 6.6 Develop Schedule
RBS (Resource Breakdown Structure)	PM.1.5 Identify and document the *Resources*: human, material, equipment and tools, standards, including the required training of the Work Team to perform the project. Include in the schedule the dates when *Resources* and training will be needed. PM.1.6 Establish the *Composition of Work Team* assigning roles and responsibilities according to the *Resources*.	6.4 Estimate Activity Resources 9.1 Plan Human Resource Management 9.2 Acquire Project Team
Budget **Customer Tender**	PM.1.8 Calculate and document the project *Estimated Effort and Cost*.	7.1 Plan Cost Management 7.2 Estimate Costs 7.3 Determine Budget
Project Plan **Mind Map**	PM.1.11 Generate the *Project Plan* integrating the elements previously identified and documented. PM.1.12 Include *Product Description*, *Scope*, *Objectives* and *Deliverables* in the *Project Plan*. PM.1.9 Identify and document the risks which may affect the project. PM.1.10 Document the *Version Control Strategy* in the *Project Plan*. PM.1.13 Verify and obtain approval of the *Project Plan*. Verify that all *Project Plan* elements are viable and consistent. The results found are documented in a *Verification Results* and corrections are made until the document is approved by PM. PM.1.14 Review and accept the *Project Plan*. Customer reviews and accepts the *Project Plan*, making sure that the *Project Plan* elements match with the *Statement of Work*. PM.1.15 Establish the *Project Repository* using the *Version Control Strategy*.	4.2 Develop Project Management Plan 5.2 Collect Requirements 5.3 Define Scope 11.2 Identify Risks

4.2 Project Plan Execution assets

Table 3 shows the assets that can be used to implement the documented plan on the project.

Table 3: Project Plan Execution assets

Asset	ISO/IEC 29110-5-1-2 Project Management Tasks	PMBOK® Processes
Progress Status Record	PM.2.1 Monitor the Project Plan execution and record actual data in Progress Status Record.	4.4 Monitor and Control Project Work
Change request	PM 2.2 Analyse and evaluate the *Change Request* for cost, schedule and technical impact. The *Change Request* can be initiated internally by the Customer or internally by the Work Team. Update the *Project Plan*, if the accepted change does not affect agreements with Customer. *Change Request*, which affects those agreements, needs to be negotiated by both parties (see PM.2.4).	4.5 Perform Integrated Change Management
Meeting Record	PM.2.3 Conduct revision meetings with the Work Team, identify problems, review risk status, record agreements and track them to closure. PM.2.4 Conduct revision meetings with the Customer, record agreements and track them to closure. Change Request initiated by Customer or initiated by Work Team, which affects the Customer, needs to be negotiated to reach acceptance of both parties. If necessary, update the Project Plan according to new agreement with Customer.	4.3 Direct and Manage Project Work

4.3 Project Assessment and Control assets

Table 4 shows the developed assets that can be used to evaluate the performance of the plan against documented commitment.

Table 4: Project Assessment and Control assets

Asset	ISO/IEC 29110-5-1-2 Project Management Tasks	PMBOK® Processes
Project Performance Report	PM.3.1 Evaluate project progress with respect to the *Project Plan*, comparing: actual *Tasks* against planned *Tasks*, actual results against established project *Objectives*, actual resource allocation against planned *Resources*, actual cost against budget estimates, actual time against planned schedule, actual risk against previously identified PM.3.2 Establish actions to correct deviations or problems and identified risks concerning the accomplishment of the plan, as needed, document them in *Correction Register* and track them to closure. PM.3.3 Identify changes to requirements and/or *Project Plan* to address major deviations, potential risks or problems concerning the accomplishment of the plan, document them in *Change Request* and track them to closure.	4.4 Monitor and Control Project Work 5.6 Control Scope 6.7 Control Schedule 7.4 Control Costs 11.6 Control Risks

4.4 Project Closure assets

Table 5 shows the assets to provide the project's documentation and products in accordance with contract requirements.

Table 5: Project Closure assets

Asset	ISO/IEC 29110-5-1-2 Project Management Tasks	PMBOK® Processes
Acceptance Record	PM.4.1. Formalize the completion of the project according to the *Delivery Instructions* established in the *Project Plan*, providing acceptance support and getting the *Acceptance Record* signed.	5.5 Validate Scope 4.6 Close Project or Phase
Historical information	PM.4.2 Update *Project Repository*.	4.6 Close Project or Phase

5 Lessons learned

In this section, the remarks we have made during the implementation of the ISO/IEC 29110-5-1-2 standard in the four companies in this clustered action are described. Regarding general aspects, we can state that:

- These small companies are fully devoted to their productive work and to solve their day-to-day survival problems. They are often unable and unwilling to devote time and efforts to define processes or assets. These companies do not have a quality department, unlike larger organizations, which is dedicated to these tasks. In small organizations, software engineers are more oriented to product, service or management instead of establishing new working practices.

- These companies were unaware of the existence of both the ISO/IEC 29110-5-1-2 and the ISO/IEC 12207 standards. Some employees knew the PMBOK® Guide but above all, they knew the existence of the PMP certification. In any case, all these standards seemed very complex and out of reach to them. We fully agree with [6] when they state that "… it is still needed often requiring intervention from software process consultants". We really think these companies need external consultancy that offers support in process development and improvement, issues that they generally unknown and considered utopian and distant.

- Small entities not only need to know what to do in order to improve their processes, but they need to have specific procedures describing in detail the work they have to perform, with a clear set of best practices and a set of assets that will help to carry them out. These procedures should be simple and applicable to the types of projects that they normally undertake.

- They spend very little effort to improve employee training and, when done, it is not according to a training plan, but as an ad-hoc action derived from a detected short-term need.

- These companies usually have no explicit procedures for the purchase or acquisition of products, services or results to be obtained outside of the company. Moreover, it has not been detected any best practice of the ISO/IEC 29110-5-1-2 *Project Management* process related to the PMBOK® Guide's Project Procurement Management area.

- While in the clustered action different ISO/IEC 29110-5-1-2 tasks for the identification of potential risks have been established, the truth is that it was not traditionally accustomed to performing risk management activities. Incidents were assumed and the companies used to react as they could.

Regarding ISO/IEC 29110-5-1-2 application aspects, we note that in our case:

- During the definition of the *Project Management* process, we had to establish a new mechanism for communication among all the stakeholders in a project. We observed that some of these practices were informally conducted but they were not defined in the organizations. Thus, although the ISO/IEC 29110-5-1-2 standard does not include the identification of the stakeholders' information needs and the definition of the communication plan, these good practices were considered in our improvement programme.

- There were as many project management methods as there were project managers in the organization. Since there was no an established procedure, each employee used the method he/she knew and that best suited to his/her way of working. In order to obtain a standard procedure, we had to consider their good practices and then agree on the most appropriate. It is well known the change resistance we all have. For this reason, and in order to generate a useful process, it is much better to reach a consensus than imposing it.

- Organizations need to have tools to become more efficient and productive in their daily work. Therefore, it was necessary to select, propose and agree a set of tools to support both collaborative work and project management tasks. Since some of the participant companies had already developed some internal software tool to support some of these aspects, each company selected the subset of tools that best suited their needs, from the list we had proposed: *Wiggio*, *TeamLab*, *Teambox*, *Zoho*, among others.

- Although it seems that ISO/IEC 29110-5-1-2 does not establish any specific practice related to the performance monitoring of team members, a core set of performance indicators and the way of monitoring them were established.

The valuation of this initiative for ISO/IEC 29110-5-1-2 implementation, pioneer in our country, is totally positive. As key strengths for the success of the project it is worth highlighting:

- The active participation, motivation and consciousness of all the participant companies.

- The willingness to share knowledge among companies. It is important to note that these companies are in the same sector and they sometimes compete to obtain a new project/client.

- The establishment of a detailed plan and its compliance with only some slight deviations. The programme had a very clear schedule with periodical monitoring in all the companies. Without these joint reviews and the understanding of the improvement actions to take by the team, the cost of the implementation of the standard would have been much higher.

Now from a technical point of view we could point out:

- The selection of a representative in each company. This person channeled the needs and requirements of his/her company and raised them to the meetings held between all the representatives and consultants. The agreements made in these meetings needed subsequent approval by each organization.

- The selection of a collaborative work tool (*TeamLab*) to support the communication and file-sharing between all the representatives and consultants.

6 Conclusions and Further Work

This paper has described the activities performed during a clustered plan of four small software development companies, which main objective was the implementation of the ISO/IEC 29110-5-1-2 standard. Since the *Software Implementation* process was already quite deployed in all the participant organizations, the article focuses on the *Project Management* process. The main result of this process improvement programme has been the definition of a standardized set of processes and procedures, and the creation of a library of process assets that are useful for the projects that are carried out within the participant organizations. The PMBOK® Guide was used to complement the ISO/IEC 29110-5-1-2 standard, in order to provide detailed knowledge about the inputs, outputs and best techniques for each *Project Management* task.

The implementation of the processes defined in the grouped plan has enabled the participant companies to increase or generate new knowledge on the projects carried out within the organization. It has represented a big innovation on their former way of working before incorporating these processes. Traditionally, the knowledge generated by projects used when developing new tenders, was only in the minds of project managers or the top manager (often also the company's owner). Now, the knowledge is transferred to the company. There is a basic but decisive set of indicators that enable the company to improve new tenders, optimize the work and, as a result, increase their competitiveness.

Regarding future actions, we want to mention that some of the actions identified at the beginning of the project need to change some aspect that has not provided the expected result when it has been applied in everyday projects. These future actions that are planned include the development of new assets, the changes in some of the existing ones and the addition of some new good practices to certain processes.

Although certification was not the main objective of the participant organizations, a company that has implemented the ISO/IEC 29110-5-1-2 standard, besides the fact that it will have met its goal of improving its internal operations, it may even choose to be certificated according to an international standard for software process improvement. We consider that it is more reasonable that these companies will select a standard created specifically for this type of organizations rather than a general purpose standard, such as ISO/IEC 15504 or CMMI. Because project management good practices must be implemented to increase the maturity of a company, this paper recommends the use of the ISO/IEC 29110-5-1-2 standard as a starting point for small and very small entities to establish contact with project management and/or organizational maturity.

Acknowledgements

7 Literature

[1] Amengual, E. and Mas, A.: Software Process Improvement in Small Companies: An Experience. EuroSPI 2007 Industrial Proceedings.

[2] Mas, A., Mesquida A.L., Fluxà B.: The long way to maturity: a road map to success. EuroSPI 2012 Industrial Proceedings.

[3] Mas, A.; Fluxà, B.; Amengual, E.: Lessons learned from an ISO/IEC 15504 SPI programme in a company. Journal of Software Maintenance and Evolution-Research and Practice. Volume 24, Issue 5, pages 493-500, August 2012.

[4] Lepmets, M., McBride, T.: Process Improvement for the Small and Agile. Systems, Software and Services Process Improvement. Communications in Computer and Information Science 301, pp. 310-318, EuroSPI 2012.

[5] O'Connor, R.V., Laporte, C.Y.: Software Project Management in Very Small Entities with ISO/IEC 29110. Systems, Software and Services Process Improvement. Communications in Computer and Information Science 301, pp. 330-341, EuroSPI 2012.

[6] Boucher, Q., Perrouin, G., Deprez, J.-C., Heymans, P.: Towards Configurable ISO/IEC 29110-Compliant Software Development Processes for Very Small Entities. Systems, Software and Services Process Improvement. Communications in Computer and Information Science 301, pp. 330-341, EuroSPI 2012.

[7] ISO/IEC TR 29110-5-1-2:2011 Software engineering - Lifecycle profiles for Very Small Entities (VSEs) - Part 5-1-2: Management and engineering guide: Generic profile group: Basic profile. International Organization for Standarization, Mayo 2011.

[8] ISO/IEC 12207:2008 Systems and software engineering - Software life cycle processes. International Organization for Standarization, Febrero 2008.

[9] A Guide to the Project Management Body of Knowledge (PMBOK® Guide) - Fifth Edition. Project Management Institute, Enero 2013.

8 Author CVs

Antoni Lluís Mesquida

Dr. Antoni Lluís Mesquida is an assistant lecturer in computer science at the University of the Balearic Islands. He is teaching in the area of Software Engineering. His research interests include software process improvement and service management. He has participated in the QuaSAR project, a software process improvement programme in software small companies at the Balearic Islands. He received his PhD in Computer Science from the University of the Balearic Islands. Contact him at Campus UIB, Edifici A. Turmeda. Ctra. Valldemossa, Km. 7,5. 07122 Palma de Mallorca, Spain. antoni.mesquida@uib.es.

Antonia Mas

Dr. Antonia Mas is a university lecturer of software engineering and project management at the University of the Balearic Islands. Her research interests include software process improvement, project management and service management. She has promoted and coordi-

nated the QuaSAR Project, a software process improvement initiative in small software companies at the Balearic Islands. She received her degree in Computer Science from UAB (Catalonia, Spain) and her PhD in Computer Science from the University of the Balearic Islands. She is a member of the Software Quality Group of ATI Spain from 1998 and she acts as the editorial board member in the REICIS journal. She has also served as program committee chair and program committee member of scientific conferences and workshops related to software quality. She is an ISO/IEC 15504 assessor and is focussed on assessing small companies. Contact her at Campus UIB, Edifici A. Turmeda. Ctra. Valldemossa, Km. 7,5. 07122 Palma de Mallorca, Spain. antonia.mas@uib.es.

Status of SPI Activities in Japanese Software – A view from JASPIC -

Hideto Ogasawara
Toshiba Corporation, 1, Komukai Toshiba-cho, Saiwai-ku, Kawasaki, 212-8582 Japan
hideto.ogasawara@toshiba.co.jp

Abstract

For the effective promotion of software process improvement (SPI) activities in the word, it is very important to establish a community beyond various social/organizational barriers. Like the auto-mobiles industry, to share various knowledge/experiences is to evolve one's own industry. To promote SPI activities in Japan, we established Japan SPI Consortium (JASPIC) in October 2000. In this paper, we describe the status of SPI practice in Japan through our experience in various activities in JASPIC, analyse current issues from software engineering point of view, and make some proposals for future action.

Keywords

Software process improvement, SEPG, Japan SPI Consortium (JASPIC)

1 Introduction

A number of software process evaluation frameworks, such as CMM, ISO/IEC 15504, etc., have been proposed and being used in many software organizations all over the world. Also, many companies have installed a functional unit called SEPG in their organization toward software process improvement. However, still there are number of companies who cannot be proceed step forward because of the shortage of human resources and/or the lack of sufficient technical/managerial knowledge necessary for SPI.

To overcome these difficulties, it is important for SEPG staff in various companies to have opportunity for exchanging information beyond the boundary of organizations. Japan SPI Consortium (JASPIC) was established as a social mechanism to promote this kind of mutual exchange among SEPG people in Japanese software industry.

2 Birth of JASPIC

The keyword "Process Improvement" became popular in Japanese software community since the translation of Watts Humphrey's book "Managing the Software Process" was published in 1991 [3]. Around that time, many software companies were trying very hard to get ISO9000s certification, but some engineers, people like SPIN (Special Interest Group on Process) members in SEA (Software Engineers Association) have already moved their concern to CMM.

SEA is a volunteer organization established in December 1985 to provide a "place for software engineers or researchers, who are working in different environments such as software houses, computer manufacturers, computing service bureaus, universities, and research laboratories, to exchange their technical experience or knowledge freely beyond the barriers of existing social organizations. It has been conducted various activities to promote software engineering practice. One of the major technical events in SEA is annual Software Symposium (SS). SEA-SPIN was kicked off at the BOF session in SS-1996. In 1998, some of SEA-SPIN members volunteered to make official Japanese translation of SW-CMM Technical Reports TR-24 & 25 responding a request from CMU/SEI. They published the result open to public via official web page of SEA.

Then SEI released CMMI. Soon, strong needs for Japanese translations of various technical documents were emerged from the industry. But the large volume of translation task seems to be far beyond the work of small volunteer group. So, a new non-profit organization JASPIC was established in October 2000 as a joint effort of several SPI-sensitive companies.

3 History of Software Engineering in Japan

In this section, we summarize the history of software engineering in Japan especially about quality management aspects of the technology [1] [2].

3.1 Before 1980s

In this period, the major concern in software industry was quality assurance of large application systems like banking, railroad seat reservation systems, etc. The major players were mainframe computer manufactures. They have applied their know-how accumulated in quality management in hardware manufacturing into software development. The result was very much successful and known as world famous "software factory system".

Also, in this period, QC circle approach, which was well known in hardware manufacturing area, was introduced into software field. It was rather easy to incorporate various support tools in software development environments because each manufacturer was using their own operating systems as the basis of environment. Many creative engineers invented their own software quality tools or management style and practiced in their projects.

3.2   1980s

In this period, IT industry's major concern was the "expansion of a scale", because of high-speed growth of Japanese economy. Many large application systems like 3rd generation banking systems were developed, and a large number of engineers from software houses were involved into these projects under Japanese style project subcontracting mechanism.

In this situation, it was necessary for system integrators (main-framers) to train subcontractors about their system for project management and quality control. For this purpose, there were intensive effort of systematizing quality control and assurance methods. The representative examples were "SWQC" of NEC, "Ayumi System" of Fujitsu, etc. NEC's approach was an example of software design improvement system all over the organization different from traditional hardware manufacturing field. Fujitsu's system was a mechanism for transferring main-framer's software quality management approach into software houses in subcontracting scheme.

This period can be summarized as the glorious age of Japanese style software quality management. Generally speaking, technology change was rather slow; OJT (On the Job Training) of quality management was successful in many organizations.

3.3   1990s

From late 1980s, a rapid trend toward open systems has started. Also, globalization of world economy brought Japanese software vendors into fierce international competition in the international marketplace. Big and long-range projects have disappeared. Many open system development project were small in size and development cycle was very short. Also, style of software business was sifted from product-oriented one to service-oriented. Traditional quality management style succeeded in the domestic market became rather out of date in this new situation and resulted lot of confusion in project management and also in training.

At the same time, International standard ISO9001 has come out. Many companies started to get ISO certification for competition in the international marketplace. Also, other new framework like CMM, Six Sigma, etc. has come in. Many software companies who did not have strong technical identity rushed into certification racing. As a result, SPI activities in field projects were slowed down.

3.4   2000s

Still the general trend in the industry is looking for the new ideas imported from overseas. But other voices for the needs of re-evaluating the strength of Japanese style quality management became louder and louder. Some people are indicating needs to establish new role of Japan in the international software engineering community.

3.5   Latest trends

SEA(Software Engineering Association) and JUSE-SQiP(Union of Japanese Scientists and Engineers – Software Quality Profession) are established in the 1980s and continue being still active. And, some communities were established on the basis of volunteers after 2000 and came to have constant

influence in the Japanese industry. A main community and the activity summary is shouwn in table 1.

Table 1: Overview of major communities in JAPAN

Community	Main Event		Characteristics
	Name	Date	
SEA (Software Engineering Association)	Software Symposium	July	Academic-Industrial Collaboration is active
JASPIC (Japan SPI consortium)	SPI Japan (Software Process Improvement Conference)	October	Under the theme of SPI, a main electrical equipment manufacturer is a member **Development of SPI Strategy Model**
ASTER (Association of Software Test EngineeRing)	JsSST (Japan Symposium on Software Testing)	January	Authorization of the test engineer by JSTQB（Japan Software Testing Qualifications Board）
JUSE(Union of Japanese Scientists and Engineers)-SQiP (Software Quality Profession)	SQiP Symposium (Software Quality Symposium)	September	Development of **SQuBOK** (Software Quality Body of Knowledge)
Agile Process Association	Software Engineer Summit	Spring and Autumn	Research and development about the agile development
AFFORD (Association For Facilitation Of Rational Derivational Development)	XDDP Conference	June	Spread and promotion of **XDDP** (eXtreme Derivative Development Process) method

The overview of SQuBOK, XDDP and SPI Strategy Modle is explained next subsection. In addition, the activity summary of JASPIC is described in chapter 4.

3.5.1 SQuBOK Guide

SQuBOK Guide as a Japan's original BOK had the following five objectives [5]:

1. To help train individuals involved with quality assurance

2. To formalize Japan's implicit knowledge concerning software quality

3. To organize and systematize new themes concerning software quality

4. To improve awareness of software quality technologies

5. To assist organizations seeking to establish software quality assurance processes

Figures 1 provides tree diagrams describing the structure of SQuBOK. The SQuBOK Guide divides knowledge areas into the three general categories of "Fundamental Concept of Software Quality", "Software Quality Management", and "Software Quality Methods" (see Figure 1). The initial category of "Fundamental Concept of Software Quality" classifies fundamental concepts and approaches concerning software quality. The next category of "Software Quality Management" classifies activities for managing quality. The final category of "Software Quality Methods" classifies specific methods, ranging from metrics and quality planning techniques to operational and maintenance techniques.
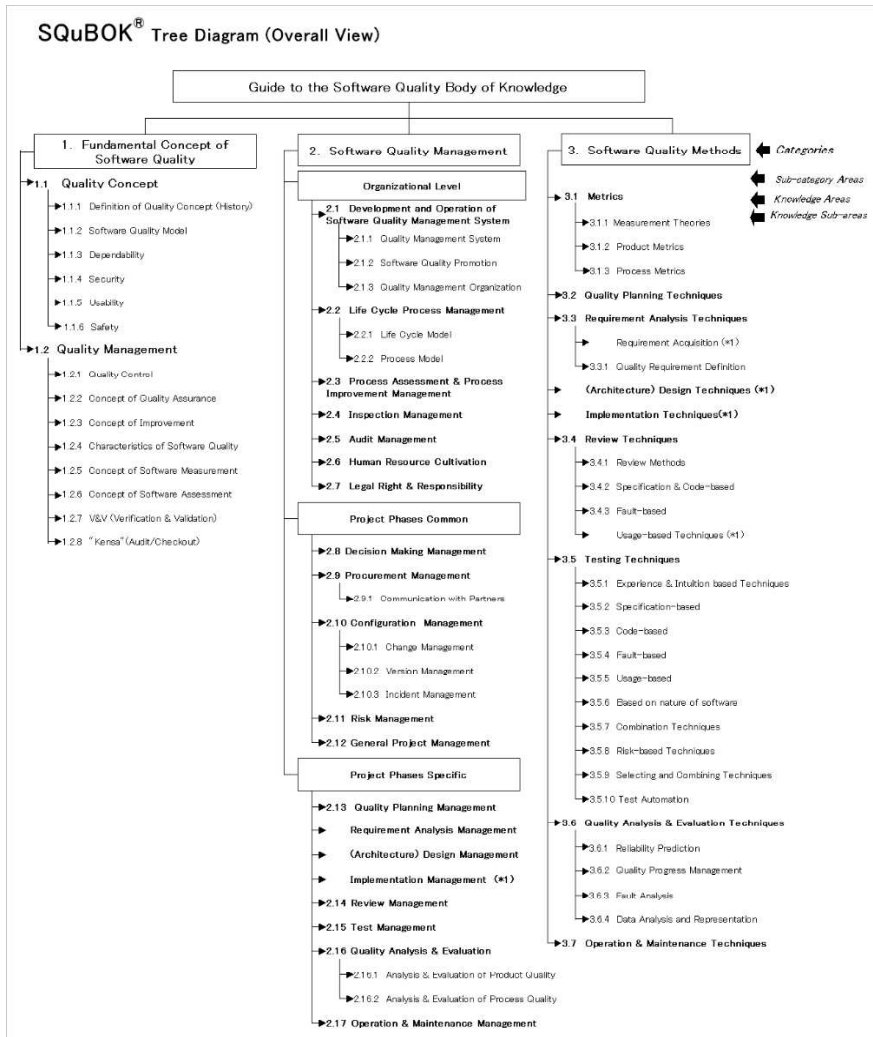
SQuBOK® Tree Diagram (Overall View)

Guide to the Software Quality Body of Knowledge

1. Fundamental Concept of Software Quality

1.1 Quality Concept
- 1.1.1 Definition of Quality Concept (History)
- 1.1.2 Software Quality Model
- 1.1.3 Dependability
- 1.1.4 Security
- 1.1.5 Usability
- 1.1.6 Safety

1.2 Quality Management
- 1.2.1 Quality Control
- 1.2.2 Concept of Quality Assurance
- 1.2.3 Concept of Improvement
- 1.2.4 Characteristics of Software Quality
- 1.2.5 Concept of Software Measurement
- 1.2.6 Concept of Software Assessment
- 1.2.7 V&V (Verification & Validation)
- 1.2.8 "Kensa" (Audit/Checkout)

2. Software Quality Management

Organizational Level

2.1 Development and Operation of Software Quality Management System
- 2.1.1 Quality Management System
- 2.1.2 Software Quality Promotion
- 2.1.3 Quality Management Organization

2.2 Life Cycle Process Management
- 2.2.1 Life Cycle Model
- 2.2.2 Process Model

2.3 Process Assessment & Process Improvement Management

2.4 Inspection Management

2.5 Audit Management

2.6 Human Resource Cultivation

2.7 Legal Right & Responsibility

Project Phases Common

2.8 Decision Making Management

2.9 Procurement Management
- 2.9.1 Communication with Partners

2.10 Configuration Management
- 2.10.1 Change Management
- 2.10.2 Version Management
- 2.10.3 Incident Management

2.11 Risk Management

2.12 General Project Management

Project Phases Specific

2.13 Quality Planning Management

Requirement Analysis Management

(Architecture) Design Management

Implementation Management (*1)

2.14 Review Management

2.15 Test Management

2.16 Quality Analysis & Evaluation
- 2.16.1 Analysis & Evaluation of Product Quality
- 2.16.2 Analysis & Evaluation of Process Quality

2.17 Operation & Maintenance Management

3. Software Quality Methods ◄ Categories

◄ Sub-category Areas
◄ Knowledge Areas
◄ Knowledge Sub-areas

3.1 Metrics
- 3.1.1 Measurement Theories
- 3.1.2 Product Metrics
- 3.1.3 Process Metrics

3.2 Quality Planning Techniques

3.3 Requirement Analysis Techniques
- Requirement Acquisition (*1)
- 3.3.1 Quality Requirement Definition

(Architecture) Design Techniques (*1)

Implementation Techniques (*1)

3.4 Review Techniques
- 3.4.1 Review Methods
- 3.4.2 Specification & Code-based
- 3.4.3 Fault-based
- Usage-based Techniques (*1)

3.5 Testing Techniques
- 3.5.1 Experience & Intuition based Techniques
- 3.5.2 Specification-based
- 3.5.3 Code-based
- 3.5.4 Fault-based
- 3.5.5 Usage-based
- 3.5.6 Based on nature of software
- 3.5.7 Combination Techniques
- 3.5.8 Risk-based Techniques
- 3.5.9 Selecting and Combining Techniques
- 3.5.10 Test Automation

3.6 Quality Analysis & Evaluation Techniques
- 3.6.1 Reliability Prediction
- 3.6.2 Quality Progress Management
- 3.6.3 Fault Analysis
- 3.6.4 Data Analysis and Representation

3.7 Operation & Maintenance Techniques

Figure 1: SQuBOK Tree Diagram (Overall view)

3.5.2 XDDP

XDDP, developed by a Japanese consultant, Yoshio Shimizu in 2007, is an enhancement-based development process [6]. The feature of XDDP is that documents made in XDDP are all about change information about base software. We must make quite new type of documents only about changes. XDDP consists of two independent processes to make the documents easily; one is for adding functions "addition process" and the other is for changing base source code "change process".

Figure 2 shows the process of XDDP. In addition process, we make "requirement specifications on

adding functions" about new tasks or functions. Then design documents are made from "requirement specifications on adding functions". The process is the same as new development process.
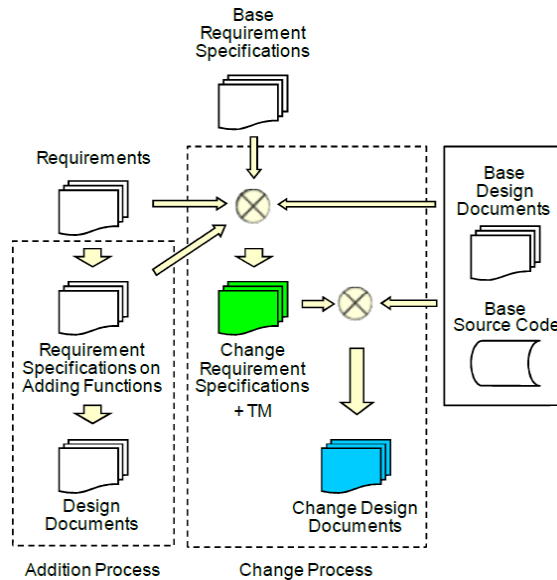

Figure 2: Overview of XDDP

In change process, "change requirement specifications" about changes in base source code are made from base documents, base source code and "requirement specifications on adding functions". "Change requirement specifications" are about changes, additions and deletions in base source code. Then "Traceability Matrix (TM)" is made to specify where change points are modified in the base source code. TM is a matrix of change specifications and the base source code. And we make "change design documents" about how to modify the base source code regarding the change points in the TM.

In XDDP, change specifications are extracted from change requirements in "change requirement specifications", analyzing related documents and base source code. Change specifications contain changes made to the base source code where new functionality is added in addition to changes and deletions in existing functionality. All change points in base source code are extracted and described in "change requirement specifications".

When a new module for new functions is developed, we make "requirement specifications on adding functions" about the module. To add the module to the base source code, we need to change base source code. The change points in accepting the module in base source code are described in "change requirement specifications". In this way, we describe all change points in the documents and review them.

3.5.3 SPI Strategy Model

To select the most effective SPI strategy for the organization, we need to identify candidates for the strategy, and analyze the differences in terms of the outcome [7]. This analysis tends to be complicated beyond a simple comparison because the SPI strategy should usually address the unique situation of the individual organization. To facilitate this effort, we devised the "SPI Strategy Model" as a framework to describe, compare, analyze, and evaluate SPI strategies.

Elements that describe the components of SPI strategy and the outcome of SPI can be grouped into

the following three categories : work to describe, compare, analyze, and evaluate SPI strategies.

(a) C : Contextual factor: elements that describe the environment and situations for the SPI activities.

(b) P : Promotional factor: elements that describe the major decisions to promote SPI activities.

(c) O : Outcome indicator: elements that indicate the degree of success as a result of SPI.

Promotional factors are the basic elements that the organization can determine to support the SPI activities. Contextual factors may not be arbitrarily selected by the organization in the short run, but it affects the decision making for promotional factors, and the conduct of SPI strategy, thus influences the outcome as a result.

When a single set of SPI activities under the environment with Contextual factors (Cn) selected the strategy that is composed of the Promotional factors (Pn), and resulted in the outcome (On), this scenario can be expressed as the following relationship.

Cn -> {Pn -> On}

(*) "->" denotes a causal relationship.

(*) Subscript "n" denotes that there could be multiple factors.

If these elements can be expressed quantitatively, this relationship can be described in a functional form, such as $O_i = F_i (C_n, P_n)$.

Although these elements can include the common factors that are found in multiple SPI cases, different organizations can adopt their own unique factors, especially when choosing outcome indicators. Also, the degree of importance for each factor (i.e. weight) with respect to the contribution to the result could vary depending on the context.

Furthermore, the categorization of these elements may change, and there could be an intricate interaction among these three categories. The organization can "select" some of the Contextual factors (e.g. the organization can set the scope of SPI activities, so this can be considered as part of the Promotional factors). Some aspects of the environment can "change" as a result of the activities, or their changes could be set as the target (e.g. "change of the organizational culture" can be one of the Outcome indicator, so the Contextual factor can be also treated as Outcome).

The objective of the SPI Strategy Model is to classify the elements of these factors, and to understand the interrelationships among them.

4 Activities in JASPIC

The fundamental operating policy of JASPIC is "grass-root" or "bottom-up". The major activity in early stage (2000 - 2001) was to support translation work of CMMI documents. After that work was finished, a number of special interested groups for discussion and information exchange have been organized as the major function of JASPIC and is active until now.

Now the number of member companies of JASPIC grew up to about 20, including almost all of major software-oriented organizations. About the half of them are embedded systems manufacturers and the other half are software houses doing enterprise systems development and maintenance.

The regular activities of JASPIC are: (1) Bi-monthly general meeting, (2) Annual general assembly meeting, (3) Special interest groups (SIG) and (4) Annual SPI Conference (since 2003).

4.1 Theme of Special Interest Groups in JASPIC

SIG in JASPIC is organized in bottom-up style based upon grass-root proposal from members. Now, the following 11 SIGs are active:

(1) SPI Promotion Know-How SIG : Exchange information how to promote SPI activities in each company.

(2) Software Process Improvement Body of Knowledge SIG : Discuss how to define SPI Body of Knowledge.

(3) Statistics Process Contrrol SIG : Study statistical analysis and quantitative management in CMM level 4.

(4) SPI Practical Know-How SIG : Exchange information got from SPI activities in each company.

(5) Assessment Know-How SIG : Exchange information got from assessment activities in each company.

(6) Core Competent Team SIG : Discuss how to introduce PSP/TSP into the company.

(7) Product Line SIG : Study software product line and exchange information got from exanples relate to product line in each company.

(8) People Process SIG : Discuss education/training issues of SPI.

(9) Requirements Development SIG : Study requirements development in CMMI level 3.

(10) SPI strategy SIG : Research SPI strategy based on the SPI practical results in each company.

(11) IDEAL SIG : Discuss effective promotion style for each phase in IDEAL model.

These SIGs represent what kind of SPI-related issues are now considered important in Japanese software industry.

4.2 SPI in Embedded Software Development

One of the uniqueness of Japanese IT industry is the strength in the embedded system product manufacturing. In the past, importance of software components in these products was rather small comparing with hardware parts. Embedded software components were developed in rather ad hoc style.

But the size and function of embedded products has been increased rapidly. For example, in the case of high-function mobile telephone, the size of embedded software is several million steps. Nowadays, software development project teams consist of several hundreds engineers, and have a number of management layers.

One reason why many embedded system manufacturers joined JASPIC is that they were not accustomed to this kind of large-scale software development and wants to transfer quality/project management know-how from other members, especially from software houses working in other industrial sectors. Current issues in embedded software development are:

- How to coordinate hardware development unit with software development unit

- How to deal with rapid growth of software size and shortening of project cycle

- How to manage large scale development project

From now on, embedded software will become more and more important along with the penetration of a variety of products into the various aspects of society like automobiles, home electric appliances, mobile terminals, etc. Japan has been leading international market in some of these fields. In the future, it will be a unique role of Japan to make contribution to the world by combining its high quality hardware manufacturing process with new style of software quality/project management method

4.3 Weakness of University Education

From industrial viewpoint, current university education of software engineering is insufficient, especially in terms of quality/project management. It is because of the large gap between academia and industry. Organization like SEA or JASPIC has been tried to narrow this gap, but still far way from completing its purpose. One other difficult social issue is the barrier of human resource exchange between two communities.

4.4 Status of SPI in Japan

In the period from late 1990s to early 2000s, many companies just tried hard to get certification of ISO9001 or to achieve CMM levels. But, presentations in SPI Conference indicate that this kind of miss understanding of models is gradually disappearing.

After entering 2000s, QCD requirement for software projects became tighter and tighter. Also, globalization of development was accelerated in terms of outsourcing to India or China. Under this situation, many software companies became more sensitive about balance of QCD in their development process, and sharpening their core skills for global competition.

4.5 SPI in future

JASPIC member companies have been very eager to construct a knowledge-sharing community beyond the organizational barrier. They all know that it is necessary to rely on community power for solving various problems in SPI.

Usually, most of requirements and functions (user interfaces) of software system to be developed are given by customer. Software developers can only show their creativity in the process of implementing these requirements and functions as a real system.

Recently, many companies are sensitive about information disclosure. Of course, we must be careful about information given from customers. But mutual exchange of process-related in-house information through mechanism is important and very useful for the development of software industry.

In JASPIC, member companies are sharing common issues in software process and jointly trying to solve problems. Rapid growth of the number of membership companies indicates that this open-process concept is accepted in Japanese software industry.

21century is the age of software. We should continue software process improvement activity to encourage software developers. For that purpose, it is needed to promote information exchange beyond various social/organizational barriers. JAPSIC is a test case to implement such a mechanism.

5 Summary

We have described short history of activity of JASPIC, and made an analytical consideration about the status of SPI in Japan. Unique character of Software Process Improvement in Japan is that it is a collection of bottom-up cooperation among software engineers beyond the organizational barriers in which they are working. JASPIC is considered as a mechanism to promote this kind of mutual knowledge exchange. We hope to enlarge this movement into international domain in future.

6 Literature

1. Hideto Ogasawara et al, "Evolution of Software Process Improvement Activities in a Large-Scale Organization", Proceedings, CONQUEST (2004) pp.213-220.

2. Michael A. Cusumano, "Factory Concepts and Practices in Software Development", IEEE Computer, January-March (1991) pp.3-32.

3. Watts S. Humphrey. "Managing the Software Process", SEI SEROES IN SOFTWARE ENGINEERING, Addison-Wesley (1989).

4. Hideto Ogasawara et al, "Proposal and practice of SPI framework -Toshiba's SPI History since 2000-, EuroSPI2012 (2012).

5. Guide to the Software Quality Body of Knowledge, http://www.juse.or.jp/software/squbok-eng.html.

6 . Keiji Kobata, Eiji Nakai, and Takahiro Tsuda, "Process Improvement using XDDP – Application of XDDP to the Car Navigation System -, 5th World Congress for Software Quality, Shanghai China, November 2011.

7. Noriko Wada, So Norimatsu, and Toshiyuki Kishi, "Analysis of SPI strategies and their modeling - Exploring the art of SPI promotion –", EuroSPI2012(2012).

7 Author CVs

Hideto Ogasawara

Mr. Ogasawara has more than10 years of experience in SPI promotion at Toshiba Corporation, where he is managing SPI activities and developing software testing techniques in Toshiba Group as a leader of Corporate-SEPG.

He is the director of the Japanese Software Process Improvement Consortium (JASPIC).

The Significance of Requirements in Medical Device Software Development

Martin McHugh[1], Abder-Rahman Ali[2] and Fergal McCaffery[1]

Regulated Software Research Centre, Department of Computing and Mathematics
Dundalk Institute of Technology, Co. Louth Ireland
[1](Martin.McHugh, Fergal.McCaffery)@dkit.ie
[2] abder.rahman.ali@gmail.com

Abstract

Software to be used in or as a medical device is subject to user requirements. However, unlike unregulated software, medical device software must meet both the user's requirements and the requirements of the regulatory body of the region into which the software will be marketed. Regulatory requirements are fixed and can be planned for; unfortunately, the same is not true with user requirements. As many medical device software development organisations are following traditional sequential Software Development Life Cycles (SDLC), they are experiencing difficulties accommodating changes in requirements once development has begun. Agile methods and practices offer the ability to overcome the challenges associated with following a sequential SDLC. Whilst the regulatory requirements are fixed, this paper presents these requirements and shows how they appear to mandate the use of a sequential SDLC. This paper also explains how   agile methods and practices can be successfully adopted in the development of medical device software without hindering the process of achieving regulatory approval.

Keywords

Agile, medical, safety critical, regulated, requirements, FDA, SDLC

1   Introduction

Software is becoming an increasingly important component of medical devices, as it enables often complex functional changes to be implemented without having to change hardware [1]. Studies in the medical industry point to the fact that software is one of the most critical factors for cutting edge products. It is expected that, by 2015, that the research and development investment in software in this area will increase from 25% of the overall budget in 2002, to 33%. As the role of software in the medical device domain increases, so do the number of failures which arise due to software defects [2].

The subject of software in and as a medical device has become an important topic for the Food and Drug Administration (FDA). This interest began in 1985 when software in a radiation treatment therapy device failed as a result of software defects resulting in the administrating of a lethal overdose radiation. The FDA then analysed recalls by fiscal year (FY) to determine how many were caused by software problems. In FY 1985, for example, 20% of all neurology device recalls were attributable to software problems, while 8% of cardiovascular problems had the same cause [3]. An analysis of medical device recalls by the FDA in 1996 found that software was increasingly responsible for product recalls. A German survey on medical device recalls indicated that software was the top cause for risks related to construction and design defects of medical device products. This analysis, from June 2006, showed that 21% of the medical device design failures were caused by software defects [2]. This was an increasing trend, as the figures from November 2005 showed software was responsible for 17% of con-

struction and design defects. This continues to be the case, and in the period: 1st January 2010 to 1st January 2011, the FDA recorded 80 medical device recalls and stated software as the cause [2]. This type of analysis, along with the results of various corporate inspections, led the FDA to conclude that some type of regulations was required, especially that the agency's review of medical device reporting (MDR) incidents and analysis of product recalls has convinced the agency that software is a factor contributing to practical problems within devices [3].

Since there are many types of software in use by the medical arena, the problem of the best way to regulate it has become an issue to the FDA. Discussions have centred on what type of software is a medical device, the type of regulations required for such software, and what could be inspected under current regulations [3].

Requirements are central to all software development projects. They are used to develop the software and to demonstrate that the software is performing as intended. However, in medical device software, requirements play an extended role. Non-regulated software must meet the requirements of the customer or end user. Regulated software must also meet the requirements of the regulatory bodies also. To accompany this, as part of the regulatory requirements, a medical device software development organisation must be able to trace all stages of development back to the requirements.

2 FDA Stance on Requirements

The FDA regulations impose stringent requirements on the process by which software systems used in medical devices are developed. These requirements translate into various software artefacts that must be made available for the software to be FDA compliant [4] and, for medical device software, the FDA is responsible for assuring that the device utilizing the software is safe and effective [3].

FDA requires medical device manufacturers to submit their device requirements before beginning development. System and software requirements are taken from the FDA medical device quality system regulation [5]. FDA regulations cover all aspects of the medical device product lifecycle, and the FDA requires medical device manufacturers to submit evidence of product safety and efficacy for FDA review and clearance before the manufacturer can market, sell, or distribute the product [6]. Thus, it is critical to obtain information from the FDA on the requirements applicable to the proposed device [7].

Validation compares the final product to the original specifications [8], and is closely related to the requirements specification. You can validate the user's requirements; this is where ambiguity reigns most of the time and where formal methods, through the use of specification languages, have the biggest strides. There is still a wide gap between what the user wants and what the developer understands that the user wants. Very often this is where one of the causes of initial system failures can be found [9]. Software validation is the confirmation that all software requirements have been met and that all software requirements are traceable to the system requirements, provided that it is not possible to validate software without predetermined and documented software requirements [10]. There are two major types of validation that come into play with medical devices - design validation and process validation. Design validation means establishing, by objective evidence, that device specifications conform to the user's needs and the device's intended uses. Process validation, on the other hand, means establishing, by objective evidence, that a process consistently produces the desired result or a product meeting the predetermined specifications [11]. The FDA requires medical device manufacturers to submit their device specifications before beginning development [12]. Thus, validation could come at early stages of development if the user's requirements could be precisely defined, and which from them the rest of the development derived [12]. Ideally, validation work would be accomplished while the requirements are being written [9]. Any safety and regulatory requirements for medical devices necessarily call for rigorous software development methods to ensure reliability and to protect public health. In addition to that, requirements and specifications based on medical practice are needed to help ensure that devices will perform appropriately [6].

The regulatory bodies request that medical device software development organizations clearly demonstrate how they follow a software development life cycle without mandating a particular life cycle [13]. In order to comply with the regulatory requirements of the medical device industry, it is necessary to have clear linkages to traceability from requirements through the different stages of the software

development and maintenance life cycles. Traceability is central to medical device software development and essential for regulatory approval. Software traceability refers to the ability to describe and follow the life of a requirement in both forward and backward direction [13]. FDA for instance states that traceability analysis must be used to verify that a software design implements all of its specified requirements [14]. Thus, traceability is particularly important for medical device companies, as they have to demonstrate this in order to achieve FDA compliance [15].

2.1 Regulations and Software Development Lifecycles

As discussed, if a medical device software manufacturer wishes to develop software, this manufacturer must adhere to the regulations of the region into which the device is being marketed. These regulations do not mandate a Software Development Life Cycle which must be followed in order to achieve regulatory approval. Initial reading of these regulations and medical device software development standards would appear to imply that software developed for use in medical devices should be developed using a sequential plan driven development lifecycle such as the Waterfall or V-Model.

The FDA Quality System Regulations (QSR) [16] Subpart C – Design Controls provide information as to the processes which must be adhered to when developing regulatory compliant software. These include:

- Design & Development Planning; (Specifications);
- Design Output; (Coding)
- Design Review;
- Design Verification; (Was the Product Built Right);
- Design Validation. (Was the Right Product Built).

As mentioned, initial reading of the QSR would suggest completing these stages sequentially for example in accordance with the Waterfall Model. However, the FDA Design Control Guidance for Medical Device Manufacturers [17] states:

"*Although the waterfall model is a useful tool for introducing design controls, its usefulness in practice is limited… for more complex devices, a concurrent engineering model is more representative of the design processes in use in the industry*"

The FDA General Principles of Software Validation (GPSV) [18] continues to further clarify that it does not mandate the use of a specific SDLC when developing regulatory compliant software:

"*this guidance does not recommend any specific life cycle model or any specific technique or method*"

Furthermore the GPSV acknowledges that activities such as Requirements Specification are likely to be performed iteratively and provides guidance on how these iterative development models can be managed.

"*Most software development models will be iterative. This is likely to result in several versions of both the software requirement specification and the software design specification. All approved versions should be archived and controlled in accordance with established configuration management procedures*"

IEC 62304:2006 [19] is harmonised with the European Medical Device Directive (MDD) [20] and is approved for use by the FDA. IEC 62304:2006 is a software lifecycle model specific to the development of medical device software. As with guidance documents, adherence to IEC 62304:2006 is not mandatory, however, if a manufacturer chooses not to follow it, they would need to provide a sufficient explanation behind not following it. IEC 62304:2006 does not address software development lifecycle models; instead, it defines processes, which consist of activities that should be conducted in each medical device software development project [21]. As with the QSR, initial reading of IEC 62304:2006 would appear to suggest it should be followed in accordance with a sequential lifecycle model such as

Waterfall Model. The publishers of IEC 62304:2006 observed that the standard appeared to mandate following the Waterfall Model and added the following to remove any ambiguity;

"it is easiest to describe the processes in this standard in a sequence, implying a "waterfall" or "once through" life cycle model. However, other life cycles can also be used"

3 Agile Methods to aid with Requirements Management

The rapidly changing business environment in which most organizations operate is challenging traditional Requirements-Engineering (RE) approaches. Software development organizations often must deal with requirements that tend to evolve quickly and become obsolete even before project completion [22]. Agile methods and practices have advantages in accommodating change due to volatile requirements, and are most applicable to projects where requirements are ill-defined and fluid, since they seek to accommodate changes easily [23]. There are different agile practices and methods that can be used in the area of requirements management:

Face-to-face communication over written specifications; effectively transferring ideas from the customer to the development team, rather than creating extensive documentation, where simple techniques (i.e. user stories) are used to define high-level requirements. Here, developers discuss requirements in detail with customers before and/or during development. Thus, customers can steer the project in unanticipated directions, especially when their requirements change owing to changes in the environment or their own understanding of the software solution [22]. All agile approaches emphasize that talking to the customer is the best way to get information needed for development and to avoid misunderstandings. The CHAOS [24] report showed the critical importance of this customer involvement, as it was found to be the number one reason for project success, while the lack of user involvement was the main reason given for projects that ran into difficulties. A key point in all agile approaches is to have the customer 'accessible' or 'on-site'. Agile methods often assume an "ideal" customer representative: the representative can answer all developer questions correctly, and is empowered to make binding decisions and able to make the right decisions [25]. This informal communication with customers obviates the need for time-consuming documentation and approval processes which are perceived unnecessary especially with evolving requirements [22].

Iterative requirements engineering; requirements here aren't predefined, instead, they emerge during development. At each development cycle's start, the customer meets with the development team to provide detailed information on a set of features that must be implemented. And, during this process, requirements are discussed at a greater level of detail. Thus, requirements are clearer and more understandable because of the immediate access to customers and their involvement in the project when needed [22].

Requirements prioritization; agile development implements the highest priority features early so that customers can realize the most business value. The feature lists are prioritized repeatedly during development as the customer's and the developer's understanding of the project evolves, particularly as requirements are added or modified [22]. And, to keep priorities up-to-date, prioritization is repeated frequently during the whole development process [25].

Review meetings; at the end of each development cycle, a meeting with developers, customers, quality assurance personnel, management, and other stakeholders is held for requirements validation. During the meeting, the developers demonstrate the delivered features, provide progress reports to the customers and other stakeholders in the organization, and the customers and QA people ask questions and provide feedback, even though the meetings' original purpose is to review the developed features and get feedback [22].

4 Integrating Agile and Regulatory Requirements

As discussed previously, cursory reading of medical device software standards and regulations appears to advocate utilising a plan driven SDLC that should be followed when developing regulatory compliant software; however, research has shown this not to be the case. Following a plan driven SDLC can prove successful when developing medical device software once the requirements are fully established up-front and there is no risk of change to them. Unfortunately, this is rarely the case and plan driven SDLCs have difficulties accommodating changes. Research has also shown that through the use of iterative development techniques, changes in requirements can be accommodated easier.

While agile methods appear to solve the problems associated with following a plan driven SDLC, how well do agile methods align with the objectives of regulatory bodies? Agile methods appear undisciplined and to advocate producing none of the necessary deliverables; however, this is not the case. The agile manifesto states:

*Individuals and Interactions **over** processes and tools*
*Working software **over** comprehensive documentation*
*Customer collaboration **over** contract negotiation*
*Responding to change **over** following a plan*

It can be seen that statements one and two appear to be contradictory to regulatory requirements, as firstly, the safety of medical device software is determined through the processes followed during the development of the software [26], and secondly, comprehensive documentation is a necessity when seeking regulatory approval. However, as highlighted in the four key principles, agile methods do not dictate that working software instead of comprehensive documentation, nor does it states individuals and interaction instead of processes and tools. The key here is the use of the term **"over"**. For example, Robert Martin, a renowned agilest, clarifies this point further with regards to documentation by stating:

"Produce no documentation unless it is of immediate business value"

In essence, with the development of medical device software, documentation is of business value; therefore it would still be produced while developing software in accordance with agile development methods. Even below the four principles on the agile manifesto [27] website, this is clarified:

"While there is value in the items on the right,
We value items on the left more"

To accompany this, an additional reason cited for not being able to adopt agile methods when developing software, is that prior to development beginning, a medical device software organisation must register the requirements of the device. This being the case, the key benefit of adopting agile i.e. handle changing requirements, become void as there can be no changes allowed. While it is the case that a device's requirements must be registered with regulatory bodies prior to development, regulatory bodies do not require the organisation to register "nuts and bolts" requirements, rather they are concerned with high level requirements.

4.1 Aligning on Goals

Agile software development methods are concerned with developing software using efficient techniques while meeting the needs of the customer. In the case of medical device software, two customers exist, the end user and the regulatory bodies. As a result, agile methods can support regulatory requirements; therefore, agile methods can be supportive of regulatory requirements rather than being contradictory

To accompany this, a key focus of agile development methods is the development of high quality software. Agile methods achieve this by increasing product development productivity and predictability. While regulatory bodies are also concerned about the development of high quality software, they

are not concerned with efficiencies used during the development; however, regulatory bodies do require medical device software organisations to produce objective evidence that the software they have developed performs exactly as described each and every time it is used. This can be achieved through the predictability delivered by agile methods.

4.2   Integration

Previous research has shown that it is not possible to wholly follow a single agile methodology when developing medical device software; however, the same research revealed that combining specific agile practices taken from multiple agile methods and combing them with a plan driven SDLC can be the most advantageous to medical device software organisation. Abbott Diagnostics integrated agile practices with a plan driven SDLC and reported cost savings of between 35% and 50% when compared to a project following a plan driven SDLC. There are a number of instances such as those that report the benefits of integrating agile practices; however, in each of the instances the organisations tailored their own SDLC with agile practices creating a proprietary SDLC. No research exists to date to supply a SDLC which combines agile practices with a plan driven SDLC which can be used by all medical device software organisation. This research will contribute to the development of such a model.

4.2.1   Tailored Software Development Lifecycle

When considering tailoring a SDLC, a foundation plan driven SDLC is required. For this purpose, we chose the V-Model for the following reasons:

- Medical device software organizations typically follow the V-Model to develop medical device software [28]. As a result, they are already familiar with the structure and phases of the V-Model and would be more willing to adopt a hybrid model based upon a SDLC with which they are familiar.
- Medical device software organizations may have received regulatory approval to follow the V-Model when developing medical device software. If these organizations move to a completely different SDLC, they may need to re-apply for regulatory approval for the new SDLC. This may be a barrier as organizations could be reluctant to undergo regulatory approval again.
- Whilst none of the regulatory requirements or development standards mandate the use of the V-Model, it appears to be the best fit with regulatory requirements, as it guides organizations through the process of producing the necessary deliverables required to achieve regulatory conformance.

Once the foundation model was chosen, an analysis of this model was performed to determine where agile practices could be integrated to overcome the problems associated with following a plan driven SDLC. A number of stages remain single pass stages in the tailored model. These stages include *"Requirements Specification"*, *"System Testing"* and *"Acceptance Tests"*. These stages must remain single pass stages to remain in line with regulatory requirements. The remaining stages *"System Analysis", "Software Architecture Design", "Software Detailed Design", "Software Unit Implementa-*
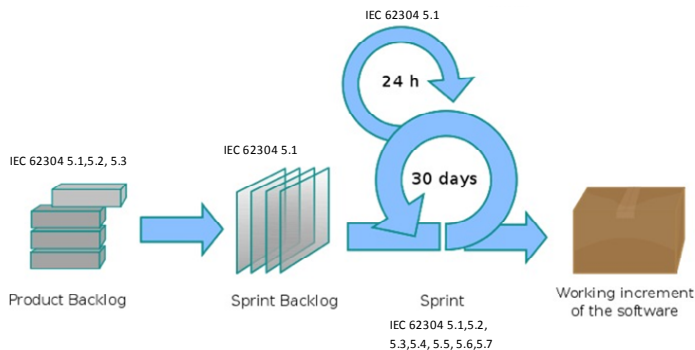
Figure 1 IEC 62304 Mapping to Scrum Lifecycle

tion", "Software Unit Test" and *"Software Integration and Integration Testing"* are integrated into a Scrum SDLC [29] (see figure 1).

IEC 62304 5.1 – Software Development Planning
IEC 62304 5.2 – Software Requirements Analysis
IEC 62304 5.3 – Software Architectural Design
IEC 62304 5.4 – Software Detailed Design
IEC 62304 5.5 – Software Unit & Implementation
IEC 62304 5.6 – Software Integration & Integration Testing
IEC 62304 5.7 – Software System Testing

A mapping study was performed in accordance with [30] which identified instances of where agile methods have been adopted in the development of medical device software. The mapping study identified 10 instances of where agile methods have been successfully used over the period of 2002 to 2012. Of these the majority of the organisations involved adopted a Scrum approach with their traditional plan driven SDLC. In figure 1 the relevant processes, in accordance with IEC 62304, are mapped to specific stages of the Scrum Lifecycle.

5 Conclusions

Regulatory bodies place a large emphasis on requirements when developing medical device software. These requirements are used to achieve traceability and to determine if the medical device software is performing as intended. In an ideal scenario, prior to the development of medical device software, all of the stakeholders in a development team could agree and sign off on the device requirements. Once these requirements are agreed, a medical device software development team could adopt a sequential plan driven SDLC to develop the software effectively. Unfortunately, the ideal scenario rarely exists, and at times, changes in requirements are unavoidable, and if a development team has begun to develope in accordance with a plan driven SDLC, they can experience great difficulties when introducing a change.

Agile methods appear to offer the silver bullet to the problem of changing requirements in development project. As a project is broken into iterations, a change can be introduced into the iteration cycle easier than compared to a plan driven SDLC. However, while agile methods appear to be the silver bullet, there remains reluctance amongst medical device software organisations to adopt them. Research has shown that where agile practices have been adopted, they have proved successful. Where they have been successfuly introduced, they have been intergrated with the existing plan driven SDLC resulting in a scenario where the organisation can rely on the stability of following a plan driven SDLC whilst reaping the benefits of agile methods such as accomodatng changes.

This paper maps stages of medical device software development to a Scrum development lifecycle. This mapping can be used by any organisation wishing to develop medical device software in accordance with agile methods whilst remaining compliant with regualtory controls.

6 References

[1]   F. McCaffery, V. Casey, M. Sivakumar, G. Coleman, P. Donnelly, and J. Burton, "Medical Device Software Traceability," in *Software and Systems Traceability*, J. Cleland-Huang, O. Gotel, and A. Zisman, Eds., ed: Springer-Verlag, 2012.
[2]   C. Denger, R. L. Feldman, M. Host, C. Lindholm, and F. Schull, "A Snapshot of the State of Practice in Software Development for Medical Devices," presented at the First International Symposium on Empirical Software Engineering and Measurement, 2007. ESEM 2007, Madrid, 2007.
[3]   R. C. Fries, *Reliable Design of Medical Devices* vol. 3rd ed. Boca Raton FL: CRC, 2012.
[4]   H. Mehrfard, H. Pirzadeh, and A. Hamou-Lhadj, "Investigating the Capability of Agile Processes to Support Life-Science Regulations: The Case of XP and FDA Regulations with a Focus on Human Factor Requirements," presented at the In Proceedings of SERA (selected papers), 2010.
[5]   F. Pavese and A. B. Forbes, *Data Modeling for Metrology and Testing in Measurement Science*: Birkhäuser, 2009.

[6]   T. H. Faris, *Safe And Sound Software: Creating an Efficient And Effective Quality System for Software Medical Device Organizations*: Asq Press, 2006.
[7]   J. K. Shapiro, "The Pathway to Market for your Medical Device: A Primer on Obtaining Information from FDA " *FDLI,* vol. Update May/June, 2008.
[8]   M. S. Sivakumar, V. Casey, F. McCaffery, and G. Colema, "Verification & Validation in Medi SPICE," presented at the The 11th International SPICE Conference Process Improvement and Capability dEtermination, Dublin, 2011.
[9]   J. O. Grady, *System Requirements Analysis* Amsterdam: Elsevier Academic, 2006.
[10]   D. Farb and B. Gordon, *Pharmaceutical Computer Validation Introduction Guidebook*: UniversityOfHealthCare, 2005.
[11]   C. T. DeMarco, *Medical Device Design and Regulation*: Asq Press, 2011.
[12]   A. Dasso and A. Funes, *Verification, Validation and Testing in Software Engineering*: Idea Group Pub., 2007.
[13]   V. Casey and F. McCaffery, "Med-Trace: Traceability Assessment Method for Medical Device Software Development.," presented at the European Systems & Software Process Improvement and Innovation Conference, (EuroSPI). Roskilde, Denmark, 2011.
[14]   J. Cleland-Huang, A. Czauderna, M. Gibiec, and J. Emenecker, "A machine learning approach for tracing regulatory codes to product specific requirements," presented at the Proceedings of the 32nd ACM/IEEE International Conference on Software Engineering - Volume 1, Cape Town, South Africa, 2010.
[15]   F. McCaffery and G. Coleman, "The need for a software process improvement model for the Medical Device Industry," *International Review on Computers and Software,* vol. 2, pp. 10-15, 2007.
[16]   FDA, "Title 21--Food and Drugs Chapter I --Food and Drug Administration Department of Health and Human Services subchapter h--Medical Devices part 820 Quality System Regulation," ed: *U.S. Department of Health and Human Services*, 2007.
[17]   *FDA Design Control Guidance for Medical Device Manufacturers*, 1997.
[18]   FDA, "General Principles of Software Validation: Final Guidance for Industry and FDA Staff," ed: *Centre for Devices and Radiological Health*, 2002.
[19]   AAMI, "ANSI/AAMI/IEC 62304, Medical device Software - Software life cycle processes," ed. Association for the Advancement of Medical Instrumentation, 2006.
[20]   *Directive 2007/47/EC of the European Parliament and of the Council of 5 September 2007*, 2007.
[21]   M. McHugh, F. McCaffery, and V. Casey, "Standalone Software as an Active Medical Device " presented at the The 11th International SPICE Conference Process Improvement and Capability dEtermination, Dublin, 2011.
[22]   C. Lan and B. Ramesh, "Agile Requirements Engineering Practices: An Empirical Study," *Software, IEEE,* vol. 25, pp. 60-67, 2008.
[23]   M. Coram and S. Bohner, "The impact of agile methods on software project management," in *Engineering of Computer-Based Systems, 2005. ECBS '05. 12th IEEE International Conference and Workshops on the*, 2005, pp. 363-370.
[24]   Standish Group, "Chaos Report," ed, 1995.
[25]   F. Paetsch, A. Eberlein, and F. Maurer, "Requirements engineering and agile software development," presented at the Enabling Technologies: Infrastructure for Collaborative Enterprises, 2003. WET ICE 2003. Proceedings. Twelfth IEEE International Workshops on, 2003.
[26]   P. L. Jones, J. Jorgens, A. R. T. Jr, and M. Weber, "Risk Management in the Design of Medical Device Software Systems," *Biomedical Instrumentation & Technology: July 2002,* vol. 36, pp. 237-266, 2002.
[27]   (2001). *Manifesto for Agile Software*. Available: http://agilemanifesto.org/
[28]   M. McHugh, F. McCaffery, and V. Casey, "Barriers to using Agile Software Development Practices within the Medical Device Industry," in *European Systems and Software Process Improvement and Innovation Conference, EuroSPI* Vienna Austria, 2012.
[29]   M. Cohn, *Succeeding with Agile - Software Development Using Scrum*. Upper Saddle River NJ: Addison Wesley, 2011.
[30]   K. Petersen, R. Feldt, S. Mujtaba, and M. Mattsson, "Systematic Mapping Studies in Software Engineering," presented at the 12th International Conference on Evaluation and Assessment in Software Engineering (EASE), University of Bari, Italy, 2008.

7   Author CVs

Martin Mc Hugh

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

Abder-Rahman Ali

Abder-Rahman received his BSc in Computer Science in 2006 from the University of Jordan and MSc Software Engineering in 2009 from DePaul University. He is interested in applying technology to medicine, and in building computer aided diagnosis systems that aid in diagnosing disease. He is currently pursuing his Ph.D. degree at Université d'Auvergne in France, as part of the ISIT lab, in the area of fuzzy clustering and discrete geometry for image analysis of MRI and ultrasound imaging sequences for Hepatocellular Carcinoma (HCC).

Fergal Mc Caffery

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

Effective Requirements Traceability

Paul Schwann (*paul.schwann@nxp.com*), **Joanne Schell** (*joanne.schell@nxp.com*)
NXP Semiconductors Austria GmbH, Mikronweg 1, 8101 Gratkorn, Austria

Abstract

In our efforts to comply with the Automotive SPICE ® [1] traceability principles (and in doing so, improve our product quality), our organization has introduced several solutions into our standardized project life cycle flow. The most prominent new component is a tool, Traceability Viewer, which takes information from a requirements database (for example, DOORS) and displays statistical information about the state of the requirements and their linkage to test cases and results. With the addition of this tool and other solutions, we have enabled 1) quicker, better-informed decisions, 2) improved accountability in requirement and test development and 3) a strong foundation for continuous process improvement. In this paper, we first discuss our previous solutions and their weaknesses. We then describe our current requirements traceability methods, including Traceability Viewer, and their benefits.

Keywords

Requirements, Traceability, Automotive SPICE®

1 Motivation

Within the automotive industry, it is crucial to maintain an overview of the traceability of requirements to tests within a product development. A traceability overview enables, for example, a status on the percentage of requirements linked to passing test results. We have seen that this visibility eases engagement with the requirement management process, especially important for engineers unfamiliar with the automotive standards. Although there are several popular requirements management databases in use, there is currently a lack of out-of-the-box vendor solutions for a requirements overview. Individual companies, therefore, have created their own overview tools. This development is challenging because 1) there are practically no similar tools to use as a starting point; thus, each company starts essentially from scratch; 2) it is not a small piece of work; 3) the user group is large and varying in its expectations.

In this paper, we address the obstacles with different overview tools and describe an approach that we have found most beneficial for the entire set of stakeholders - our best practices, based on 2 years of experience. We also touch on learnings related to requirements management in general.

We use the Automotive SPICE® [1] V-model as our reference for requirements structure.

The tool extension method: One common solution to enable a traceability overview is to extend the requirements database (e.g. DOORS) to show the traceability overview directly within the database itself. A basic drawback with this technique is that the tool license allows access to only a limited set of stakeholders, reducing visibility (generally for management) to a traceability overview. Additionally, the extension is specific to the requirements database; the introduction of a new requirements database would necessitate the costly development of an additional extension. And requirements data from the various sources cannot be easily combined.

The spreadsheet approach: Another solution is to export the requirements data and then use a spreadsheet to process and visualize the linkage information. However, the table-based algorithms within spreadsheets are not well-suited to the tree structure inherent in traceability data. The table-based formulas result in complicated, error-prone spreadsheet code. This complexity is challenging for most stakeholders, making view adjustments (for example, to see link status for a particular block) annoying at best. Manual adjustment of the views can be reduced by introducing static graphs specific to a viewing requirement, e.g., one status graph per block. Unfortunately, this often leads to spreadsheet bloating — a large number of graphs, making the visual experience unwieldy.

The problem with the methods above: For our environment, the fundamental issue with each of these techniques is that various stakeholders miss a current, readily-accessible, comprehensive (or detailed, as the case may be) view of the state of the design. This lack of data will likely impact decision-making, efficiency, and, in the end, quality.

Our solution: The solution introduced within our organization and used extensively over the past year is a lean, customizable, license-free tool with a GUI (called *Traceability Viewer*) suitable for tracking and analyzing linkage information. With this tool, we have seen a cultural shift with regard to requirements management, because, for the first time, we have traceability data and metrics readily available. And measurement is, as expounded on in the Kaizen [2] and CMMI [3] methodologies, a fundamental element of continuous process improvement. Adherence to Automotive-SPICE ® requirements has been greatly simplified in our environment both through efficient tools and a more engaged workforce.

2 Traceability Viewer

This section describes the Traceability Viewer tool in more detail, the first part focuses on how the requirements data and traceability information are created and processed to be suitable for the Traceability Viewer. The second part illustrates how the traceability data are actually presented within the tool.

2.1 Data Flow

Test Reports
per Module
CSV

Upload
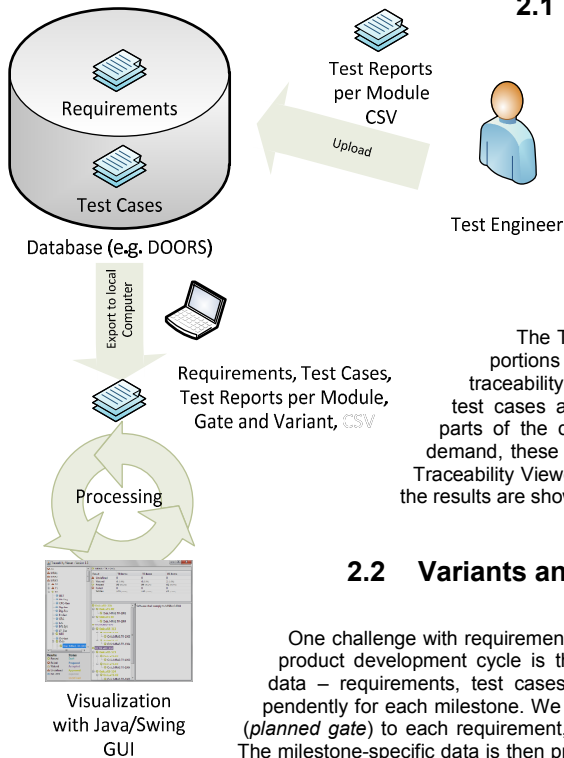
Test Engineer

The data flow for the Traceability Viewer is illustrated in the picture to the left. Project stakeholders (e.g. customers, architects, test engineers) enter requirements and test data into a database[1] (e.g. DOORS®), which serves as the single source of truth for requirements/test information. The data is properly linked and maintained therein.

Requirements

Test Cases

Database (e.g. DOORS)

Export to local Computer

Requirements, Test Cases, Test Reports per Module, Gate and Variant, CSV

Processing

Visualization
with Java/Swing
GUI

The Traceability Viewer is used to visualize portions of the database content, namely the traceability information between requirements, test cases and test reports. In order to do that, parts of the database are exported into CVS. On demand, these files are loaded and processed in the Traceability Viewer, implemented with Java/Swing, and the results are shown to the user.

2.2 Variants and Milestones

One challenge with requirements that must be addressed early in the product development cycle is the management of milestone-specific data – requirements, test cases and test results may change independently for each milestone. We address this by assigning an attribute (*planned gate*) to each requirement, each test case and each test result. The milestone-specific data is then processed and displayed in the Traceability Viewer.

A similar challenge relates to variants of a product; frequently, multiple successive versions of a product are developed with only small differences for requirements and test cases between the versions. To enable differentiation of the versions, a *variant* attribute is attached to each requirement, test case and test result.

A benefit of this structure (using attributes) is that it facilitates *concurrent* development across gates and product variants – since the data is separated correctly, there is no concern that data will be overwritten and lost.

2.3 Views within the Tool

The layout of the GUI has been designed such that the majority of users (engineers, architects, managers) can use it intuitively, based on the "Principles of user interface design" [4]. It shows a navigation tree to the left and specific data views based on the selected data to the right. Similar layouts can be found in many of today's popular business and engineering applications like Outlook and Thunderbird email clients or Eclipse and Visual Studio IDEs. Even the requirement management tool used by our organization – DOORS – uses such a GUI layout to present data.

---

[1] Although we use only one requirements management database, it would be possible to have multiple databases as the Traceability Viewer tool is independent of the source(s).

The following sections describe the different views of the requirements traceability data in the Traceability Viewer.

Milestone MRA4

Requirements

Module MDI

Individual Requirements
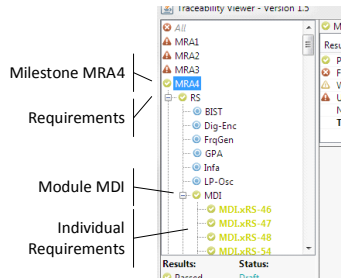
2.3.1 Milestone View

The *Milestone View* serves as the main navigation entity through the traceability data as well as a quick indicator for the requirements coverage and test status (based on icons and colors of the individual elements.)

It shows all items (requirements, test cases and test reports, modules and milestones) in a tree–like structure. For every (planned or already passed) milestone of a project, all items linked to it are shown. To make the navigation easier, the items are separated by the type (requirements, test case and test report) and the module they belong to. Each item may show up for multiple milestones (most requirements, for example, last for the entire project and are linked to all milestones).

Selecting any of the nodes affects immediately the content shown in the *Statistics View* and in the *Details View*. The icon for each node represents the combined test results for all nodes below it. The color of the specific requirements and test case nodes represent the status of definition.

2.3.2 Statistics View

The *Statistics View* reveals the actual statistics of the requirements traceability. It is shown as soon as a node in the *Milestone View* is selected and displays a table which content depends on the node selected in the *Milestone View*.

For a milestone node and all nodes belonging to test reports, the table shows the test status of the test reports and how it maps to test cases and finally to requirements.

For requirements and test case related nodes, the table shows the status of definition (according to the life cycle specified in our organization) and how many items are covered. A requirement is considered to be covered if it is linked to a test specification and vice versa — horizontal bi-directional coverage.

The items counted for the statistics depend on the selection in the *Milestone View*. If a milestone itself is selected, all items for that milestone are considered. If a module is selected, only the items of that module are considered for the statistic. This mechanism allows a quick look-up of traceability and coverage data for individual modules or the whole milestone.

Path of current data selection

117 Requirements:
116 Approved, 1 Rejected

All active requirements linked to test cases

94% of the requirements have passing test results linked to them

2.3.3 Details View

The *Details View* is shown as soon as a specific requirement, test case or test report node is selected in the *Milestone View*. It consists of two elements, a tree to the left and a text view to the right.

The tree shows the context of the selected item, for example, which requirements make use of a selected test report or which tests are specified for a selected requirement. It also demonstrates the vertical traceability within the design.
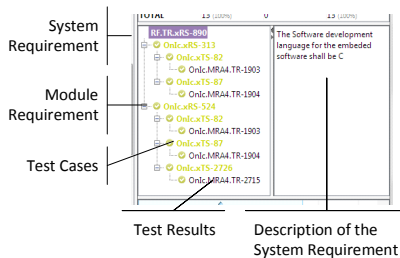
The text view shows the description text of the currently selected item (in the tree to the left). It is empty if no description is available.

System Requirement

Module Requirement

Test Cases

Test Results   Description of the
                      System Requirement

3 Requirements management hurdles and how we solved them

While the Traceability Viewer has proven to address many of the requirements traceability issues we have found, it does not solve all of them. The following sections give answers to other requirements management and traceability issues.

3.1   Managing bi-directional traceability from architecture to requirements and test cases

We do not show architecture-to-requirements traceability directly in Traceability Viewer. In order to facilitate the linkage of architecture to requirements, we use the same block names in the architecture description as in the requirements database.

We also include requirement IDs in the architecture description document itself. We execute a script regularly that compares:

- requirement IDs in the requirements database to those in the architecture description
- requirement IDs in the architecture description to those in the requirement database

…and highlights any differences.

All requirements must have test cases and results; this linkage is visible in the Traceability Viewer. In this way, we ensure that architectural-related requirements are linked to test cases and results.

3.2   Why is it necessary to have test cases at every level?

Within requirements management, we encountered the idea that if we have, for a set of hierarchical requirements:

- 100% vertical coverage
- 100% horizontal coverage at the lowest level of the V-model
- 100% successful passing test results at the lowest level

….then, by induction, the highest level requirements are tested. The problem with this line of reasoning is that the requirements may be imperfectly specified at the highest level and lowest-level testing would not help to catch that error. Developing tests at every level helps to mitigate the risk of incorrectly understood requirements.

3.3   The difficulty of traceability into code

According to the Automotive SPICE® V-model, traceability is required between requirements and implementation:

ENG.6.BP9: Ensure consistency and bilateral traceability of software requirements to software units.

This traceability helps ensure that the implementation and requirements remain consistent with each other. However, maintaining requirement IDs in the implementation is resource-intensive, especially for a product with thousands of requirements. Some of the challenges include:

- Determining placement of the requirement ID since the implementation of a requirement may extend over numerous functions. For example, for re-usable libraries, numerous requirements may require the same library code for their implementation.

- If the requirements are used in a next version of the product, an update of the requirement ids may be required. If the same code base is used for multiple products and/or milestones, it may become necessary to attach this information to the requirement ID.

For some applications (e.g., safety/security relevant), this traceability is crucial (and the challenges above must be addressed.) For other types of applications, however, an indirect traceability may be sufficient to ensure the continual consistency of requirements to implementation. We implemented this indirection by establishing and running a nightly regression suite. Since the associated tests are linked to requirements and the tests exercise the implementation, a failing test would highlight a potential breakage of the requirements-to-code linkage. In this way, we are able to detect inconsistencies between requirements and implementation in an efficient and timely manner.

3.4   The "bureaucratic" hurdle

One of the most difficult challenges in requirements management is the common attitude that achieving traceability is just additional paperwork in an already tight schedule. With the addition of the solutions in this paper, the usefulness of transparent traceability has been demonstrated by:

- Better accountability through readily available status information on requirements, test cases and test results

- Re-use: a good tool standardizes metrics across projects and encourages a common understanding and vocabulary on a topic

4 Future Activities

We are currently piloting three improvements for the Traceability Viewer, involving more advanced processing and visualization of the data.

Historical View: The first improvement incorporates trend information into the Traceability Viewer. Requirements, test cases and test results are snapshot-ed on a weekly basis. The Traceability Viewer uses the historical data to build a trend chart for the different GUI views. The chart below illustrates an example.

Requirements Status Trend
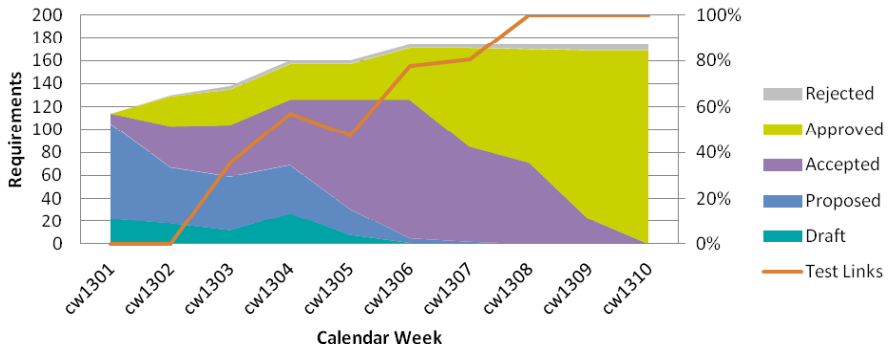
The *Test Links* line within the graph indicates the percentage of the requirements linked to at least one test case.

The historical view gives us a quick indicator on the amount of activity within a project in the space of the V-model.

Actual vs. Planned View: The second improvement for the Traceability Viewer takes these trend views and maps it to the project planned milestones. A line is introduced into the graph which represents the expected status of the requirements, test cases and test results until the next milestone, in order for the related goals of the milestone to be met. With these automatically-generated graphs, tracking of progress is greatly simplified. See example below.

Requirements/Test Result Trend
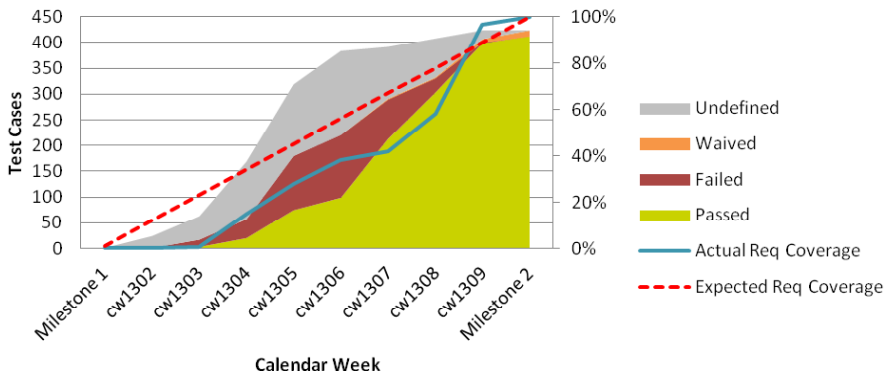
The *Actual Req Coverage* line in the graph shows the percentage of requirements linked to passed test results. The *Expected Req Coverage* line indicates the expected progress between Milestone 1 and Milestone 2. In this way, we can easily judge the degree of alignment to project plans.

Cluster View: We are also piloting a simple scripting change that allows a clustering of *otherwise-separated* blocks to enable tracking of a set of related requirements, for example, analog requirements. With this change, we are able to track progress *in one view* of requirements, test cases and

results for a cluster of requirements that are not captured conveniently in one module of the architecture.

One additional improvement that we will investigate in the future is better visualization related to vertical traceability. Currently, we determine the completeness of vertical traceability, manually, by clicking on individual requirements with the Traceability Viewer. Because our hierarchy is relatively flat, this has been a satisfactory solution so far but we anticipate that will change in the future.

5   Summary

In our environment, we have assessed and trialed different approaches to improve requirements traceability, including spreadsheets and database extensions.

The Traceability Viewer has enabled us to resolve many of the hurdles we have encountered in the management of requirements. The tool processes exported requirements data — from DOORS only, for example, or even a combination of different sources. It displays, among other things:

- Lifecycle status (draft, proposed, accepted) of requirements and test cases

- Percentage and number of requirements linked to passed test cases/results

This information is viewable on a block or system basis and by project milestone. Historical snapshots will allow progress tracking of requirements development and verification.

Although Traceability Viewer does not address every Automotive SPICE® specification related to traceability (e.g. architecture and implementation traceability), we have found suitable solutions and may in the future update the tool to handle those aspects.

Before we introduced Traceability Viewer, requirements traceability was considered the domain of architects and expert users of DOORS. Determining requirements to test-result linkage was a time-consuming and error-prone exercise.

Traceability Viewer (and the other described solutions) has proved its effectiveness in multiple ways:

Improved decision-making: Traceability View provides our organization, for the first time, a clear record of requirements traceability for a product, indicating the degree of tested-ness or requirements readiness. We can now make quick, better-informed decisions because we have a factually-based, immediately-current state on traceability. We can now more speedily answer questions like:

- How stable are the requirements? (Do we need more resources?)

- How much more testing is necessary? (Will we tape-out on time?)

Accountability: We now have better accountability related to requirements and can answer questions like:

- Why doesn't Requirement 231 have a test case?

History information will allow us to query, for example, the reasons for no changes to the requirement database in the past 2 weeks.

Standardization: Having an easy-to-use tool enables re-use because engineers do not try to solve the linkage problem in their own way to get around a clumsy solution. Re-use promotes standardized vocabulary and metrics – we can compare linkage statistics across projects. Re-use also allows easier resource movement between projects.

Continuous process improvement: The metrics available through the Traceability Viewer form the basis of continuous process improvement. In fact, some of the improvements in this paper were inspired through the metrics overview.

Cultural shift: Formerly, traceability was often viewed as basically a bureaucratic necessity for audits. We now see a broader participation and ownership of the traceability status. Progress tracking at pro-

ject and sub-project level is greatly simplified. There is a raised awareness and understanding in general of the requirements traceability process and its power from the engineer to executive level.

Through the effectiveness of these solutions, we are able to improve the efficiency of our project life cycle and the quality of our final product.

6 Literature

[1] Automotive SPICE® version 2.5, www.automotivespice.com

[2] Imai, Masaaki (1986). *Kaizen: The Key to Japan's Competitive Success*. New York: Random House.

[3] CMMI for Development, Version 1.3 (pdf). CMMI-DEV (Version 1.3, November 2010). Carnegie Mellon University Software Engineering Institute. 2010. Retrieved 16 February 2011.

[4] Principles of user interface design, Wikipedia (January 2013)

7 Appendix

7.1 An Overview – Traceability Base Practices and Their Support

For purposes of completeness, we show a table below listing Automotive SPICE® best practices that relate to traceability. The right column explains how we support the practice in our organization.

Automotive SPICE® version 2.5 Base Practice	Reporting to Support Execution of Base Practice
ENG.2.BP6: Ensure consistency and bilateral traceability of customer requirements to system requirements.	(2) Traceability Viewer (vertical)
ENG.3.BP6: Ensure consistency and bilateral traceability of system requirements to system architectural design.	(1) Script
ENG.4.BP6: Ensure consistency and bilateral traceability of system requirements to software requirements.	(2) Traceability Viewer (vertical)
ENG.4.BP7: Ensure consistency and bilateral traceability of system architectural design to software requirements.	(1) Script
ENG.5.BP9: Ensure consistency and bilateral traceability of software requirements to software architectural design.	(1) Script
ENG.5.BP10: Ensure consistency and bilateral traceability of software architectural design to software detailed design.	(4) Manual check (vertical)
ENG.6.BP8: Ensure consistency and bilateral traceability of software detailed design to software units.	Detailed design is documented within the implementation.
ENG.6.BP9: Ensure consistency and bilateral traceability of software requirements to software units.	See section **Fehler! Verweisquelle konnte nicht gefunden werden.**, page **Fehler! Textmarke nicht definiert.**.
ENG.6.BP10: Ensure consistency and bilateral traceability of software units to test specification for software units.	(3) Traceability Viewer (horizontal)
ENG.7.BP7: Ensure consistency and bilateral traceability of software architectural design and software detailed design to software integration test specification.	(5) Manual check (horizontal)
ENG.8.BP5: Ensure consistency and bilateral traceability of software requirements to software test specification.	(3) Traceability Viewer (horizontal)
ENG.9.BP7: Ensure consistency and bilateral traceability of system architectural design to the system integration test specification.	(5) Manual check (horizontal)
ENG.10.BP5: Ensure consistency and bilateral traceability of system requirements to the systems test specification.	(3) Traceability Viewer (horizontal)

(1) **Script**: Compares list of *database* requirement IDs to *design-document* requirement IDs (and vice-versa) to ensure that all bi-directional linkage is in place. The script outputs any inconsistencies.

(2) **Traceability Viewer (vertical)**: Vertical linkage visualized in GUI.

(3) **Traceability Viewer (horizontal)**: Statistics available on horizontal linkage.

(4) **Manual check (vertical)**: A visual inspection comparing the block names in the architecture and lower level design documents.

(5) **Manual check (horizontal)**: A visual inspection comparing the block names in the architecture and lower level design documents to those in the test suite. (Tests are clustered according to block names.)

Author CVs

Dipl.-Ing. Paul Schwann

Since 2007, Paul Schwann has held the position of a project leader within business units Automotive and Identification at NXP Semiconductors. Since 2010, Paul is, as the group and project leader for software, responsible for all software needs of the products areas of Car Access and Immobilizers.

From 1992 to 1997, Paul Schwann attended the Dresden University of Technology and graduated with a Diplom-Ingenieur (equivalent to MSc) in Electrical Engineering.

Paul is a certified Project Management Professional and Certified Scrum Master.

From 1997 through 2007, Paul Schwann has worked in the area of hardware/software co-design for communication systems for Dresden University of Technology, Systemonic, Philips and NXP Semiconductors.

MSc Joanne Schell

Since 2010, Joanne Schell has held the position of Quality Assurance Officer within the Business Unit Automotive at NXP Semiconductors. Her scope of responsibility includes ensuring conformance of engineering processes to relevant standards (ASPICE, ISO/TS 16949, CMMI, AIAG FMEA, VDA 6.3) for the product areas of Car Access and Immobilizers.

From 1995 to 1997, Joanne Schell attended the Texas State University in San Marcos, Texas and graduated with a Master's Degree in Computer Sciences.

Joanne is certified in Foundation PRINCE2 (Projects in Controlled Environments) and as a Project Management Professional, Certified Scrum Master, Provisional ISO/TS16949 Assessor, and Automotive SPICE® Provisional Assessor.

From 1997 through 2007, Joanne Schell has worked, primarily in the area of EDA tool development, for Motorola, Freescale, ARM Limited and, through internships, at Siemens Automation and Sprint Tele-communications in various locations in the USA and Europe. For 7 years, she was the company representative and active participant in the OpenAccess Coalition, a 35-member organization developing and driving standards across the semiconductor industry.

Implementation of Traceability Best Practices within the Medical Device Domain

Gilbert Regan, Fergal Mc Caffery, Kevin Mc Daid, Derek Flood
{gilbert.regan, fergal.mccaffery, kevin.mcdaid, derek.flood}@dkit.ie

Abstract

Requirements validation, compliance verification and impact analysis are important activities that are performed during the software development lifecycle. Traceability of requirements through the software development lifecycle (SDLC) is essential in the development of safety critical software. Organisations such as the Food and Drug Administration and the Federal Aviation Authority in the United States require traceability as part of their approval process. However, despite its criticality there is extensive digression in the practices and usefulness of traceability across development projects. Many projects' traceability efforts are simply focused on satisfying regulations and do not leverage the many benefits of traceability. Traceability, if fully implemented is an important tool for managing system development and there are a number of published best practices to help companies with this implementation. By means of a literature review we record a list of the commonly accepted best practices for traceability implementation. Furthermore, through interviews with two medical device companies we report that a number of these practices are unfamiliar to these companies and why an even greater number of these practices are not applied.

Keywords

Requirements traceability, Traceability best practices, Medical device, Safety critical

1 Introduction

Traceability is the ability to establish links (or traces) between source artefacts and target artefacts [1]. Requirements tracing is concerned with recovering the source of requirements and predicting the effects of requirements [2]. In general, traceability is about understanding a design right through from the origin of the requirement to its implementation, test and maintenance. Traceability allows us to understand aspects such as to whether the customers' requirements are being met, the specific requirements that an artefact relates to, and the origins and motivation of a requirement. Traceability aids impact analysis, component re-use, change management and generally supports an improvement in competitive advantage. Traceability helps ensure that 'quality' software is developed [3].

Safety critical industry standards, such as the medical device standards mandate that traceability be implemented e.g. the Food and Drugs Administration (FDA) state that the validation of software typically includes evidence that all software requirements have been implemented correctly and completely and are traceable to system requirements and to risk analysis results, that the software design implements all of the software requirements and that all code is linked to established specifications

and established test procedures[4]. IEC 62304:2006 [5] is a harmonised standard which defines the life cycle requirements for medical device software and requires traceability between system requirements, software requirements, software system test, and risk control measures implemented in software, and that the manufacturer shall create an audit trail whereby each change request, relevant problem report and approval of the change request can be traced. In addition to these requirements regulation normally requires critical systems are certified before entering service. This involves submission of a safety case - a reasoned argument and supporting evidence that requirements (non-functional requirements such as safety, availability and reliability) have been met and that the system is acceptably safe [6].

However, despite its many benefits and regulatory requirements, traceability is a tool that often is grudgingly implemented, and, if implemented is often not leveraged to take advantage of the information it can provide to a development or validation team. Numerous reasons have been identified for reluctance in implementing traceability including cost and complexity. Reasons include the task of building a requirements trace matrix (RTM) is time consuming, arduous and error prone [7], there are few metrics for measuring the return on investment for traceability, stakeholders within an company have differing perceptions as to the benefits of traceability [8], the need for documentation can cause resentment among developers who may fear that traces could be used to monitor their work [9], difficulties with trace tools including selecting between available tools, and difficulties configuring a general purpose tool or developing a custom tool [10]. Finally almost no guidance is available for practitioners to help them establish traceability in their projects and as a result, practitioners are ill-informed as to how best to accomplish this task [11, 12].

Notwithstanding the lack of guidance, there are a number of commonly accepted best practices which can direct practitioners with the implementation and maintenance of traceability. In section 2 of this paper we present twenty three best practices which we have identified from literature [8, 13-23]. In section 3, we   discuss our findings from interviewing two small to medium sized (SME) medical device companies about their application of these best practices. We then analyse our findings in section 4 before providing our conclusions in section 5.

2   Traceability best practices

A review of the literature has revealed the following commonly accepted best practices (BP) for the efficient implementation and maintenance of requirements traceability through the SDLC.

BP 1: Adopt a company policy for traceability [8, 14]. A company must adopt consistent practices for requirements management, including traceability. Different stakeholders can have different viewpoints on the need/value of traceability.If all stakeholders buy into the traceability policy, it greatly increases the chances of success. As Kannenberg and Saiedian note, "Perhaps the best way to deal with the problem of different stakeholder viewpoints on traceability is to create an organizational policy on traceability to apply uniformly to all projects."

BP 2: Implement a standard operating procedure (SOP) for traceability. In clinical research, the *International Conference on Harmonization* (ICH) defines SOPs as "detailed, written instructions to achieve uniformity of the performance of a specific function". SOPs help address ' the lack of detailed guidance on how to implement traceability [11]' as they are an integral part of a successful quality system as it provides individuals with the information to perform a job properly, and facilitates consistency in the quality and integrity of a product or end-result and help ensure compliance with governmental regulations.

BP 3: Develop a traceability information model (TIM) [15]. A TIM models the traceable artifact types (i.e. requirements, design, code etc.) and their permitted trace links as a unified modeling language (UML) class diagram. The benefits of a TIM are that it [16]:
- Ensures consistent results in projects with multiple stakeholders;
- As traceability is also used by people who did not create it, these people need to know how it has been defined and what to expect from it;

- As tracing is a complex task, a TIM provides a guideline to ease its set up and allows for the validation of changes;
- Coverage analysis is only possible after having defined what the expected coverage is;
- A TIM is a necessary precondition for automated traceability handling, validation and analyses

BP 4: Provide tool support for traceability [15]. Creating and maintaining traceability can be a time consuming and complex task. Requirements management/ traceability tools provide features for establishing, maintaining and navigating trace links and can display information in matrix or trace slice format. It is worth considering tool support when [1]:
- A project has many requirements,
- When more than one person, site or company is doing the engineering and requirements management work, and where there is a need to share and align artefacts and traces.
- When requirements and other engineering artefacts, including their traces, are being used and reused in multiple ways, such as within other projects and within product families.
- When the engineering personnel are performing repetitive and administrative tasks to enable requirements management.
- When a long project or product life is expected, or when there are many customers with likely change requests to manage.

BP 5: Keep it simple [18] Capture "just enough information". Limit number of data points. Capturing too much information makes the system burdensome, complex and error prone, however one must ensure that enough information is captured so as to make it useful.

BP 6: Create traces incrementally [15]. Stakeholders are empowered to create trace links incrementally within the context of their daily work. This avoids the situation of traceability being deferred until the end of a project where its perceived purpose is for regulatory reasons. Creating traceability as the project progresses allows stakeholders to benefit from traceability knowledge throughout the project.

BP 7: Unique identifiers must be adopted for requirements and business rules [14, 18-20]. To permit traceability, each requirement must be uniquely and persistently labelled so that you can refer to it unambiguously through the project. Unique identifiers follow the requirement throughout the workflow and are never reused or reassigned.

BP 8: A responsible party must take ownership of traceability [8, 14]. Gathering and managing requirements traceability data must be made the explicit responsibility of certain individuals or it won't happen. Further, errors will occur if someone who is not familiar with the system or the requirements attempts to make updates, errors will abound. Updates must be practised consistently or traceability will degrade and become untrustworthy.

BP 9: Practice value based requirement tracing instead of full tracing [8, 14, 17]. Value-based requirement tracing prioritises all of the requirements in the system, with the amount of time and effort expended on tracing each requirement depending on the priority of that requirement. This can save a significant amount of effort by focusing traceability activities on the most important requirements. However, value-based tracing requires a clear understanding of the importance of each requirement in the system; it may not be an option if full tracing is a requirement of the customer or the development process standards used for the project.

BP 10:Clearly identify stakeholder or source [18]. Associate each requirement with a named person or customer or other source of the requirement such as a regulation or requirement from a standard.

BP 11: **Educate the team about the concepts and importance of requirements tracing** [8, 19]. Many companies do not train their employees regarding the importance of traceability and traceability is not emphasized in undergraduate education.

BP 12: Centralise - Requirements traceability should be documented centrally using some type of log e.g. a traceability matrix [21].

BP 13: Audit/Review the traceability information periodically to make sure it is being kept current [19, 21].

BP 14: Inventory Current Processes and Tools [22]. Review methods and practices to ensure they are not outdated? There are always new tactics and development methodologies that can give you a competitive advantage for each product or project, but are they compliant?

BP 15: Start tracing at the beginning of analysis [18]. It is considered much harder to implement traceability if you wait until the end of the project. Less value to the team and company

BP 16:Periodically communicate improvements from the practice of traceability [18]. Quantify savings from early defect detection or elimination of missing requirements

BP 17: Never use traceability as a measurement in performance reviews [8, 18, 23]. Individuals may be concerned that traceability data will be used against them in performance reviews or as a threat to their job security.

BP 18: Bidirectional Traceability [13]. Good traceability practices allow for bidirectional traceability, meaning that the traceability chains can be traced in both the forwards and backwards directions.

BP 19: Identify the key individuals [19] who will supply each type of link information and the personnel who will co-ordinate the traceability activities and manage the data.

BP 20: Explore Terminology and Meaning [22]. Get all participants to agree on one naming convention per artefact (if possible), especially if there are legal or compliance reasons for consistent artefact names. There may be situations where an agreement is not possible, in which case these artefact names should be documented or mapped to each other, even if they are essentially the same. For example, customers may call something a bug, QA may call it a defect, Development may call it an issue, and Engineering may call it an anomaly, but ultimately each group may be referring the same thing. Map these terms to each other and move on.

BP 21: Model traceability queries [15]. Traceability queries cover basic life-cycle activities such as finding all requirements associated with currently failed test cases or listing all mitigating requirements associated with a given hazard.

BP 22: Visualize trace slices [15]. In safety critical systems trace links established between hazards, faults, mitigating requirements, design, implementations, and test cases are of particular importance. Therefore, instead of presenting traceability material in the form of trace matrices, generate visualizations of trace slices in which the hazard is the root node, and all direct and indirectly traced artefacts that contribute towards mitigating the hazard are shown as a tree.

BP 23: Evaluate traces continually using a dashboard [15]. Tracing benefits are often not realized directly by the people performing the tracing tasks. Furthermore, the current status of the traceability effort is often not visible to individual stakeholders or to the project manager. A dashboard that displays the tracing progress for a project can be effective for tracking and managing the tracing goals of the project and also for motivating team members to create appropriate trace links.

3   Best practices in practice

3.1   Introduction

To examine the application of traceability best practices we interviewed two SME companies operating within the medical device domain. These two companies were selected because they were both SME's developing medical devices which contained software or developed devices which are standalone software. While both companies were considered SME's they were a different size (20 employees as against 70) and had different levels of experience in developing medical device software.   The contrasting size and level of experience of these companies was chosen to provide a broader overview of the experience of traceability in the medical device domain.

Prior to conducting the interview with the companies, a questionnaire was developed to examine the company's overall approach to traceability. The questionnaire includes a set of direct questions were used to determine the companys' application of traceability best practices e.g. have you adopted a company policy on traceability or have you considered adopting one? One direct question was used for each of the twenty three best practices.

In addition to examining the best practices for traceability adopted by the companies, the interview also examined their compliance with the medical device standards, the difficulties that are encountered while implementing traceability and opportunities that may exist for improving their traceability process. A full discussion of these topics is beyond the scope of this paper, however the interested reader is referred to [24]for more details.

3.2   Company Profile

Company A

This company is solely located in Ireland and has a total workforce of less than 20 people. The software systems are provided on-line and have been available for up to 11 years now. The company have 2 full time developers mainly working on upgrades or specific customer requests. Their products are rated as software safety classification I, meaning they may not present much risk of illness or injury. The company uses the Waterfall model for their software development.

Two people were interviewed together and they answered questions both together and individually. The first individual's position was as Quality Control Manager and they had been with the company for eighteen months and had many years' experience in this position. The second individual was a Project Manager and had been with the company for 6 months.

Company B

This company's headquarters are based in the UK, but it has a research and development and manufacturing facility based in Ireland. This study was carried out in the Ireland facility. The company employs 60 to 70 people and sometimes employs contractors on a part time basis so the numbers can fluctuate. The products are marketed globally into the primary care market, secondary care, occupational health, sports medicine and clinical trials. Their products are rated as software safety classification II, meaning non-serious injury to the patient or operator of the device is possible due to a defect in the device. The company uses the V model for their software development.

Two people were interviewed separately from this company. The first person was the Chief Technical Officer and had been with the company for approximately ten years. The second person was the Software Development Manager and he had been with the company for approximately ten years.

3.3   Findings

Table 1 below summarises Company A & Company B's responses to questions about their application of traceability best practices. The first column refers to the best practice numbers as detailed in section 2 of this paper.

Table 1: Implementation of Traceability Best Practices

Best Practice	Company A	Company B
BP 1	No company policy on traceability in place and no plans to implement one. Have never considered it.	No company policy on traceability in place and have not considered implementing one.
BP 2	No traceability SOP in place. No plans to implement one.	A traceability SOP is in place. It basically details how to complete the trace matrix.
BP 3	No TIM in place. Have never considered its implementation.	No TIM in place. Would consider its implementation.
BP 4	No tool support for traceability except general purpose spreadsheet. Not considering trace tool support at present.	No tool support for traceability except general purpose spreadsheet but are actively considering the purchase of a tool.
BP 5	Had a consultant to review this. Find it difficult to know exactly what links to make.	Knowing how much information to capture while at the same time being efficient is difficult.
BP 6	Not something the company has thought about. Would probably need to review	Trace matrix mostly deferred towards the end of the project so stakeholders don't benefit from traceability knowledge throughout the project
BP 7	Each requirement is uniquely identified.	Each requirement is uniquely identified
BP 8	One of the developers is fully responsible although this is not documented.	It is the responsibility of the software development manager to ensure that traceability gets implemented.
BP 9	Don't practice VBRT. Trace every requirement because operating in safety critical domain.	Don't practice VBRT. Trace every requirement because operating in safety critical domain.
BP 10	Source of requirements is identified but not in trace matrix. A developer keeps a record of this himself (not in a company document).	Source of requirements is identified.
BP 11	No education of employees as to the benefits of traceability..	No education of employees as to the benefits of traceability
BP 12	A traceability matrix is in place.	A traceability matrix is in place
BP 13	No formal audit/review of traceability information. It is left to the developer to ensure it is correct so any audit is on an ad-hoc basis.	No formal audit/review of trace information except at end of project where it is signed off by the software develoment manager.
BP 14	Had a consultant who reviewed traceability technique but made no significant change. No formal plan in place for review/audit of traceability approaches.	Company are currently reviewing their approach by considering purchase of trace tool. No formal plan in place for review/audit of traceability approaches.
BP 15	In most cases the matrix gets completed towards the end of the project.	The matrix does not get maintained as and when it should.
BP 16	Improvements made due to traceability are not communicated within the company.	Improvements made due to traceability are not communicated within the company.
BP 17	Traceability data is never used in staff performance reviews.	Traceability data is never used in staff performance reviews.
BP 18	Bi-directional tracing available through trace matrix but difficulty in tracing back from Technical Spec. to Functional Spec.	Bi-directional tracing available through trace matrix and in-document tracing.
BP 19	One developer is responsible for all links	Key individuals were identified.
BP 20	The company had yet to engage in naming conventions and this issue was causing some confusion.	The company has engaged in naming conventions.
BP 21	Had not considered or were aware of this practice	Had not considered this practice
BP 22	Had not considered or were aware of this	Had not considered this practice

	practice	
BP 23	Had not considered or were aware of this practice	Had not considered this practice

4 Discussion of Traceability Findings across both Companies

The first point to consider is the number of practices that each company implement with Company A implementing eight practices and Company B implementing 10 practices. The companies agreed that this was quite a low number considering the authors stated twenty three best practices and offered two main reasons for this. The first reason is that the companies were unaware of some of the practices and the defense for this was that they did not have the in-house traceability expertise. The second reason is that the companies generally viewed traceability as "a pain in the backside" and that "they would probably not bother with it if it wasn't for regulatory purposes."

Another point to note is that Company B implemented every practice that Company A implemented (i.e. BP 7, 8, 10, 12, 14, 17 , 18, 19) plus BP 2 and BP 20. The reason for this commonality may be that it is difficult to have any form of traceability in place without implementing these common practices i.e. traceability will not happen unless someone is made responsible and the easiest way to centrally document traceability is through a trace matrix which also somewhat facilitates bi-directional tracing. With regard to BP 14 Company A have had a consultant review their processes (including their traceability process) because as a result of an amendment to the Medical Device Directive [25] (where stand alone software can now be considered a medical device) their product is now considered a medical device. Company B have reviewed their process and have concluded that the addition of a trace tool would improve their process and are currently considering different tool options.

Company B have implemented two practices more than Company A, one of which is a SOP for traceability and we believe the reason for this may be due to the following two factors. Company B have been developing medical device software for more than thirty years and so has a lot of experience in developing medical device software relative to Company A. In addition to this Company B, being a bigger company, has more resources than Company A.

The lack of interest in traceability (outside of regulatory compliance) which the companies openly admit to, is perhaps better illustrated by the best practices that have not been applied (e.g. having no company policy or SOP in place). Being unaware of or not considering certain best practices only emphasises this indifference. The companies were not interested in the many benefits that requirements traceability offers with one interviewee stating 'I know I should say things like impact analysis but the truth of the matter is we only use traceability for ensuring all requirements have been tested completely and the matrix is very useful when an auditor comes in'.

5 Conclusion

The implementation and maintenance of traceability varies greatly between development projects. Many reasons have been put forward for this including cost, complexity and lack of guidance on how to implement traceability. However from the literature we have identified twenty three best practices, all or some of which a company can use to improve its traceability.

To assess the application of traceability best practices within the medical device domain we interviewed two medical device companies and found that Company A and B applied eight and ten practices respectively, with the eight practices been applied by Company A also been applied by Company B. This commonality of best practice application is mostly due to these common practices been basic practices necessary for any form of traceability implementation. Company B is a bigger and more experienced company in medical device software development and this is a contributor to it applying more practices than Company A. Another finding was that companies did not implement some of the practices because they simply were unaware of them, due mainly to the fact that they did not have the traceability expertise in-house. Finally companies found traceability to be 'a pain in the backside' and 'would not bother with it if it wasn't for regulatory reasons' and so were only interested in the minimium implementation necessary. This viewpoint did not stimulate the application of traceability best practices and meant that these companies were not availing of the full benefits that traceability has to offer.

6 Acknowledgement

7 References

1. Gotel, O. and P. Mader, Acquiring Tool Support for Traceability, in Software and Systems Traceability, J. Cleland-Huang, O. Gotel, and A. Zisman, 2012, Springer.

2. IEEE, Guide to the Software Engineering Body of Knowledge, SWEBOK, Editor. 2004.

3. Regan, G., et al., Impact of Standards on the Role and Application of Traceability in the Medical Device Domain, in EuroSPI. 2012, DELTA: Vienna, Austria. p. 2.11.

4. FDA, General Principles of Software Validation; Final Guidance for Industry and FDA Staff. 2002, CDRH: Rockville.

5. ANSI/AAMI/IEC, 62304:2006 Medical device software—Software life cycle processes. 2006, AAMI: Arlington, VA.

6. Mason, P., On Traceability for Safety Critical Systems Engineering, in Proceedings of the 12th Software Engineering Conference. 2005, IEEE Computer Society. p. 272-282.

7. Cleland-Huang, J., Just Enough Requirements Traceability, in Proceedings of the 30th Annual International Computer Software and Applications Conference - Volume 01. 2006, IEEE Computer Society. p. 41-42.

8. Kannenberg, A. and D.H. Saiedian, Why Software Requirements Traceability Remains a Challenge. CrossTalk The Journal of Defense Software Engineering, July/Aug 2009.

9. Jarke, M., Requirements tracing. Commun. ACM, 1998. 41(12): p. 32-36.

10. Regan, G., et al., The Barriers to Traceability and their Potential Solutions: Towards a Reference Framework, in 38th Euromicro Conference on Software Engineering and Advanced Applications. 2012, IEEE: Cesme, Turkey. p. 319- 322.

11. Caffery, F.M. and V. Casey, Med-Trace: Traceability Assessment Method for Medical Device Software Development, in EuroSPI 2011: Denmark. p. 1.1 - 1.8.

12. Mader, P., O. Gotel, and I. Philippow, Motivation Matters in the Traceability Trenches, in Proceedings of the 2009 17th IEEE International Requirements Engineering Conference, RE. 2009, IEEE Computer Society. p. 143-148.

13. Westfall, L. Bidirectional Requirements Traceability. 2006 [cited 2013 06/02/2013]; Available from: http://www.compaid.com/caiinternet/ezine/westfall-bidirectional.pdf.

14. Marchis, A. An Introduction to Requirements Traceability. Requirements.com 2011 [cited 2013 06/02/2013]; Available from: http://www.requirements.com/Topics/RequirementsTraceability/tabid/87/Default.aspx.

15. Mader, P., et al., Strategic Traceability for Safety Critical Projects. IEEE Software, 2013(In Press).

16. Mader, P., O. Gotel, and I. Philippow, Getting back to basics: Promoting the use of a traceability information model in practice, in Proceedings of the 2009 ICSE Workshop on

Traceability in Emerging Forms of Software Engineering. 2009, IEEE Computer Society. p. 21-25.

17. Rochimah, S., W.M.N. Kadir, and A.H. Abdullah. An Evaluation of Traceability Approaches to Support Software Evolution. in Software Engineering Advances, 2007. ICSEA 2007. International Conference on. 2007.

18. Thomas, B., Ensuring Project Success with Requirements Traceability, in Software Systems Best Practice Conference. 2012: Anaheim, California.

19. Morckos, M., Requirements Traceability. 2011, University of Waterloo: London.

20. Tvete, B. Introducing efficient requirements management. in Database and Expert Systems Applications, 1999. Proceedings. Tenth International Workshop on. 1999.

21. CDC. Unified Process Practice Guide - Requirements Traceability. 2006 [cited 2013 Feb 20]; Available from: http://www2.cdc.gov/cdcup/library/practices_guides/default.htm.

22. Seapine, Six Exercises to Strengthen Traceability. 2011, Seapine Software.

23. Ramesh, B., et al. Implementing requirements traceability: a case study. in Requirements Engineering, 1995., Proceedings of the Second IEEE International Symposium on. 1995.

24. Regan, G., et al. Investigation of Traceability within a Medical Device Organization. in International SPICE Conference on Process Improvement and Capability dEtermination in Software, Systems Engineering and Service Management. 2013. Bremen, Germany.

25. Council, E., Medical Device Directive 2007/47/EC. Official Journal of the European Union, 2007.

8   Author CVs

Gilbert Regan
Gilbert Regan is a PhD student at Dundalk Institute of Technology. He is a member of the Regulated Software Research Group in Dundalk Institute of Technology and a member of Lero. His main research subject is Software Process Improvement with particular focus on Traceability within Medical Device Software.

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

Derek Flood
Dr. Derek Flood is a post doctoral research fellow with the Regulated Software Research Group at Dundalk Institute of Technology (DkIT). Derek was formally a post doctoral research assistant at Oxford Brookes University on the PACMAD project. His current research interests include Software process improvement for Medical Devices and Usability

Kevin McDaid
Kevin is a lecturer in Computing at Dundalk Institute of Technology. He leads research in the area of Spread sheet engineering in the Software Technology Research Centre. His background is in Mathematics and Statistics and his research focus on the application of statistical methods to decision problems in software development.

Challenges experienced by Medical Device Software Development Organizations while following a Plan-Driven Software Development Life Cycle

Martin McHugh and Fergal McCaffery

Regulated Software Research Group, Department of Computing and Mathematics,
Dundalk Institute of Technology & Lero, Ireland
{martin.mchugh, fergal.mccaffery}@dkit.ie

Abstract

Medical device software organisations face challenges not faced by generic software development organisations. These challenges include the adherence to regulatory controls. Regulatory bodies require medical device software organisations to provide objective evidence that the software they are developing is safe and reliable. To produce this, regulatory bodies require a number of deliverables which must be achieved. However, they do not dictate which Software Development Life Cycle (SDLC) must be followed in order to achieve these deliverables. Despite not dictating which SDLC must be followed when developing medical device software, organisations typically develop their software in accordance with a Plan-Driven software development lifecycle. By conducting semi structured interviews with seven medical device software organisations, we gained a deeper insight into how the challenges experienced impact on the development of medical device software. The interviews also attempted to learn from the participants how they believe the challenges experienced can be overcome. The aim of this paper is to explain the methodology used to perform interviews with medical device software organisations and to present these interviews.

1 Introduction

Medical device software organisations experience difficulties not experienced by non-regulated software development organisations. Anecdotal evidence suggests the biggest challenge experienced by medical device software organisations is regulatory controls. Medical device software, regardless of the time or money spent on the development of the software, can be deemed useless if it fails to achieve regulatory approval. Medical device software organisations therefore may be reluctant to adopt new techniques to improve efficiencies, fearing that it may hinder their chances of achieving

regulatory approval. Ongoing research will present practices to medical device organisations which aim to achieve efficiencies without hindering the process of securing regulatory approval.

To learn the areas in which difficulties are experienced semi structured interviews were conducted with medical device software organisations. These interviews gave an insight into the real differences between developing medical device software and generic software and also the challenges faced by these organisations. Once this understanding was gained, appropriate recommendations can be made as to how these challenges may be overcome and efficiencies can be successfully introduced.

The remainder of this paper is structured as follows, Section 2 discusses the methodology used to create the interviews, Section 3 presents the participants of the interviews, Section 4 outlines the findings of the interviews, Section 5 discusses the recommendations and in Section 6 the conclusions are presented.

2   Research Methodology

The interviews were conducted in accordance with Wengraf [1]. The interviews were performed on a semi structured basis. This form of interview is known as a Semi Structured Depth Interview (SSDI). SSDIs are characterized by the following features:

- *"The interview is a research interview, designed for the purpose of improving knowledge.*
- *It is a special type of conversational interaction: in some ways it is like other conversations, but it has special features which need to be understood.*
- *It has to be planned and prepared for like other forms of research activity but what is planned is a deliberate half-scripted or quarter-scripted interview: its questions are only partially prepared in advance and will therefore be largely improvised by you as an interviewer. But only largely: the interview as a whole is a joint production, a co-production, by you and your interviewee.*
- *It is to go into matters 'in depth'".* [1]

SSDIs are further categorized into two classifications, Heavily Structured Depth Interviews and Lightly Structured Depth Interviews. The degree of structuring is determined by the degree to which the questions and interventions are pre-prepared by the researcher. Figure 1 shows the relationship between structured and unstructured interviews.

Figure 1 Spectrum from Unstructured to Fully Structured Interviewing, and Possible Relationship to Phases in the Development of a Theory [1]

2.1   Pyramid Model

In accordance with Wengraf, the interview was broken into four elements. These elements are:

- Research Purposes (RP);
- Central Research Question(s) (CRQ);
- Theory Questions (TQ);
- Interview Interventions (II) / Interview Questions (IQ).

The RP is the motivation behind the research being conducted. For this research, the RP is to gain a deeper insight into difficulties experienced when developing medical device software. The CRQ is the primary question(s) to which answers are being sought as a result of the interview being conducted. The TQ are high level questions. These questions are not asked directly to the interview participant. TQ are used to formulate the actual questions that will be asked of the participant. II/IQ is what is actually asked of the participant during the interview. The information gleaned from the responses is compiled to answer the TQ which in turn answer the CRQ which ultimately supports the RP. The relationship between each of these elements is shown in the pyramid model shown in figure 2.

In Dillon [3], the author discusses the various types of questions and non-questions in interview scenarios. He describes that interventions[1] in an interview can be more beneficial than pre-prepared questions. As a result they are included in the pyramid model.
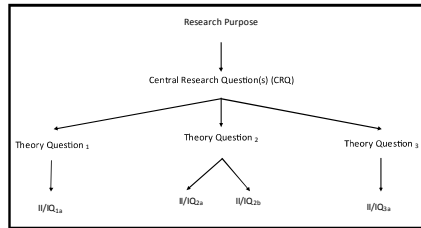
Figure 2 CRQ > TQ > IQ/II: Pyramid Model [1]

3 Interview Participants

A difficulty often associated with SSDI's is the process of achieving representative sampling. Patton [4] presents several different sampling methods used as part of conducting SSDI's each valid to specific methods of research. This is known as Patton's Typology of Randomised and Purposive Sampling [4]. Based upon Patton's Typology of Randomised and Purposive Sampling the most appropriate method of sampling for this research is "Purposeful Sampling" employing "Maximum Variation Sampling". Using this approach, organisations of varying structure and size each of which representing a sector of the medical device software development industry were identified.

Seven medical device software organisations participated in the interviews. Within each organisation, employees best placed to answer questions relating to the development of medical device software were interviewed. The positions which these employees hold within the different organisations varied. Below is the list of roles which the participants perform:

- Chief Technical Officer;
- Head of Development;
- Quality Manager;
- Co-Founder/Director;
- Senior Research and Development;
- Principal Engineer;
- Chief Executive Officer;
- Electronic Design Engineer.

In accordance with Wengraf the following questions i.e. IQ, were established prior to the interviews being conducted. The following tables show how each of the IQ relates to a TQ and in turn how the TQ contributes to answering the CRQ. The CRQ for this research is "*How does the development of medical device software differ to the development of non-regulated software?"*

---

[1] Interview Interventions are described as questions or statements made during the interview to elicit responses that are not prepared prior to the interview.

Table 1 Research Questions

CRQ	RQ1 What are the issues with developing medical device software?	TQ1	IQ1a.
			IQ1b.
	RQ2 What are the issues with developing medical device software using a traditional software development lifecycle?	TQ2	IQ2a.
			IQ2b.
		TQ3	IQ3a.
			IQ3b.
		TQ4	IQ4a.
			IQ4b.

Table 2 Theory Questions

CRQ	RQ1	TQ1 Does regulatory conformance directly impact the development of medical software?	IQ1a.
			IQ1b.
	RQ2	TQ2 Are you following a Plan-Driven software development lifecycle and if so why?	IQ2a.
			IQ2b.
		TQ3 Does following a Plan-Driven software development lifecycle meet all of the organisation and regulatory requirement?	IQ3a.
			IQ3b.
		TQ4 Would a tailored lifecycle be more appropriate than moving to a different software development lifecycle completely?	IQ4a.
			IQ4b.

Table 3 Interview Questions

CRQ	RQ1	TQ1	IQ1a. How do you believe the development of medical device software differs to that of the development of generic software?
			IQ1b. How do these differences impact on the development of medical device software?
	RQ2	TQ2	IQ2a. For your current software development project which software development lifecycle are you following?
			IQ2b. Why are you following this lifecycle?
		TQ3	IQ3a. What difficulties are you experiencing as part of your current software development project?
			IQ3b. Why do you believe you are having these problems?
		TQ4	IQ4a. Do you believe there is a way to overcome these problems?
			IQ4b. What do you believe are the barriers to moving away from your current software development lifecycle?

4 Interview Findings

The interviews conducted yielded rich qualitative data. However, a method was needed to extract the findings from this qualitative data. The results of the interviews were analyzed in accordance with Wengraf's, Interview Material to Answers to Theory Questions to an Answer to the Central Research Question (IM-ATQ-ACRQ) model [1]. Whilst the CRQ > TQ > IQ/II model utilizes a top down approach, the IM-ATQ-ACRQ model utilizes a bottom up approach to determine the answer to the central research question. This method was used as it complimented the method employed for the creation of the interview questions i.e. RP > CRQ > TQ > IQ/II. The results were also analysed in accordance with Miles and Huberman's [2] method of analyzing qualitative data.

4.1   Qualitative Data Analysis

Miles and Huberman [2] present three stages for qualitative data analysis. The three stages are:

- Data Reduction;
- Data Display;
- Conclusion Drawing and Verification.

The relationship between these stages of qualitative data analysis and data collection is shown in Figure 2.

4.1.1 Data Reduction

SSDIs produce a large amount of data. To navigate all of the data collected the volume must be reduced. Data reduction is a continuous process happening before the data is collected. Before data collection occurs "Anticipatory Reduction" takes place. The interviewer attempts to pre-empt the information being collected and selects questions in an attempt to reduce unnecessary information from being collected.
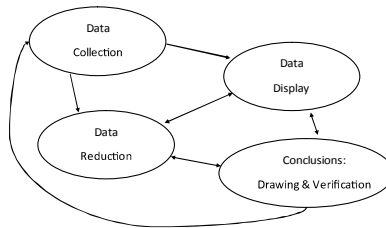
Figure 2 Components of Data Analysis: Interactive Model [2, p.12]

4.1.2 Data Display

As SSDIs can produce large amounts of raw data, in this case over 30 pages of interview transcripts, a method was required in which this data can be displayed in a form easily understood by a person. To achieve this matrices and graphs were employed. A key element of data display is that data display is not separate from data analysis. In fact, it is part of overall process of data analysis [2].

4.1.3 Conclusions Drawing and Verification

The process of drawing conclusions involves examining the collected data and analyzing the implications this data has on the research being conducted. Miles and Huberman [2] discuss that, whilst a final conclusion is created once all of the collected data has been analyzed, conclusions appear very early on in the data collection process and that whilst the conclusions may appear to be established inductively, external influences can have an impact on the development of early conclusions. Conclusions can be derived from analyzing data once. These conclusions are verified by analyzing the data multiple times. The conclusions are deemed to be verified once the results after each analysis are the same.

4.2 Results

After each of the three stages of qualitative data analysis were performed as outlined in the previous section, the results were produced (See table 4). The number sequence after each response correlates to a specific interview and at which point in the interview the response was given.

Table 4 Responses Received during SSDIs

	Central Research Question							
	RQ1		RQ2					
RESPONSES RECEIVED	TQ1		TQ2		TQ3		TQ4	
	IQ1a	IQ1b	IQ2a	IQ2b	IQ3a	IQ3b	IQ4a	IQ4b
	There is a level of authorisation surrounding the developing of medical device software which adds a certain amount of complexity [6,015].Production of clinical evidence, being ISO 13485 compliant and following IEC 62304, production of documentation [5,008,012] [1,008] [3,004] [2,002] [7,0034]Risk [4, 002] [1,008] [3,004] Safety [1,008]Prescribed set of deliverables before you start [3,004] Usability [3,004] Process is more strict [2,002]	If you have the right processes in place and you built it into the company it wouldn't take as long as it currently does for us [5,016] The cost associated with making sure the traceability is current all of the way through [4,004] Increased processes infrastructure required and you need to have a disciplined approach [1,020] Large effort put into testing, integration and validation [3,006]Increased time to market, improved quality more expensive to produce [2,006, 008] [7,008]	Waterfall Model [6,021] [1,024] V-Model [5,022] [4,006] [3,014] [2,012] [7,016]	Residue from some of the activities we performed in the automotive industry [4,009] The majority of our customers have asked us to do it [1,028] Auditors are familiar with the V-Model, document outputs are in line with guidance and regulations [2,014] [7,018]	Requirements Changes [6, 031] Validation is probably something and making sure test cases are linked back to the requirements [4,011] Interdependency between stages and impact on other areas [2,015] Requirements or design, retrofitting or filling gaps [1,034] Traceability of requirements [3,002][7,022]	We don't have a strong enough structured approach and we cannot formalise the requirements capture [4,013] It wasn't specified up-front [1,040]	Work better up front on capturing requirements precisely [1,042] Modifying the quality management system [3,032] Improve requirements management up front [2,018] combine life cycles i.e. agile and Plan-Driven [7,026]	No barriers at present [6,037] We chose the V-model as it best reflects our processes but we could move away from it [5, 036] Yes, our clients would struggle to understand and it would be difficult to quote on an agile project [4, 15] I think the barriers would come from key stakeholders [1,046] We wouldn't have a problem making a change and introducing a new system to get over what difficulties we come across [3,032] Increase cost in retraining staff and redefining new processes [2,020] [7,028]

5   *Discussion*

Based upon the findings shown in table 4, it can be seen that the organisations involved identified a number of the same problems and challenges as shown by the multiple number sequences after specific responses. The two most cited difficulties are regulatory constraints and managing requirements changes.

5.1   Regulatory Constraints

Software developed for use, in or as a medical device must adhere to strict regulatory controls. These controls are put in place to ensure the safe and reliable functioning of the software. Medical device software organisations must provide objective evidence to regulatory bodies that their device is safe. This evidence is achieved through the production of comprehensive documentation. The production of this documentation can become burdensome for software organisations. One of the interview participants noted that whilst the production of documentation and adherence to regulations can be burdensome, it can also be beneficial. The burden of adherence can act as a barrier to entry into the medical device software development industry potentially reducing the amount of competition within the medical device industry.

To accompany the requirements to produce adequate documentation, medical device manufacturers are advised to adhere to a quality management standard such as ISO 13485 [5] when developing medical device software. Whilst in Europe, it is not mandatory to follow this standard, should a device manufacturer choose not to follow this standard, they must prove to the regulatory bodies that the method which they used to ensure the quality of their device is equally comparable to ISO 13485.

The interviews showed that medical device software organisations typically follow a Plan-Driven Software Development Life Cycle (SDLC) such as the V-Model. Plan-Driven SDLCs produce the necessary deliverables required when seeking regulatory conformance; however, Plan-Driven SDLCs are not seen as efficient [6] and can be difficult to apply to a medical device software development project in a practice. Section 5.2 discusses how these Plan-Driven SDLCs can be modified to become more efficient whilst still producing the necessary regulatory deliverables.

5.2   Managing Requirements

Previous research [7, 8] and the interviews conducted as part of this paper has shown that medical device software is typically developed in accordance with a Plan-Driven SDLC. Plan-Driven SDLCs such as the Waterfall and V-Model are typically performed in a sequential manner with very little scope for revisiting stages. Plan-Driven SDLCs dictate that requirements are gathered up-front prior to any development beginning. However, a medical device software development project can potentially take a number of years to be completed and it can be very difficult to ensure that there will be no change in requirements throughout this period.

Each of the organisations involved in the interviews identified that a major problem they experience is accommodating changes once development has begun. To accommodate changes a number of stages may need to be revisited, having a knock-on effect of increasing rework and therefore increasing cost. When asked in the interviews how to resolve the problems associated with changing requirements a number of responses were given. One organisation suggested the establishment of an incubation period prior to the requirements analysis stage. This incubation period would allow the customer time to consider all potential features they wished to include in the software and ideally removing the need for a change to be implemented once the project has begun. Another organisation suggested placing greater emphasis on up-front planning and again making sure all of the necessary requirements were captured. One organisation suggested "placing manners on the customer" and preventing them from introducing a change once development has begun.

Each of these suggestions has their own merit, however these are proactive steps, none of the organizations were able to suggest a reactive response to when a requirements change was unavoidable. Current Plan-Driven SDLCs are rigid and therefore have difficulty accommodating a change. Typically, when a change is introduced, a number of stages of development need to be revisited to

accommodate the change. This can require a lot of rework therefore increasing cost and development time. Agile practices and methodologies promote the ability to be able to accommodate changes. The agile manifesto states *"welcoming changing requirements, even late in development"* [9]. This would suggest that utilising agile practices in the development of software could offer the *"silver bullet"* to problems associated with late changes in requirements.

However, research [10-12] has shown that it is very difficult to fully adopt a single agile methodology such as Scrum or XP, as no single agile methodology produces the necessary deliverables required when seeking regulatory approval. To overcome this, research suggests that combing agile practices with a Plan-Driven SDLC can reap the most significant rewards as the organisation would still benefit from the structure associated with following a Plan-Driven approach whilst also gaining the efficiencies associated with utilising agile practices.

6 Conclusions

Medical device software organisations face challenges not faced by non-regulated software development organisations. We conducted interviews with seven medical device software organisations to gain a deeper insight into these challenges. We selected organisations of varying size, structure and criticality to act as a broader representation of the medical device software industry as a whole. These organisations included medical device manufacturers, software suppliers to medical device manufacturers and organisations providing design services to medical device software suppliers and manufacturers. Whilst these organisations ranged in maturity, size and software criticality, the challenges experiences by each of them are very similar. The biggest challenge identified is the adherence to regulatory controls. This adherence brings with it the overhead associated with producing large amounts of documentation. It also brings with it the perception that following a Plan-Driven SDLC is required in order produce the necessary deliverables required when seeking regulatory approval[2].

The CRQ of this research is to determine the differences between the development of medical device software and the development of non-regulated software. The interviews revealed that the key difference is the need to adhere to regulatory controls. Regulatory controls appear to restrict medical device software organisations to follow a sequential plan driven SDLC. However, following a plan-driven SDLC can introduce problems such as having difficulties introducing requirements changes.

As medical device software is typically developed in accordance with a Plan-Driven SDLC, medical device software organisations experience the inherent problems associated with following this type of lifecycle. The most identified challenge by the participants of the interviews associated with following a Plan-Driven SDLC is accommodating requirements changes. As Plan-Driven SDLC are completed in a sequential manner, if a stage is completed and development has moved on it can be very difficult to revisit a stage such as *"Requirements Management"*.

To overcome this challenge, medical device software organisations are advised to move to a SDLC which can better accommodate requirements changes. Agile methodologies boast the ability to welcome changes throughout the development lifecycle. However, research has shown that it can be very difficult to fully move away from a Plan-Driven SDLC to an agile methodology as no single agile methodology produces the necessary regulatory deliverables. Based on this, future work as part of this research will involve developing and validating a hybrid SDLC which combines a Plan-Driven SDLC with agile practices to introduce efficiencies and to overcome the difficulties associated with requirements management in medical device software development projects.

---

[2] Whilst it is perceived that medical device software regulations and standards require medical device software to be developed in accordance with a Plan-Driven SDLC there is no direct instruction with the regulations or standards dictating the use of a specific SDLC

7 References

[1]   T. Wengraf, *Qualitative Research Interviewing*. London: Sage Publications, 2001.
[2]   M. Miles and A. M. Huberman, *Qualitative Data Analysis: An Expanded Sourcebook*. London: Sage, 1994.
[3]   J. Dillon, *The Practice of Questioning*. London: Routledge, 1990.
[4]   M. Patton, *Qualitative Evaluation and Research Methods (2nd Edn)*. Newbury Park: Sage, 1990.
[5]   ISO, "*ISO/IEC 13485:2003 Medical devices -- Quality management systems -- Requirements for regulatory purposes*," ed. International Organisation for Standards - Geneva, Switzerland, 2003.
[6]   J. Cadle and D. Yeates, *Project Management for Information Systems*: Pearson Education, 2008.
[7]   F. McCaffery, D. McFall, P. Donnelly, and F. G. Wilkie, "Risk Management Process Improvement for the medical device industry," presented at the Conference on Software Development (SWDC-REK-2005) Iceland, 2005.
[8]   M. McHugh, F. McCaffery, and V. Casey, "Barriers to Adopting Agile Practices when Developing Medical Device Software," presented at the The 12th International SPICE Conference Process Improvement and Capability dEtermination, Palma, Majorca, 2012.
[9]   R. C. Martin, *Agile Software Development - Principles, Patterns and Practices*: Prentice Hall, 2003.
[10]   R. Rasmussen, T. Hughes, J. R. Jenks, and J. Skach, "Adopting Agile in an FDA Regulated Environment," presented at the Agile Conference, 2009. AGILE '09. , Chicago, IL 2009.
[11]   P. A. Rottier and V. Rodrigues, "Agile Development in a Medical Device Company," presented at the Proceedings of the 11th AGILE Conference. AGILE '08., Girona, Spain, 2008.
[12]   H. Mehrfard and A. Hamou-Lhadj, "The Impact of Regulatory Compliance on Agile Software Processes with a Focus on the FDA Guidelines for Medical Device Software," *International Journal of Information System Modeling and Design,* vol. 2, pp. 67-81, 2011.

8 Author CV's

Martin Mc Hugh

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

Fergal Mc Caffery

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

Understanding the Software Product Line Derivation Process: An Industrial Example

Leandro Oliveira de Souza[1, 3,5](*los2@cin.ufpe.br*), Pádraig O'Leary[2,5](*padraig.oleary@lero.ie*), Eduardo Santana de Almeida[3,4,5](*esa@rise.com.br*), Silvio Romero de Lemos Meira [3,5] (*srlm@cin.ufpe.br*)

[1]*Federal Institute of Bahia (IFBA), Irecê-BA, Brazil*
[2]*Lero – the Irish Software Engineering Research Centre, University of Limerick, Ireland*
[3]*Federal University of Pernambuco (UFPE), Recife-PE, Brazil,*
[4]*Federal University of Bahia (UFBA), Salvador-BA, Brazil,*
[5]*Reuse in Software Engineering (RiSE),*
Recife-PE, Brazil

Abstract

Software Product Line Engineering has emerged as a software engineering strategy aimed at helping industry achieve business goals. Nevertheless, in order to ensure the return of investment with the *Software Product Line (SPL)* approach, a well-defined *Product Derivation (PD)* process is important. Without this process, the products are instantiated in an ad-hoc manner with success relying on the effort of a few individual members. This may increase the production costs and time-to-market.

Despite its importance, when compared to the vast amount of research on developing product lines, relatively little work has been dedicated to the process of product derivation. Additionally, there are few available reports about how software development organizations derive their products from a product line.

Thus, this study presents the findings gathered through to the case study methodology in order to enhance understanding of how product derivation is performed in industrial settings, including its key phases and activities in the product derivation process.

Keywords

Case study, Product derivation and Software Product Line Engineering.

1 Introduction

A growing number of software development organizations are adopting strategies that emphasize proactive reuse, interchangeable components, and multi-product planning cycles [1]. In this way, the Software Product Line (SPL) approach has emerged as a software reuse approach, in which reuse is planned, enabled, and enforced. It applies a strategy that plans the use of assets in multiple products rather than ad-hoc approaches that reuse assets only if they happen to be suitable [1].

The SPL approach makes a distinction between domain engineering (where a product is derived based on the platform components [2]) and application engineering (when individual products using the platform artefacts are constructed). The process of creating these individual products from a product line of software assets is known as product derivation [3].

An effective product derivation process may contribute to ensuring that the effort required to develop the platform assets is less than the benefits delivered through using these shared artefacts across the products within a product line [3]. However, despite the importance of product derivation, there are several difficulties associated with the process, such as: the process is slow and error prone [2], it has an inherent complexity [4], [5] and it is still a time-consuming and expensive activity in many organizations [3]. In addition, there are few reports [6], [3] available describing how SPL organizations derive products from a product line.

Due to these difficulties, we performed a case study in an industrial SPL project within the medical information management systems domain. The case study investigated the key phases and its activities in the product derivation process.

The remainder of this paper is organized as follows. Section 2 describes the case study background. Section 3 presents the research results, i.e. an overview of company product derivation process. In Section 4, the results are discussed. Section 5 discusses related work on product derivation. Finally, conclusions and futures directions are presented in Section 6.

2 Case Study Background

This section describes the case study conducted at MedicWare Informatic Systems (http://www.medicware.com.br/) located in Salvador, Brazil. MedicWare Systems has been developing integrated management systems for the medical domain since 1994.

The MedicWare product line (SMART) is composed of 52 modules (sub-systems), including more than 918 features. It provides thousands of possible variations among its different features and enables the instantiation of customized products within the medical domain. Thus, a company costumer can choose within *SMART portfolio*, the set of modules and features that satisfy their needs.

The products built on top of the *SMART Platform of Core Assets* are large and complex technical software systems with hundreds of features. Their infrastructure is composed of several parameter calls, which enables the selection of components and features during product derivation.

2.1 Research Question

The main goal of this study is *to investigate and collect information about the MedicWare product derivation process*. Evidences gathered in the organization were used to understand how the process is conducted. Hence, the research question of this study is stated as follow:

- *RQ: What are the key phases and activities in the product derivation process?* According to recent literature [7], [8], these issues are important aspects to be investigated in the area.

2.2 Case and Subjects Selection

The MedicWare systems LTDA was selected as our case study organization because it had a platform of reusable core assets, a product derivation process, and a considerable number of customers to which it provided customized products.

Regarding to the study subjects selection, a set of 15 subjects were selected from different units and areas (Development team, Product Customization Team, Analysis, Deployment and Business). It is important to involve different roles and personalities in a study like this one to get different and complementary point of views [12].

3 Product Derivation Process

The Medicware product derivation process is composed of five phases: *Commercialization, Modelling, Analysis, Customization*, and *Deployment*.

3.1 Commercialization

Commercialization consists of four activities: *Define Scoping, Commercial Presentation of the Product, Technical Presentation of the Product* and *Prepare Proposal/Negotiation*.

Each activity is detailed in Table 1.

Activity	Purpose
Define Scoping	Information about the customer is elicited. An initial survey is performed to establish the reusability of features relevant to the customer.
Commercial Presentation of the Product	To determine the set of appropriate models and features to be instantiated and integrated to compose the *Base Configuration*. *Director of Sales* discusses and presents the *Base Configuration* and identifies customer's needs that are not supported by the *SMART Platform*.
Technical Presentation of the Product	To present an overview of the features offered by each module and what should be instantiated for the specific customer. Before this presentation, the *Product Expert* configures assets variants via globally accessible database tables (*TableINI*) and *Configuration Files*. Thus, during the *Technical Presentation of the Product*, each module is described within the customer scope, i.e., each module is described as a base configuration.
Prepare/Negotiate Proposal	To present an overview of the features offered by each module and what should be instantiated for the specific customer. The customer confirms if the product and features to be implemented are aligned to their needs. With the review meeting, new features can be included, excluded or reprioritized in the feature list. Moreover, a module list that represents the *Partial Product Configuration* is defined.

Table 1. Commercialization Activities.

3.2 Modelling

The modelling phase involves two activities: *Modelling* and *Kick Off Meeting*.

Each activity is detailed in Table 2.

Activity	Purpose
Modelling	Map the customer requirements and workflow to define the level of reuse in the Base Configuration and create a Partial Product Configuration. The customer needs are elicited and relevant features are selected. The process of customer needs elicitation involves mapping the workflow of the medical unit (customer domain) in Medical Unit Workflow Diagrams which is compared with the Workflow of Medical Domain Diagrams (i.e. workflow supported by platform features) to identify which features will be integrated in the Partial Product Configuration. Deciding which features should be selected is supported by a questionnaire for each module and variation point resolution.

At the end of this activity, the customer scope is defined. From this modelling, the core assets that will be instantiated to compose the Partial Product Configuration are defined. Additionally, the features defined will be customized (platform assets that will be adapted) and developed from scratch during the product customization phases. |
| Kick Off Meeting | In order to obtain agreement from the stakeholders on the feature list and product scope, the Partial Product Configuration is demonstrated by the Product Expert to key stakeholders.

The Scoping Declaration and Deployment Chronogram are presented and the customer specific requirements which cannot be satisfied by the Partial Product Configuration are negotiated. |

Table 2. Modelling Activities.

3.3 Analysis

The Analysis phase consists of two activities: Specify Requirements and Analyse New Features.

This phase occurs when it is necessary to implement new features or adapt existing ones which cannot be satisfied through a configuration of the SMART Platform of Core Assets. When new features are identified, the New Features Analyst interacts with the Platform Architect, in order to analyse and approve feature development. The approved feature is then classified as: Specific Feature, Reactive Feature or Proactive Feature.

Each activity is detailed in Table 3.

Activity	Purpose
Specify Requirements	The new features for the customer's product are specified and detailed. The *New Features Analyst* or *Requirements Analyst* can specify the new features during the *Modelling* activity or the *Analysis* phase. In order to do it, they make observations, collect documentation and interview potential system users.
Analyse New Features	Customers can request requirements which are not supported by the platform. When new features are identified, the *New Features Analyst* interacts with the *Platform Architect*, in order to analyse and approve the feature development. The approved feature is then classified as: *Specific Feature*, *Reactive Feature* or *Proactive Feature*. The features are classified according to: potential for reuse, its level of complexity,

Table 3. Analysis Activities.

3.4 Customization

During the Analysis phase, features are classified in categories to define their implementation form.

Thus, features classified as *Specific Feature* will be implemented by the *Product Development Team*. On the other hand, new features classified as either *Reactive Feature* or *Proactive Feature* are implemented by the Platform Development Team using *Reactive Product Customization* or *Platform Evolution*.

Figure 1 shows the relationship between features classification, product customization approach and parameters analysed.
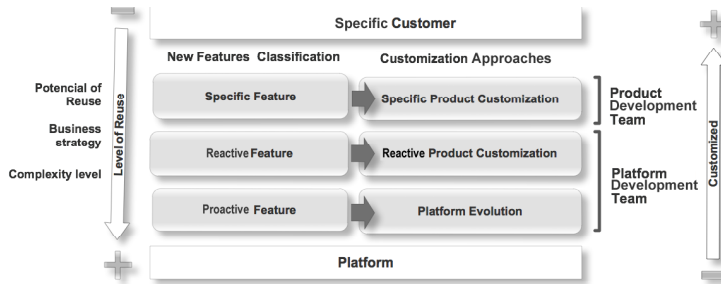
Fig.1. Relationship between feature classification, product customization approach and analysis parameters. ".

In *Reactive Product Customization*, new features are implemented with incorporated configuration mechanisms that are configured only in the specific product. *Reactive Product Customization* results in platform evolution without impact to other customers. These *configuration mechanisms* enable the configuration of variants for instantiation of the reusable assets.

Similarly, in the *Platform Evolution*, new features with incorporated configuration mechanisms are implemented and integrated into the *SMART Platform of Core Assets*. These features are then available to other customers as new updates. For *Specific Product Customization*, new features are implemented and integrated in a specific product without configuration mechanisms and therefore cannot be integrated within products of other customers.

Each activity is described in Table 4.

Activity	Purpose
Specific Product Customization	To perform specific product customization including testing and integration of components. During the process, the *Product Developer* is responsible for component customization. This includes development of new components or adaption of existing platform components. Typically, this type of customization is necessary only for a specific customer and therefore is not integrated with the platform. During *Specific Product Customization*, developers select components which will be adapted to the customer requirements and adjustments are made to each component to accommodate customer's needs. After customization, the components are tested and integrated to compose the customized product.
Reactive Product Customization and Platform Evolution	Implement or adapt reusable software assets based on the customer's needs. Both *Reactive Product Customization* and *the Platform Evolution* activities are performed by the *Platform Development* Team. The customization process is similar for these two activities with exception of *the Integration of Components*. Once implemented, tested and documented, the new features will be compiled and integrated. However, for the *Reactive Product Customization*, the parameters for configuration, created during component implementation, will be enabled in this specific product. During component customization, the *Platform Development Team* interacts with the Platform Architect to obtain detailed information about core asset evolution and the

constraints of the Platform Architecture. The *Platform Development Team* implements new features based on customer requirements. In order to implement reusable new features, the component code incorporates a *parameterization mechanism*. This enables the selection of variants that enable the reusable components.

After implementation and testing, the new feature is documented in the *Core Assets Documentation*. This includes a description, characteristics of its use, and configuration parameters necessary to enable it. Finally, a release of the module with the new feature is generated. It is integrated in the *SMART Platform of Core Assets*, and released to a specific customer (if the request has been classified as a *Reactive Feature*) or to other company's customers (if the request has been classified as a *Proactive Feature*).

Table 4. Customization Activities.

3.5 Deployment

The *Deployment* phase occurs incrementally, as each customer feature supporting a module is selected, it is installed and configured. This phase involves three activities: *Instantiate Database, Configure Product* and *Simulation*.

Each activity is described in Table 5.

Activity	Purpose
Instantiate Database	A database is selected from the *Database Templates*. The *SMART Platform of Core Assets* has five *Databases Templates*, where each model supports a specific medical speciality and can be customized according to the customer's needs. With the mapping complete (Modelling activity), the *Database Analyst* selects the database, which best fits the customer domain. After this selection, the *Database Analyst* adjusts the base model to support customer needs. Then, the database is installed and available for data entry.
Configure Product	The product is assembled from reusable assets, which are built by reusing existing platform assets, implementation of non-existing assets in the platform, or implementation of product-specific assets. Iteratively and incrementally, as each module is configured, it is installed and configured. This process continues until all modules which compose the *Specific Product* have been configured and integrated. The *Technical Deployment* instantiates each module and configures the correspondent parameterizations. Thus, from this configuration, the variabilities are resolved and a *Specific Product* is derived based on the customer's requirements.
Simulation	Certifies the data and detects whether there are non-conformity within the product derived. During this activity, the users run the system using as parameters the workflows mapped. Thus, the final product is validated.

Table 5. Deployment Activities.

4 Discussion

The paper reports the results of the case study methodology application to elicit the PD process and understand how it is performed within an industrial setting. In order to maximize the benefits from the available sources of evidence, this study followed three principles as defined by Yin [11]: (i) use of multiple sources of evidence; (ii) creation of a case study database; and, (iii) maintenance of a chain of evidence. In this way, we applied three different data collection methods: interviews, documentation, and participant-observation [11]. These data collection methods allowed us to triangulate the evidence, increasing the precision of the empirical research. The approach allowed us to looking at

the studies outcomes in different ways, capturing a set of interesting insights and issues.

Based on the number of customers, features reused and tests performed, the company platform was considered stable. However, regarding to the organization maturity level, none specific CMMI (Capability Maturity Model Integration) [10] process area was found. The process provides an iterative form of product derivation which enables product assembly to occur incrementally. This is one of the important characteristics for a product derivation approach as highlighted by Rabiser et al. [8].

The case study process proposes some interesting insights for product derivation, such as: the product derivation process begins during the sales process, the use of workflow mapping for elicitation of customer requirements, the analysis and implementation of new features in a SPL environment, the use of incremental deployment, and the role of training as part of the product derivation process.

Finally, although the process is deemed effective with the case study, we identified some issues. Firstly, the process is not formally described which can lead to confusion as to individual responsibilities. Secondly, there is no standardization of tools across the domain and application engineering teams. This can make it difficult to re-allocate staff occurring to organizational needs. Finally, the process is heavily dependent on expert knowledge particularly for control of dependences among core assets, traceability, and variability management. There is a high risk of losing important process and technical knowledge if these experts leave the company.

From a comparative analysis with [8], we observed that the MedicWare process provides full or partial support for the key activities in product derivation. The analysis we conducted showed that the details provided by the process and how each activity is performed can be used as a basis for building or improving existing product derivation approaches.

5 Related work

Rabiser et al. [7] identified that there is a growing interest by researchers and practitioners in product derivation. However, there is a lack of research reporting how software development organizations derive their products from a product line and the associated problems [8],[13].

Two relevant industrial case study reports on product derivation were published [6] and [3]. In the first, the authors present five problems and three issues associated with product instantiation. In the second one, the goal was to investigate the source of problems associated with the derivation of individual products from shared software assets.

These studies can be considered good sources of information in the area. However, important aspects associated with the industrial product derivation process and practice [20] were not covered.

Our study presents the results of a case study performed in industrial environment, describing how a product derivation process occurs and what practices are used. The study definition and reporting was structured based on [9] and [11], according to well-defined guidelines which allows the study replication and extension.

6 Conclusion

This paper presents the results of an exploratory case study on a SPL company working in the healthcare domain. We investigate industrial product derivation practices and document our findings. In particular, this paper provides further knowledge of the product derivation area. We present knowledge of industry product derivation. This paper provides empirical findings to demonstrate industrial product derivation practices.

The case study process proposes industrial insights on product derivation, such as: the product derivation process begins during the sales process, the use of workflow mapping for elicitation of customer requirements, the analysis and implementation of new features in a SPL environment, the use of incremental deployment, and the role of training as part of the product derivation process. Finally,

although the process is deemed e effective with the case study, we identified some issues.

The findings presented can serve as a comparison for product derivation reporting. Researchers can use this work as a basis for defining, adapting or evaluating their product derivation approaches. Moreover, we expect that other researchers can use our work as a starting point for new industry reports, presenting their experiences with product derivation.

Acknowledgements

7 Literature

1. J. McGregor, L. Northrop, S. Jarrad, K. Pohl, Initiating software product lines, Software, IEEE 19 (4) (2002), pp. 24-27.

2. L. Hotz, A. Günter, T. Krebs, H. C. Fachbereich, A knowledge-based product derivation process and some ideas how to integrate product development, in: in of Software Variability Management Workshop, (2003), pp. 136-140.

3. S. Deelstra, M. Sinnema, J. Bosch, Product derivation in software product families: a case study, J. Syst. Softw. 74 (2005), pp. 173-194.

4. M. L. Griss, Implementing product-line features with component reuse, in: Proceedings of the 6th International Conerence on Software Reuse: Advances in Software Reusability, Springer-Verlag, London, UK, (2000), pp. 137-152.

5. R. Rabiser, P. Grunbacher, D. Dhungana, Supporting product derivation by adapting and augmenting variability models, in: Proceedings of the 11th International Software Product Line Conference, IEEE Computer Society,Washington, DC, USA, (2007), pp. 141-150.

6. J. Bosch, M. Högström, Product instantiation in software product lines: A case study, in: Proceedings of the Second International Symposium on Generative and Component-Based Software Engineering-Revised Papers, Springer-Verlag, London, UK, (2001), pp. 147-162.

7. R. Rabiser, P. Grünbacher, D. Dhungana, Requirements for product derivation support: Results from a systematic literature review and an expert survey, Inf. Softw. Technol. 52 (2010), pp. 324-346.

8. R. Rabiser, P. O'Leary, I. Richardson, Key activities for product derivation in software product lines, J. Syst. Softw. 84 (2011), pp. 285-300.

9. P. Runeson, M. Höst, Guidelines for conducting and reporting case study research in software engineering, Empirical Softw. Engg. 14 (2009), pp. 131-164.

10. Capability maturity model integration (cmmi) homepage,.(2010) [Online]. Available: http://www.sei.cmu.edu/cmmi/.

11. Yin, R. K., Case Study Research: Design and Methods (Applied Social Research Methods). Sage Publications, Inc.(2008).

12. Wohlin, C., Runeson, P., H¨ost, M., Ohlsson, M. C., Regnell, B., Wessl´en, A.. Experimentation in software engineering: an introduction. Kluwer Academic Publishers (2000).

13. Rabiser, R., Gr¨unbacher, P., Dhungana, D. Requirements for product derivation support: Results from a systematic literature review and an expert survey. Inf. Softw. Technol. 52, (2010) 324–346.

8 Author CVs

Leandro Oliveira de Souza

M.Sc. Leandro Oliveira de Souza is an assistant professor at Bahia Federal Institute of Education, Science and Technology (IFBA) and researcher at the Reuse in Software Engineering (RiSE) group based in the Federal University of Bahia (UFBA). He has a M.Sc. in Computer Science from Federal University of Pernambuco (UFPE), Brazil. Contact him at los2@cin.ufpe.br.

Pádraig O'Leary

Dr. Pádraig O'Leary is a Research Fellow at the University of Limerick, Ireland. He is a member of Lero - the Irish Software Engineering Research Centre and ARCH – the Centre for Applied Research in Connected Health. Contact him at padraig.oleary@ul.ie

Eduardo Santana de Almeida

Dr. Eduardo Santana de Almeida is an assistant professor at Federal University of Bahia and head of the Reuse in Software Engineering Labs. Contact him at esa@dcc.ufba.br.

Silvio Romero de Lemos Meira

Dr. Silvio Romero de Lemos Meira is a full professor at Federal University of Pernambuco and chief scientist at the Recife Center for Advanced Studies and Systems. Contact him at silvio@cesar.org.br.

Transforming Requirements of IEC 80001-1 into an ISO/IEC 15504-2 Compliant Process Reference Model and Process Assessment Model

Silvana Togneri MacMahon, Fergal Mc Caffery, Frank Keenan

Regulated Software Research Centre, Department of Computing and Mathematics
Dundalk Institute of Technology, Co. Louth Ireland
{Silvana.MacMahon, Fergal.McCaffery, Frank Keenan}@dkit.ie

Abstract

Efficiencies in patient care can be achieved through interoperability of medical devices. Patient safety is the key concern during the design and manufacture of medical devices with medical devices being subject to stringent regulation in the region in which the device is to be marketed. However, with medical devices increasingly being designed to be incorporated into an IT network, the process of networking the device can introduce risks that may not have been considered during the design and manufacture stage. IEC 80001-1 was developed to address the risks associated with the incorporation of a medical device into an IT network. This paper presents how the requirements of IEC 80001-1 were used to develop a Process Reference Model (PRM) and Process Assessment Model (PAM) which are compliant with the requirements for PRMs and PAMs as outlined in ISO/IEC 15504-2.

Keywords

IEC 80001-1, ISO/IEC 15504-2, ISO/IEC 20000-1, Process Assessment, Risk Management, Medical IT Networks

1 Introduction

In 2003 and 2004, the FDA received reports of a cluster of cyber-attacks on hospitals. The attacks acted as a catalyst for the FDA to produce cyber security guidance for medical device manufacturers for networked medical devices containing off the shelf software [1]. Having developed this guidance to address the cyber security risks, it was recognized that the wider area of risk management of networked medical devices needed to be addressed in a more comprehensive way. Traditionally if a medical device was to be incorporated into a network, the device manufacturer would provide the device and the network. This led to a situation where a hospital could have a plethora of self-contained private networks. In order to allow true interoperability of devices and achieve efficiencies in patient care, medical devices are increasingly being developed to be incorporated into the general IT network of the Healthcare Delivery Organisation (HDO). These networks can carry traffic from life critical patient information to general email traffic. The incorporation of a medical device into the HDOs general network creates a medical IT network which can introduce risks that may not have been considered during the design and manufacture of the device [2] . To address the risks which are specific to the incorporation of a medical device into an IT network, it was recognized that guidance would need to be addressed not only to the manufacturer of the device, but also to the HDO who are responsible for the establishment and maintenance of the medical IT network (referred to in the standard as Responsible

Organisations (RO)) and also to the providers of the HDO networks and other information technology. Guidance in this area would need to focus on promoting a high level of communication among these groups, but also among the various risk management stakeholders within these groups such as IT and clinical staff within a HDO [2]. This was to be the origins of IEC 80001:1 2010 Application of risk management for IT-networks incorporating medical devices - Part 1: Roles, responsibilities and activities [3]. Section 2 of this paper discusses the requirements as described in IEC 80001-1. Section 3 of this paper discusses the development of the approach to the PRM and PAM while Section 4 discusses the step by step approach of how the requirement of IEC 80001-1 were transformed into the resultant ISO/IEC 15504-2:2003 [4] compliant PRM and PAM. Section 5 presents the conclusions of this paper and future work to be carried out in this area.

2 IEC 80001-1 – Application of Risk Management for IT Networks Incorporating Medical Devices

IEC 80001-1 addresses 3 key properties of a medical IT network – Safety, Effectiveness and (Data and System) Security. Safety is freedom from unacceptable risk of physical injury or damage to the health of the patient or the user of the device or damage to property or the environment. Effectiveness focuses on the ability of the networked device to provide the intended result both for the patient and for the RO. Data and System Security ensures that information assets are reasonably protected from degradation in terms of confidentiality, integrity and availability. IEC 80001-1 takes a lifecycle approach to risk management and applies when a medical IT network is established, when a medical device is added or removed from a network or during any modification or maintenance activities. The lifecycle approach requires the appointment of an appropriately qualified medical IT network risk manager who will ensure that a risk management policy is established and documented and that all risk management activities throughout the lifecycle of the network are carried out in accordance with the risk management policy. All documentation which is produced during the performance of risk management activities must be maintained within the medical IT network risk management file. Roles and responsibilities for each of the stakeholder groups involved in the performance of risk management activities are detailed within IEC 80001-1.While IEC 80001-1 provides guidance on the performance of risk management activities, there is no method available which can be used to allow ROs to assess the capability of their risk management practices with regard to the requirements of IEC 80001-1. Our research to date has focused on the development of a PRM and PAM for IEC 80001-1.

3 Approach to the Development of the PRM and PAM

To develop the PRM and PAM for IEC 80001-1, a review of the following areas was undertaken:

- A detailed review of the requirements of IEC 80001-1
- A review of Process Assessment standards focusing on ISO/IEC 15504-2 and ISO/IEC 15504-5 [5]
- A review of Process Assessment models that have been developed to assess against similar standards and how they were developed

The approach to the review of the requirements of IEC 80001-1 is detailed in section 4 below which was undertaken using the Tudor IT service management Process Assessment (TIPA) [6] transformation process. The TIPA transformation process is a goal oriented requirements engineering technique which was developed by CRP Henri Tudor to develop the TIPA framework which is used to assess service management processes. The TIPA transformation process provides guidance on how to transform domain requirements into PRMs and PAMs which are compliant with the requirements of ISO/IEC 15504-2 and ISO/IEC TR 24774 [7]. TIPA can be used to assess the capability of Service Management processes against the requirements of ISO/IEC 20000-1 [8] or the IT Infrastructure Library (ITIL) [9]. The TIPA transformation process was analysed during a review of models which have been developed for similar standards for its ability to be applied to the requirements of IEC 80001-1.

The TIPA transformation process has been used for the development of the PRM and PAM for IEC 80001-1 due to the similarities between IEC 80001-1 and ISO/IEC 20000-1 which are identified in Annex D of IEC 80001-1. Both IEC 80001-1 and ISO/IEC 20000-1 take a lifecycle approach to addressing the requirements of the standard. Annex D of IEC 80001-1 details process areas which are common to both standards such as "Configuration Management" and also highlights areas where while the terminology appears to be different the underlying role, document or process is similar. The TIPA transformation process is discussed in detail in section 4

4 Development of the PRM and PAM using the TIPA transformation process

4.1 The TIPA Transformation Process

The TIPA transformation process is a goal oriented requirements engineering technique. The TIPA transformation process was developed in recognition of the fact that while ISO/IEC 15504-2 is detailed in its description of the requirements for PRMs and PAMs, it does not provide guidance on how to transform the input - the domain requirements into the output – the PRM and PAM [10]. The transformation process advocates identifying elementary requirements and organising these requirements into requirement trees. These requirement trees are then then oriented around the business goals to which they are related to form goal trees. The transformation process uses the requirements of ISO/IEC 15504-2 [4] combined with the requirements of ISO/IEC TR 24774 to develop the final PRM and PAM. ISO/IEC TR 24774 Systems and software engineering - Lifecycle management - Guidelines for process description is a standard which provides guidelines for the elements used most frequently in describing a process as a means to ensuring consistency in standard process reference models. The guidelines expressed in this standard can be applied to any process model developed for any purpose.

The steps in the TIPA transformation process are summarised below:
1. Identify elementary requirements in a collection of requirements.
2. Organise and structure the requirements.
3. Identify common purposes upon those requirements and organise them towards domain goals.
4. Identify and factorise outcomes from the common purposes and attach them to the related goals.
5. Group activities together under a practice and attach it to the related outcomes.
6. Allocate each practice to a specific capability level.
7. Phrase outcomes and process purpose. (Apply ISO/IEC TR 24774 guidelines)
8. Phrase the Base Practices attached to the Outcomes. (Apply ISO/IEC TR 24774 guidelines)
9. Determine Work Products among the inputs and outputs of the practices.

The TIPA transformation process was used in the development of the PRM and PAM which will be discussed in the next two sections of this paper.

4.2 Development of the PRM

To provide a template to inform the development of the PRM for IEC 80001-1, the PRM for ISO/IEC 20000-1 which is contained in ISO/IEC TR 20000-4 [11] was reviewed. The document was reviewed to assess if the set of processes contained within the PRM for ISO/IEC 20000 could be used to assess against IEC 80001-1. While both standards follow a lifecycle approach, the processes detailed within ISO/IEC 20000-4 do not adequately address the aspects of risk management that are particular to the incorporation of a medical device into an IT network. On this basis, ISO/IEC 20000-4 was used to inform the structure of the PRM for IEC 80001-1 while not using the same set of processes. In reviewing ISO/IEC 20000-4, it was clear that the lifecycle approach of using a "Plan, Do, Check, Act" ap-

proach, as used in ISO/IEC 20000-4, could also be used to address the lifecycle approach advocated in IEC 80001-1. This approach has been maintained as illustrated in Figure 1.0.
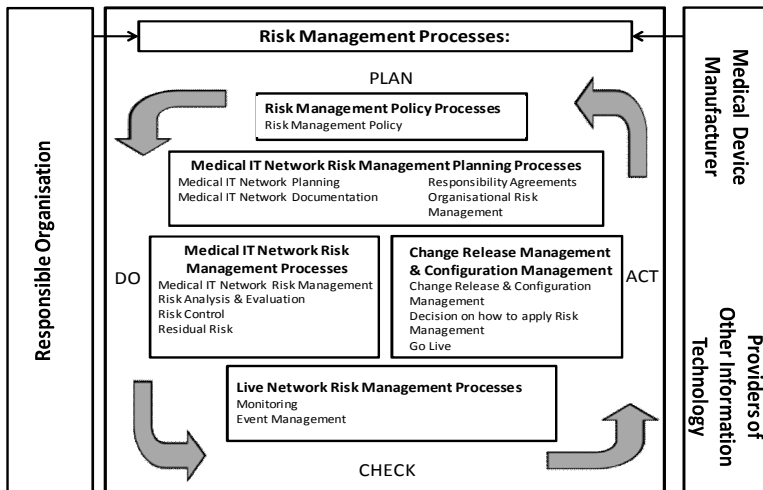

Figure 1.0 IEC 80001-1 Processes

Using the ISO/IEC 20000-4 as a template, the next stage of the development of the PRM was to structure the requirements according to the TIPA transformation process. Step 1 of the transformation process requires that elementary requirements are identified within the collection of requirements. In order to isolate elementary requirements, IEC 80001-1 was reviewed line by line and each item that was considered to be a requirement was identified and placed in a requirement catalogue. Elementary requirements have a single verb, object and complement and do not contain conjunctions. One hundred and sixty one requirements were initially identified. In order to maintain traceability, the source of each requirement was noted making reference to the section of the standard and the line number. The requirements catalogue was updated as required to ensure that only requirements that would form the PRM were included. For example initially requirements were noted that prescribed that certain actions should be carried out by the holder of a specific role. However these requirements had to be updated in order to comply with the requirements of ISO/IEC 15504-2 that requires that processes within the PRM are defined in term of the purpose and the outcome of the process and are not concerned with who performs the process.

Once all elementary requirements had been identified, the next step in the TIPA transformation process is to organize and structure the requirements. The process groups within ISO/IEC 20000-4 were reviewed to understand if these groups could also be used to structure the requirements for IEC 80001-1. While there are some process areas which are common to both standards (e.g. Release Management and Configuration Management), the structure of the processes needed to be adapted to take into account that IEC 80001-1 solely contains requirements for risk management activities throughout the lifecycle while risk management is a single process in the lifecycle approach to Service Management within the ISO/IEC TR 20000-4 PRM. Various approaches were taken to the organization of the requirements. The approach that was considered most suitable was to follow the structure of the IEC 80001-1 standards and to use the different sections of the standard to isolate the domain goals which would eventually form the processes. The domain goals informed the definition of the process purpose. In structuring the requirements in this way, step 3 of the transformation was completed simultaneously. The requirements of ISO/IEC TR 24774 were also considered during this stage and process descriptions were formulated accordingly.

Having defined the processes and process purpose through steps 1, 2 and 3, step 4 focused on the definition of the outcomes to be attached to each of the identified process purposes. A process outcome is a measurable, tangible technical or business result that is produced as a result of the performance of the process. In order to ensure the completeness of the list of outcomes associated with any process, the complete set of outcomes were reviewed to ensure that the achievement of all of the

outcomes would result in the fulfillment of the process purpose. This step was completed in conjunction with step 7 of the transformation process which ensures that outcomes and purposes of the process are phrased in a manner which is compliant with the requirements of ISO/IEC TR 24774. The completion of steps 1, 2, 3, 4 and 7 were necessary for the development of the PRM. The remaining steps of the transformation process are associated with the development of the PAM.

4.3   Development of the PAM

In order to develop the PAM for IEC 80001-1, the process within the PRM are extended with the addition of a measurement framework. This framework consists of 5 levels which range from "Incomplete" to "Optimising" and is defined in ISO/IEC 15504-2. In order to be able to make an assessment against this measurement framework, the remaining steps of the TIPA transfomation process were carried out to complete the development of the PAM. Step 5 of the TIPA transformation process consists of grouping activities under a practice. In order to complete this step, the process outcome and purpose were reviewed and a practice was defined that would result in the production of each outcome. Practices consist of base practices and generic practices. A base practice is an activity that addresses the purpose of a particular process. Base practices are also process performance indicators that indicates the extent of achievement of the process purpose and process outcomes. Generic practices are the principal indicators of process capability and practices that are established to support the process performance as it is characterized at level 1.

Once the practices had been identified, to complete step 6, each of the identifed practices was reviewed and was assigned to a specific capability level. This was done by reviewing the outcome of each parctice and the effect its perfomrnce would have on the process purpose. On the basis of this review, each practice was determined to be either a base practice, and related to a process performance and a capability level of 1, or a generic practice, and related to process capability and the achievement of a capability level upper than 1. For generic practices, the specific capability level was detemined by a review of the 9 process attributes associated with each capability level. Capability levels are based on the achievement of these process attributes. Process attributes and the evidence required to achieve them are defined in Clause 4.3.2 of ISO/IEC 15504-5. Capability levels are detailed in Table C.2 of Annex C of the IEC 80001-1 PAM which detailes the association of IEC 80001-1 requirements with capability levels and base practices.

Step 8 requires that base practices were linked to the outcome that would be achieved by the performance of the practice. These practices should be phrased according to the requirements of ISO/IEC TR 24774 and it should be noted that a single practice may produce and therefore be linked to a number of outcomes. The previous steps were completed for each of the 14 processes resulting in the identification of 70 base practices.

Having identified the practices associated with each of the proceses, step 9 required that each of the processes were reviewed in order to determine work products among the inputs and the outputs of the practices. A review was undertaken of specific and generic practices contained in ISO/IEC 15504-5. Applicable work products were used within the IEC 80001-1 PAM with additional work products related to specific risk management activities as per the requirements of IEC 80001-1 being added as required. A list of generic and specific inputs and outputs and their characteristics are contained within Annex B of the IEC 80001-1.

The competion of all 9 steps of the TIPA transformation process allowed the domain requirements as expressed in IEC 80001-1 to be developed into a PRM and PAM. A sample process from the PAM is shown in Table 1.0. The resultant PRM and PAM are compliant with the requirements of ISO/IEC 15504-2 and ISO/IEC TR 24774. The PAM can be used for assessment against IEC 80001-1 and can be used to determine the capability levels of risk management processes for the incorporation of medical devices into an IT network. These capability levels can then be used as a basis for process improvement which will inturn increase the safety, effectiveness and security of the medical IT network.

Process ID:	CRCM.3
Name:	Go-Live
Context:	The process is to allow the responsible organisation to manage the Go-Live Phase of the project and to consider the decision to go live in terms of the residual risk.
Purpose:	The purpose of the Go-Live Process is to allow the responsible organisation to manage the transition of the IT network to the live environment and to allow the responsible organisation to manage the risk management activities associated with the Go-Live phase of the project.
Outcomes:	As a result of the successful implementation of Go-Live Process: 1. Medical IT-network residual risk is reviewed prior to going live. [[IEC 80001-1, 4.5.3]]. 2. Residual risk summaries are reviewed for acceptability of risks associated with interactions of recent or pending projects or changes. [[IEC 80001-1, 4.5.3]]. 3. The specified change to the medical IT-network is approved prior to go-live by the medical IT-network risk manager. [[IEC 80001-1, 4.5.3]]. 4. The approval of the medical-IT network residual risk is documented in the medical IT-network risk management file. [[IEC 80001-1, 4.5.3]].
Base Practices:	CRCM.3.BP1: Review residual risk. Review Medical IT Network residual risk summaries for acceptability of risk associated with interactions of recent or pending projects or changes, prior to going live. [IEC 80001-1, 4.5.3] [IEC 80001-1, 4.5.3] [Expected Result: 1, 2]. CRCM.3.BP2: Approve specified change. Approval is given for the specified change by the medical IT Network Risk Manager prior to go-live. [IEC 80001-1, 4.5.3] [Expected Result: 3]. CRCM.3.BP3: Document approval of residual risk. Document the approval of the medical IT Network residual risk in the Medical IT network risk management file. [IEC 80001-1, 4.5.3] [Expected Result: 4].
Inputs:	
13-03 Risk Benefit Analysis Record [CRCM.3, BP1, 2] [Expected Result 1, 2, 3]	
Outputs:	
08-02 Change Request Approval Record [CRCM.3, BP.2, 3] [Expected Result 3, 4] 16-02 Medical IT network Risk Management File [CRCM.3, BP.3] [Expected Result 4]	

Table 1.0 – Sample Process from IEC 80001-1 PAM

5 Conclusions and Future Work

The focus of research to date has been on the development of an ISO/IEC 15504-2 compliant PRM and PAM for assessment against IEC 80001-1. This will allow HDOs to assess the capability of their risk management processes against the requirements of IEC 80001-1 with regard to the incorporation of a medical device into an IT network which can then be used as a basis for process improvement. The PRM and PAM which have been developed as part of this research have been presented at a meeting of IEC SC62A JWG7 in September 2012. The PRM and PAM have been raised as a new work item proposal and will be published as a technical report as part of the IEC 80001-1 family of standards. This will establish the PAM as the standards method of assessment against IEC 80001-1.

Future work in this area will focus on the development of an assessment method for IEC 80001-1. A PAM cannot be used in isolation to perform an assessment against IEC 80001-1. To allow an assessment to be carried out, an assessment method will be developed which will allow for a standardised approach to performing the assessment and provides a set of questions which will allow a capability level to be determined for each of the practices related to the processes.

The PRM, PAM and assessment method will be validated in a number of ways. The PRM and PAM will be validated for structure and content with regard to their ability to assess against IEC 80001-1 requirements and their compliance with the requirements of ISO/IEC 15504-2. This validation will be conducted through expert opinion by eliciting feedback from the developers of the TIPA framework and through the resolution of comments made by members of JWG7 during the comment resolution phase of the technical report. Validation of the PRM, PAM and assessment method will also be carried out from the HDO perspective and medical device manufacturer perspective. Validation will be performed form the HDO by mapping of the processes within the PRM and PAM processes to a previously implemented network project within a large ICU and in a smaller clinical context. A medical device manufacturer will be asked to provide feedback on the processes which are addressed to medical device manufacturers. The research will take a design research approach by placing the artifact, in this case the PRM, PAM and assessment method in the context in which they will ultimately be used in order to assess their effectiveness. The required changes, based on feedback during the validation process, will be incorporated into the final model.

Acknowledgements

6 Author CVs

Silvana Togneri MacMahon

Silvana Togneri MacMahon received a Higher Diploma in Science in Computing in 2011 from Dundalk Institute of Technology having worked as a User Acceptance Test Lead in the Retail Banking sector. She is currently undertaking research for her PhD in the area of risk management of networked medical devices focusing on the development of a process reference model, process assessment model and assessment method for IEC 80001-1.

Fergal Mc Caffery

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

Frank Keenan

Dr Frank Keenan lectures in Software Engineering in the Computing Department at Dundalk Institute of Technology. His current research interests include requirements engineering, agile development and context analysis. A particular interest is the transfer of knowledge gained to form innovative delivery approaches to subjects taught.

7 *References*

[1]   *Guidance for Industry - Cybersecurity for Networked Medical Devices Containing Off-the-Shelf (OTS) Software,* Food and Drug Administration, 2005.

[2]   T. Cooper*, et al.*, *Getting Started with IEC 80001: Essential Information for Healthcare Providers Managing Medical IT-Networks*: AAMI, 2011.

[3]   IEC, "IEC 80001-1 - Application of Risk Management for IT-Networks incorporating Medical Devices - Part 1: Roles, responsibilities and activities," ed. Geneva, Switzerland: International Electrotechnical Commission, 2010.

[4]   ISO/IEC, "ISO/IEC 15504-2:2003 - Software engineering — Process assessment — Part 2: Performing an assessment," ed. Geneva, Switzerland, 2003.

[5]   ISO/IEC, "ISO/IEC 15504-5 - Information technology — Process Assessment — Part 5: An exemplar Process Assessment Model," ed. Geneva, Switzerland, 2006.

[6]   B. Barafort*, et al.*, *ITSM Process Assessment Supporting ITIL : Using TIPA to Assess and Improve your Processes with ISO 15504 and Prepare for ISO 20000 Certification* vol. 217. Zaltbommel, Netherlands: Van Haren, 2009.

[7]   ISO/IEC, "ISO/IEC TR 24774:2010 - Systems and software engineering — Life cycle management — Guidelines for process description," ed. Geneva, Switzerland, 2010.

[8]   ISO/IEC, "ISO/IEC 20000-1:2011 - Information technology —Service management Part 1: Service management system requirements," ed. Geneva, Switzerland, 2011.

[9]   The Cabinet Office, "ITIL 2011 - Summary of Updates," ed. Norfolk, England: Crown Copyright, 2011.

[10]   B. Barafort*, et al.*, "A transformation process for building PRMs and PAMs based on a collection of requirements – Example with ISO/IEC 20000," presented at the SPICE Nuremberg, Germany, 2008.

[11]   ISO/IEC, "ISO/IEC TR 20000-4:2010 - Information technology — Service management - Part 4: Process reference model," ed. Geneva, Switzerland, 2010.

Europe wide Industry Certification Using Standard Procedures based on ISO 17024 – European Job Roles

Michael Reiner[1], Richard Messnarz[2], Gabriele Sauberer[3]
[1]IMC FH Krems, Austria, michael.reiner@fh-krems.ac.at
[2]ISCN GesmbH/LTD, Austria & Ireland, rmess@iscn.com
[3] Termnet, Austria, gsauberer@termnet.org

Abstract

ECQA (European Certification and Qualification Association) is the result of a series of EU funded projects from 2005 – 2012. This included European projects such as EQN (European Quality Network, 2005 – 2007), EU Certificates Campus (2008 – 2009) and DEUCERT (Dissemination of EU Certification), the ECQA nowadays acts as an organization that is independent from funding. The members of ECQA are widely spread all over Europe and vary from universities to companies as well as individuals. ECQA is aimed at a demographic problem of education and training in the European Union. For people at 40 – 50 it is many years ago that they attended the university and required skills nowadays are changing every 2-3 years. This means that their skills get outdated and we experience in Europe a growing unemployment from the age of 45 upwards. Universities in Europe are currently not addressing this problem.

EQN developed a so called role based qualification concept where e.g. an existing software engineer (who studied informatics some 15 years ago) can identify job roles to upgrade so that he remains a value for the organization. He might receive additional industry qualification for e.g. a safety architect (additionally learning how to enrich existing software architecture with functional safety aspects). This way the person, would for instance, grow into a safety architect position and a younger person who knows more about new programming techniques gets his old position. EQN then developed certification mechanisms for this role based approach for university and industry educational partnerships.

EU Cert Campus collected about 15 job roles, structured the corresponding skills sets and established online services. The online services comprise skills browsing, skills assessment, and online training. So people from industry can attend job role based qualification training from the work place.

DEUCERT   established ambassadors for this new job role based qualification concept Europe and worldwide. DEUCERT also helped to create a critical mass of partners. At the moment we do have 26 job professions (ready and in progress) that are certified all over Europe and already outside of the European Union. The concept is meanwhile supported by approx. 60 universities and training bodies in Europe.

The processes of the ECQA are mapped onto the ISO 17024 international standard for the certification of persons.

ECQA supports industrial programs (e.g. AQUA: Certification of functional safety, six sigma, and Automotive SPICE skills with Automotive industry) and universities (e.g. BPM-HEIup: Certified Business Process Modeller as part of university lecturing programs).

Keywords: European qualification standards, European certification strategy, European exam systems, European learning portals.

1 Introduction

While university programs address people in the age of 19 - 26 they do not address the demographic problem of increasing unemployment for people older than 45 at all. We already experience in nearly all countries in the European Union the growing age of people and in parallel that the number of unemployed people older than 50 is increasing as well. This can lead to a high demographic risk because most economies also plan to increase the age of pension and this will not work if people do not stay employed with the growing age.

The project EQN [3],[4],[5] was a strategic network project in education in which industry and educational institutions from 13 countries did networking to propose a solution to this demographic problem. Also the project should deliver an initiative of innovation in education as well [6],[8].

This resulted in:

Job Role Based Qualification Strategies [5],[7]. While the universities teach for a domain (e.g. informatics engineer) the job role based qualifications are short courses which in 2-3 weeks re-qualify people in industry (access from the workplace) to sustain their value for the company and remain employed. Job roles are described in form of skills sets (similar to the skill cards in the Department of Trade and Industry in the UK), and skill sets are mapped onto training, exercises and tests.

If someone studies informatics engineer this takes approx. 5 years at a university. Assuming this as a basis a job role qualification is for instance to upgrade to a safety architect, software process improvement manager, etc.

Modular Certification. It was also assumed that people in industry at the work place are under time pressure and cannot do the whole training at once. They might do a part in year 1 and the rest in the next year. They (assuming the European mobility strategy) could do the first part from Spain the rest from a German work place.

Thus the skills sets were structured into skill elements and each skill element has so called performance criteria. Each certificate lists elements and a full certificate is achieved if all elements are passed.

European Mobility Strategy. If, for instance, in the current economic situation Spanish workers would move to Germany for employment and later would move to UK and then back to Spain, all certifications should be added up and accepted across all countries.

This led to the concept of so called lifelong learning accounts for people. Each person in Europe can register at the system and in one lifelong learning account (like an international bank account but for education) can do self assessment, achieve skills profiles, and receive training. The account sustains and is a central service across EU countries.

This infrastructure was later established in the EU Cert campus project.

Europe Wide Standardized Approach. To be able to roil out this strategy in all countries the design of standard guidelines was important. They were translated in all major European languages.

As a result of this the ECQA [3],[4],[5],[9],[10] has established a set of standard quality guidelines for

- Certification of a new job role
- Certification of training bodies
- Certification of trainers
- Certification of people

See www.ecqa.org and cooperation guidelines under downloads.

A job role committee is an international working group who signed the standard ECQA JRC agreement and annually maintain the skill card and test questions pool.

Attendees of courses do an ECQA based exam and receive an ECQA Certificate.

In the guidelines of ECQA the fulfillment of the ISO 17024 standard for certifications of persons has been considered. A mapping to this international standard can be found in the ECQA guidelines.

This standard process requires that training organizations and examiners are separated. ECQA supports that by

- Establishing a Europe wide exam system which generates tests randomly per person (each person gets a different test and cannot be controlled by the training body). The exam questions are assigned to skills elements of job roles.

- Using certified exam bodies who provide examiners to organize the exams.

- Automatic corrections through the ECQA test system so that none can interfere the tests and the results personally.

- Job Role Committees elaborate and annually update this pool of multiple choice test questions.

2 EUROPEAN WIDE INFRASTRUCTURE

To support the concept of lifelong learning accounts (a strategy developed in the EQN project) the EU Cert Campus project (above 20 partners from 18 countries) developed an infrastructure which supports standard procedures for lifelong account management, self ass4essment of skills, exams and administration of certificates [2],[4],[5],[9].

Once this infrastructure was developed and in place (now used by above 11000 professionals and managers in European industry) it was important to find a body maintaining these services. This resulted in the legal foundation of the ECQA (European Certification and Qualification Association, 2008).

ECQA is not for profit and the income model is based on the certification business. ECQA follows the processes of independent exams as outlined in ISO 17024 and earns a share of each certificate sold. The income is used to further maintain the lifelong learning accounts and exams infrastructure.

The members of ECQA (and owners) are more than 60 universities and training bodies from Europe.

The Notion of Skills Profiles. The future vision of EQN was that each person in Europe might have an educational card like a bank card. On the chip we store the skills profiles of the person. This strategy was called "skills-card" strategy for Europe. The original strategy stems from the EU funded FP project CREDIT, 1998 – 2011, where ISCN was the partner for the technical architecture.

In EQN and EU Cert this strategy was adapted to a skills profile online which a person can maintain in a private lifelong learning account. Instead of a physical skill card with a chip on the cards, there are now lifelong learning online accounts.

A skills profile (= skill card as designed by the EU project CREDIT) is a representation of the coverage of competencies of a person in different skills elements. Each job role contains skills elements, the coverage of skills is shown in form of a percent mark.

   Exam results Illustrating the Coverage of Skills Profiles. Exam results are displayed in the form of a skills profile, demonstrating the coverage per skills element. See Figure 1. To be certified you must reach 66% in each skills area.

Figure 1: Typical Profile as Result of an Exam

The exam system portals are continuously updated and new functions to guarantee stability of the system during exams is extended since more than 6 years now.

3   A SPACE FOR EUROPEAN LIFE LONG LEARNING

Once the ECQA [2] was founded and the exam procedures were in place the next problem to solve was to allow a smooth integration of a critical mass of LLP projects into the platform. The EU LLP program finances a few hundred VET projects per year. The strategy was to empower a selected set of these projects to structure the skills and learning results into ECQA standards and integrate them into the platform.

Knowledge and Learning Cluster for IT and Services. The partnership of ECQA decided that we cannot accept every LLP project or new job role. Firstly, they must satisfy the ECQA standards and quality criteria and secondly they should relate to a specific cluster. So it was decided that in the first years of rollout (2008 – 2014) we especially support the sector of IT and services.

Some of the universities who are ECQA members decided to not use the multiple choice tests but to extend the system to support the APL (Accreditation of Prior Learning) procedures in the industry-university partnerships for education.

Thus a further new developed software function is the assessment of prior learning function. Here a panel of advisers (assessor of skills) can review existing knowledge of a person and assess a specific skills element to be fulfilled. In this case the exam is only to be done for skills elements where the assessment did not show coverage of the skills.

The ECQA platform allows people from the work place to attend online skills portals, receive training, do exams, and receive a certification [3],[4],[5],[9].

Figure 2: Skills Browsing and Self Assessment

With one login people can register for one or more job roles. When they register for the job role they can browse the skills and do a self assessment.

The self assessment highlights in which areas the learner needs to fill some skills gaps. In these areas the system allows to connect to an e-learning system.

Figure 3: Integrated E-Learning Courses

The courses include multimedia lectures, learners notes, exercises and online discussions of the exercises.

The skill card structure defines units (competence areas), learning elements (learning objectives), and performance criteria (what the learner must be able to demonstrate). The learning approach is therefore "learning by doing" based and thus task based leaning is a method which ECQA supports.

Once the training has been performed the learner can register for an exam. The exam is computer generated, and will be corrected automatically. Exams are random generated and different per learner. Questions are generated from a Europe wide shared test questions pool.

The implemented learning approach is described in Figure 4
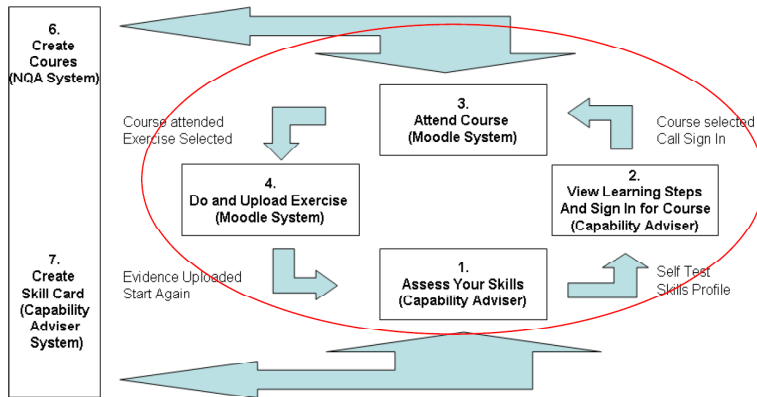
Figure 4: ECQA based integrated learning cycle

Figure 5: ECQA – Multiple Choice Based Exams

4 EUROPEAN DIMENSION and OUTLOOK

The current economic situation in Europe clearly illustrates that the demographic problem outlined in the abstract of the paper is a realistic scenario. The awareness that universities have a social responsibility to also think about solutions about how to upgrade skills of people older than 45 is increasing but still nearly non existing Europe wide. They still just focus on young people aged 19 – 26.

The concepts developed in EQN (2005 – 2007) and implemented with ECQA (as a not for profit initiative) represent a possible solution to this problem. This growing awareness leads to a growth and more and more universities and training bodies are expected to join.

The growth of the ECQA initiative is shown in Figure 7 where we see the growth trend of managers in Europe that have been ECQA certified after attending an ECQA test.

Training Bodies comprise universities, commercial training companies, chambers of commerce who became active providers of ECQA certified courses.

The described demographic situation is the major driver for growth in ECQA. Also by the number of many thousand certified people and their connections with other people in Europe we expect a wave of growing interest.

Also the number of LLP funded projects applying ECQA guidelines is currently increasing per year.

If you also plan to join this initiative please contact the ECQA president, Prof. Michael Reiner, University of Applied Sciences, Krems, Austria, Email: michael.reiner@fh-krems.ac.at.
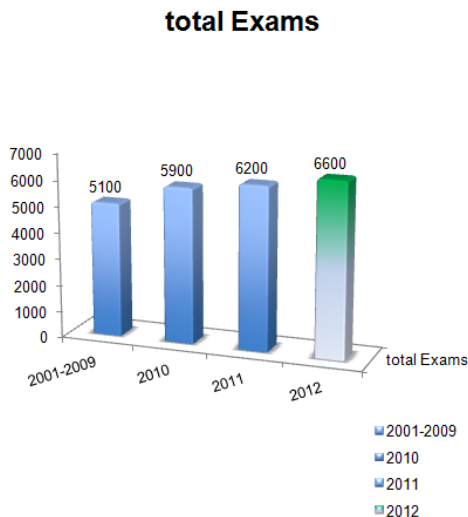
Figure 6: ECQA –Growth Trend of Exams in Europe

5 ACKNOWLEDGEMENTS

We are grateful to the European Commission who has supported the establishment of ECQA by vari-

ous EU projects such as CREDIT (1998 – 2001), EQN (2005 – 2007), EU Cert Campus (2008 – 2009), DEUCERT (2010 – 2011), BPM-HEI-up (2011-2013).

We are also grateful to all the ECQA members (currently 60 training bodies and universities) who helped to establish this qualification strategy in Europe.

This publication reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

6 Current ECQA Initiatives

Internationalisation. In 2012123 ECQA became a world wide certifier with active training bodies and certificates in Shanghai / China, South America, India, and meetings with Japanese industry and research groups to launch ECQA certificates in Asia.

So far the job role committees were set up by European representatives, in 2013 we started to extend the job role committees with representatives from Japan, India, South America, etc.
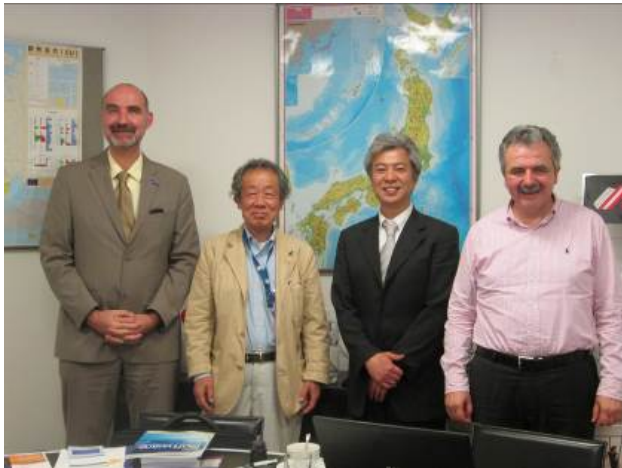
Figure 7: Meetings at Austrian Embassy Commercial Section in Tokio, Japan, May 2013

Impact on European Universities. Similar to the Microsoft certification program ECQA offers certificates where lecturing programs cover specific modules of an ECQA job role and students attend the exams. With BPM-HEIup an additional transfer of competencies was achieved by applying the ECQA certified business process modeling competencies for the optimization of university management processes.

Clustering Strategies. ECQA became part of strategic clustering projects where ECQA certificates are supporting clusters of industry groupings and representatives (e.g. AQUA – Automotive industry and Automotive Academies, QUALETRA – DG Justice and network of connected law institutions).

7 Literature

[1]   R. Messnarz, D. Ekert, M. Reiner, M.A. Sicilia, Europe Wide Industry Certification Using Standard Pro-
        cedures based on ISO 17024, in: Proceedings of the TAEE Conference, June 2012, IEEE Computer So-
        ciety, Vigo, Spain

[2]   European Union Lifelong Learning website, accessed on November 15th, 2011;

        http://ec.europa.eu/education/lifelong-learning-programme/

[3]   European Certification and Qualification Association, www.ecqa.org

[4]   R. Messnarz, From process improvement to learning organisations (p 287-294), Wiley SPIP Journal

        (Software Process Improvement in Practice), Volume 11 Issue 3 , Pages 213 - 335 (May/June 2006)

[5]   Richard Messnarz, Damjan Ekert, Michael Reiner and Gearoid O'Suilleabhain, Human resources based

        improvement strategies—the learning factor, Wiley SPIP Journal, Volume 13, Issue 3, (pages 355–362),

        July/August 2008

[6]   O'Keeffe, T., & D. Harrington, 2001. Learning to Learn: An Examination of Organisational Learning in

        Selected Irish Multinationals. Journal of European Industrial Training, MCB University Press, Vol. 25:

        Number 2/3/4

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

        Vocational Qualification Standards and Levels

[8]   Gemünden H.G., T. Ritter, Inter-organisational Relationships and Networks, Journal of Business Re-
        search, 2001

[9]   R. Messnarz, et. al, Assessment Based Learning centers, in : Proceedings of the EuroSPI 2006 Confer-
        ence, Joensuu, Finland, Oct 2006, also published in Wiley SPIP Proceeding in June 2007

[10]   ISO 17024, International Standardisation Organisation, Standard for the Certification of Persons

AGILE APPROACHES AND SPICE ORGANIZATIONAL MATURITY MODEL IN TURKISH SOFTWARE INDUSTRY

Kerem KEMANECİ
SPICE Assessor (Provisional)
TURKISH STANDARDS INSTITUTION
kkemaneci@tse.org.tr

Abstract

Making analyzes over the compatibility of agile approaches for Turkish Software Industry**.** Diagnosing the agile characteristics and agile minds of local sector gained agile characteristics and agile way of working with experiences from the past. Deciding agile approaches are necessary for software companies or not. Expectations from agile approaches in Turkish Software Industry. Applying agile methods and integrating agile approaches with SPICE Organizational Maturity Model, against domination of Total SMEs in Turkish Software Industry understanding the necessity and suitability of implementing maturity models.

Keywords

Agility, Agility skills of Turkish Industry, Deciding Appropriate Maturity Level, Agile Approaches, Scrum, XP, Extreme Programming, Agile Modeling Crystal, Waterfall, Plan Driven, Maturity Models, Requirements, Development Strategy, Software Quality in Turkey, SPICE, ISO 15504,Iterative Development, Technology Development Zones in Turkey, General Profile of SW companies in Turkey, SMEs, compatibility, maturity model, adaptability, expectations, CMMI.

1 Motivation

Quality Systems are implemented by many organizations in Turkey like Total Quality Management, Six Sigma for many years in all sectors for many years. IT sector in Turkey is very young with average ages of companies and with average ages of labor force.

There are roughly 2300 IT companies in Turkey. About % 51,5 of the number is just accepted as "micro" sized companies, 35,7% is accepted as "small" sized companies, %9,8 is accepted as "medium" sized companies and %3 of those enterprises are accepted as "large" sized companies. According to statistics of January 2013, there are 2174 companies (%53 IT Companies) in total 34 Technology Development Zones and total number of employee is about 19498 (16015 of those are R&D staff, rest for support services)[1]. This means an average company has about 9 employees. We can conclude that software development teams are very small. Thus teams can be managed to work coordinated and synchronized.

Turkish industry is very adaptable with special "know how" thanks to negative-positive experiences in very turbulent atmosphere of commerce. This makes people to be more agile personally and makes companies to be more agile to stand alive.

The growth rate of Local Turkish Software Industry is about % 75 in last 4 years and IT sectors average annual growth rate is % 14 [2]. But still Turkish Software Industry is lacking of sufficient investment funds to boost the growth [3].

Most of the software products are produced custom-made on purchase, so every software project is a new development. Reuse programs are limited. Great acquirers are mostly government institutions. This brings contracted and standard time scheduled projects, these conditions and also manner of contracts are based on plan driven project conditions.

By the time everything is getting more mature with experiences. The experiences from projects finished either successful or unsuccessful, experiences from management of human resources, experiences of marketing, experiences of planning, risking, appropriate usage of needed tools, lessons learned etc… Now companies are more intelligent, more professional, more mature, specializing in special business areas and emerging "know how"s from business areas and software projects. Turkish Software Industry is on the edge to next level, with its "likely to" agile software development and more mature companies.

2 Agility

In general, **agility** is defined as flexibility and shorter cycle times, to be able to respond changes effectively and quickly [4]. Flexibility gives a company adaptability skill against rapidly changing conditions and customer needs. Agility is a good skill against rapid changes of technology, financial conditions, human needs, new opportunities etc.

2.1 Shorter cycle times bring advantages like

- Processes get faster and respond times to customers get shorter.
- Time of reaching to working prototype phases become shorter.
- Gives more chance for revisiting every phase.
- By revisiting every phase earlier, defects will be detected earlier and cost of change will decrease relatively.
- Feedbacks from customers can be collected earlier since early delivery of working prototype is available, in that case cost of defects fall.
- Easier to collect correct requirements and easier to eliminate requirements those never to be used.
- Focusing on real functional requirements and producing real thing (what customer expected).

2.2   Agile Skilled Industry of Turkey

Dr. Can Erbil[1] "In trade Turkey is number one in terms of adaptability comparing to the world. If Turkey cannot sell jacket, they cut sleeves and sell as vest. If you don't buy they find someone else to sell. If some skirts remained in stocks, they sew the middle of skirt and converting to trouser and pushing into market in very short time, because they have been forced to work in high inflation conditions[2] in Turkey for 30 years. After all Turkey gained agility skills by lessons learned and experience gained in high inflation conditions. People did not know long term investments because no one was able to find long term credits so production and trade decisions always had to be for short term. As a nation we have advantages that we can act fast and agile in crisis' hazy conditions in terms of making right decisions. " [5]

With this paragraph, we can understand that in Turkish industry agility is already a weapon, but still have to implement agile software development approaches.

2.3   Agile Software Development Principles

The Agile Manifesto was published Feb. 2001 by 17 independent practitioners of different programming methodologies at a summit. Those people agreed on 4 main values and 12 principles underpinning those 4 values.

"We are uncovering better ways of developing software by doing it and helping others do it.

Through this work we have come to value: Individuals and interactions over processes and tools
Working software over comprehensive documentation

Customer collaboration over contract negotiation
Responding to change over following a plan

That is, while there is value in the items on the right, we value the items on the left more." [6]

There are Special Agile Development Methods like Scrum, Agile Modeling, Extreme Programming (XP), Crystal, RUP, and FDD etc. those are getting along with Agile Manifesto principles.

2.4   Key Agile Practices [7]

Incremental and multigrain planning: "Coarse" and "fine" grain plans developed and used to guide work. Continual refinement of plan: Plans are continually refined as new information is acquired.

Short iterative development cycles working closely with customer: To help ensure requirements are understood, work is done in short cycles using frequent customer feedback to aid course correction.

Daily team standup meetings with current work kept visible to the full team: Ensures team is communicating and staying on the agreed-to course.

Teams self-manage the work: Team members measure their own velocity/productivity and commit to work based on the team's measured performance.

Frequent delivery of working product to customer based on customer priorities: Helps team stay focused on customer high value items.

Time-boxing work: Schedules are maintained by reducing delivered functionality, if necessary.

---

[1] Dr. Can Erbil's (Economics - Education and Research, Adjunct Associate Professor (Boston College) interview published on Habertürk Newspaper in Turkey.
[2] Inflation rates are under %10 since 2004.

Team retrospectives: Team periodically reflects on its processes, frequently making improvements the team agrees can help their performance.

Rapid response to changing customer needs: By keeping the iterations short and continually communicating with the customer, the team is able to change priorities on shorter cycles, if required.

3 Using Agile Approaches in Turkish SW Industry

According to Agile Turkey's 2013 Software Productivity Report, %64 of Turkish software companies are using agile methodologies, %41 of those companies are using Agile Methodologies less than 2 years. Adding the companies, those never used but planning to use Agile Methodologies in future percentage is reaching %77. Agile development practices are not used widely, however those practices will take more place while usage of agile software development methodologies are reaching to maturity [8].

3.1 Agile Approaches

- There are learned agile and lean production skills from other industries so far:
Turkish companies are working for 30 years in turbulent economic conditions, forced to work adaptable in rapidly changing atmosphere.
- Turkish Software companies' average number of staff is estimated 9. (Technology Development Zones):
Agile Approaches require relatively small teams which should work co-located, collaborated, concurrent and well communicated.
- Young people and young companies in Turkish Software Industry:
Agile approaches are implemented easier by open minded younger people, because old habits die hard.
- Competent, trained staff:
Working in IT sector is special, only competent and smart staff can find a place in software development teams.

3.2 Agile Approaches, Necessary?

- High growth rates in Software and IT Sector:
High growth rates show us that change of company sizes, increasing competitor companies and difficulty of keeping teams fixed. Point is not only agile development but also agile companies.
- Rapid changes in country:
Growth rates are also high in other sectors. Software needs of 1, 5 million SMEs of Turkish Industry and increasing foreign investment… Obviously there would be great opportunities for software companies.
- Rapid Changes in Technology:
Rapidly changing technology leads to new requirements.
- Rapid Changes in needs:
In competitive environment needs of customers will always go higher. Companies have to improve products and services continuously to stay alive.
- Market is mostly running on custom-made purchases:
Building close relationships and well established communication channels is very important. SW Companies have to understand customer well and have to respond fast against customers requirements.
- Lack of funds for large investments and importance of Return on Investment (ROI):
Lack of sufficient funds make companies fight against budget challenges, so the target should be eliminating every unnecessary requirement which won't add any value to the product. Researches show that every step, every activity of development should bring value to the product. Cost of rework and

cost of defects should be decreased. Cost of quality should be a measure for companies. A study by Standish Group shows that in typical software systems 45 percent of the features never actually used and another 19 percent are only used rarely. [9]

"I know when i see it" (IKIWISI) law:

At the beginning before starting everything, it's very difficult to ask for requirements from customer [10]. Before prototyping and usage of the product there always be something missed or something to discard. IKIWISI says that customers will see the product early before the final delivery, and then customers can express ideas over prototypes better and describe what they want more accurate.

3.3   Agile Approaches, Expectations?

While applying agile approaches

- Lean thinking, simple design and killing waste of time and waste of effort.
- Eliminated and prioritized requirements and achieving simple design which is also easy to test against quality issues.
- Well established communication channels between development team members
- Close relationship with customers built on high trust.
- Testing the software recurrently by each iteration and producing the right thing and working software without errors.
- Satisfied and faithful customers.
- Every software project has its own domain requirements, it looks like a bottleneck for small sized teams to get into special development domains in analyze phase. With Iterative programming models it's easy to communicate with customers and produce the right thing. [3]
- Increasing motivation of individuals working as member of SW team because of more initiative given.

4 Integrating Agility and Maturity

4.1   Agility and Discipline

Every successful venture in a changing world requires both agility and discipline. "If one has strong discipline without agility, the result is bureaucracy and stagnation. Agility without discipline is the unencumbered enthusiasm of a startup company before it has to turn a profit. " [11]

Everything is changing so fast in the world, much faster than ever. We are witnesses of huge companies' success stories stuck into 3 years, vice versa some disasters of huge companies stuck into 3 years, months. Being so rigid can take companies to the end… And also unpredictable projects, overrunning budgets, exceeded schedules… in short "Chaos" would also take you dead end.

It's two-sided sword, agility and discipline. Need to find some middle ground to adjust balance. There are software maturity models like SPICE and CMMI. While agile companies are flexible they need to success this with smart estimations, well managed resource usage, well managed risks, and well calculated budgets.

The idea of responding every request of customers sounds great, but while trying to satisfy every customer till the last request, company should take in to account that persistence of own existence. Nobody (customers) would kill "The Goose That Laid the Golden Eggs"[4] while being satisfied.

Agile companies need more to implement organizational capability maturity model while applying agile methodologies. Those companies need to take their agile capabilities to a mature level.

---

[3] (See IKIWISI Law.)

[4] Aesop's Fable - The Goose That Laid the Golden Eggs

4.2 Implementing Organizational Capability Maturity Model for Agile Approaches

Maturity Models are helping to determine processes, capability of processes and guiding to take the process capabilities to high maturity levels. People who are intending to apply agile approaches should know that those models are neither control lists, nor written laws. Those models are offering best practices, and guiding to companies. The goal is not having certificates but measuring, improving and assessing processes. If agile methods would improve and strengthen processes then it's a good option to apply agility is not a handicap for maturity.

4.3 4.3 SPICE for Turkish SW Companies

SPICE ISO/IEC 15504 is quite flexible to implement, SPICE asks "what" and doesn't ask "how". Answer of "how" depends on companies own policies. Considering flexibility of SPICE Maturity Model implementation won't force companies to do certain practices.

First step is to determine processes and practices to improve the processes, starting from the core of production& development. SPICE definitely focuses on engineering processes from user requirement elicitation to product release. Requirement analysis, design, construction, integration, testing of both system and software. Then Installation and maintenance processes. With roughly % 80-% 85 rates of SMEs in Turkish Software Industry, SPICE is an opportunity for software companies of Turkey. Because even implementing SPICE Organizational Capability Maturity Model at level-1 will bring companies a good quality in production & development. While implementing SPICE, agile practices would help to improve processes. But also its open to use some extra quality tools to improve quality.

Quality Function Deployment could be used to prioritize requirements, and to choose most important requirements while prototyping. It gives a chance to eliminate average %45 unnecessary requirements (According to Standish Group).

80/20 Law (Pareto Analysis): % 80 of defects are contributed by % 20 of resources, % 20 of defects are contributed by %80 of resources. This is another quality tool can be used in scope of SPICE to avoid average % 40-%50 rework [12]. Fishbone Diagrams can be used to localize the contributors of defects. Control charts, histograms etc. will help to see the direction of the progress. Nobody would drive a car that has no speedometer.

In Turkey, people are scared of quality systems, because there is a perception that having a quality certificate will require lots of documents and unnecessary audits. SPICE focuses on real evidences, real applications, traceability, testability, integration, communication, durability, reliability etc. Course always documents will be asked somewhere.

Technology also helps to improve and manage the quality. There are effective tools for tracking, measuring and controlling the processes. Especially Engineering processes can be managed very well; fortunately those tools are giving option to export reports.

Agility needs disciplined work, in case of losing control higher SPICE level can be picked as an objective. Agile companies of companies those are using agile practices should never avoid risk management against risks. Controlling budget limits, time limits, human resources limit, unaccounted exceeded requirements are very important for company's persistence. SPICE will set the stage for fighting against those risks.

Most of the software products are produced custom-made on purchase, and acquirers are government institutions. Thus some of the regulations, contracts and signed time schedules are inevitable as a de facto. Against this constraints SPICE will help companies to manage the achievements.

Top - Down vs. Bottom - Up Improvement Approach [13]; agile approaches satisfies and volumes up of practitioners, but top of the company needs information in the name of management. While satisfying customers with good products and services, agile satisfy developers and SPICE will provide self-confidence to managers.

5 Conclusion

It can be little confusing since there are number of Agile SW Development Methods. Companies can choose proper method according to skills, management style, size of company, development domain etc. Agile development methods will bring a new style of working; big changes are inevitable in rapidly changing conditions. While being agile, companies need to keep the balance between agility and discipline. To increase the possibility of permanent success with agile methods SPICE is a very useful model for SMEs and large organizations of Turkey.

6 References

[1] Teknoloji Geliştirme Bölgeleri Hakkında Genel Bilgiler -January 2013
http://www.sanayi.gov.tr/ServiceDetails.aspx?dataID=107&catID=305 (2013)
[2] http://ekonomi.haber7.com/sektorler/haber/965700-yazilim-sektorunde-goz-kamastiran-buyume (2013)
[3] Türkiye Bilişim Derneği 2012 Değerlendirme Raporu
http://www.tbd.org.tr/usr_img/raporlar/TBD_2012-Degerlendirme%20Raporu.pdf (2012)
[4] Kalite Yönetimi, Assoc. Prof. Dr. Mesiha Saat Ersoy, Prof. Dr. Abdullah Ersoy, İmaj Yayınevi (2011)
[5] http://ekonomi.haberturk.com/para/haber/833816-abdde-turkiye-deyince-herkes-dikkat-kesiliyor (2013)
[6] Agile manifesto is online at http://agilemanifesto.org/
[7] Integrating CMMI and Agile Development, Paul E. McMahon (2010)
[8] AgileTurkey 2013 Software Productivity Report is online at
http://www.agileturkey.org/Haberler/Detay/Yazilim-Uretkenligi-Raporu (2013)
[9] Jim Johnson, "ROI, It's Your Job!" Published Keynote Third International Conference on Extreme Programming (2002)
[10] An Introduction to Agile Software Development, Victor Szalvay (2004)
[11]Barry Boehm, Turner – Balancing Agility and Discipline
[12] Measuring the Cost of Software Quality of a Large Software Project at Bombardier Transportation: A Case Study, Software Quality Magazine, June 2012, Claude Y. Laporte, Nabil Berrhouma, Mikel Doucet and Edgardo Palza-Vargas (2012)
[13] CMMI or Agile: Why Not Embrace Both, SEI Technical Note, Hillel Glazer, Jeff Dalton, David Anderson, Mike Konrad, Sandy Shrum (2008)

7 Author CV

Kerem Kemaneci

Computer Engineer (BSc.)
Certified Automotive SPICE Assessor
TSE-Turkish Standards Institution
Information Technologies Certification Department
Graduated from Computer Engineering Program at Selçuk University in 2008. Started working at IT Department of Culture and Tourism Ministry of Turkey in Jan 2009. Then transferred to IT Department of State Supply Office. Now working for TSE since October 2010 and still studying at Production Management Master Program (MSc.) at Gazi University in Ankara.