EuroAsiaSPI² 2016 Proceedings

Proceedings

The papers in this book comprise the industrial proceedings of the EuroSPI² 2016 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.

WHITEBOX, formerly DELTA Series, about Process Improvement – ISBN 978-87-998116-6-3

EuroSPI²

EuroSPI² is a partnership of large Scandinavian research companies and experience networks (SINTEF, WHITEBOX [formerly DELTA], FiSMA), iSQI as a large German quality association, the American Society for Quality Software Division, the ECQA (European Certification and Qualification Association), 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 23rd 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.

Science, passion, technology. Graz University of Technology has built up an impressive record of achievements in teaching and research over 200 years. Five areas of strength – the Five Fields of Expertise – go to form the unmistakable academic fingerprint of the Styrian University at the heart of Europe. Currently Graz University of Technology co-edits the books, hosts the conference, and supports the conference by the political and industry network.

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	Christian Kreiner, Graz University of Technology, Austria
Industrial Programme Chair	Jørn Johansen, WHITEBOX, Denmark
Industrial Programme Chair	Risto Nevalainen, FiSMA and STTF, Finland
Industrial Programme Chair	Morten Korsaa, WHITEBOX, Denmark
Industrial Programme Chair	Nils Brede Moe, SINTEF, Norway
Industrial Programme Chair	Stefan Göricke, ISQI, Germany
Industrial Programme Chair	Michael Reiner, Gabriele Sauberer, ECQA, Austria
Organizing Chair	Adrienne Clarke, ISCN, Ireland
Co-Organizing Chair	Eva Christof, ISCN Austria

Industrial Programme Committee

Bæk Jørgensen Jens	Mjølner Informatics A/S	Denmark
Baer Cristina	Continental Engineering Services GmbH	Germany
Barafort Béatrix	Luxembourg Institute of Science and Technology (LIST)	Luxembourg
Breske Eva	Robert Bosch GmbH	Germany
Daughtrey Tazewell	James Madison University	USA
Dreves Rainer	Conti Temic microelectronic GmbH	Germany
Ekert Damjan	ISCN GmbH	Austria
Fehrer Detlef	SICK AG	Germany
Hallikas Jarmo	Falcon Leader Oy	Finland
Ito Masao	Nil Software Corp.	Japan
Johansen Jørn	WHITEBOX	Denmark
Kaynak Onur	Innova Bilisim Cozumleri	Turkey
Kemaneci Kerem	Turkish Standards Institute	Turkey
Kreiner Christian	Graz University of Technology	Austria
Larrucea Xabier	Tecnalia	Spain
Mayer Nicolas	Luxembourg Institute of Science and Technology	Luxembourg
Messnarz Richard	ISCN Ltd	Austria/Ireland
Morgenstern Jens	-	Germany
Much Alexander	Elektrobit Automotive GmbH	Germany
Nevalainen Risto	FiSMA	Finland
Poth Alexander	Volkswagen AG	Germany
Reiner Michael	FH Krems	Austria
Riel Andreas	Grenoble Institute of Technology, Laboratory G-SCOP	France
Rozman Tomi	BICERO	Slovenia
Sauberer Gabriele	TermNet	Austria
Sechser Bernhard	Method Park	Germany
Siakas Kerstin	V. Alexander Technological Educational Institute of Thessaloniki	Greece
So Norimatsu	JASPIC	Japan
Sporer Harald	pewag International GmbH	Austria
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

WHITEBOX, http://www.whitebox.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

Jørn Johansen, WHITEBOX, Denmark

Morten Korsaa, WHITEBOX, Denmark

Eva Christof, ISCN, Austria

Damjan Ekert, ISCN, Austria

Supporting Partners

Graz, University of Technology, Campus Inffeldgasse, http://www.tugraz.at

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 System, Software and Services 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 cooperation with publishers to support thematic topics of EuroSPI (SPRINGER CCIS series, Wiley annual volume about process Evolution, Volume in the SQP Journal of the ASQ, and more).

EuroSPI² is a partnership of large Scandinavian research companies and experience networks (SINTEF, WHITEBOX [formerly DELTA], FiSMA), 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 (System, Software and Services Process Improvement Network) in Europe.

EuroSPI² conferences are attended 50% by industry and 50% by research and the idea is to create an innovation space between industry and university. Also EuroSPI does not promote just one method, EuroSPI is a platform supporting a whole cluster of improvement and innovation methods in **systems, software and services process improvement (S^3PI)** because we believe that methods need to be integrated and also experiences need to be integrated to achieve success. 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".*

EuroSPI is organising the conference every year in a different country of Europe (or Asian partner associated with Europe) and the hosting country helps to organise social events demonstrating the local culture and history. This way the participants learn to bridge different cultures in Europe, and we do that since 23 years.

A cluster of European projects (supporting ECQA and EuroSPI²) contribute knowledge to the initiative, including AE (Automotive Engineer), and AQU (Automotive Quality Universities). 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 Alexander Poth – Co-editor of EuroSPI Books

Alexander Poth

Volkswagen AG,

Germany

In my opinion we can have more effective and efficient IT solutions with better methods for professional development and improvement of software systems and services.

I think a key to reaching this goal is to focus on methods that ensure the explicit and implicit demanded quality of IT systems and services leading to more customer and user satisfaction. In a user perspective added value by IT solutions is based on adequate quality. To deliver adequate quality we have to continuously realign the quality of the IT solution with the current quality demands of the users.

EuroSPI is a platform that brings together people from the industry and academic world to address this demand for more effective and efficient high quality IT solutions. This is the reason why I'm an active member in the EuroSPI community. My personal objective is to give ideas and feedbacks to the EuroSPI community to improve innovative concepts and methods for usage in the daily IT business to realize a higher added value with IT solutions.

Alexander Poth received the Dipl. Ing. (Master) degree in 2004 in computer engineering from the Technical University of Berlin. He is IT Quality Manager at Volkswagen AG.

Contact: Alexander Poth, Volkswagen AG, Germany, e-mail: alexander.poth@volkswagen.de

Welcome by WHITEBOX, Editors of the Improvement Series

Jørn Johansen

WHITEBOX, Denmark

EuroSPI is the best and most efficient European conference for news and experience in process improvement and innovation. Here you meet all the experts, the researcher and companies with the deep interest for this topic.

I have taken part in EuroSPI form the very beginning more than 20 years ago – and it has always been my best source for knowledge and inspiration.

1½ year ago, Whitebox was spun out of DELTA, mainly because the 3 partners in Whitebox wanted to continue to help companies to improve their professionalism in product development – turn the blackbox, which the development department often is, into a whitebox.

EuroSPI is the conference, where we can present our experience and get new inspiration for our work.

Jørn Johansen has been working with Software Process Improvement (SPI) for more than 25 years including maturity assessment according to BOOTSTRAP, SPICE and CMMI.

He has an M.Sc.E.E. in IT. He has worked in a Danish company with embedded and application software as a Developer and Project Manager for 15 years.

For 20 years he has worked at DELTA as a consultant and registered BOOTSTRAP, ISO 15504 Lead Assessor, CMMI Assessor and ImprovAbility™ Assessor. 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. The Talent@IT project developed the ImproveAbility™ model, which help organisations to improve more efficient. Latest Mr. Johansen was the Project Manager of SourceIT project, an 18 person-year research project focusing on outsourcing and maturity.

Mr. Johansen is also the co-ordinator of the Danish knowledge network Tecpoint (www.tecpoint.dk).

Mr. Johansen was lead editor on ISO/IEC 33014 Guide for process improvement, which was published November 2013.

Contact: Jørn Johansen, WHITEBOX, Denmark, e-mail: jj@whitebox.dk

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

Welcome to the 23rd EuroSPI^2 Conference in Graz, Austria

Christian Kreiner,

Graz University of Technology, Austria

Since 22 years, EuroSPI is a most effective exchange platform for innovation between industry and academia. Carefully addressing the needs of both worlds lead to a sustained trust relationship within the EuroSPI community. This allowed to plant, grow, and mature many ideas like the ECQA, the SPI manifesto, many project initiatives like AQUA mentioned further down, and enlarging an European initiative to a true global scale EuroSPI community that we see now.

For over 200 years, Graz University of Technology has built up an impressive record of achievements in teaching and research. Some 13,000 students and 3,300 staff continue to carry forward its power of innovation and vision into the future. Five Fields of Expertise, internationalisation, and co-operations with science and industry go to form the profile of Graz University of Technology. Sharing a common spirit with EuroSPI, Graz University of Technology is proud to be the host the EuroSPI conference in 2016.

AQUA is one of the successful initiatives in EuroSPI. It started as an alliance for integrated Automotive quality and engineering skills, covering the fusion of Automotive SPICE, Functional Safety (ISO26262), and Design for Six Sigma, just like experienced in practical engineering of nowadays complex Automotive systems. As a European Skills Alliance project, AQUA received high visibility by the European Commission, Automotive clusters, and industry, in particular suppliers and their European Association CLEPA. Almost from start, the AQUA alliance was growing - in terms of geographic coverage, by extending the initial focus on vocational training to university and technical school education programmes, and extending the scope of topics, e.g. incorporating security design in practice to harden Automotive systems against malicious attacks. Follow-on projects "Automotive Quality Universities" and "Automotive Engineer" currently roll out and extend AQUA skills in Universities across Europe and technical schools - to grow and strengthen the AQUA alliance.

Dr Christian Kreiner serves as head of the competence group “Industrial Informatics” at Graz University of Technology, Institute of Technical Informatics. Research topics are architecture and quality engineering methods for industrial networked embedded and process management systems with special focus on functional safety and security. This includes flexible platform architectures, middleware, model-based techniques in engineering and run-time, domain specific languages, and integrated development tool chains. Christian Kreiner is an intacs certified Automotive SPICE assessor, coordinator and trainer of ECQA job roles Automotive Sector Skills Alliance AQUA (ECQA Automotive Quality Skill integrated) and Functional Safety Manager. Christian Kreiner also has a long history in automated logistics systems as company co-founder, software and product line architect, and R&D head (previous Salomon Automation GmbH, now SSI Schäfer Salomon).

Contact:
Christian Kreiner (E-Mail: christian.kreiner@tugraz.at)

Welcome from the ECQA President

Michael Reiner

ECQA, Austria

The European Certification and Qualification Association (ECQA) is a not-for-profit association joining together institutions and several thousand professionals from all over Europe and the world. The association provides a world-wide unified certification schema for numerous professions. The same exam pool, exam rules and the same electronic exam system are used for certification exams in any participating country. It joins experts from the market and supports the definition and development of the knowledge required for job roles. ECQA defines and verifies quality criteria for Training organizations and trainers to ensure the same level of training all over the world.

Nowadays it is important that training courses are really recognised and attendees receive a certificate valid for all European countries. As a backbone of this initiative the EU supported the establishment of the ECQA almost 10 years ago.
The European Certification and Qualification Association (ECQA) is the result of a number of EU supported initiatives in the last ten years where in the European Union Life Long Learning Program different educational developments decided to follow a joint process for the certification of persons in the industry.

The overall objective of the project was to establish the ECQA which is supported by training organisations from European countries (currently organisations from 18 countries participate) developing and maintaining a set of quality criteria and common certification rules which are applied across the different European regions in the Life Long Learning scope in the IT and services, engineering, finance and manufacturing sectors.
This resulted in a pool of professions in which a high level of European comparability has been achieved by an Europe-wide agreed syllabus and skills set, an European test questions pool and European exam (computer automated by portals) systems, and a common set of certificate levels and a common process to issue certificates.

Through the ECQA it becomes possible to attend courses for a specific profession in one country and perform a Europe-wide agreed examination at the end of the course. The certificate will be recognized by European training organizations and institutions in 18 member countries 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.).

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 NAP (ERP), Knowledge Management, Business Intelligence, Web 2.0, social networks and VR&AR. Moreover, Mr. Reiner coordinates and participate various EU projects.
In the last nine 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. This expansion on the one hand enriches ECQA and its job roles with new views and different cultural aspects but also shows that there be the need of approaches for the solution of international certification schemas.

I wish you a good time and the EuroSPI² 2016 in Austria, a lot of interesting networking partners and exploratory meetings.

Contact: Michael Reiner, President of ECQA and Lecturer of IMC University of Applied Sciences, Austria, e-mail: ecqa_president@ecqa.org

Table of Contents

Experience Session 1: SPI and Safety and Security

*Integrating Automotive SPICE, Functional Safety and Cybersecurity Concepts – A Cybersecurity Layer Model (*SQP) *1.1*
Richard Messnarz, ISCN Austria, Christian Kreiner, Graz University of Technology, Austria, Andreas Riel, EMIRAcle, France

Functional Safety Certification from Automotive to Medical (SQP) *1.3*
Alastair Walker, Lorit Consultancy, Scotland

Integrating Assessment Models for ASPICE, Functional Safety and Cybersecurity (SQP) 1.5
Christian Santer, AVL LIST GMBH, Austria, Richard Messnarz, ISCN Austria, Alexander Much, Elektrobit AG Germany, Damjan Ekert, ISCN Austria, Andreas Riel, InnoPlusPlus, France & ISCN Group

Experience Session 2: SPI and Organisational/Process Improvement

Process Management for Electromechanical Systems Development on SPICE Level 3 and ASIL-D at VW..... *2.1*
Fabian Wolf, Volkswagen, Germany, Philipp Lackmann, Volkswagen, Germany, Christian Steinmann, Synspace, Germany

Scope and secrets of reviews within the automotive supplier industry (ABSTRACT) *2.17*
Norbert Merk, ZF, Germany, Bernhard Krammer, ZF, Germany

Terminology, Technical Documentation and Standards: Safety and Security for Industry and Engineering Environments *2.19*
Frieda Steurs, KU Leuven, Belgium & TermNet, Hendrik J. Kockaert KU Leuven, Belgium, Gabriele Sauberer, TermNet, Austria, Blanca Nájera Villar, TermNet, Austria

Experience Session 3: SPI and Automotive Engineering

A Compact Introduction to Automotive Engineering Knowledge (Springer) *3.1*
Andreas Riel, InnoPlusPlus & Grenoble Institute of Technology, France, Monique Kollenhof, Symbol BV, Netherlands, Sebastiaan Boermsa, Summa, Netherlands, Ron Gommans, Roc Ter AA, Netherlands, Damjan Ekert, ISCN, Austria, Richard Messnarz, ISCN, Austria

Functional Safety Considerations for an In-wheel Electric Motor for Education (Springer)..... *3.3*
Miran Rodic, University of Maribor, Slovenia, Andreas Riel, EMIRAcle France & ISCN Group, Richard Messnarz, ISCN, Austria, Jakub Stolfa, Technical University of Ostrava, Czech Republic, Svatopluk Stolfa, Technical University of Ostrava, Czech Republic

The Need for Policy Rationale *3.5*
Joanne Schell & Paul Schwann, NXP Semiconductors, Austria

Experience Session 4: SPI and HW Safety and Testing

A GSN Approach to SEooC for an Automotive Hall Sensor A Compact Introduction to Automotive Engineering Knowledge (Springer) *4.1*
Xabier Larrucea, Tecnalia, Spain, Silvana Mergen, TDK-EPC AG & Co. KG, Germany, Alastair Walker, Lorit Consultancy, Scotland

An advanced testing approach to validate software changes in complex hardware environments..... *4.3*
Domenik Melcher, Graz University of Technology, Austria, Thomas Puchleitner, NXP Semiconductors, Austria

Experience Session 5: SPI and Organisational and Human Factor

Method to establish strategies for implementing process improvement according to the organization's context (Springer) *5.1*
Mirna Muñoz, University of Zacatecas, México, Jezreel Mejia, University of Zacatecas, México, Gloria P. Gasca Hurtado, University of Medellin, Colombia, Maria C. Gómez-Álvarez, University of Medellin, Colombia, Brenda Durón, University of Zacatecas, México

Self-What?—the single most important success factor..... *5.3*
Danilo Assmann, Vector Informatik, Germany, Melanie Klemenz, DOGA, Spain

Experience Session 6: SPI and SW Measurement

More Effective Sprint Retrospective with Statistical Analysis..... *6.1*
Muhammed Emre PEKKAYA, Onur ERDOGAN, Halime GÖK, TUBITAK-BILGEM-YTE, Turkey

Software quality measurement and evaluation framework for innovation projects..... *6.13*
Marcin Wolski, Poznan Supercomputing and Networking Center, Poland, Bartosz Walter Poznan Supercomputing and Networking Center, Poland & Poznan University of Technology, Poland, Patryk Prominski, Poznan Supercomputing and Networking Center, Poland, and Szymon Kupinski, Poznan Supercomputing and Networking Center, Poland

Experience Session 7: SPI and Innovation Strategies

Forming a European Innovation Cluster as a Think Tank and Knowledge Pool (Spinger) *7.*1
Richard Messnarz, ISCN Austria, Andreas Riel, EMIRAcle France & ISCN Group, Gabriele Sauberer, TermNet, Austria, Michael Reiner, University of Applied Sciences Krems, Austria

Innovative Marketing in low-tech micro companies - Lessons learned from study projects (Springer) *7.3*
Michael Reiner, University of Applied Sciences Krems, Austria, Christian Reimann, FH Dortmund, Elena Vitkauskaite, Kaunas, University of Technology, Lithuania

User Orientation through Open Innovation and Customer Integration (Springer) *7.5*
Dimitrios Siakas, Citec Finland, Kerstin Siakas, Alexander Technological Educational Institute of Thessaloniki, Greece

Experience Session 8: Process Improvement

Proof: Maturity matters: Higher maturity gives higher productivity 8.1
Jørn Johansen, Whitebox, Denmark, Morten Korsaa, Whitebox, Denmark

Process Improving by Playing: Implementing Best Practices through Business Games (Springer) 8.11
Antoni-Lluís Mesquida, University of the Balearic Islands, Spain, Milos Jovanovic, University of Novi Sad, Serbia & University of the Balearic Islands, Spain, Antònia Mas, University of the Balearic Islands, Spain

Experience Session 9: SPI and Medical Safety

Infinite Demands and Constrained Methods - A Unified approach towards delivering Large Volume 'Quality' Automotive Software (ABSTRACT) *9.1*
Aradhana Sivan & Leena Safeer, TataElxsi Limited, Bangalore, India

Safety Analysis of a Hemodialysis Machine with S# *9.3*
Johannes Leupolz, Axel Habermaier, and Wolfgang Reif, University of Augsburg, Germany

A Preliminary Systematic Literature Review of the use of Formal Methods in Medical Software Systems *9.15*
Silvia Bonfanti & Angelo Gargantini, Università degli Studi di Bergamo, Italy, Atif Mashkoor, Software Competence Center Hagenberg GmbH, Austria

Experience Session 10: SPI and Automotive Safety and Security

Integrating HARA and TARA – How does this fit with Assumptions of the SAE J3061 (SQP) *10.1*
Georg Macher, AVL List GmbH, Austria, Andreas Riel, EMIRAcle, France, Christian Kreiner, Graz University of Technology, Austria

Automotive Security: Challenges, Standards and Solutions (SQP) *10.3*
Alexander Much, Elektrobit Automotive GmbH, Germany

Merging FMEA and FTA for safety analysis of sensors for automotive applications *10.5*
Silvana Mergen, W.J. Schreiber-Prillwitz, Philipp Schmidt-Weber, TDK-EPC, Germany

Formal Methods & Functional Safety (Springer) *10.17*
Micheal Mac An Airchinnigh, ISCN Ireland

Appendix: Selected Workshop Papers

A Virtual Glucose Homeostasis Model for Verification, Simulation and Clinical Trials *11.1*
Neeraj Kumar Singh, University of Toulouse, France

Model-based offline and online testing for medical software. *11.11*
Paolo Arcaini, Prague, Charles University, Czech Republic, Elvinia Riccobene, Università degli Studi di Milano, Italy, Angelo Gargantini, Università degli Studi di Bergamo, Italy

Modelling bio-compatible and bio-integrative medical devices *11.21*
Didier Fass & Dominique Méry, LORIA, France

Integrating Automotive SPICE, Functional Safety and Cybersecurity Concepts - A Cybersecurity Layer Model

Richard Messnarz[1], Christian Kreiner[2], Andreas Riel[3]

[1] ISCN GesmbH, Liebenauer Hauptstrasse 2-6, Graz, Austria
rmess@iscn.com
[2] Graz University of Technology, Inffeldgasse 16, 8010 Graz, Austria
Christian.kreiner@tugraz.at
[3] EMIRAcle c/o Grenoble Alpes University. 46 av. Félix Viallet, 38031 Grenoble, France
andreas.riel@emiracle.eu

Abstract

Automotive companies need to develop more and more functionalities to stay competitive and already more than 80% of functions in a car are controlled by electronics and software. Automotive projects need to implement standards which help to cope with this new complexity where more than 100 ECUs (Electronic Control Units) are networked by a bus system, and vehicle functions are implemented by a real time sequence of commands to these ECUs actuating several subsystems. In volume 17, Issue 3, June 15 of the Software Quality Professional magazine we discussed the implementation of Automotive SPICE and Functional Safety in an integrated approach. In this paper we extend this approach by integrating concepts of considering Cybersecurity threats and requirements as well.

Keywords

Automotive SPICE, functional safety, cybersecurity

Published in: ASQ Software Quality Professional

Functional Safety Certification from Automotive to Medical

Alastair Walker, Functional Safety Consultant. LORIT CONSULTANCY, Edinburgh, Scotland
alastair.walker@lorit-consultancy.com

Abstract

The medical device sector has many international standards and guidance documents; it is also a very wide ranging product sector. This paper aims to suggest a strategy for assessing systems including either or both electronic hardware and software, that utilises some of the techniques introduced in the ISO 26262: 2011[1] automotive functional safety standard. The reason for suggesting this approach is to recommend processes that will help improve and simplify the risk assessment and development activities of safety relevant medical devices.

The approach here is very much systems focussed on and relates to medical devices that are within the remit of IEC 60601-1[2] and hence are defined as ME EQUIPMENT or ME SYSTEMS (devices transferring energy to or measuring energy from the patient). Here there are strong parallels with the functional safety strategy used in the automotive sector.

Not all products are deemed to be ME EQUIPMENT or ME SYSTEMS nor is IEC 60601-1 relevant for all medical devices, others may be e.g. in-vitro or implantable devices. Not all software that falls within the remit of the software life-cycle standard IEC 62304[3 is relevant to IEC 60601-1 e.g. standalone software can be a medical device. [4], [5]

Keywords

Functional safety, ME SYSTEM, ME EQUIPMENT, risk analysis, HARM, HAZARDOUS SITUATION.

Published in: ASQ Software Quality Professional

Integrating Assessment Models for ASPICE, Functional Safety and Cybersecurity

Christian Santer[1], Richard Messnarz[2], Alexander Much[3], Damjan Ekert[2], Andreas Riel[4]

[1]AVL LIST GMBH, A-8020 Graz, Hans-List-Platz 1, Austria
christian.santer@avl.com
[2]ISCN LTD/GmbH, A-8010 Graz, Schiessstattgasse 4, Austria
rmess@iscn.com
[3]Elektrobit AG, Tennenlohe, Erlangen, Germany
alexander.much@elektrobit.com
[4]InnoPlusPlus, ISCN Group, 2 av. des Jeux Olympiques, 38100 Grenoble, France
ariel@iscn.com

Abstract

The ISO 26262-2 Chapter 6.4.8 demands one or more functional safety audits during the safety lifecycle, also a suitable ASPICE (Automotive SPICE ®) level is claimed in most of automotive projects from suppliers by all major OEMs. In order to reduce the effort of audit and assessment activities a combination of an ASPICE ISO/IEC 15504 Assessment and a Functional Safety ISO Audit is strongly recommended. This paper will introduce a framework to extend the Process Assessment Model (PAM) based on ISO 15504 (ASPICE) to meet the requirements of ISO 26262 for a functional safety audit. Also the paper will discuss current work in SOQRATES (www.soqrates.de, a working party with participation of leading Automotive suppliers) integrating the new requirements from the Cybersecurity standard SAE J3061.

Keywords

Automotive SPICE, functional safety, cybersecurity, assessments and audits

Published in: ASQ Software Quality Professional

Process Management for Electro-mechanical Systems Development on SPiCE Level 3 and ASIL-D at VW

Dr. Fabian Wolf (fabian.wolf@volkswagen.de)
Philipp Lackmann (philipp.lackmann@volkswagen.de)
Christian Steinmann (christian.steinmann@synspace.com)

Abstract

In the context of safety critical software development for automotive steering systems at Volkswagen, the establishment and maintenance of a web-based process management system has lead to a set of experiences, best-practice approaches, new insights, and major certificates for process maturity which shall be addressed in this paper.

The necessary prerequisites for the modelling system like the process meta-model to define the relation amongst the different process elements and the different views for organising and visualising processes are presented. To create the project specific instances of the processes, there are smart solutions for tailoring elaborated.

A selection of aspects of process modelling – especially those which may lead to challenging situations – will be addressed and suggestions for (re-)solutions presented.

Another topic is to guarantee completeness and consistency of the model with respect to requirements coming from Automotive SPiCE and topics like functional safety for road vehicles. The mapping approach - which peaked in a certificate for ISO 26262 compliance up to ASIL D by SGS TÜV – will be explained.

Finally, a structured change management of the process modelling system and the methods to involve user feedback to ensure continuous improvement are outlined.

Keywords

Embedded Systems Development; Process Management; Engineering Process Modelling; Continuous Improvement; Automotive SPiCE; Functional Safety; ISO 26262; Automotive;

1 Introduction

The development of electromechanical systems including embedded hardware-software systems is one of the most difficult engineering challenges. Especially handling software complexity and safety aspects are main factors for project success. Software process quality is an important factor and should be a main focus. In the automotive domain, software features are a major part of the cars functionality while software size and complexity rise continuously. As a solution for rating process quality Automotive SPiCE was established in the automotive industry for more than a decade. Additionally, for safety relevant development like steering or braking systems the ISO 26262 is of high importance and a 'must have' for the product development.

Based on this context, selected relevant experiences in developing electromechanical steering systems are presented. The engineering context is safety critical systems containing Automotive Safety Integrity Level D (ASIL D).

The development experiences are documented in a web-based process management system called KEEP ('Komponenten Elektromechanik Entwicklungs-Prozess') including about 60 processes, 500 activities, 50 roles, 120 method descriptions and 250 document templates with a total of more than 3000 chapters. As a novelty, all three disciplines (hardware, software, *and* mechanical engineering) are fully covered and deployed at development teams working in different locations and domains.

Such a comprehensive process model handling that complexity and simultaneously giving the opportunity of an individual process tailoring for every project and focussing on user experience is – to our knowledge - unique in the automotive domain.

KEEP builds upon Microsoft SharePoint combined with Microsoft Visio for graphical modelling. The basic structures and functions for this system are enabled by an additional software suite (see [QUAM]). This combination enables active directory integration, bidirectional connections between different elements, different context types and workflows.

In the first part of this paper the theoretical backgrounds like modelling issues are discussed, i.e. meta-model, process hierarchy and in- and output concept. The chapter 'modelling dilemmas' discusses design decisions to be made answering issues in the context of process complexity and modelling concepts. Afterwards a tailoring mechanism resulting in project specific process instances is introduced. The next chapters focus on covering Automotive SPiCE, ISO 26262, and how do maximize the resulting benefit. The last two chapters discuss organisational structures for establishing the processes and their continuous improvement.

2 Process Meta-Model

One of the starting points for setting up the process modelling system is the development of the process meta-model which defines the general relations amongst different types of process elements like activities, processes, roles, documents, templates, methods, milestones, trainings and guidelines.

Using a process meta-model has – based on our own experiences – several advantages:

- **Focus:** The meta-model helps to focus to put the information to the right elements e.g. explanations 'what to do' should be put into activities, explanations for 'how to do' should be put into method descriptions
- **Life-cycle:** By defining the borders of each element, it is easier to define the life-cycle (concerning versioning, baseline, release) of the different process elements
- **Tailoring:** The clear description of the relations amongst the process elements implicate several constraints concerning the tailoring of the process for the use in projects. Knowing these constraints (e.g. you can't drop role X because it is responsible for activity A) is elementary for doing correct and consistent tailoring.

Together with the advantages of the meta-model, there comes a set of disadvantages: Whenever there is a need to document some integral process steps which incorporate activities, methods, guidelines and tool descriptions, you have to split that information to the correct process elements.

Amongst the various available and established models (see [Meta-Models]), the aspects which promised the best balance between low complexities and fulfilling the requirements were picked and put together to develop the KEEP meta-model (Figure 1).

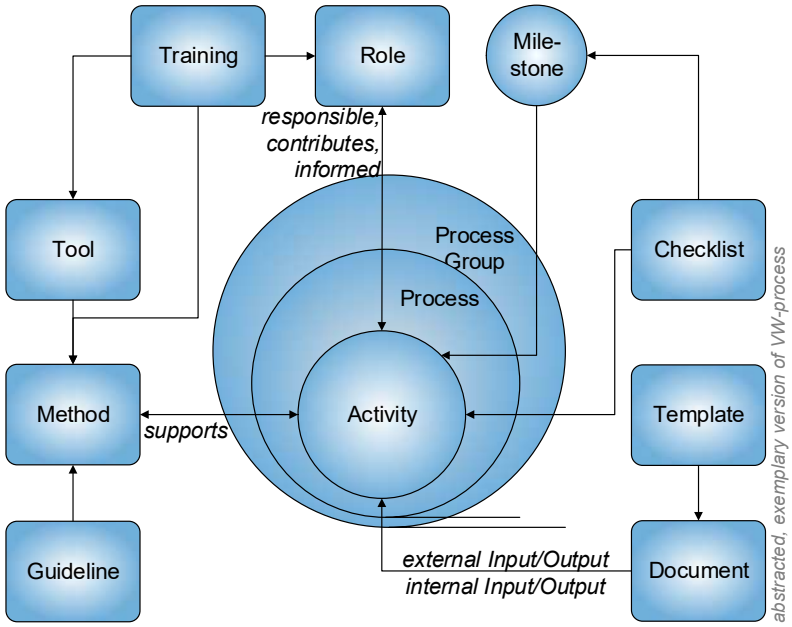


Figure 1: KEEP meta-model

The model is straight forward and can be read as follows:

- Role X is responsible for performing activity A
- An activity A is supported by method M
- Method M can be performed using Tool T
- For Method M there is a guideline G available

Two aspects need additional explanations: process hierarchy and input/outputs.

2.1 Process Hierarchy

The process hierarchy is defined by ‘process group’, ‘process’ and ‘activity’. Activities are the lowest/smallest elements of the hierarchy. The activities belonging together are grouped within a ‘process’ and describe the steps that shall be performed.

At the beginning of setting up the whole system, processes belonging together were grouped within a ‘phase’. This was convenient for some engineering processes – but soon turned out to be too clumsy for managing such big blocks in projects. So the projects started their planning based on processes - and not on phases. The process modellers renamed the term ‘phase’ to ‘process group’ and now use it as the top level hierarchy for the process modelling.

The meta-model implies a strict design which is bound to have exactly three layers in the hierarchy, but the solution which was implemented proved to be much more flexible: the two lower layers ‘activities’ and ‘processes’ are the fixed layers, where all the attributes and rules according for the element types apply. But the layer on the top, the ‘process group’ can be used in a flexible way: a process group can be part of another process group; a process group can refer to processes belonging to other process groups. So this layer can be used, to create multiple, different views of the processes. A simple view for a process group showing the dedicated processes can be seen in Figure 2.

Figure 2: Simple view of a process group

Usually the amount of information displayed on one page or in one chart should be limited for easier understanding. But we should not underestimate the users: engineers who are used to design and implement complex systems. So based on feedback it makes sense to provide a summary view showing all process groups with all their processes in one picture. Even if this view is almost unreadable for the beginner, the engineers welcome it, because it is shaped like a 'V' – representing the development approach – and contains *all* twelve process groups and *all* of the 60 processes (blue boxes) on *one* page. Such a view can be used as the starting point for the user to quickly navigate into the model (see Figure 3).

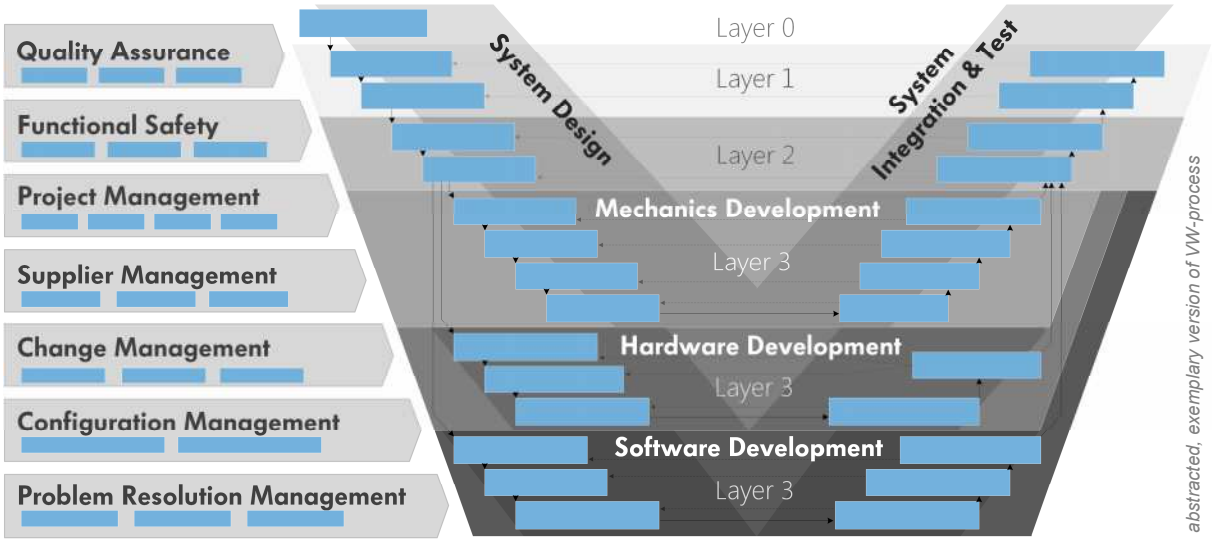


Figure 3: Summary view of all process groups and processes

With this approach, it is possible to build several views e.g. showing only the supporting processes or showing only the engineering processes without compromising the underlying process groups, processes or activities.

2.2 Internal and External Inputs/Outputs

Concerning the input and output of processes, the KEEP meta-model defines: inputs and outputs are assigned at the level of activities. Nevertheless, all inputs and outputs of a process can be shown by simply putting all inputs/outputs of the activities – belonging to the process – together.

There is an additional distinction: *internal* versus *external* input/output:

- An *internal* input is an input that is produced by a preceding activity within the same process.
- An *internal* output is an output that is used as an input by a subsequent activity within the same process.
- An *external* input is an input, which is produced by an activity from another process or an external source (e.g. customer or supplier).
- An *external* output is an output of an activity, which is used as an input in an activity belonging to another process or an external stakeholder (e.g. customer or supplier).

Consequently, the external outputs are those, which matter for achieving a milestone. Looking into the processes there is usually an activity producing a relevant artefact (output) – which is still modelled as an *internal* output, because subsequently it is used as an *internal* input for a review activity including the release of the artefact. So this review activity now has an *external* output – which is still the same artefact in our model, but labelled as 'external'.

An example for the activity ‘develop systems architecture’ – which is related to the SPiCE ENG.2 process - is depicted in Figure 4.

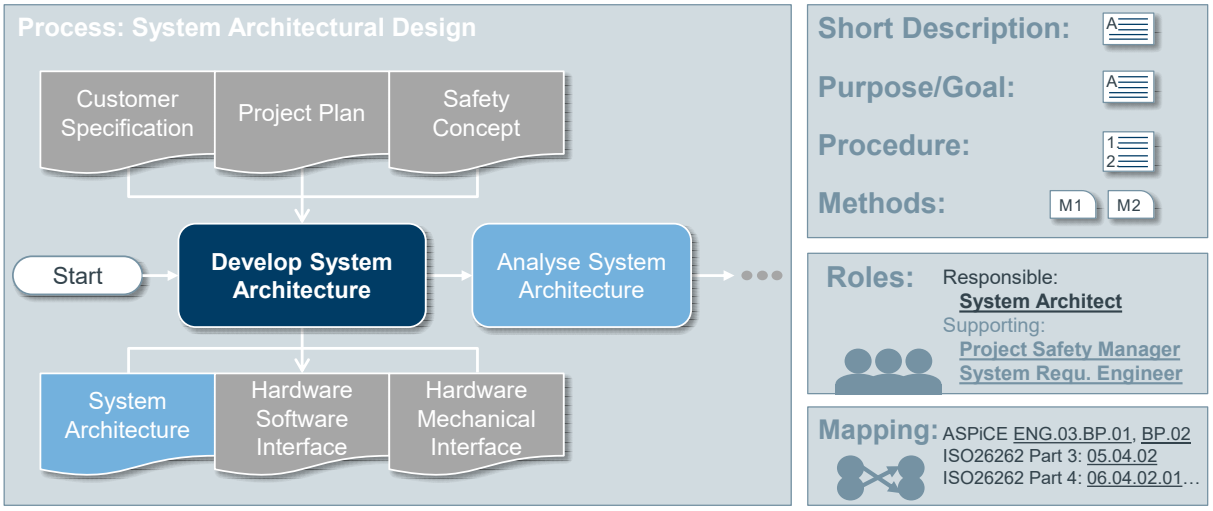


Figure 4: Exemplary view of activity with inputs/outputs and additional information

3 *Modelling Dilemmas*

Although the basic concept for modelling is based on BPMN and supported by the tool [BPMN], [QUAM], the approach for process modelling is determined by several goals for optimisation. Those goals are different – and sometimes counteracting - depending on the involved stakeholders like engineers, project managers, quality staff, and persons doing process modelling [BPMN1].

Such goals would be e.g.:

- To maximize readability of process descriptions,
- To minimize the time needed to understand the process description,
- To minimize ambiguity and misinterpretation,
- To minimize the length of the descriptions,
- To minimize the effort for maintaining the process descriptions.

For some contradicting goals, an acceptable balance can be achieved in the sense of optimisation – and not as a bad compromise. For example, if you want to minimize misinterpretation of process descriptions, you usually add additional explanations to guide the reader into the right direction of interpreting the process correctly and in the intended way. This directly constricts with other goals like to minimize the length of descriptions. But between the extremes, there *should* be a solution where the text is long enough to prevent *most* of the possible misinterpretations but at the same time is short enough to be accepted by *most* of the users.

There remain serious design decisions, where the available alternatives don’t unfold to a win-win solution or an acceptable compromise. This is a typical dilemma situation and a few examples are given in the next chapters.

3.1 Universalism versus Particularism:

If the processes are modelled in such a (universal) way, that only a minimum set of processes will be needed, then there is the problem, that these processes are not able to serve all different variants of

(particular) projects.

On the other hand, if there were individual (particular) solutions for each variant of the process (e.g. hand code vs. auto code) to maximise applicability in each special case, this would increase the number of processes and also increase redundancy, because several processes would be very similar.

It turned out, that there is no absolute rule, which path to take, but based on the meta-model (see Figure 5), it is possible to solve the dilemma up to a certain degree. If there are several processes/activities, where the pattern of the process itself is in principle the same, then one should not duplicate the processes and activities, instead several different methods should be created which support the needed variants of the process. Additionally each method description should contain guiding information when to apply the method.

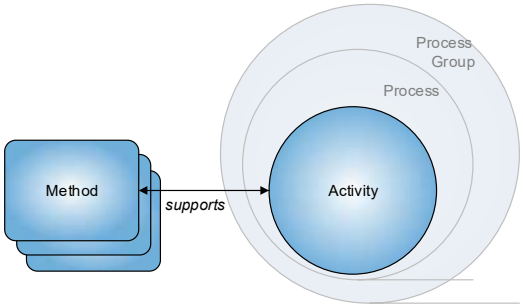


Figure 5: Relation between Activities and Methods

This solution has some constraints, e.g. that the assigned roles and responsibilities for the activity and all the input/output artefacts need to be the same.

3.2 Maintaining Consistent Role Descriptions

The standard organigram for a project may contain *fifty* roles – which seems to be a lot, but it includes all the necessary roles for all three disciplines: hardware, software, and mechanics development. From the modelling system's point of view, the only way to go is to provide the finest acceptable granularity for role descriptions. It does not make sense to combine several roles into one role – just to lower the total number of roles – because then there will be big problems in assigning clear responsibilities when modelling the processes and activities. So it is necessary to have that huge number of roles described in the modelling system and a combination of roles is only done at the project level, when a person plays two or more roles defined by the system. This also allows for different patterns of assigning the roles depending on the type of project.

Still, one problem remains: There will be lots of similar roles, e.g.: SW-Unit-Tester, SW-Tester, HW-Tester, Mechanics-Tester, Component-Tester, and System-Tester. So this example covers six roles for 'Tester' – but on different levels or in different domains – and there would be another six similar roles for doing the integration-testing. Here the challenge is to find a method to maintain consistency.

The proposed solution is to use *generic roles* (or abstract roles) as a pattern for creating the specific roles like those above. In the given example, this means there would be a generic role description for 'Tester' which is formulated in such a way, that there is no restriction concerning the level or domain the tester is working on. This generic role should not be used in any processes or activities and should not be part of any organigram. The specific roles for testers in the different domains or levels now can *inherit* the properties (like description of work, necessary qualifications, escalation) from the generic role. Additionally those descriptions can be enhanced, to add the specific parts of the role – if necessary. Of course, the name of the role, its abbreviation and the short description are not inherited from the generic role.

Whenever a generic role is altered, all specific roles which were inherited from that generic role will be changed automatically. So there is only a single source of information and thus, consistency is always guaranteed.

Generic roles make sense not only for testing roles, also for architects or requirement engineers on different levels or domains, for (sub-) project managers,and for test managers. However, the concept of generic roles should not be used as often as possible; it should be used only if it saves time for maintaining consistency, e.g. as a rule of thumb if there are at least three roles which have a common core.

3.3 Finding the Right Number of Layers

Between the top level of the engineering processes (called the system level) and the domain specific levels on the bottom (HW/SW/Mechanics), there is usually the need for additional layers, which reflect the principle of decomposition of the system to be able to manage the development.

There are two concepts for representing these layers between top-system and domain engineering:

Concept A: Simply define that the system layer (layer 1) is self-containing in regard to sub-systems. So the parts of the system (as known as the sub-system or components, speaking of layer 2) can be seen and handled like a system (layer 1) again.

Concept B: Explicitly define an additional layer 2 below the system (layer 1) with processes for analysis, architecture, design, integration and test which is called 'component development'. Note that this layer 2 needs to be self-containing to be able to have additional decomposition layers; but layer 1, the system level is unique.

From a process modelling point of view the advantages of concept A are: no duplication of similar processes or activities; the model is as small as possible.However, there are disadvantages: All the descriptions of the processes and activities on the top layer shall be formulated in such a way, that they can be applied to the system level as well as to (any) sub-system level. This decreases readability and understandability of the model.

Concept A should not be used at all, when the processes on system level have *relevant* differences to the processes on component level: e.g. on system level, the whole process is interwoven with the processes concerning management of functional safety and concerning qualification for different prototype and release levels.

So in that case, it would be wise to use concept B with a separate layer for component development between the top layer system development and the three domain specific layers for HW/SW/Mechanical development.

The component layer should be iterated as necessary, to create the number of layers appropriate for the project management of the product development.

There would be some redundancy in the description of the processes and activities amongst system and component level, but these redundancies are not that hard to manage concerning consistency, because there are not plenty of them, just the system and the component layer. So in this case it is not worth implementing a mechanism similar to the generic roles (see 3.2) for processes and activities. The only shortcut – compared to fully duplicating the layers – is to re-use all the templates of the system level at the component level in case they fit.

3.4 Splitting SW-Development into Two Paths

For the implementation of safety critical software, there are additional requirements concerning design methods, implementation methods and testing methods coming from ISO 26262. To reflect these requirements, there can be two alternatives for designing the SW process:

Alternative A: Single path. Design the whole SW development process from SW requirements engineering, SW architecture, SW design, SW implementation up to SW integration and SW testing modelled in one single path – but whenever there are requirements concerning safety software, design the process that way, that these requirements are addressed - but can be left out, if software is developed, which is not safety related.

Alternative B: Two paths. Duplicate all processes and activities, which have additional requirements coming from ISO 26262. So there will be one path leading through the SW development process for SW which is not safety related (usually called 'functional software') and there is a second path for SW development in respect to functional safety called 'safety software'.

From the modelling point of view, alternative A would be the one to choose, because this would be the solution which minimises redundancy – which is always good if you strive for consistency.

The disadvantages of alternative A are the readability and understandability of the described process: there are a lot of engineers involved in creating functional software (not the safety related part) – but the process always talks about what has to be done for safety related software. And because the process modelling is not done for the sake of the modellers – it should fit the users (in this case the SW engineers). Considering that, alternative B - having two paths for SW development – would be the better choice.

The disadvantage is obvious: many activities (the exact number is 19) are almost identical – but now there are two instances of them: one instance belonging to functional software, the other instance belonging to safety software. The second instance is the one which is enhanced to contain the additional methods to support and comply with ISO 26262.

Having this as starting point, the descriptions of the activities for functional software development can be fine-tuned for easy reading and understanding – and the user's feedback will witnesses it: this would be the right decision.

During the maintenance of the processes, additional requirements can be brought up:

- To be able to get an ISO 26262 certification (see 6. ISO 26262 Certification of KEEP) the process for SW development of safety related functions shall follow a strict change management *always* involving functional safety personnel.
- The functional SW development process needs to deal with alternative methods (e.g. auto code, hand code, assembler, configurable SW-modules) and has a higher change rate, because adapting to projects' needs more frequently.

Both requirements can be satisfied with the solution B. Of course there will still be the duplication and redundancy, but it is better to have some known redundancy which can be managed than to hinder any process improvements, because the concept of processes is too rigid.

4 Tailoring Concept and Project Instances

Tailoring an engineering process is an opportunity for reducing the engineering and development effort, because tailoring of processes means to fit the process based on the project's needs. Processes describe which outputs must be provided, depending on multiple project conditions like the safety relevance and the type of development product. In the following a tailoring concept is presented:

The first step is to define the possible sets for tailoring – i.e. to define views of the process elements which include only a subset of the whole system. The definition of views is straight forward: for each type of process element (see 2. Process Meta-Model) there are two properties called: 'project type' and 'aspect' which refer to two lookup lists containing the available project types and the aspects for tailoring. The property 'project type' is a rough filter used on the higher levels of the process hierarchy; the property 'aspects' does the fine-filtering which is used on activity and document template levels.

In a second step, during the setting up of the project specific instance of the process model, the designated project manager selects the appropriate 'project type' (only one type of project can be selected) and then the applicable 'aspects' which best fit the intended type of development. Because of the huge amount of contents of the whole system, there is a preview mode, where the effect of the selection is shown on each web-page of the system – but only for the current user. After the visual inspection of the tailoring, the project's instance of the processes is generated as a sub-web based on the latest baseline.

One of the special features is the system's capability to provide tailoring of document templates down to chapter-level. In KEEP, each document template consists of one or more chapters arranged in a hierarchy below the template. Each chapter is a separate data item with individual properties for project types and aspects. Thus it is possible, to design e.g. one template for e.g. for documenting a release – and depending on the type of project (QM/ASIL A-D), there are different blocks (chapters) for

e.g. the signature, defined by the selected tailoring.

The only challenge which needs clarification is the distinction between 'not doing' something (e.g. tailoring or crossing out HW-Development, because the project is only SW) versus 'have it done by suppliers'. Concerning the first case, one can simply drop all activities, roles, templates etc. belonging to HW-development processes. In the second case a special tailoring should be provided, where all of the preparing activities (e.g. writing specifications), the accompanying activities (progress reviews, controlling) and the approval activities (review, test) are still part of the process instance and only those activities are dropped, which are fully in the responsibility of the supplier. In that case, the process itself will still be part of the project instance, but on the lowest level activities will be greyed.

5 Covering Automotive SPiCE – HIS Scope – on Capability Level 3 and ISO 26262 up to ASIL D

When moving from a proprietary, file-based process modelling system to a database driven system like KEEP, the process modellers could already build upon a sound set of processes which were assessed on SPICE Capability Level 2 for HIS-Scope. So there was a sound basis, but additional advantages can be gained from switching from files to database: A mapping between KEEP process elements and Automotive SPiCE process elements can be established.

On the SPiCE side, the mapping refers to processes, outcomes, base practices and management practices.

On the KEEP side, the mapping refers to different process elements like activities, processes, process groups, roles, document templates, guidelines, and methods.

As can be seen in Figure 6, multiple references between any elements of the SPiCE structure (process groups, processes, base practices BP, process attributes PA, and generic practices GP) to any elements of the KEEP structure are allowed – and there are no restrictions concerning the level of hierarchy.

The mapping is used in both directions:

- During an assessment, the assessor can navigate into the SPiCE structure and receives links to activities, roles, templates and methods associated with the currently selected process or practice.
- During regular quality checks of the process modelling system - inspired by [BPMN2] - the coverage of all relevant practices from SPiCE is calculated and thus can be analysed to fill gaps if they appear.

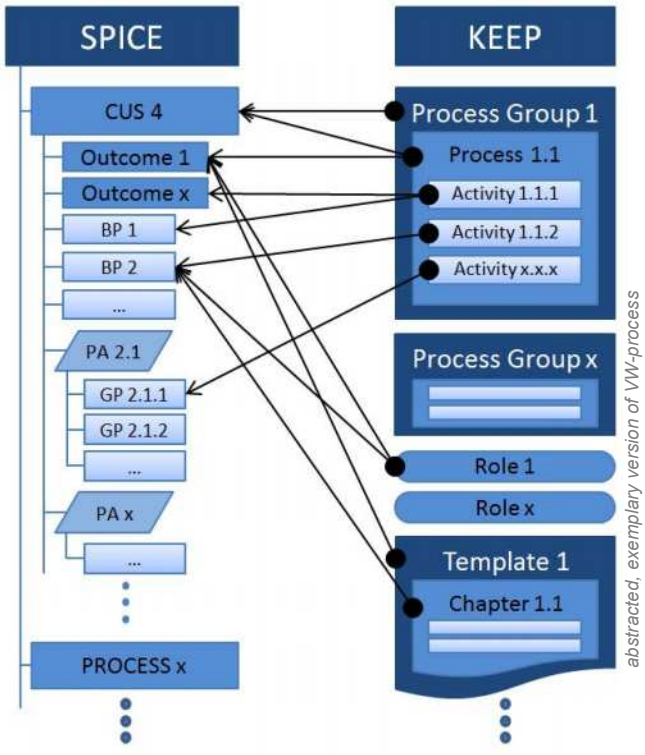


Figure 6: Granularity of SPiCE mapping

Additionally, it is also highly recommended to perform an audit by external experts, reporting the ability to fully support capability level three for projects based on KEEP processes.

Complementing Automotive SPiCE, the most important normative reference for safety relevant electrical/electronical car systems is ISO 26262. As mentioned before, referencing and linking to the requirements from an international standard is a good way to proof consistency. This also applies to ISO 26262. Maintaining a full mapping of activities, processes, methods and document templates on a detailed level for each relevant requirement and work product of ISO 26262 is not only useful in as-

sessments: Focussing on completeness a mapping of process elements supports project specific tailoring while simultaneously being sure that all relevant normative requirements are fulfilled. In this case, a full mapping means about 1200 elements from KEEP are mapped via about 1900 relations to 1100 chapters from ISO 26262 covering part 2 to 9.

As a conclusion, mapping norms and standards towards process elements and documenting this in a coverage matrix is a good way to check consistency and completeness of normative requirements.

6 ISO 26262 Certification of KEEP

A highlight in contributing to process quality comes from certifying the KEEP process for compliance with ISO 26262 up to ASIL D.

Reviewing and discussing the process model with external experts gives the opportunity of examining the workflow and artefacts from another point of view. This can help becoming more efficient, closing gaps and starting also discussions about alternative methods.

To maximize the benefit of certifying, it makes sense to choose a global, overall approach focusing on the KEEP process and thus covering all projects and not only a single project. Consequently all projects using the KEEP process now can refer to this certification if needed.

The challenge is that ISO 26262 usually requires a process audit to be project specific. This can be solved by focusing the certification procedure on the process model and providing representative samples from projects, by giving context information and documents, and by explaining, how the process is executed and how to work with the templates. This demonstrates the usability of the process model and demonstrates a sound process context.

Having the processes certified saves money in every safety assessment, because the processes themselves don't need to be assessed anymore. Only the correct appliance of the process needs to be proven. This can be done with a reasonable amount of quality records.

The following shows a possible approach for certification:

- In a first step an unattended review of the process model would be a good start. This makes sure the process and its documentation is understandable without further context. This can give important hints for the clearness of processes' descriptions and the way the process is documented.
- As a second step there should be a feedback to talk about findings and impressions. This step can be done iteratively throughout the certification processes and helps to avoid misunderstandings during analysation of the process model.
- As a third step a document template review is recommendable because the projects are directly using these documents. The templates are the most important artefacts, which should be consistent to ISO 26262 requirements. To speed up the process of certification there should be a continuous feedback loop between the auditors and der process team to fine-tune the process while the review process is still going on.
- Finally an on-site assessment is a perfect way to review changes made through certification process and to build a basis for future process improvements.

7 Managing the Process Model

The ongoing improvement of the processes and their associated descriptions, roles, templates, and methods needs to respect the principles of sound configuration management: Every change is based on a change request (CR). The CRs are processed in a Process Change Control Board (PCCB) with defined roles (PCCB-Manager, responsible persons for certain process groups) and follow a stringent CR workflow containing analysis, implementation, review and release activities. The workflow is an integral part of the process management tool and is constructed in a way that process elements can't

be modified without a CR explicitly assigned to the element and having the status ‘implementation’. This mechanism effectively prohibits any uncontrolled changes to the process elements.

Additionally each process element has to have a built-in versioning mechanism, which allows roll backs of one or more changes – if necessary. The versioning mechanism is connected to the roles and rights assigned to each user so modellers and reviewers can see all of the latest modifications – which is necessary during editing the processes – but normal users still see the last valid version, which is part of the last baseline.

During times of intensive modifications of the process descriptions (e.g. when a process is split into two processes) the whole system would not be useable as a blue-print for creating process instances. Therefore you need to have a consistent state, where the relations amongst the process elements and their descriptions ‘fit together’. This consistency is achieved by defining and releasing baselines on a regular basis (e.g. two months). From the technical point of view, a baseline is a copy of the whole contents of the process modelling system at a certain point of time, stored as a read-only sub-web. Of course, the intended level of consistency is not only reached by simply creating the baseline – it is achieved by implementing and reviewing those CRs, which shall be finished up to the baseline date. Automated consistency checks are helpful for judging upon the degree of consistency: e.g. each activity needs to have at least one output; or to identify activities with no input.

8 Keeping Processes Alive - Continuous Improvement, User Feedback and User Acceptance

For continuous improvement of the processes, it is necessary to implement a mechanism which allows user feedback in an easy way.

Four mechanisms for generating user feedback in the process management context are recommended:

- The first on is a feedback sheet which can be send to the process team directly per mail or can easily be printed. It shall be accessible form every process element. So wherever the user has some issues there should be a possibility to directly fill in the feedback sheet and send it to the process management team.
- The second mechanism would be of organisational nature. As mentioned in 7 there are defined responsibilities different processes. So every process has an ‘owner’. In each part of the organization the process is used there are defined responsibilities for the process. Regular meetings called ‘process rounds’ including process owner and organisational responsibilities discussing process usability and improvements are established. The discussion of best practices which can lead to organisational comprehensive development process improvements is one of the most important benefits in the context of an efficient development. It furthermore supports standardization of development methods and tools, because every part of the organization is interested in the most efficient workflow.
- The third one is user questionnaires to force contact with the users. Experience has shown that a lot of users do not give feedback on their own initiative. Surveys towards specific topics are a good possibility to get a wider feedback spectrum. In this case it is important to give the user a chance for answering in free text. A checkbox system is useful for getting easy metrics but restricts the user’s possibility for answers. Best practice is a mix of checkboxes and free text with a maximum of seven to ten questions.
- User meetings and trainings are an important medium as well to communicate processes and changes in processes for multiple reasons. It is a good method to talk about the processes with the engineers. This avoids the usual prejudice (which comes from missing knowledge, not from facts) that processes were developed far away from practice and don’t fit reality. Depending on the system’s grade of detail it is important to talk about best practices documented in the process descriptions and methods to make the users especially familiar with the documented workflows.

Experiences show that several process improvements are the result of the process rounds and the user meetings. Meaning direct communication between process modellers, process owners and users is the most rewarding way for continuous process improvement.

Additionally to technical correctness a main point for user acceptance of a process model is usability of the process management tool and the way of presenting the processes. Especially in large process models with a lot of detailed information a user focused approach is highly recommended.

In large process models, which also contain methods, work instructions and tool guidelines (see Chapter 2 Meta Model) a good structure of information is essential for success. The challenge is placing the information, where users expect them to be. A use case based thinking while designing the user interface is a good opportunity to make the navigation as simple as possible and place all relevant information in the right context. This also avoids overloading the user with too much information. Especially the entry point of a process model is of importance, because there should be navigation possibilities for power users searching for a specific process element and new users which want to make themselves familiar with the process for the first time.

In the following some examples will give an impression of the user interface:

- A hierarchical navigation menu (‘breadcrumb path’ like Windows Explorer offers) allowing access to all process elements makes it possible to navigate to every process element, meaning activity, template or even chapter with even one mouse click. This is especially helpful for power users knowing exactly which element to look for.

- As a new user a structured overview is important. Therefore a combination of long scrollable sides and a horizontal navigation bar is useful. The navigation bar gives the opportunity to easily scroll throw the side showing the different entry points of the process model, like the process landscape, role model, documents, trainings and methods overviews. Having all elements on one page in a flat structure improves clarity and structure.

- It is important to balance the quantity of information on different grades of details in the process. For example RACI-Matrix and a short description on process level are useful for a process overview. Detailed information like engineering how-to and manuals towards specific topics like generating a test specification or writing a change request are placed much better at the corresponding activities. Resulting from this a view concept supporting the information structure for different process elements is necessary.

Automotive SPiCE and ISO 26262 require a set of output documents for every project. They are supplemented by the organizations’ standards and best practices. These documents can be provided in a generic template system. Making them available for every project increases reusability and saves best practices and experiences for future projects. Providing templates is not an innovation, but especially in this context a database based way of providing and managing templates will show advantages:

Holding chapters in a database give the opportunity to generate documents based on the same chapters, without copying them over, just by linking them together. This allows consistency between different documents. The following example explains the benefit:

All review protocols in the engineering processes contain a checklist for formal aspects. If there are new requirements for formal aspects of a document, the related changes must be implemented in just one chapter and all documents containing this chapter are updated automatically. This offers consistency in an easy way and reduces process work significantly.

Starting a new project, one of the first steps is generating the basis for the project documentation. Especially formal aspects like project name, name of the organization, location, name of factory are pieces of information which do not change. Generating this information automatically reduces workload (and reservation) for creating and reviewing the documents.

The automatic document generation should contain table of content, list of figures, correct header and footer information, and object id from the database. This combines the benefits of a classic office document and a database based content management system.

9 Summary and Conclusion

To our experience, there is no ‘golden way’ in modelling processes. A lot of decisions depend on the goals of the organisation including established workflows and processes. This paper shows a meta-model for an engineering process including comprehensive documentation and engineering methods focusing on real life activities as the centre of information. The method-based tailoring concept allows different workflows and tools in different projects using the same basic processes. A concept for generic roles helps keeping consistency and reduces workload for modelling specific roles. Another point to guarantee completeness and consistency is mapping requirements derived from international standards to the process elements. This enables structured consistency checks and supports the correctness of tailoring.

Additionally to the issues arising from process content, there is the aspect of managing the whole process system, which is supported by a structured change management process. Concerning process changes, the *users* need to be in focus and should be involved. They are the key of a well-documented, established and informative process which improves development.

As a conclusion, a capable process management tool enables a lot of new possibilities especially when it shall be used consistently in several locations and – as a novelty – covers the three disciplines hardware, software, *and* mechanical development. The actual application of KEEP on all levels of hierarchy enables Volkswagen to cope with the rising challenges of safety critical electronics development.

Literature

[Automotive SPiCE PAM 2.5] Automotive SPICE® Process Assessment Model; PAM 2.5; Released 2010-05-10; The Procurement Forum/ Automotive SIG http://www.automotivespice.com/fileadmin/software-download/automotiveSIG_PRM_v45.pdf

[Automotive SPiCE PAM 3.0] Automotive SPICE® Process Reference Model / Process Assessment Model Version 3.0 Released 2015-07-16; VDA QMC Working Group 13 / Automotive SIG http://www.automotivespice.com/fileadmin/software-download/Automotive_SPICE_PAM_30.pdf

[HIS-Scope] Results from the HIS (Herstellerinitiative Software) Woking-Group 'Process Assessment': http://portal.automotive-his.de/images/pdf/ProcessAssessment/his_process-scope_automotivespice_v01.pdf

[Lintra15] Vom Prozess zum Projekt - Erfahrungsbericht mit QUAM zum SPiCE Level 3, Lintra Anwenderforum, Oktober 2015, Magdeburg

[ISO 26262] Road vehicles -- Functional safety; Part 1-10; 2011-2012;

[Meta-Models] An Overview of Industrial Process Meta-Models, Erwan Breton, Jean Bézivin, ICSSEA 2000-14, Breton

[QUAM] Product information /technical data for QUAM: https://www.lintra.de/produkte/quam-prozessmanagement

[BPMN] Business Process Model and Notation™ Specifications, http://www.omg.org/spec/BPMN/

[BPMN1] Opportunities and constraints: the current struggle with BPMN, Jan Recker (2010) Business Process Management Journal, Vol. 16 Iss: 1, pp.181 - 201

[BPMN2] Analysis of Most Common Process Modelling Mistakes in BPMN Process Models, Tomislav Rozman, Gregor Polančič, Romana Vajde Horvat (2007), EUROSPI'2007: industrial proceedings, European Software Process Improvement, At Potsdam, Germany

10 Author CVs

Philipp Lackmann

Philipp Lackmann is a software engineer in a centralised department for electronics development and process management. He holds a degree in information systems engineering from technical university of Braunschweig. His work focuses on software processes for embedded systems with functional safety aspects.

Christian Steinmann

Christian Steinmann is a senior consultant for software process improvement and director of SynSpace's office in Austria named HM&S IT-Consulting GmbH. He holds a degree in Technical Mathematics from TU Graz. Since 1995 he supervises the development of the SPiCE 1-2-1 assessment tools. He conducted more than 100 assessments for SPiCE and CMMI and is certified instructor for CMMI-DEV 1.3 and Scrum Master. On Fridays he is lecturer for Informatics and Software Development at the FH-Joanneum degree programme Automotive Engineering.

Dr. Fabian Wolf

Dr.-Ing. Fabian Wolf is a manager at Volkswagen AG. Dr. Wolf received the diploma in 1996 and the PhD in 2001 both in electrical engineering from the technical university of Braunschweig. His main research areas have been design automation tools for WLAN base stations and software timing analysis. From 2001 to 2008, he has been working on design automation and test for engine control software at Volkswagen Wolfsburg. Since 2008 he has been working on steering control software and the according engineering and management processes at Volkswagen Braunschweig.

Scope and secrets of reviews within the automotive supplier industry

Norbert Merk, norbert.merk@zf.com
Bernhard Krammer, bernhard.krammer@zf.com

Abstract

The basic idea of reviews is simple but effective: a team inspects work products or processes and detects discrepancies early on, long before something is tested or even delivered to a customer. Based on the experience of several successful product developments within the truck and bus division of ZF Friedrichshafen AG the time was right to work on a bachelor thesis exploring the current activities, methods and approaches uncovering key factors of this quality-ensuring activity.

Starting with an intensive literature research including current standards like ISO/IEC 15504, IEEE 1028 and ISO 26262 the goal was to establish a new internal directive ensuring the full effectiveness of reviews for all future development projects. It is really simple to just perform reviews since they are easy to learn and do not cost much except time and brainwork of the attendees. But why is it that there seems to be a huge difference in usefulness and acceptance of reviews among different departments or topics? Maybe it is the various methods applied or the actual phase of the project when the review is performed? Is there a golden rule to the number of participants or who should be invited after all? And should a project manager actively ask for reviews as a way to ensure success or simply trust the specialists?

Many questions arise but what are the answers? ZF Friedrichshafen AG is trying to figure out what works best in their environment between blindly following rules set by common standards and a long company tradition based on trust.

Keywords

Review, Assessment, Audit, ISO/IEC 15504, ISO 26262, IEC 61508, IEEE 1028, VDA 6.3, ZF Friedrichshafen AG, Quality assurance

Terminology, Technical Documentation and Standards: Safety and Security for Industry and Engineering Environments

Frieda Steurs
KU Leuven and TermNet - the International Network for Terminology
frieda.steurs@kuleuven.be

Hendrik J. Kockaert
KU Leuven
hendrik.kockaert@kuleuven.be

Gabriele Sauberer
TermNet - the International Network for Terminology
gsauberer@termnet.org

Blanca Nájera Villar
TermNet - the International Network for Terminology
bnajera@termnet.org

Abstract

The efficient and effective use of specialised language is a prerequisite for successful communication in industry, education and scientific communities. Specialised communication plays a crucial role in engineering /automotive / industry environments, where terminology management has a direct impact on safety and security issues and, as a consequence, on the liability of products and services. In the automotive industry, user manuals are an integral part of the product, and the manufacturer is legally responsible for it. In order to avoid safety and quality issues, terminology in user manuals and in technical documentation need to be managed and the quality of texts and translations needs to be measured.

There is a correlation between process quality, service quality and product quality. Whenever organisations design their business processes and work-flows according to general or industry-relevant quality standards, this has an influence (both direct and indirect) on the quality of their processes, products and services. Terminology management, quality assurance in technical documentation, state-of-the-art standards and training of professionals are key success factors. Whenever an organisation tackles one or two or all three of these aspects, this has an impact on the other aspects or the entire quality system.

Keywords

Specialised language, engineering, technical documentation, safety and security, terminology, quality, product liability, standards and training.

1 Terminology and specialized communication

Terminology is the core component of all specialized communication. It is about the linguistic and non-linguistic designations for concepts and objects in a given subject area. Ontologies are the most important way to describe the relation between concepts and to model a conceptual system. Through ontologies, we can make terminologies operational.

According to ISO 1087-1, terminology work is the systematic collection, description, processing and presentation of concepts and their designations. This means that terminology is concerned with concepts and conceptual systems, making them explicit by means of definitions and designations as well as phrases within languages for special purposes. Terminology science provides the basic concepts and best practices for terminology work and terminography, i.e. for the systematic documentation and maintenance of terms. Terminology and the study of language for special purposes is situated on the intersection of various fields of knowledge (logic, ontology, linguistics, information science, language policy, language planning etc.).

Terminological units can be seen not only from the **linguistic dimension** (that examines *existing linguistic forms as well as potential linguistic forms that can be created in order to name new concepts*) but from the **cognitive dimension** (which examines the concept relations and how the concepts constitute structured sets of knowledge units or concept systems in every area of human knowledge, as well as the representation of concepts by definitions and terms) and the **communicative dimension** (that focus on the *use of terms as a means of transferring* knowledge to different categories of recipients in a variety of communicative situations and covers the activities of compilation, processing and dissemination of terminological data in the form of specialized dictionaries, glossaries or terminological databases, etc. (Sager 1990)).

Referring to specialist communication, terminology can be defined as the entirety of all concepts and terms of a specific subject field. Efficient communication with regard to technical language or standardization of concepts is not possible without an exact definition of the concepts under discussion. That means that an onomasiological approach is needed; i.e. the starting point is the concept that has to be defined in an unambiguous way. Once the concept is defined within a concept system, such as a taxonomy, the terms can be correctly assigned and different terms in different languages can be correctly linked to the concept under discussion.

Communication is getting more and more complex, not only between specialists and laypeople, but even between experts in one and the same discipline. This is especially true when communicating across and beyond language and cultural borders. Today, technical and specialist communication comprises around 80% of all information exchanged across the new communication paths of a borderless and multilingual information society. Within the given economic context, each company and institute faces a growing technical, scientific and legal complexity. As a consequence, a lot of very sophisticated documents have to be written and translated: e.g. legal and administrative documents, technical specifications, spare parts catalogues, user manuals, procedures, reports etc.

1.1 Terminology management in engineering and technology environments

The machinery sector as important part of the engineering industry can serve as example of the need for professional terminology management. Machinery consists of an assembly of components, at least one of which moves, joined together for a specific application. The drive system of machinery is powered by energy other than human or animal effort. Therefore, important EU Machinery Legislation has come into force. One of the main legislations governing the harmonisation of essential health and safety requirements for machinery at EU level is the Machinery Directive 2006/42/EC

The Directive promotes the free movement of machinery within the Single Market and guarantees a high level of protection for EU workers and citizens. Correct and consistent terminology is a quality and safety factor.

New products must be accompanies by information, most often as an Instruction Manual. All European product safety Directives require information to be made available to end users to enable the safe use of products. Not only end users are implied, also the installers need technical information to enable the product to be safely installed. User instructions should be comprehensive, easy to understand, and in the user's own language. Other information provided on the product such as warnings, which may be given in pictorial form, should be explained in the user instructions. User instructions are essential for safety and should be provided in printed form. Quality control in this field includes terminology management in order to come to a clear understanding of the technical instructions.

1.2 Terminology management in medical and health industry environments

Health matters - in all communities, all over the world. Access to health services, quality care, and safe medical practices and equipment is important to people everywhere. ISO has over 1.300 standards that focus on health in many sectors, ranging from dentistry to medical devices**,** and health informatics to traditional medicines.

Standards help improve health in many ways, including:

- Promoting global harmonization of medical practices
- Protecting the health and safety of patients and healthcare providers
- Supporting efficient exchange of information and protection of data and
- Improving the quality of care.

In the domain of medical supplies, drugs, and clinical studies, due to the critical nature of the information, we have a lot of challenges in communication. Starting with the source text quality, every step in the communication process is crucial and must be monitored carefully.

We can refer to companies such as Medtronic, who implemented a perfect terminology management systems starting with the source text, defining all the concepts they were handling in critical medical devices, and then coining the terms in the different target languages.

Another aspect of medical communication is the field of clinical studies. This involves specialist to layman communication, and includes medical and ethical aspects. Clinical trials are research studies that explore whether a medical strategy, treatment, or device

is safe and effective for humans. These studies also may show which medical approaches work best for certain illnesses or groups of people. Clinical trials produce the best data available for health care decision-making. The purpose of clinical trials is research, so the studies follow strict scientific standards. These standards protect patients and help produce reliable study results. Clinical trials are one of the final stages of a long and careful research process. All of these results are important because they advance medical knowledge and help improve patient care. However, the Contract Research Organizations (CRO) behind these procedures have to cope with huge challenges to improve the quality of the communication. Faulty communication and the lack of terminology management slows down their processes considerably.

2 Terminology Management as a crucial quality factor in technical communication and documentation

In many organizations, the amount of content developed grows exponentially. The amount of technical documents, legal and administrative information etc. is huge.
This has two reasons:

1) The impact of high-level technology in many products and processes triggers more complex texts and information
2) Large companies have a complex structure with different departments dealing with the products and processes on different levels. This is the reason for multiple versions of the same content.

Managing the source texts in the company (i.e. the texts in the mother tongue) is a first challenge where terminology management comes in as a key issue in non-ambiguous information. The multiplication of the problems when dealing with multilingual issues and translation will be touched upon later in this paper. Content management is a new important issue in many large organizations and companies. And in the same way, terminology is the key to good content management and a major organizational asset.

The quantity and difficulty of specialist texts have increased, along with the demands on the technical and specialist documentation (laws, norms, customer and corporate language). Experts in technical documentation must become familiar with the terminology of their field. Frequently, parts and components have different names in one and the same company. Often much time is lost before the clear terms established there find their way into the linguistic usage of technical languages, not to mention the fact that there is no possibility for all technical terms to be standardized. Thus, for the good of specialist communication, it is very important that the meaning of complex terms be defined as early as possible, the results be documented and made available to potential communication partners. An example: one small modification, such as changing part of a technical component, will affect all models in which this part can be found. This means that all language versions of all model descriptions must be revised. This can be very expensive and conceals the risk of errors and confusion among all stakeholders.

Both at European and at national level, lawmakers place special requirements on the development of terminology, especially in the area of technical documentation. EU standards, product liability, and certification require companies to deliver, as an integral part of their products, documentation that meets safety requirements. Defective documentation is considered a product defect that leads to complaints or even claims for damages.

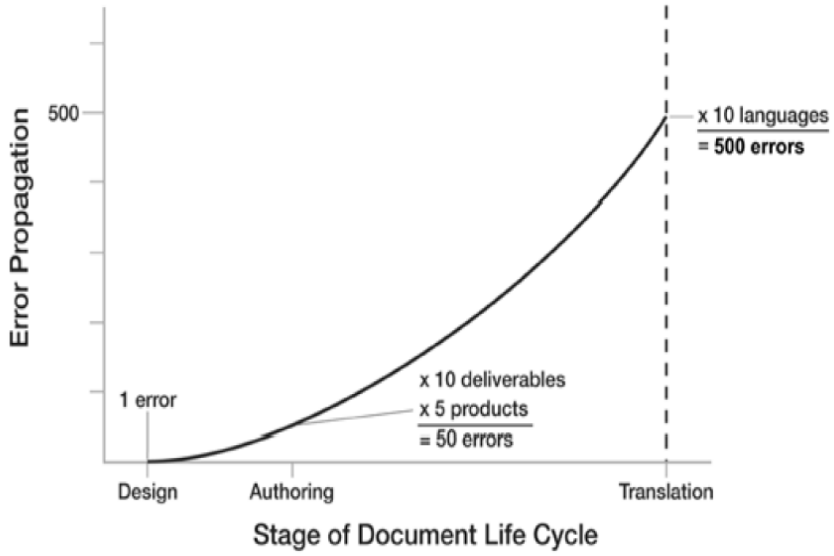


Illustration 1: Stage of Document Life Cycle

Illustration 1 is an example of the costs of incorrect terminology (by Kjeldgaard): Outsourced translations tend to cost up to 50% more if the terminology in the source text is inconsistent.

The real wealth in a company is the knowledge that is handled and carried by the different employees. This wealth is at the same time the liability in every commercial institute; the companies will strive to make the implicit knowledge explicit, the knowledge in the heads of the individual employees. Good information handling is guided through correct communication, with clearly defined concepts and the terms related to these concepts in the different languages.

Terminology is clearly seen as having a crucial role in good communication. This holds both for internal AND external communication in a particular company. Not only the communication between different departments or units within a particular organization has to be clear and unambiguous; the customer must equally benefit from clear handouts, manuals and other technical communication.

The solution to this type of complex data management can be found in the creation of a central knowledge repository, where the concepts can be defined for the different subdomains under discussion, and the relevant information and terminology can be added.

Strong brands build on consistent communication, and this is impossible without terminology management. Content management and retrieval of information thrives on maintaining terminological consistency. If companies want to communicate both on internal and external level in a clear way, they need an advanced content management to be in place, together with a localization strategy to address customers and technical agencies all over the world.

First there is the need to examine the quality of the source texts used in the company, both for internal and external communication. This may be done using a terminology extraction tool to create a first "lexicon" of the in-house terminology used. This

terminology database has to be made available to all departments and all the employers in the company. The terminology has to be updated constantly, and new terms have to be processed and included. This type of source text terminology management is extremely relevant in the early stage of the product design: the authoring system and the work of technical writers and other specialists will benefit from this step considerably. The better the quality of the source text, the less confusion and mistakes will appear in the translations. According to JDEdwards (Xerox), it costs 10 times more to fix a term at the end of the production cycle than at the beginning.

Once the concepts and the correct terms have been defined, then the attention can go to the multilingual target markets. How to make standard industry glossaries for individual target markets? The multilingual terminology has to be integrated in a translation memory, and has to be available for sharing with freelance translators. According to Cerrella Bauer, S., eDITion, 2/2007, when localising an application where the character strings contain approximately 20% of technical terminology, the use of a terminology database containing approx. 70% of the technical terminology to be translated resulted in a time saving of approx. 32%.

A typical example where inconsistent terminology use may occur, is the launch of new software products, where all the terminology used in the user interface, the product documentation, the help file, internal communication and training materials have to be consistent. Inconsistent terminology makes it difficult for the user or the trainer to use to product.

Companies tend to be hesitant to embark on a systematic approach to terminology management, but this can be seen as a smart investment. It is not only beneficial to the pure communicative and translation workload, but it leads to better processes, higher quality and shorter turnaround time during the localization process.

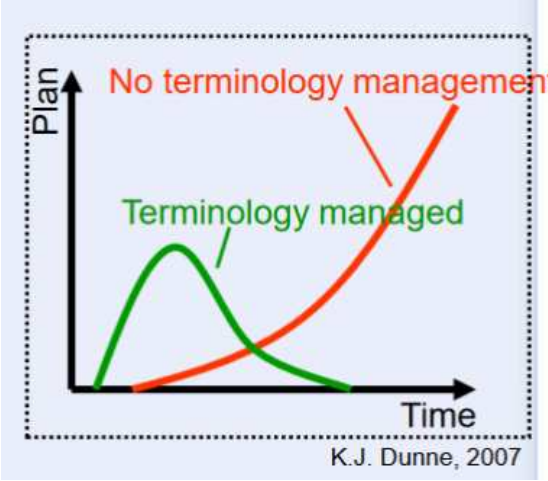


Illustration 2: How to present the business case for terminology project according to Dunne, K. J. In: Multilingual April/May 2007, S. 32-38

3 Quality control methods, standards and metrics for engineering environments

Metrics have always been used to help guide managers with decisions about their organizations. The quality of information in technical and user manuals has always attracted attention. Companies are very careful as to how to monitor and control the quality of their technical documentation. In this paper, we briefly touch upon two of these metrics: the SAE J2450 and the ISO 17100.

3.1. The SAE J2450 metric for quality control of multilingual document management for the automotive industry

In fall of 1997, "J2450 Task Force on a Quality Metric for Language Translation of Service Information" was formed under the auspices of the Society of Automotive Engineers, Inc. (SAE). Its task was to create a standard to measure translation quality of service manuals. Initially, the task force consisted of representatives from General Motors, Ford and Chrysler as well as different translation service providers.

The purpose of the task force was the development of a standard for the automotive industry which objectively measures and evaluates the translation quality of automotive service manuals. The typical target groups were automotive technicians.

The standard was designed to measure the translation quality regardless of the source language, regardless of the target language, and regardless of how the translation was performed (i.e., human translation or machine translation). In contrast to the approaches described below (ISO 17100), this standard is not a quality standard defining processes and procedures, but a standardized, quantitative metric for the linguistic quality of a translation.

Main Category	Abbreviation	Sub-Category	Weight s / m
1. Wrong Term	WT	Serious error or minor error	5 / 2
2. Wrong Meaning	WM		5 / 2
3. Omission	OM		4 / 2
4. Structural Error	SE		4 / 2
5. Misspelling	SP		3 / 1
6. Punctuation Error	PE		2 / 1
7. Miscellaneous Error	ME		3 / 1

Illustration 3: Interpretation of the SAE J-2450 quality metric

Before SAE J2450, quality measurement on language translation in the automotive industry has largely been subjective, if such measurement was undertaken at all. If an automotive company did set up a quality process with its translation suppliers, the quality of translated service information would generally be reviewed by in-country validators designated by the automotive company. Markups of the translated documents were provided back to the translation supplier for correction and editing. There would

likely not be any standardized measurement metrics for determining or rating quality in a manner similar to methods used in the manufacturing side of the automotive business. The risks of low-quality translations of service information include erosion of customer confidence, higher warranty costs, and (at an extreme) damage to vehicles or injury to people.

Thus, the objective of the metric was to establish a consistent standard against which the quality of translation of automotive service information can be objectively measured. The metric allows an evaluator to tag errors in a translation and compute a weighted, numeric score that represents the quality of the translation.

The standard is applicable for any other industry where safety and security matters – and where texts are written in specialized language with a high percentage of terminology. Today, many industries, such as the automotive, aviation or health industries, assess the quality of translated technical documentation and user manuals on a regular basis. After lost cases and damaged reputation these industries are aware of their liability for the translation of the technical documentation as part of the product.

3.2. ISO 17100: Translation services -- Requirements for translation services

The first multi-national standard to define requirements for the provision of quality translation services has been the European Standard EN 15038 "Translation services – Service requirements" published by the European Committee for Standardisation (CEN) in 2006.

The standard specified requirements with regard to human and technical resources, quality and project management of a Translation Service Provider (TSP). Special attention was payed to the competences of translators and to the "revision" of translated texts: As default, every translation shall be checked by a second, equally qualified translator (see below).

For the first time, also the TSP/client relationship was tackled, requesting the TSP to actively communicate with customers, in order to understand their needs, such as the purpose and target audience of the translation, delivery dates, commercial terms and conditions of the project, etc.

The procedures in translation services were specified and a major issue in EN 15038 was - and still is in ISO 17100 – the requirement to revise translations, i.e. to apply the 4-eyes-principle through a second, equally qualified translator who compares the translated text with the source text.

According to Peter Jonas from Austrian Standards (Jonas and Sauberer 2014), the two standards EN 15038 and ISO 17100 are identical with the exception of only a few minor points. The most striking difference to EN 15038 is the "Terms and definitions section" in ISO 17100. Whereas the European standards defines only a few terms which are used in the standard and thus are needed to understand the standard itself, ISO 17100 goes far beyond that approach by defining in total 42 terms which are commonly used in the global translation industry. Thus, apart from being a standard for service provision, ISO 17100 may be considered as a terminology standard for the provision of translation services.

Thus, to use ISO 17100 terminology in internal and external communication, has become a quality criteria for TSPs, because unclear communication with customers, with concepts and terms continuously mixed up, results in misunderstandings and in poor quality of products and services. "Revision" is a crucial step in the service provision process where a second translator examines the translation output for any errors and other issues, and

its suitability for purpose. Revision is a bilingual process, whereas editing, reviewing or proofreading are all monolingual processes.

4 Certification and qualification of professionals

Professionals dealing with technical writing do terminology work and need professional terminology management. The European Certification and Qualification Association (ECQA) has the appropriate certificate meeting these needs: The ECQA Certified Terminology Manager (CTM) basic, advanced, and for special industries, such Engineering/Technology and Medical/Health, to be followed by Software Process Improvement, etc.

TermNet joined the ECQA in 2007. Since then, certification schemes and a skill cards for the job role ECQA Certified Terminology Manager have been developed by a consortium of international terminology experts and TermNet members. The ECQA, together with TermNet and their cooperation partners, guarantee that all certificate and preparation courses worldwide are built according to the same structure and standards.

The training initiative ECQA CTM combines the various competences of professionals active in the areas of information & communication, classification & categorization, and translation & localization, offering training and qualification not only for language and terminology professionals, but also for all experts in industries where terminology management is a critical factor for quality and safety.

The ECQA CTM also considers the fact that terminology managers in today's companies and organizations don't necessarily have a university degree in linguistics, translation or terminology.

The training is design in different levels to address the needs of different groups of professionals dealing with terminology projects. The basic level of CTM focusses on understanding how terminology work is embedded in organisations and work environments and the basic principles and methods of terminology management. The ECQA Certified Terminology Manager – Advanced addresses professionals who already have some experience in working with terminology; either as a full-time specialist or as part of their regular job.

Since 2015, CTM is available also as specialised qualification for professionals in various industry environments (engineering/technology, automotive/aviation and medical/health terminology. Since 2010, more than 300 professionals all over the world qualified as "ECQA Certified Terminology Managers".

5 Conclusions

In an increasingly international, globalized world, effective and unambiguous multilingual communication has become crucial. Terminology plays an essential part in specialised communication. Terminology management can be considered as integral and quality assuring part of any product and service in the areas of

- information & communication
- classification & categorization
- translation & localization

The international information society has created a huge demand for multilingual technical, scientific and legal documents. Professionals from all industries are faced with

the need to care for high quality translations of texts, user manuals, etc. To know the respective standards and quality requirements is highly relevant for all industries.

Competitive business environments with dynamic development processes, such as Agile and SCRUM, make it necessary to constantly maintain and update corporate / organisational knowledge and to make it accessible for changing users and user groups.

Terminology management and the use of reliable terminology resources play a crucial role in quality assurance, consumer protection and cost savings.

1) Terminology helps to guarantee the quality of the product and process

- All companies produce terminology in written and oral form.
- Terminology is a part of the process and, also, of the product.
- Terminology management is crucial at the production of source texts (cooperate knowledge) and technical documentation and should be implemented though out the process.
- All stakeholders in a company should be included in the terminology policy planning.
- Terminology is an asset for a company and helps the company to stand out from the competitors.

2) Terminology helps to protect costumers

- Clarity and safety are key premises in technical documentation for the medical industry.
- Bad terminology management can influence the quality of the product and lead to legal claims.
- Technical documentation and, as a consequence, terminology are part of the product.
- Quality assurance is a key point for the industry and technical documentation.
- Wrong terminology in technical documentation can cause damages to the costumers.

3) Terminology management saves costs

Terminology management in technical documentation can lead to:

- 5% reduction of the translations costs,
- 10% reduction of the general cost through a 100% matches in the translation memories,
- 10% reduction of the work,
- 50% less in translation work,
- reduction of 60% in the translations questions and queries.
  (Schmitz und Straub, 2010)

6 Bibliography

- Dalla-Zuanna, Jean-Marc (2010): "Evaluierung fremdsprachiger technischer Dokumentation im Bereich Vertrieb After Sales der Marke Volkswagen Pkw. Direkte Qualitätsmessung", in: MDÜ 2/2010, pp 20-25.
- Karsch, Barbara Inge and Sauberer, Gabriele (2011): "Terminological Precision - A Key Factor in Product Usability and Safety", in: Design, User Experience, and Usability. Theory, Methods, Tools and Practice. Lecture Notes in Computer Science Volume 6769, 2011, pp 138-147.
- Kockaert, H.J; Steurs,F. (eds) (2015) Handbook of terminology. John Benjamins (Amsterdam/Philadelphia)
- European Certification and Qualification Association: www.ecqa.org
- B. Nájera Villar, D. Brändle: There Is No Knowledge without Terminology: Key Factors for Organisational Learning. EuroSPI 2012: 300-309
- ECQA Certified Terminology Manager – Engineering Skill Card: http://www.termnet.org/english/products_service/ecqa_ctm-engineering/2015_online/programme.php
- ISO 1087-1:2000 Terminology work -- Vocabulary -- Part 1: Theory and application
- ISO 17100:2015 Translation services -- Requirements for translation services: http://www.iso.org/iso/catalogue_detail.htm?csnumber=59149
- SAE J2450 - 200508: Translation Quality Metric: http://standards.sae.org/j2450_200508/
- The Wüster Archive - a special node in a European digital archive network. In: E. Oeser, C. Galinski (Ed.): Eugen Wüster. Leben und Werk. Ein österreichischer Pionier der Informationsgesellschaft. Vienna: TermNet, 1998, p 169-174 [2] S.E.
- Budin, Gerhard: Ontology-driven translation management, in: Dam, Helle V.; Engberg, J.; Gerzymisch-Arbogast, H. (Ed.) Knowledge Systems and translation. Berlin: de Gruyter, 2005, pp. 103-123
- Directive 2006/42/EC of the European Parliament and of the Council of 17 May 2006 on machinery, and amending Directive 95/16/EC
- Sager, Juan C. (1990), A Practical Course in Terminology Processing, John Benjamins Publishing Company
- Jonas, Peter and Sauberer, Gabriele (2014): „Translation Quality Goes Global: From EN 15038 to ISO 17100, in: Proceedings oft he XXth FIT World Congress, Berlin 2014, Volume II, p 941-947.

7 Author CVs

Frieda Steurs, KU Leuven

Frieda Steurs is a Full Professor in Terminology, Technical Translation and Language Technology and a member of the QLVL research group of KU Leuven.. She teaches Terminology and Translation Technology. Her research includes terminology management, language technology and specialised multilingual documentation. This has lead to several projects with industrial partners and government organisations. She is the founder of NL-TERM, the Dutch terminology association for both the Netherlands and Belgium. She is also the head of the ISO TC/37 standardisation committee for Belgium and the Netherlands. She is a research associate at the University of the Free State (SA) and the president of TermNet.

Hendrik J. Kockaert, KU Leuven

Prof. Dr Hendrik J. Kockaert lectures French Linguistics, Terminology, Legal Translation, and Translation Technology at KU Leuven, Faculty of Arts, in Antwerp. Since August 2015, he is Dean of the Faculty of Arts on Campus Sint Andries in Antwerp. He is a Research Associate at the University of The Free State, Republic of South-Africa.

He is a certified LICS Auditor for granting ISO 17100 certification to translation services, and he is the Editor-in-Chief of The Journal of Internationalisation and Localisation [JIAL], published by John Benjamins Publishing Company. He is the Chairperson of ISO TC 37 SC 1 and a member of NBN, the Belgian Standardization Institute. He is an expert in the development of ISO terminology standards. He is a certified ECQA (European Certification and Qualification Association) Terminology Manager Trainer and Job Role Committee Provider. He was the coordinator of the following projects financed by the European Commission: QUALETRA (JUST/2011/JPEN/AG/2975), and LIT Search (JUST/2013/JPEN/AG/4556). He publishes in the areas of terminology, translation quality assurance, and translation technology. Together with Prof. Dr Winibert Segers and Televic, he develops TranslationQ, a CAT tool that delivers an automated evaluation and revision process.

Gabriele Sauberer, TermNet – International Network for Terminology

Gabriele holds a PhD in Linguistics and a Master Degree in General Management. After pursuing an interdisciplinary bundle of studies with focus on Eastern European Languages and many years of scientific project management at the University of Vienna, Gabriele finished the post graduate course "European Project Management (EUPROMA)" and successfully manages the International Network for Terminology since 2002. Gabriele is Head of the TermNet Group (TermNet, TermNet Business GmbH and TermNet Americas). For the European Commission she acted as consultant to European eContent and 6th Framework Programmes and designed many projects and trainings at European, regional, national and international level. From 2007 to 2010, Gabriele was teaching diversity management, intercultural management and project management at the Centre for Translation Studies of the University of Vienna.

Gabriele is a certified quality auditor, EN 15038 and ISO 17100 lead auditor and expert in several standardization committees (Terminology, Translation, Human Resources, Diversity management, Corporate Social Responsibility).

Gabriele is a pioneer in Innovation and Quality in the Language Industry. Together with Austrian Standards, she founded the international certification platform LICS in 2007, today´s world-market leader for quality certificates in the language industry. She is active as:

- Director of the International Network for Terminology (TermNet);
- Managing Partner of the private companies TermNet Business GmbH in Austria and TermNet Americas in Canada;
- Co-founder and international partner of the Language Industry Certification System (LICS);
- Founder and Vice-President of the Forum European Diversity Management (FEDM);
- Owner of the private company Dr. Sauberer European Business Consultancy;
- Vice-President of the European Certification and Qualification Association (ECQA).

Blanca Nájera Villar, TermNet – International Network for Terminology

Blanca Nájera Villar holds an university degree in translation and interpreting gained from the University Alfonso X, el Sabio in Madrid. She is highly skilled freelance translator, with strong experience in the profession after working as a project manager and in-house linguist with several translation agencies across both Spain and Germany. In 2012 she finished her postgraduate Master on Terminology at the University Pompeu Fabra, Barcelona, with a final master project on the ECQA terminology (ECQA Term).

She works for TermNet, the International Network for Terminology, since 2004 as project and event manager with focus on terminology strategies for the industry, quality assurance, certification and innovation and also as accredited trainer for ECQA Certified Terminology Manager- Basic. Blanca is Deputy Director of TermNet since October 2011.

A Compact Introduction to Automotive Engineering Knowledge

Andreas Riel[1], Monique Kollenhof[2], Sebastiaan Boersma[3], Ron Gommans[4], Damjan Ekert[5], Richard Messnarz[5]

[1]InnoPlusPlus & Grenoble Institute of Technology, France
andreas.riel@grenoble-inp.fr
[2]Symbol BV, the Netherlands
monique.kollenhof@symbolbv.nl
[3]ROC Summa College, the Netherlands
sc.boersma@summacollege.nl
[4]ROC Ter AA, the Netherlands
r.gommans@roc-teraa.nl
[5]ISCN LTD/GesmbH, Ireland/Austria
{dekert, rmess}@iscn.com

Abstract

Professionals in the automotive industry, teachers at Vocational Education and Training institutions (VETs), and training and consulting organizations from all over Europe developed a curriculum of basic skills needed to assume modern Automotive Engineering job roles. Based on this curriculum, defined in the form of two skill sets, training materials, as well as a text book and exercise book have been authored. For VETs and industry pilot trainings were organized.

The training not only prepares automotive students for their future jobs in the automotive industry but also enables professionals in the automotive industry to teach their newly graduated engineering employees in specific and fundamental knowledge and skills that form the basis of the growing variety of engineering job roles in the automotive industry. This article describes the experiences so far and looks ahead to the near future. The proposed Automotive Engineering curriculum provides a value-added springboard for engineers to assume engineering job roles in the modern automotive industry.

Keywords

Automotive engineering, automotive professional, vocational education and training

Published in: Springer Communications in Computer and Information Science (CCIS) vol. 663

Functional Safety Considerations for an In-wheel Electric Motor for Education

Miran Rodic[1], Andreas Riel[2], Richard Messnarz[3], Jakub Stolfa[4], Svatopluk Stolfa[4]

[1]University of Maribor, Faculty of Electrical Eng. and Computer Science, Slovenia miran.rodic@um.si
[2]EMIRAcle & ISCN Group, France
andreas.riel@grenoble-inp.fr
[3]ISCN LTD/GesmbH, Ireland/Austria
rmess@iscn.com
[4]Technical University of Ostrava, VSB, Department of Computer Science, Czech Republic
{jakub.stolfa, svatopluk.stolfa}@scoveco.cz

Abstract

The European Automotive Quality Sector Skill Alliance AQUA has been establishing practice-oriented education and training program on modern automotive engineering challenges since 2012. In the context of their most recent activity, they transfer their certified industry training program on integrated automotive quality engineering and management to higher education. This article introduces a practical example that is used in this new curriculum to explain the functional safety dimension of integrated automotive quality. Since the students are highly interested in electric and hybrid vehicles technologies, the example of an e-motor control for the drivetrain was chosen as a case study.

Keywords

Functional Safety, Higher Education, Quality in Automotive

Published in: Springer Communications in Computer and Information Science (CCIS) vol. 663

The Need for Policy Rationale

Joanne Schell (joanne.schell@nxp.com), Paul Schwann (paul.schwann@nxp.com)

NXP Semiconductors Austria GmbH, Mikronweg 1, 8101 Gratkorn, Austria

Abstract

Within the automotive industry, many important standards exists around the development process. In order to effectively adopt these standards (or any standards, for that matter), it helps to understand why the procedure have been established in the first place. The standards offer some rationale for the policies but not always or it's not compelling or the rationale is not easy to find. This paper describes the common approaches to policy justification and the reasons justification is crucial for a robust adoption of policies (including changes in policy.) The paper also provides an efficient method to communicate policy rationale, providing an example of the method with some of the more common policy hurdles that the authors have encountered, based on many years of experience in the automotive industry.

Keywords

Justification, Change Management, Implementing SPICE, CMMI, VDA, Standards, Automotive, Requirements Management, Configuration Management, Risk Management

1 Introduction

Procedures, as defined by standards, are an essential part of most industries; they are ideally a consolidation of best practices. They help ensure predictability and, at the same time, provide a foundation to improve upon. Procedures give an organization a standard language to discuss the way of working. In the automotive industry, there are a number of development methodologies and standards, like ISO/IEC 15504, ISO 26262, CMMI and VDA. And while many of the policies described in the standards are intuitive for persons in the industry, other policies are not. One example is the requirement to write down estimation rationales. If an engineer is new to estimating, it is not always clear why one should bother documenting the justification for the estimate. *Who's going to read it anyway?* is a not infrequent challenge. If an organization is attempting to adopt a new policy or move to a new standard, the lack of policy justification can be crippling.

The fact that so little attention is paid to policy justification is baffling, given the available research on motivation. "All motivating messages, from Apple's marketing to Martin Luther King's "I Have A Dream" speech, do the same thing: they start with *Why.* People are engaged and motivated by why we do things more than what we do." People are more engaged in something they think is important. [1]

And there are other benefits to understanding why:

- It allows an engineer to fill in the gaps and take initiative on decisions or situations where there is ambiguity in the policy. [2]
- Clarifying the justification ensures that incorrect justifications are addressed. For example, developers may think that a policy is required solely because a standard dictates its use. Generally, a policy is important for other reasons and this should be assessed and communicated. Why? the standard could, for some reason, be removed, the associated behavior would then be stopped and the unforeseen benefits with it.
- If the rationale for the policy is understood, it helps ensure that the policy fits the purpose. If the purpose changes or becomes obsolete, the policy can be tuned properly.
- Justification is also important because, unfortunately, there is a common perception that policies are bureaucratic. Perhaps, this perception is in part due to the general lack of policy rationales – no one explains why the policy is necessary. Or perhaps, your organizational policies really are bureaucratic and may need tuning. In any case, determining the rationale for policies is a valuable exercise.
- Apprehending the policy motive is useful in getting the policy implemented, more so than with most types of development work, because the policy's *raison d'être* may be comparatively less obvious. If an engineer implements code, it is generally apparent that the code is necessary for the product; that doesn't need explaining. If an engineer tests code, it is usually clear to the engineer that testing helps identify bugs – crucial for quality! But, if an engineer has to *document* a solution for a bug, then the reason for the effort starts to be less apparent. Policy justification is your friend in these situations.
- The costs versus benefits of a policy can be better assessed by an organization when the policy rationale is understood.

In our experience, we have seen various methods to address the motivation for policies:

- The most common: Ignore the justification for the policy altogether and leave people in the dark.
- "We've always done it like that."

Both are dangerous answers because, in the authors' experience, they discourage use and ownership of policies and consequently reduce continuous process improvement.

Other methods:

- Provide an individual answer to the policy challenge impromptu and without documentation. Not particularly efficient and can lead to varying (and incorrect or unchallenged) justifications.
- Provide a policy training with the relevant justification described. An important method because it places the motivation with the policy that needs to be motivated. However, trainings can be very time-consuming for the trainer and inconvenient for the trainee (wrong time for the trainee.) Reading training slides might be a substitute but it lacks the dynamism that comes with interactive training.
- Provide a managed FAQ wiki with the policies most frequently challenged.

The use of an FAQ wiki has multiple advantages [3] compared to the other methods of communicating policy justification:

- It is interactive over a long period, unlike trainings.
- Unlike informal explanations to individuals, writing (an FAQ) has comparative permanency; the re-use factor is much higher.
- It is more efficient in communicating the rationale to a large and growing organization than individual answers or one-time trainings. When deployed properly, an FAQ wiki can touch more people more easily. (Admittedly though, speaking to a person or group has a compelling nature that a written communication lacks; but the other advantages of a wiki would help compensate for the deficit. We recommend that the wiki be instituted in addition to trainings.)
- A wiki has easy editing, allowing the answer to be improved over time within the development community, promoting community ownership.
- It builds a standardized answer (same understanding across the organization) compared with an impromptu, undocumented response.
- It allows for continuous improvement of the justification through peer review.

An FAQ on challenging policy topics can demonstrate clearly how the policy benefits the engineer and consequently can help speed up acceptance and use of policies within your organization. We illustrate such a policy FAQ below. Please note that the answers are based on experience with larger companies and may not apply to your organization entirely.

The target audience for the FAQ is the engineer and not policy experts; hence the answers may not always be instructional for experienced quality personnel.

Before implementing the FAQ, consider introducing a survey to baseline the current attitudes on policies, including the justification. Questions might include the following, with a rating from one to five:

- Do you have an influence, feel ownership on policies?
- Do you know how to suggest changes to the quality management system?
- Do you understand the rationale for policies?
- Are the policies continuously improved?

The survey might be executed before and sometime after the FAQ introduction to help evaluate its impact and usefulness. Based on results on the above questions from various companies, we conclude that an FAQ is beneficial and very much appreciated by the users.

2 FAQ

The following sections contain an example for a Frequently Asked Questions wiki, containing policy challenges we have often received from the development community. The FAQ should be tailored by

the organization making use of it.

2.1 Why do we have to make tickets for Problem Reports?

Reference: *Verband der Automobilindustrie, Quality Management in the Automotive Industry Process Audit Part 3*, Section P2.4, Is change management in the project ensured by the project organization?

Problems need to be reported from the party finding it to the party that will fix it. Telephone or e-mail might seem sufficient and more efficient for this purpose and thus, the use of a more complex PR database system (with ticketing) is regularly questioned by engineers new to this topic.

If problem reports (PR) are captured in a persistent and widely accessible way (via a database), the related information can be shared in the community and everyone is able to contribute to the solution; email exchanges generally limit the visibility (including long-term) within the team and organization.

Handling of duplicate findings is possible only if all PRs are persistently recorded in a searchable database. If you want to ensure that you or your co-workers are not solving the same problem a second time, recording the PR in a database is indispensable.

A basic activity in a project is to analyze PRs from the past in order, for example, to understand potential risks or plan for a new project: are there old, unresolved bugs we need to fix in this project? Without a PR data base, this becomes a painful exercise.

Such a data base is also able to require specific, formalized input for a new PR to ensure all the proper information required for quick and successful analysis and solution are given. Something, a simple e-Mail can never do.

2.2 Why do I need to write a project management plan?

Reference: *A Guide to the Project Management Body of Knowledge 2000 Edition*, Section 4.1, Project Plan Development

We have heard the project management plan questioned over the years: “Why do we need to write a project management plan? It is so much effort and nobody reads it!” One reason for writing a project management plan is to prompt the appropriate actions (serving a checklist function) from the project lead and consolidate the information in one place for easy reference, both for the project team but also for the future; if a bug is found after the project is closed, it may be necessary to re-construct the project environment and having a project management plan can greatly ease that high-pressure experience.

If no one is reading the project management plan, then perhaps a reading sign-off of the project management plan can be introduced in the team to help promote familiarity with the document. Or maybe a re-evaluation of the content would make sense, to ensure it brings value to the project.

2.3 Why should I prepare for risk sessions?

Reference: *Potential Failure Mode and Effects Analysis Reference Manual Fourth Edition,* page 11, Identify Functions, Requirements, and Specifications

Initial risk sessions in a project are often approached with limited enthusiasm, an exercise to get through. However, identifying risks in a project is vital:

- Customers require detailed risk management.

- Our standards require detailed risk management; standard adherence allows us to continue supplying within the automotive industry.

But, even if we did not have those reasons, we would continue detailed analysis and management of risks to protect ourselves and our customers from the risk of failures. (At this place in an FAQ, a sample of relevant industry failures could be helpful to list.)

2.4 Why do we make estimation rationales?

Reference: *CMMI for Development, Version 1.3*, Project Planning Process Area, Specific Practice 1.4, Determine Estimates of Effort and Cost, Estimate the project effort and cost for the work products and tasks based on estimation rationale.

In order to develop a proper plan for a project, estimating the effort (time, resources etc.) is a clear element of that process. A wide range of possibilities exist to get estimates for individual tasks, building blocks or entire projects: simple expert judgement, 3-point estimates involving the whole team and analysis of historical data are used.

What is forgotten to write down many times is the rationale for the estimate. Why, for example, a certain tasks takes 10 man days and not just 5? What were the ideas of the experts judging the effort? Which historical data showed that it will take 10 days?

Recording the answers to those questions is important for several reasons: First, the rationale increases confidence and creates a common understanding about the estimates with the people involved in the project. But also people not directly involved can benefit: management may challenge those estimates to ensure a project is a proper investment. Recorded rationales can help to defend the estimates made by a project team.

Properly recorded estimation rationales are especially beneficial if the estimation data are reused in another project. They explain why a certain estimation has been made. An assessment can then be made, based on experience, on whether the explanation was correct and the new estimate can be improved. Also, the team members don't need to go through the effort of re-thinking the rationale – there it is already, with the original task and estimate.

2.5 Why do we have to write down a configuration item list?

Reference: *CMMI for Development, Version 1.3*, Configuration Management Process Area, Specific Practice 1.1, Identify Configuration Items

Configuration items (CI) are the fundamental structural unit of a configuration management system [2]. They are essentially the list of project deliverables and can serve to clearly divide the work in a project. For design-driven engineering communities, CIs typically are:

- Design Data
- Documentation and reports
- Development and infrastructure tools

Identifying these items within a project and writing them down in a configuration item list can be helpful for every member of the project. The key to maximum benefit is not just to name each configuration item but in addition, to add proper attributes to them.

Design data and documents should not just contain the name but, in addition, the location of the files (including the version control system) and the owner of the configuration item, which allows other engineers to quickly find the person in case there are problems or questions for the CI.

For each of the documents and design data CI, a short name is to be specified in the CI list, identifying

the CI across the whole project documentation – for example, Software Architecture Description, SAD; this makes project communication more efficient, especially for new-comers.

Each tool required to create the project's deliveries becomes an element in the CI list. For software engineering projects, for example, the software tool CI contains not just the name but also the version used and the download URL. With such a detailed CI list of tools, it can be ensured that every engineer in a project uses the exact same specific set of tools. Engineers joining the projects can create the environment quickly since the exact tools are well defined. Project managers get a list of the software licenses required to provide the team with the proper tools.

Besides the points above, there is another critical reason for CI lists – imagine if you did not have a list of CI items for a particular project. The project closes, and 10 years later, a severe bug is found on the corresponding product and the customer is very unhappy. The Failure Analysis team urgently needs to analyze the bug, find the root cause and put a solution in place. They discover that the version control system can be restored for the product and, oh joy! they find a reasonable-looking baseline tag and are able to populate project files. But, the file that is needed – an architectural description - is missing! Someone forgot to check it in and there was no control to ensure its inclusion in the tag. That is where a CI list comes to the rescue. It helps ensure that all the project deliverables are in the version control system and there when you need them.

2.6 Why do we need formal decision-making and documentation?

Reference: *CMMI for Development, Version 1.3,* Decision Analysis and Resolution Process Area, Specific Goal 1: Evaluate Alternatives.

Having a documented explanation on important decisions and a required set of stakeholders is vital for several reasons:

- In case the decision needs to be re-visited in the future, a record of why the initial decision was made is very helpful. Without it, the team may not remember the reasons and neglect important aspects of the new decision.
- A critical decision generally requires complex analysis by multiple persons. Recording the details of the decision-making process protects that investment (especially if persons are no longer in the company); it can serve as idea re-use when similar decisions are required in the future.
- A required set of stakeholders and a formal structure for the decision (including criteria and decision method) reduces the subjective nature of a decision.

2.7 Why do we need control on hardware/software interfaces?

Reference: *Automotive SPICE Process Assessment / Reference Model, Version 3.0*, SUP.10.BP5, Approve change requests before implementation; and internal project policy: The project freeze date for signal names is the silver milestone.

Silos often exist between hardware and software groups; hardware might freely make changes to interface elements (such as a register or bit) and have very little understanding of the impact of the change on the software code, using the same interface element. Because of this common software/hardware disconnect, it is crucial to have dedicated control and review on the interfaces in order to avoid uncontrolled consequences on the software schedule and stability.

2.8 V-Model

Reference: *Automotive SPICE Process Assessment / Reference Model, Version 3.0,* Annex D.4, Traceability and Consistency

2.8.1 Why don't we have vertical links on the test side?

Within the Automotive SPICE depiction of the V-model, there are no links on the right side, between the tests levels. However, we have seen right-side linkage introduced and explained to indicate re-use of test cases; this is not the verification goal, according to the V-model. The aim instead is to create tests for each specific V-model level to ensure that level-specific goals are addressed. Exclusively re-using tests could mean maintaining any errors/holes in requirements interpretation. That is, if your upper-level requirement is improperly broken down into lower-level requirements, re-running the set of lower-level tests for the upper-level requirement won't highlight that breakdown problem; you would need a different test for the upper-level requirement.

2.8.2 Why do we need requirements breakdown?

Breaking down a requirement into lower-level requirements helps organize the work into manageable tasks for the team. It can foster an independent structure in the work breakdown structure that reduces unnecessary interaction with other team members and gives clear ownership.

With the linkage from upper to lower level requirements clearly established, the impact of any proposed change can be more easily determined.

A breakdown of all requirements helps ensure that a component is not forgotten. In the rush of a release, a component, which is not linked to an upper requirement, may not be properly instantiated. If there is no link to an upper requirement, there also may be no linked test case. If no specific tests are executed for the component, it could be overseen entirely at the time of the release.

2.8.3 Can upper-level requirements be satisfied with only lower-level tests?

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. [4]

(This is true in the other direction; top-level tests don't sufficiently cover lower-level requirements. Top-level tests might not finely test every input combination for the bottom level units – a bug at the low level may not be triggered.)

3 Results

Since we began to communicate policy rationale more actively, including through a wiki, we have seen the engineering community take a more active role in driving changes, becoming a self-learning organization. (No doubt that the self-learning is attributable to multiple factors but users have clearly said a justification is helpful.)

One example to help illustrate the point: previously, we had problems with schedule slippage due to

misalignment on the hardware/software interface elements - in this case, the registers, including its name and functionality. There was a policy in place that froze register names at a particular development milestone and an unwritten rule that the changes would be limited. This policy worked well for a while until the software started to become very complex and new persons entered the team (unaware of the unwritten rule.) The hardware team would make many changes to register names and the software team would struggle to update their code in time for release. Both sides thought they understood the situation. The hardware team thought the changes were warranted and that the software team was exaggerating the impact. The software team thought the hardware team was making frivolous changes, careless of the software impact.

When we clearly discussed the justification for the freeze date of the register names – the far-ranging and complex impact of a register change on software but also the hardware need for changes to register names - we could appropriately update the policy: we introduced earlier reviews of potential register changes. In the reviews, the hardware team explained the need for changes and the software team discussed the impact; the teams then negotiated which changes would be accepted (also involving marketing and management for tougher decisions.) The updated policy met the goals of both sides: the hardware designers could make necessary (and agreed) changes to register names and the software team could keep their schedule under better control.

A crucial element of this story: the hardware and software teams closely cooperated to effect the policy changes, made easier by grappling with the policy rationale.

4 Deploying and Maintaining

Properly deploying and maintaining an FAQ is just as important as writing an FAQ in the first place. Publishing the FAQ wiki in the company intranet is essential. From our experience, those intranet systems are not always equipped with a proper search and find functionality. Thus, make sure the FAQ can be easily found with various search terms.

Advertising the FAQ at the organization's internal home page with a "New" icon – at least for some initial period - helps to publicize the page. Communicate the FAQ in kick-off meetings, new employee trainings, workshops on related topics and how-to emails to the entire organization.

Once the FAQ is successfully deployed, it's important to keep it up-to-date, to maintain and improve its content. To help with any necessary clean-up and monitoring, nominate an owner for the page and ensure the owner's name is clearly visible on the wiki page.

5 Conclusion

Understanding the rationale for a procedure is a requisite aspect for its acceptance by many engineers. It is also vital for the organization to understand the benefits versus costs of policies to maintain its efficiency and effectiveness. In this paper, we have described the advantages of clarifying the policy reasons to the development organization. We provide an example of an FAQ wiki with policy questions and answers and describe the benefits of capturing the FAQ in a wiki format versus other means of transmission. Results of this approach are encouraging and briefly described; we plan to promote the wiki to a wider organization. The authors envision that the approach would work well in other industries and welcome any feedback on its application in general.

3 Literature

[1] http://time.com/53748/how-to-motivate-people-4-steps-backed-by-science/

[2] https://soapboxhq.com/blog/help-employees-take-ownership-work

[3] http://wikieducator.org/Wikieducator_tutorial/What_is_a_wiki/Advantages_and_disadvantages

[4] *Effective Requirements Traceability,* Industrial Proceedings, 20th EuroSPI² Conference, June 27th, 2013, Paul Schwann, Joanne Schell [4]

4 Author CVs

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 Secure Car Access.

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.

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 Secure Car Access.

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.

A GSN Approach to SEooC for an Automotive Hall Sensor

Xabier Larrucea[1], Silvana Mergen[2], Alastair Walker[3]

[1]TECNALIA, Bizkaia, Spain. xabier.larrucea@tecnalia.com
[2]TDK-EPC AG & Co. KG, Stahnsdorf, Germany. silvana.mergen@epcos.com
[3]LORIT CONSULTANCY, Edinburgh, Scotland. alastair.walker@lorit-consultancy.com

Abstract

One of the key challenges for manufacturers of automotive systems, hardware components and software products is not only the process of defining explicit and implicit requirements but also the ability to satisfy safety requirements such as those specified in ISO 26262. From an element point of view, the Safety Element out of Context (SEooC) defined in ISO26262 is becoming a reference for developing systems, elements and components in the automotive sector. Integration teams have limited prior knowledge of how these third party devices have been defined, the assumed requirements used during the validation and verification phases. Goal Structuring Notation (GSN) can be used to define and document the assumed SEooC requirements in a graphical manner. However, development teams are facing several challenges for example how different requirements are implemented in SEooC, or how far GSN is able to represent SEooC definitions. This paper provides a GSN based approach to represent SEooC requirements in a practical example of an automotive hall sensor.

Keywords

Argument, assurance case, claim, Safety Element out of Context

Published in: Springer Communications in Computer and Information Science (CCIS) vol. 663

An advanced testing approach to validate software changes in complex hardware environments

Domenik Melcher, Graz University of Technology, Austria, d.melcher@student.tugraz.at
Thomas Puchleitner, NXP Semiconductors, Austria, thomas.puchleitner@nxp.com

Abstract

Projects in the Semiconductor branch consist of hardware devices and also software parts, that communicates with them. The software that communicates directly with the hardware is called firmware and gets written for every hardware device separately. In addition to that, a middleware gets written once, which can deal with several hardware devices. Changing the middleware for a new release, will raise the problem of ensuring that the software still works for all available hardware devices, without losing the quality level. Because of the high quality requirements and the close connection between hardware and software in this environment, testing the software framework gets a complex task. To avoid the execution of the whole test framework, which contains often several thousand tests, on several hardware chips, this paper proposes an efficient solution to face this problem. For that, the problem will be first split into three sub-problems, to reduce the complexity and also to focus on each one individually. After that, the first two approaches will describe how to select the set of test cases, which cover the modified parts of the software and ensures that all side effects will be tested, at the same time. In the end, the third approach will show how this data can be used to automate the testing process. All three approaches combined will form the solution that ensures the functionality of a software change upon several hardware components.

Keywords

Test management, test case selection, Test plan generation

1 Introduction

Semiconductor components are nowadays wide spread on the market. Often found in mobile phones, they are represented as stand-alone hardware chips that provide an interface for other components in an embedded system. These hardware components are usually developed for a specific use case, such as establishing a connection, using RFID technology. Therefore, the chip and also the software that communicates with it, a so called firmware and middleware, are mostly developed by the same company, to ensure a save connection and also to save costs. The solutions, provided by companies in the Semiconductor branch often implement several OSI layers, to fulfill customer needs. Having that, a customer has the possibility to choose which layers should be used and which should be customized, to create a flexible end solution. However, all of the supported layers need to be tested, to ensure a certain level of quality, independent to if it will be used in the end or not. Therefore it is obvi-

ous that the testing process in the development process of hardware/software system engineering has high priority, to fulfill the quality criterions.

As hardware and software components are often engineered by different teams within a company, designing the chip and developing the dependent software are done in parallel. This makes the testing process more difficult, as the final hardware is not available most of the time for real tests and needs to be simulated in that case. Simulating the hardware device makes it hard to deal with hardware specific behavior like timeout, which can affect the test result and must be taken under consideration in the implementation of the software as well. Also, the hardware components that shall be used in the end are in terms of computational performance lower dimensioned than classic personal computers, as they only fulfill one specific purpose. Simulating the hardware device in a powerful working environment, makes it therefore possible to execute function tests on the software. However, there are also tests, such as stress tests, that need to be executed directly on the hardware chip as they test the system under real stress conditions for several hours.

A company in the Semiconductor environment, has several such hardware chips in place, to fulfill similar use cases with slightly different requirements. As the firmware needs to work for one specific hardware chip, the common middleware that uses this firmware needs to execute safely for all available hardware components that are provided by the company. For that, the code base gets extended every time, a new hardware component gets added, to support the new features. This means, the more hardware components a company offers, the more versions of the firmware will be implemented, that need to be supported by the middleware. Thus ensuring that this software works on several hardware components without losing the quality standard gets a complex task, as it needs to be cross-checked with every available firmware and therefore tested on all hardware devices. Because of that, this paper will focus on this problem and presents an efficient approach to ensure that the modified software version for a new release will always work safely for all provided hardware devices by only testing the changed code parts.

2 Current situation and problem definition

Companies in the Semiconductor branch are confronted with the problem of having a big test framework, to ensure high quality standards, which needs to be executed for every new hardware chip that gets released. Releasing a new hardware component means to develop or extend the firmware and update the middleware, to support the new functionalities. However, the changed software needs to work for all produced hardware devices that are currently on the market. Therefore the whole test framework gets executed for every available hardware chip, to check if the performed code changes for the new hardware component did not cause any side effects. As executing always several thousand test cases for several hardware chips before a new release takes a lot of time, an automated approach is needed to select only those tests that are directly associated with the new code changes. How to ensure that the software works for all available hardware components, by only testing the code parts that got changed for a new release?

To analyze this question, the method of Systems Engineering gets chosen. This method follows in general the approach of splitting a big problem into several sub-problems. [1] These problems will be afterwards analyzed and solved separately. In the end, theses separate solutions will be combined to form the solution for the main problem. For the question stated above, three sub problems got identified which will be discussed in this chapter. As the final solution needs to be integrated as easy as possible into the current developing process, the software components which are currently in use will be analyzed to determine how they can be combined. The idea is to select a set of test cases that cover the modified changes of the middleware that needs to work for all available hardware devices. For that, the first paragraph will therefore analyze the potential of the used version control system, followed by the current test case setup. As third sub-problem, the test plan management will be analyzed.

2.1 Managing the code base - Version Control Systems

As soon as the developing process of a software package involves more than one person, a centralized place where a team can store the written code occurs as beneficial to make it accessible easily for every team member. Software solutions, such as Apache Subversion [2], in short SVN, or Git [3], are used to store software in so called code repositories. These repositories provide a backup mechanism by storing the source files on a centralized server which can be accessed easily directly via command line or tools, such as Tortoise SVN [4], or SourceTree [5]. In addition, a version control system holds information about the files contained in the repository associated with user accounts. Beside security aspects, this makes it possible to determine who changed which file and which lines of the file were affected. Every change made is stored in a so called revision which is included in a visual graph that shows the changes in a timeline. This functionality is very beneficial, as every revision displays which lines of code in a file got changed to the previous state. With that, a developer can determine quickly which function got affected in case of maintaining. This revision graph does not only document the changes, within a software project, it is also possible to go back to a specific revision, if needed. To determine which revision was used for the released version, it is general practice to highlight this change in the graph with a Tag, which holds a customized comment. But how to identify all functions that have changed between two specific releases, using a Version Control system? That needs to be clarified to use that given features for filtering the code by parts that are of interest.

2.2 Test cases and call-graphs

One way to validate the quality of a program is to test it via test cases. Having a given function under test, a test case, represented in most cases by a program routine that calls the function with set input values, determines via direct feedback if it meets the requirements, defined in the test, or not. Tools, such as the NUnit framework [6] provide automated functionality to run test cases either via a plugin for an integrated development environment (IDE) or as stand-alone solution via an external library. Depending on the size of the software project, there can be several test case clusters that might be assembled in a test framework. The bigger the system under test gets, the more test cases are in this framework that might be double or test equal code parts then other tests. Therefore it might get complicated when it comes to the selection of a specific test case, which covers code parts of interest for a new release.

The testing framework, used by a company in the Semiconductors branch contains approximately several thousand tests that get fully executed for every hardware component, which is currently available on the market. Using the V-model for describing the validation process of a software program as displayed in figure 1, shows that testing bottom up will call first independent module tests for each component, followed by function tests that will call parts of these module tests again, to test the whole execution path in a specific context, down to the hardware abstraction layer (HAL), which communicates directly with the underlying hardware component. Having multiple chips available, all of these test cases get executed several times for a code change that might only affect a few functions. Following this approach will raise the execution time steadily with every new hardware component that gets released. Therefore, a solution is needed to reduce the amount of executed test cases. The idea is to execute only these test cases that call the functions which got changed for the new release. How to generate a call-graph for a specific function? Answering that, makes it possible to associate the changed functions with the test cases that needs to be called, to validate the functionality. Having such a call-graph available makes it also possible to determine possible execution paths that might cause side effects. Figure 2 shows the general idea of such a call-graph, which displays the calling path of a test case to an added function for a new release with all dependencies.

2.3 Test execution

To organize a list of test cases, a test plan can be implemented as structured way of clustering a set of test cases for a specific purpose. A test plan is in general a logical entity that gives information about

which test cases will be executed in a test run, to fulfill certain quality criterion. Therefore it can be represented either written on paper or depicted as software solution. Defining which test cases should be tested for a given quality criteria keeps the overview of the test procedure, defines clearly if a software component meets the requirements and provides also the possibility to recycle the test plan for similar future purposes. Test management tools, such as TestLink [8], provide the functionality to automate the execution of different test plans. The system can associate test cases directly to test plans. However, TestLink only stores logical entities that describe the test framework and organizes it. To execute the test case entity, which is stored in a test plan object of TestLink, with a programmed test routine, a separate system must be used such as Jenkins that combines these two aspects. This means the information, stored in test plan and test case objects of TestLink is by default independent to the source code and needs to be maintained separately. Therefore the question is, how to automatically generate a test plan based on a given set of test cases in TestLink?

Taken the Systems Engineering approach into account, all these three problem fields shall be analyzed and solved separately. Outcomes of these findings shall then be combined into a generic approach for reducing test efforts based on an on need basis. Main goal is to keep the test and therefore system quality at highest level with reducing test time.

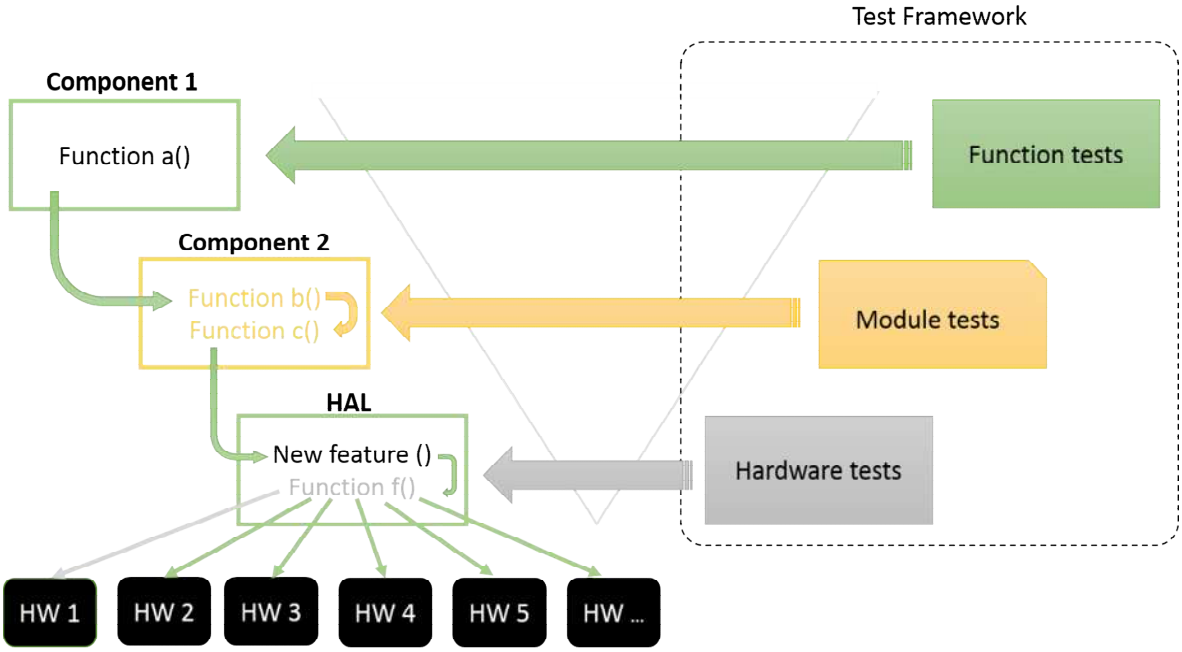


Figure 1: Validation process with several hardware components

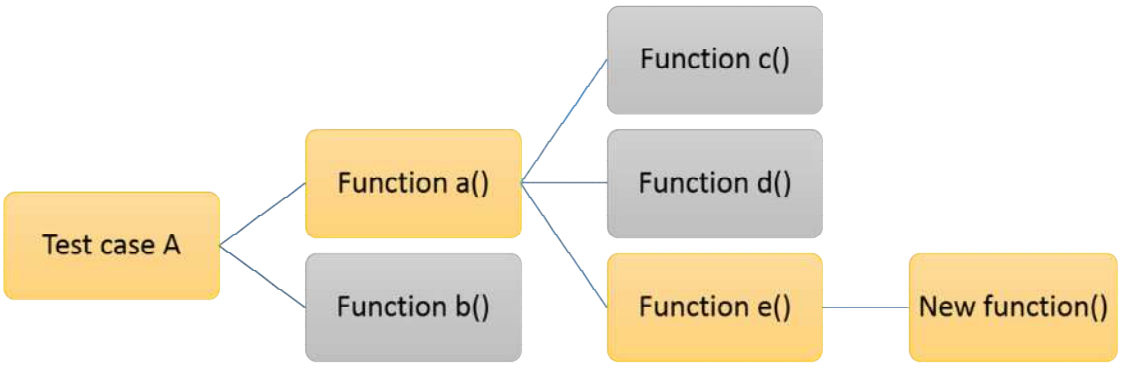


Figure 2: Call-graph with highlighted path to the depending test case

3 Theoretical background and problem analysis

This part of the paper will focus on possible approaches that are state of the art, to solve each of the questions, defined in the previous chapter. According to the approach of Systems Engineering [1], it will again discuss each question separately. The first section will describe possible solutions for determining the functions that got changed since the last release, followed by approaches to create call-graph that lead to their dependent test cases. The last section will describe possibilities of how to maintain the test plan objects, stored in TestLink.

3.1 Analyzing changes between Revisions

Determining which code parts changed between two different releases follows in general a straight forward approach. The easiest solution for that would be to simply compare two different working states of a file and detect which parts differ. However, there are two types of version control systems available that provide a diff mechanism that allows comparisons between files, dynamically out of the revision history. The first version control technique uses a centralized repository and is set up as server-client solution. This means, that a user always works on a local copy of the repository content that is located on an external server. For associating the local working copy with revision content, such as committing or retrieving changes, the repository server needs to be accessed via a client program. A free software solution that works on that basis is Apache Subversion. The diff command that comes out of the box with it compares by default the current working copy of a file with the latest revision in the repository. By using command parameters, it is possible to define that the working copy should be compared to a specific revision or to show the differences of the file between two revisions [9].

The second version control system, uses a distributed version control technique, which brings the possibility of having a repository either locally, on a remote server, or both. The main difference to centralized repositories is that a user does not have to communicate to a remote server to be able to access repository functionality such as accessing the revision history, as the local copy represents itself a fully repository. However, it is still possible to sync the local changes to a remote server to share the code and have a backup available. A software solution that implements this kind of technology is for example Git, which is available for free. Calling the diff command of Git without any parameters will compare the working copy with the version that is currently in the index. The index contains all files of the local repository that are marked for an upcoming commit. It is therefore a layer between the working copy and the repository itself. In comparison to SVN, the diff command of Git comes with a big list of available options. Git also provides the possibility to include an overview of changes, or a so called patch, for each revision when showing all available commits with the git log command. To enable that functionality, the parameter –p, or –patch must be added to the log command. [10] The output, these commands produce is standardized in the so called unified diff format. This standard is well documented and ensures therefore, that everybody that uses the tool can understand the printed result easily and may use it for post processing. The command can be either launched via the included command line extension or from a tool that provides graphical user interface, such as TortoiseSVN [4] or SoureTree [5].

There are also articles available that describe how one can extract information out of the log from a concurrent version system (CVS), to determine changes by a commit. As a CVS recognizes changes for each file separately, these commits need to get associated to form a context, similar to a revision. Former approaches used that data to highlight which functions were changed in a software. [12,13] Based on that preprocessing outcome, an IDE plugin like ROSE can be designed that is able to tell connections within changed code parts, to raise the awareness of a developer about which functions got changed together in the past, by keeping track with the information from the CVS. [14] As the technique of a CVS got obsolete, subversion replaced it as successor by bringing new features, such as revisions, out of the box. [15] Nevertheless, these approaches describe general strategies, that can be used mostly system independent.

3.2 Analyzing the program structure

In general, creating a call-graph follows the approach of analyzing source files that are of interest, to establish connections between the caller and the callee of the contained functions. This process can be either done statically, without compiling and running the corresponding program or dynamically by profiling the program during runtime. However, static code analyzes works most efficiently for program languages without generic datatypes, as the concrete type of a generic type gets defined during runtime which makes it hard to determine statically. Also determining the program flow, as well as creating code coverage during test case execution can be only achieved dynamically, as different function parameters, create different call-graphs. If one wants to create a call-graph using static code analyzes, tools such as GNU cflow, egypt and CodeViz are available freely for the programming language C. The programs CodeViz and egypt also provides C++ support as well as Doxygen. [16, 17, 18, 19] All of the mentioned programs create a parseable output file, which can be used for further analyzes or as input for the program Graphviz to create a graphical representation of the call-graph.

Another possibility to collect the needed data to create a call-graph of a program is to profile it during execution. To achieve this, two general approaches are available. The first approach profiles the needed data by injecting specific functionality into the binary during the compiling process. According to the functionality of the tool, this injected code parts will produce profile output during the execution of the binary, which can be used for post-processing steps to extract information about the program behavior. A tool that can be used for this approach is for example GNU gprof. As part of the GNU Binary Utilities, which contains as well the GNU Compiler Collection (gcc), gprof comes together with the compiler, one needs to compile a normal C/C++ program. However, to profile a program using gprof, the compiler flag –pg needs to be set during the compilation of the source files. Using the output file of the executable, as input for gprof to perform post-processing steps, will produce a call-graph in text format. [20] The second approach can be invoked directly at the program startup as program parameter, to create ready-to-use profiling output. To ensure, that the executable produces profiling data, the program gets dynamically recompiled during the execution, to inject the profiling algorithm. This brings the benefit, of being able to profile every available program that got written in C/C++ without necessarily having the source code to compile it. However, recompiling the program dynamically takes a lot of time, compared to normal execution. A solution which implements that technique is the program Callgrind, which comes together with the Valgrind software suite. For invoking the tool, the command line parameter –tool=callgrind needs to be included when starting a program, followed by parameters that are documented on the official Callgrind homepage. Doing so will analyze the program during execution and produce at the end a call-graph in readable text format. [21] As Callgrind is only available on Linux, there is a Windows alternative called Very Sleepy that provides also a graphical user interface. [22]

The topic of generating call-graphs is very well researched and consist of documents that describe strategies starting with a general approach of how to create a call graph to complete frameworks that provide automated enterprise solutions. [23, 24, 25] In addition to that, there are also empiric studies available of static analyzing tools, as well as comparisons of different kind of call graphs. An aspect that is often included in these former approaches are the different levels of precision during the creation of a call-graph and how to use interprocedural analyzes to improve the quality. The more a call-graph focuses on the context, the more precise it gets, but also the more expensive. A context defines all possible varieties, the input values of a function can get as well as all possible types a generic variable can be. Therefore, a call-graph that displays different paths for every possible scenario according to the context, is known as context-sensitive. As an input value of a function or concrete type of a generic variable might depend on actions a previous function performed, creating such graphs using statically analyzes would take a lot of time, consume memory and cannot guarantee a precise context. Because of that, a profiling approach is used to store the context of a program during its execution. Displaying the plain calling path of a function, can be performed easily via statically analyzes and is known as context-insensitive. [26] To be able to deal with polymorphism in object oriented languages, there are also different approaches available that describe strategies of how to resolve code that include virtual functions and function pointers in a context-sensitive graphs. [27, 28]

3.3 Analyzing possibilities to customize TestLink

Depending on the provided possibilities, customizing a stand-alone system, or its parts, can be done in general either by using built-in maintaining options of the program or externally by using an Application Programming Interface (API). As TestLink is a Test Management tool, that organizes logical entities, such as test plans with an underlying set of test cases in a MySQL database, both possibilities are available. However, synchronizing the data of TestLink with a big Test Suite manually via the provided web interface takes time and will get outdated as soon as the set of test cases will change. Therefore the API, will be used to update the database according to the test cases that cover the changed code parts of a new release. Originally provided in PHP, the API communicates with TestLink using the XML-RPC standard [29] and is well documented by the community members. However, adaptations that extend the support for different programming languages, such as C#, Python, Java, and more are also available for free. These APIs are all open source and written by different people, who provide the functionality for the active TestLink community. [30, 31, 32]

Former approaches describe ways to customize TestLink to achieve automated testing environments, similar to the approach presented in this paper. For that, TestLink is mostly used as one part of a bigger test process, beside the system under test. In addition, there can be also external components that trigger the execution of the provided test plan with specific input parameters, or to validate the outcome of the test run, to serve customized information of the result. By default, TestLink only provides logical entities that describe a dependent test case which should be executed. However, to link a test plan with source files, which contains the implementation of a test routine that can be executed automatically, an external tool is usually needed. To include this mechanism, a solution is available that adds this part directly into the implementation of TestLink. The solution describes a completely out-of-the-box environment where a user can link the dependent source files, located on the server, when creating a test case entity and execute it using a customized dashboard. [33]

4 Definition of an advanced testing approach

The former chapter elaborated which technologies are available for each problem individually. Having that, this chapter will define what gets chosen to form the approach that gets implemented. The first paragraph will describe the technique that will be used to determine the functions that got changed in between of two specific releases. Having these functions, the next part will show the approach of how to generate a call-graph out of them and in the end, a solution will be given how to combine these two datasets, using TestLink to ensure automate test case execution. Figure 3 shows an overview, how the separate components shall work together.

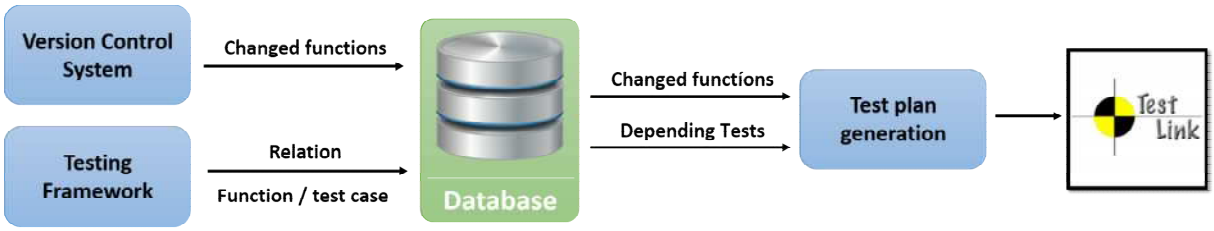


Figure 3: Component Architecture Overview

4.1 Determining changed functions

To determine the functions that changed between two specific releases, the version Control system Apache Subversion will be chosen. To extract the list of functions, two separate steps will be performed. First, the diff command will be used, to generate a document in the unified diff format that holds all changes to the last release. For that, the latest commit will be compared with the Tag of the

last release. After that, an external program will be used to parse this document and extract the function names out of it. These extracted function names will be afterwards forwarded to a database that holds a relation between function and depending test cases. To ensure, that the data is always up to date, a SVN hook will be used, that triggers at every commit a defined script, that calls the actions described. Figure 4 shows the general idea of the approach and describes the components.

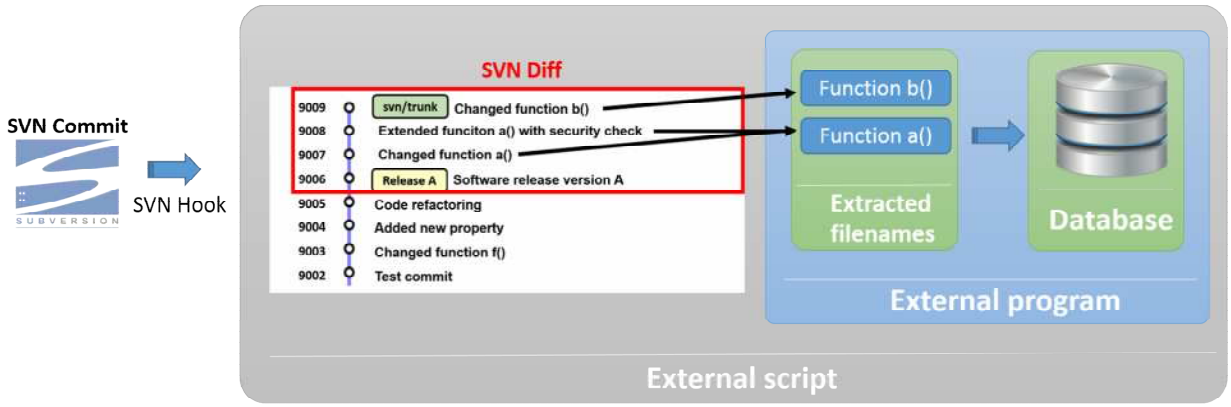


Figure 4: The components that extract the functions that changed since last release

4.2 Creating relations between Test Case and called Functions

For the generation of the call-graph, the profiling tool GNU gprof will be used. It can be invoked into nightly builds, for automating test execution using Nunit. As these test runs always perform a full test of the whole test framework, the created call-graph displays all possible functions that can be called by a test case. This data will be analyzed with an external program afterwards to associate the test cases to the called functions. As a result, this relations will be also pushed to the database. Figure 5 visualizes the general idea behind the approach. Together with the data about the functions that got changed since the last release, it is now possible to associate the functions of interest to test cases that cover them. With that an efficient solution for test case selection is done, as in the end only test cases will be executed that code interesting code parts. To keep the data up to date, this mechanism will be executed automatically as post-process after every nightly test run.

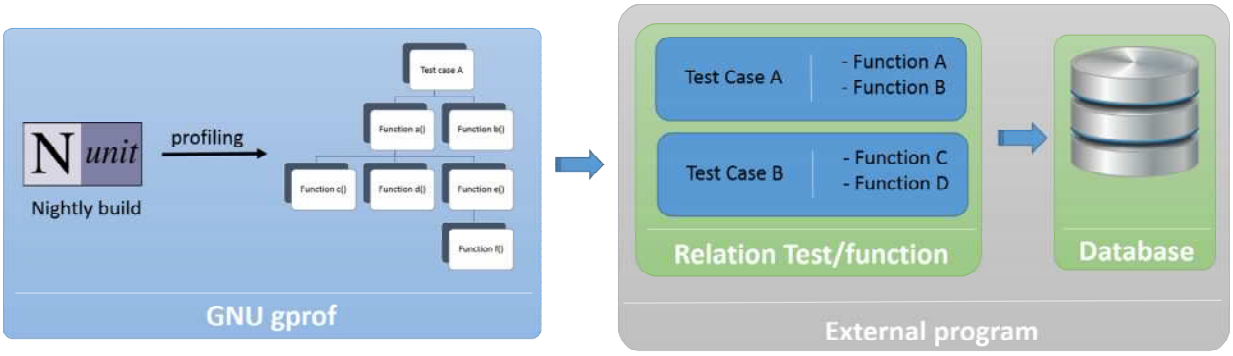


Figure 5: The external program that processes the test plans for TestLink

4.3 Automated Test plan generation for TestLink

Having the functions that changed since the last release and the related test cases, the TetsLink API will be used to create test plans for TestLink out of this data. This test plan will represent therefore one release and will hold a list of test cases that test the changed functions. Therefore an external program will used that communicates with TestLink and the database that holds the input data for the test plans. As the database holds relations of all possible functions that can be called by the test framework, a preprocessing step needs to be performed beforehand. For that a SQL query gets executed, that creates an intersection between the dataset with the changed functions and the dataset with the test case relations, to filter the test cases which cover the changed functions. Having this data prepared, the TestLink API can be called to insert a new test plan that holds the list of test cases, created with the data given. Figure 6 shows the general approach and describes how to components work together.

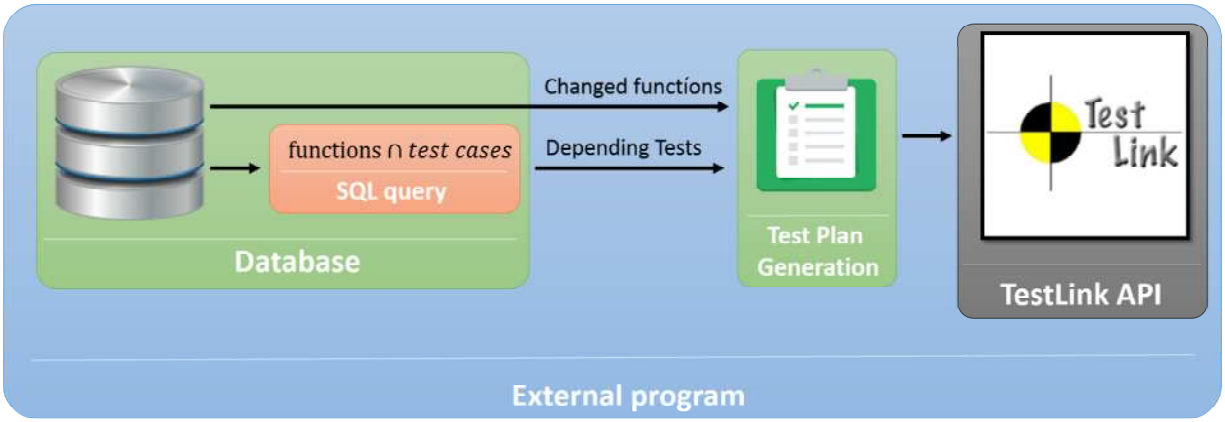


Figure 6: The program that creates the test plans for TestLink

5 Conclusion

This paper focuses on an efficient solution which ensures that a modified middleware will still work for all available hardware components. For this, the method of Systems Engineering was used to split the complex main problem into three sub-problems that got solved separately. [1] Out of a test framework, the first two approaches collect the data which is needed to select the set of test cases that validate the changed functionality and its dependencies of the middleware. For that, the first approach extracts the function names out of a version control system that got changed since the last release. To find the test cases, which test these functions, the second approach creates call-graphs of every executed test case and saves the relation between the test case and the functions that got called. To conclude the test case selection, an intersection will be made between the changed functions and the test case relations, to create an association between these two datasets. This preprocessed data will be used in the third approach to create and maintain the test plans in the test management tool TestLink for automating the execution of these test cases. However, the limitations for this approach is the usage of a version control system as well as having a testing framework that contains executable test cases for the functions of the software. The higher, the coverage of these test cases, the more precise the test case selection will be. The next step is to create a testable solution of this approach, as at this point, this paper just describes the general idea about the approach.

6 Literature

[1] Haberfellner, R. and Stelzmann E. 2008. Systems Engineering: neu überdacht. WING-Business (2008) 3/08: 18 – 25.

[2] Apache™ Subversion. 2011. The Apache Software Foundation. Retrieved April 7, 2016, from https://subversion.apache.org/#site-overview

[3] Git. 2016. Software Freedom Conservancy. Retrieved April 7, 2016, from https://git-scm.com/
[4] TortoiseSVN. 2016. The TortoiseSVN team. Retrieved April 7, 2016, from https://tortoisesvn.net/
[5] Free Mercurial and Git Client for Windows and Mac | Source Tree. 2016. Attlassian. Retrieved April 7, 2016, from https://www.sourcetreeapp.com/
[6] NUnit. 2015. NUnit.org. Retrieved April 7, 2016, from http://www.nunit.org/
[8] TestLink. 2014. TestLink Development Team. Retrieved April 7, 2016, from http://testlink.org/
[9] Pilato, C. M., Collins-Sussman, B., Fitzpatrick, B. W. 2008. Version Control with Subversion. O'Reilly Media. ISBN-10: 0596510330
[10] Loeliger, J. and McCullough, M. 2012. Version Control With Git. O'Reilly Media. ISBN-10:1449316387
[12] Ball, T., Jung-min, K., Porter, A. A. and Siy, H. P. 1997. If Your Version Control System Could Talk...
[13] Zimmermann, T. and Weissßgerber, P. 2004. Preprocessing CVS Data for Fine-Grained Analysis. 1st International Workshop on Mining Software Repositories (MSR)
[14] Zimmermann, T., Zeller, A., Weissgerber P. and Diehl S. 2005. Mining version histories to guide software changes. IEEE Transactions on Software Engineering 31(6): 429 - 445
[15] CVS - Concurrent Versions System. 2006, 1998. Price, D. R., Ximbiot (2006) and Free Software Foundation (1998). Retrieved April 7, 2016, from http://www.nongnu.org/cvs/
[16] GNU cflow. 2011. Free Software Foundation. Retrieved April 7, 2016, from http://www.gnu.org/software/cflow/
[17] egpyt. Date not defined. gson. Retrieved April 7, 2016, http://www.gson.org/egypt/
[18] CodeViz: A CallGraph Visualiser - Skynet. Not defined. Skynet. Retrieved April 7, 2016, from http://www.csn.ul.ie/~mel/projects/
[19] Doxygen. 2015. Doxygen. Retrieved April 7, 2016, from http://www.stack.nl/~dimitri/doxygen/index.html
[20] GNU gprof. 2016. Free Software Foundation. Retrieved April 7, 2016, from https://sourceware.org/binutils/docs/gprof/index.html
[21] Callgrind: a call-graph generating cache and branch prediction profiler. 2015. Valgrind™ Developers. Retrieved April 7, 2016, from http://valgrind.org/docs/manual/cl-manual.html
[22] Very Sleepy. 2016. Richard Mitton. Retrieved April 7, 2016, from http://www.codersnotes.com/sleepy/
[23] Ryder, B. G. 1979. Constructing the Call Graph of a Program. IEEE Transactions on Software Engineering SE-5 (3): 216 – 226
[24] Graham, L. S., Kessler, P. B. and Mckusick, M. K. 1982. Gprof: A call graph execution profiler. ACM SIGPLAN Notices - Proceedings of the 1982 SIGPLAN symposium on Compiler construction 17(6): 120-126
[25] Ondvrej Lhoták. 2007. Comparing call graphs. PASTE '07 Proceedings of the 7th ACM SIGPLAN-SIGSOFT workshop on Program analysis for software tools and engineering: 37-42
[26] Grove, D. and Chambers, C. 2001. A framework for call graph construction algorithms. ACM Transactions on Programming Languages and Systems (TOPLAS) 23(6): 685-746
[27] Milanova, A., Rountev, A. and Ryder, B. G. 2002. Precise call graph construction in the presence of function pointers. Source Code Analysis and Manipulation, 2002. Proceedings. Second IEEE International Workshop on Source Code Analysis and Manipulation: 155 - 162
[28] Bairagi, D., Kumar, S., Agrawal, D. P. 1997. Precise call graph construction for OO programs in the presence of virtual functions. Parallel Processing, 1997., Proceedings of the 1997 International Conference on Parallel Processing: 412 - 416
[29] XML-RPC.Com. 2011. Scripting News, Inc. Retrieved April 7, 2016, from ttp://xmlrpc.scripting.com/
[30] The C# API to talk to Testlink via XML/RPC. 2016. Jfrog Bintray. Retrieved April 7, 2016, from https://bintray.com/smeyn/Gallio-Testlink-Adapter/Testlink-API/view
[31] TestLink-API-Python-client 0.4.7. 2015. Python Software Foundation. Retrieved April 7, 2016, from https://pypi.python.org/pypi/TestLink-API-Python-client/0.4.7
[32] TestLink Java API. 2016. GitHub. Retrieved April 7, 2016, from https://github.com/kinow/testlink-java-api
[33] Menezes, F. L. 2015. A Regression Testing prioritization Component for TestLink.

7 Author CVs

Melcher Domenik, studying Software Development and business Management at Graz University of Technology, writes currently his master thesis at the company NXP Semiconductors. Besides studies he is working at Andritz AG, to use the knowledge he gains at university in global software projects. In addition, he is active in the European wide student association, called Board of European Students of Technology (BEST), where he is currently the head of the local group in Graz.

Thomas Puchleitner received his doctoral degree in Information Systems in 2014 from the Technical University of Graz, Austria. He also holds a MBA degree from the Joseph-Schumpeter-Institute in Wels, Austria. In his position as System Architect for NFC infrastructure products at NXP he is responsible for the definition of product specifications as well as product quality. He is author of research articles published in international journals and conferences proceedings.

Method to establish strategies for implementing process improvement according to the organization's context

Mirna Muñoz[1], Jezreel Mejia[1], Gloria P. Gasca Hurtado[2], Maria C. Gómez-Álvarez[2], Brenda Durón[1]

[1]Centro de Investigación en Matemáticas, Av. Universidad no 222, 98068 Zacatecas, México
{mirna.munoz, jmejia, brenda.duron} @cimat.mx
[2]Universidad de Medellín, facultad de Ingeniería, Cra. 87 No. 30-65, Medellín, Antioquia, Colombia.
{gpgasca, megomez}@udem.edu.co

Abstract

Software process improvement has become a logical way to address the growing need of increasing the competitiveness in software development organizations. Unfortunately, not all process improvement implementations have the desired results, because the existing models and standards focus their attention on "which activities to implement" without addressing "how to implement them". However, identifying which are the activities to implement is not enough; the knowledge of how to implement them is required for a successful implementation of software process improvement initiatives. This paper shows a method that provides strategies for the implementation of software process improvements based on the contextual aspects in which the software is developed, so that, the strategy is provided according to the organization needs and their work culture regarding project management.

Keywords

Software Process Improvement; Strategies for Implementing Improvements; Software Development Organizations; Organizational Work Culture.

Published in: Springer Communications in Computer and Information Science (CCIS) vol. 663

Self-What?— the single most important success factor

Danilo Assmann
(Vector Informatik GmbH, Ingersheimer Str. 24,70499 Stuttgart, Germany
danilo.assmann@vector.com)

Melanie Klemenz
(DOGA, S.A., Autovía A-2, Km. 583, 08630 Abrera – Barcelona, Spain
melanie.klemenz@doga.es)

Abstract

Software development is a complex task. At the latest since COCOMO we know that many different cost drivers exist, which we need to control to keep a project successful. Each of this cost drivers addresses a different aspect and needs attention. The search is on for the "magic silver bullet", which influences all of them. Our hypothesis is that no technical silver bullet exists, but a soft one. Looking at psychology and considering several established models, one key factor for destructive (costly) behavior is a lack of self-worth. Therefore, the question is: which cultural or organizational patterns influence this basic element, and how?

Keywords

Effectiveness; win-win; self-worth; improvement; productivity; comfort zone; transactional analysis

1 Introduction

Since 20 years, we are working in the area of change, process, and process improvement. Another 10 years older is the fascinating work on COCOMO from Barry Boehm (Boehm1986) (Boehm2000). One part of the publication is an overview of cost factors in software development.

We do not want to discuss COCOMO or the question if the data of Boehms overview is correct or still valid. We want to focus on some lessons learned from this overview. Two things that belong together, somehow two sides of the same medal.

- there is a huge number of influencing factors: influencing the cost and the success of a project; this leads to the finding that software development is a complex topic (Figure 1)
- there is no such thing as a “magic silver bullet”, because even the most influencing factors can provide only a relative small effect

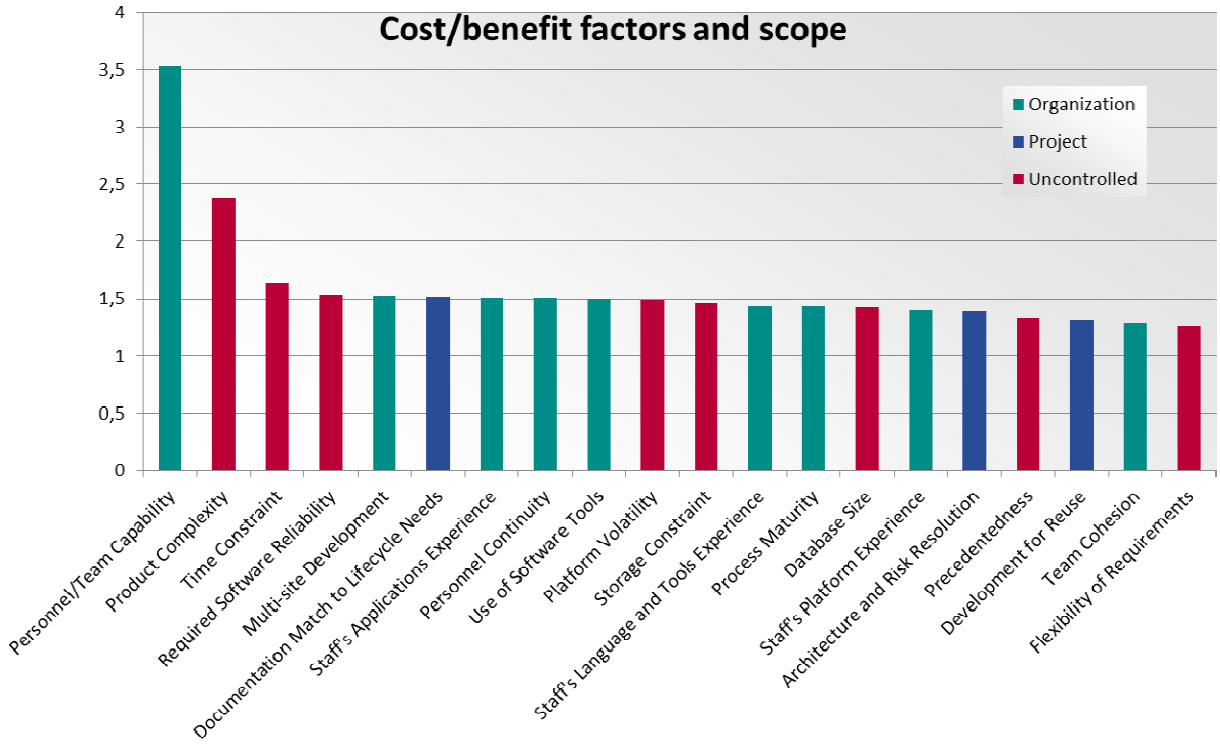


Figure 1: Cost driver factors with scope. How to read it: a factor of 1.5 can increase cost (effort) by 50%. Therefore, the topics with the highest values have the highest potential for savings and the highest variation in the effect of good and bad performance.

If we look at the chart, we can see another thing, process people do not like to hear so much: processes and process maturity is not one of the most influential factors. It is only one of several.

So how can we talk about a “single most important factor”?

Let us do some research and start with the mentioned scopes:

- Organization: organizational issues coming from the organization, which can also be solved in the organization
- Project: similar to organization, but in the project—project issues coming from the project, which can also be solved by the project

- Uncontrolled: this summarizes all technical issues

There is a quote we have heard and used quite often: “all problems can be traced back to management”. And from our experience this is true. Maybe not helpful, but true.

Gerald Weinberg states in his Second Law Of Consulting (Weinberg1985): "No matter how it looks at first, it's always a people problem."

The basic notion behind this law is that there are no such things as “technical problems”. Technique and technical solutions can be challenging, and some requests may be impossible (looking to the basic problems of the theoretical computer science, talking about undecidable or non-calculable problems). But that’s it. They are not a problem in itself. The problem comes through an organizational (management) system. Promises made, pressure build up based on business needs, not technical reality.

So far, so true. Still not helpful.

When we talk about the “single most important factor”, we talk about the single element influencing all others. This is the element organizations and projects are made of: humans. This leads us to the two hypotheses we want to look at in this paper:

Hypothesis 0: Humans are the single most important factor for productivity.

Hypothesis 1: All problems in humans and human interaction result from self-worth (self-esteem), or more precise from a lack of self-worth.

To check these hypotheses, we first look at a definition of self-worth and delimit it from similar terms.

2 Self-Worth and value

2.1 Definition

To get a clear picture fast, we start with the opposites. In (Ellis2016) three key beliefs of disempowerment are mentioned. Sadly, today these are common basic beliefs.

- Unworthiness: you believe that your worth is dependent on proving that you are worthy.
- Victimhood: you believe that others can harm you in some way.
- Powerlessness: you believe that you have little or no power to control your life.

There is an inherent logic, how these elements are interrelated. If you take a look at the social upbringing in the western world, key elements are “obeying to rules” and “performing to standard (expectation)”. You find this already in kindergarten, school, during college/university, and later also in work life. To be “okay” you have to fit in the expectation. You have to deliver your results and get the grades you need. There is always someone else (first the grownups, later the teacher, then your manager), who is judging (evaluating) you.

What you learn is: your value (worth) depends on your performance. There is always someone else how judges your performance. This builds the inner beliefs of unworthiness and victimhood. In case, you accept these beliefs in a broader view (which depends on your experiences), it will lead to the feeling of powerlessness, which is “I am not in control of my life”.

These key beliefs keep people from personal growth, and let us stay in our comfort zone. (Luckner1997) The risk with that is that when time passes I am not used anymore to step in the learning zone. The learning zone, as well as the comfort zone becomes smaller, see Figure 2. I will panic about everything outside my comfort zone. I will panic and will fight or flight according to the emotional type I am.

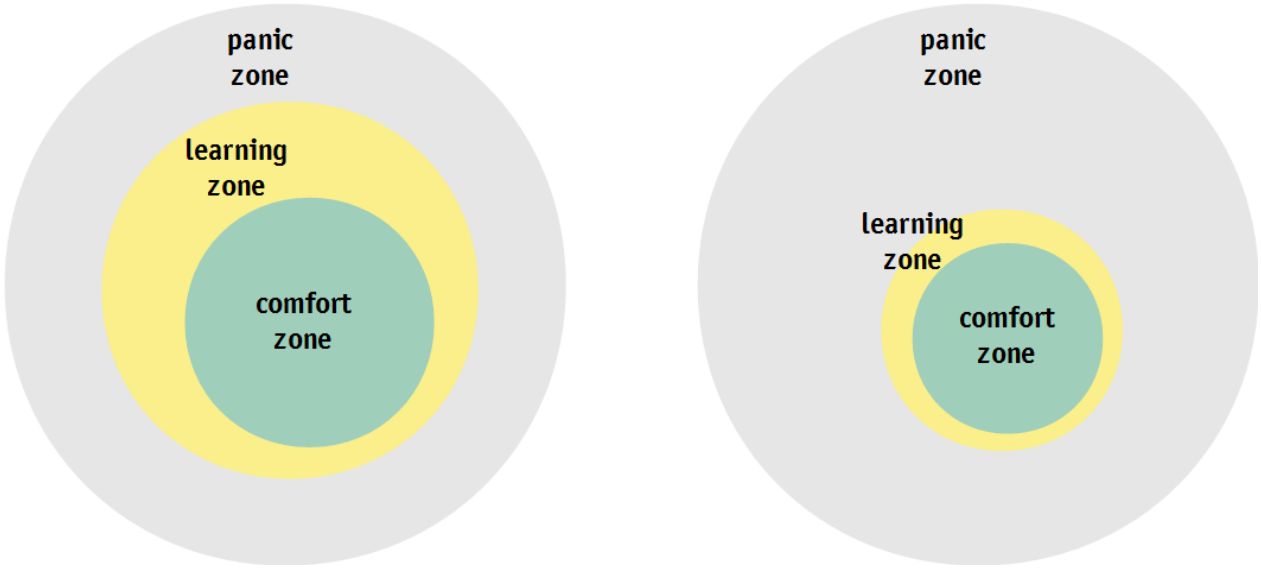


Figure 2: Basic model of the comfort zone (left). The concept of a collapsing comfort and learning zone on the right.

The definitions we look for are the reversals of these three key beliefs:

- self-efficacy which is the reversal for powerlessness: we see ourselves and others mastering skills and achieving goals. This is the confidence that, if we learn and work hard in a particular area, we will succeed; and it's this type of confidence that leads people to accept difficult challenges, and persist in the face of setbacks (Bandura1977)
- self-confidence is the reversal for victimhood: based on the definition of the Oxford Dictionary, self-confidence relates to self-assurance in one's personal judgment, ability, and power. It is the idea that I am able to control my life by what I think and do. I am able to shape my life. I can act and not only react. The more confidence I have, the more active I am, and the more control of myself, situations, and others do I have. The confidence comes from my experience and the feedback of others. So confidence is based on my looks, my abilities, my knowledge, my possession/status, my work/job, my achievements, and so on—and the reaction of others to each of them. It is a very useful and valuable part of me, but nothing stable, because everything it is based on can be lost. So it should only be a result of personal grows, never a goal in itself.
- self-esteem/self-worth is the reversal for unworthiness: the concept of self-worth is the fundamental pattern. It describes the inherent, not removable, not losable value you have as an individual. Actually, it is based on nothing, but your mere existence. It includes all your strength and weaknesses. Moreover, even in the worst case of complete immobility, you are still a valuable individual. You do not have to proof anything. You are valuable (worthy). Even if we rationally accept the basic pattern (it is included in nearly all modern laws), it is rarely felt. The basis for the feeling and personal acceptance of this concept comes from unconditional love and acceptance. So the self-worth is the not changeable (losable) value each of us has. We cannot increase or decrease it. There is no difference between anyone of us.

And one important basic element to find self-worth is:

- self-acceptance: As Robert Holden puts it in his book (Holden2015) “Happiness and self-acceptance go hand in hand. In fact, your level of self-acceptance determines your level of happiness. The more self-acceptance you have, the more happiness you will allow yourself to accept, receive and enjoy. In other words, you enjoy as much happiness as you believe you're worthy of.” We mention this additional concept, because it is the realistic enhancement of self-confidence. Confidence focuses on strength and positive feedback. Self-acceptance includes all shades of my personality. (Seltze2011) It is the first step to a healthy and holistic thinking and feeling.

The different concepts are related and interact with each other. Nevertheless, they address different aspects. I want to stress the two different layers: the value that is based on my actions and any form of

external feedback. This is constantly changing. The second layer is my connection to the inherent value I have. No matter what happens and what anyone thinks and does.

Now the challenge: many people out there (this may include us) never established a connection to their inner value or lost it in a context (culture) of conditional acceptance; or a culture in which victim-hood, powerlessness, unworthiness are accepted and practiced mindsets.

2.2 The bigger context

After we explained the basic mechanisms, we want to look at the effects, which these beliefs (and behaviors) have on the organization. We focus here mainly on the work about organizational health (Lencioni2002) (Lencioni2012) and organizational culture (Logan2012).

The concept of tribal leadership presents 5 levels of cultural patterns:

- Alienated (1): this is basically a hostile culture in war with “the world out there”
- Separated (2): not in war anymore, but passively antagonistic; seen everything fail and no hope for nothing
- Personal (3): the culture of heroes; I have to gather knowledge and save the world (so I am important)
- Collaboration (4): people are fully themselves and everyone seems happy and inspired; a culture of interdependence
- Team (5): it’s the culture of living a meaningful life; it is about following a bigger sense and having impact; change the world and see the unlimited potential

In the long-term study performed on organizational units, it showed that only 2% of organizations have a level 5 culture. On the collaboration level (4) are 22% of the organizations. So there is a huge potential wasted. (Logan2012)

Organizational health is the next level, which gives some indicators, why the different tribes behave as they do. Therefore, we have certain characteristics teams can work on. Key characteristics are (Lencioni2002):

- trust and vulnerability
- open and unfiltered conflict around ideas
- commitment and unambiguity
- accountability and high standards
- focus on achievements and collective results

2.3 All the way down to the core

What we want to do now, is to close the link between the organizational observations, and our initial topic. The models above discuss the behaviour of and in groups of people. This is the highest observable level. But as in physics, modelling the interaction of elements on this level is quite difficult, if not impossible.

In psychology we have two powerful approaches how the peer to peer interaction can be described. And the type of the peer to peer interaction, defines the culture of the group.

We use the game theory and the transactional analysis. In game theory (Mérö1998) (Spangler2003), and also applied for management (Covey1989), we find the patterns of win and lose (win/lose).

		I	
		win	lose
You	win	Win/win is the belief that everyone can win in a situation. It is the attitude that involves caring about both the other person and about yourself. Win-win is about believing that there is plenty of success to go around.	Lose/win is a situation where you give in rather than to try to express your feelings or needs. You compromise your values and standards to make others happy.
	lose	Win/lose is an attitude in which everything and everyone is a competition. While competition can be a very good thing, people working with this attitude view life like a game – there can only be one winner. Relationships are not as important as being the best.	Lose/lose is a situation where no one comes out with what they need or want. It is a negative situation for everyone involved.

Figure 3: The simplified win/lose model.

Covey mentions that there are (extreme) situations, in which each of these patterns is acceptable. In a collaborative (interdependent) reality, the only viable alternative is win/win. Inside our relationships (organizations, projects) we always have to think win/win. Therefore, we need the win/win kind of people, which have developed to live beyond their ego.

As you can easily see, these four patterns map quite well to the first four organizational levels. The fifth level is based on the win/win attitude, but with an added sense of life/purpose. Covey addressed this topic in (Covey2004).

Inside an organization or system, the only positive attitude is win/win. As long as we stay in one system, the overall balance for all other combinations is zero or negative.

Win/win can be achieved with two strategies. The first is two operate on two different dimensions. This is the standard use case in sales: you want the product and the producer wants your money. As long as you value the product more than your money (and the producer the money more than the product), you achieve win/win, because you use two independent dimensions. The second strategy is the “third way”. Conflicts arise because there are two alternatives and every party wants to have it “their way”. The third way is an alternative solution, which addresses the basic needs behind the proposed alternatives and tries to merge/fulfill them. This is much more than a compromise (where both parties lose somehow, which makes a compromise lose/lose). The third way should be consents, that mean that both parties get their needs fulfilled, which makes it win/win.

The second model we use to explain group culture based on peer to peer interaction is the transactional analysis (Berne1964) (Harris1976). Transactional analysis distinguishes four life positions, which reflect in the form of interaction we have. The life positions are based mainly on how I see myself, and how I see others.

		I	
		Okay	Not Okay
You	Okay	This is the healthiest position about life and it means that I feel good about myself and that I feel good about others and their competence.	In this position the person sees him/herself as the weak partner in relationships and feels that others in life are definitely better than the self. The person who holds this position will unconsciously accept abuse as OK.
	Not Okay	In this position I feel good about myself but I see others as damaged or less than me and this is usually not healthy.	This is the worst position to be in as it means that I believe that I am in a terrible state and the rest of the world is bad as well. Consequently, there is no hope for any ultimate support.

Figure 4: Basic Okay/Not Okay dimensions.

Equipped with these tools we go the chain down to the core. The elements of the transactional analysis lead us already the right way: the basis for my interaction is, how I see myself and how I see others. So we have the chain:

- behavior inside a group is based on
- the peer to peer interaction I chose, which is based on
- how I see myself (and feel about myself) and how I compensate.

So let's take a look at the core of behaviour.

2.4 Core levels

Computer scientists love levels. Therefore, we provide a level model for self-worth or value. How valuable do you feel? Moreover, why is it that way? Still more important: How can I identify the level of others based on their behavior.

Following the concept of the four levels of self-esteem/worth (Satir1991) (Pellier2012) (Martin2011)

- supplicative (1): these people see no value in themselves. Very sad for the person, but otherwise nice to handle. This people depend on other people (reflect their value), and are quite nice because they buy stuff or do things to get value (earn your attention). (They accept lose/win)
- combative dynamics (2): these people do not feel as good as the others (they think they have a lower value), therefor they try to decrease value from others. This destroys relations and team spirit–nothing you want to have in your team, and especially not between your manager/leader. This kind of people is most difficult to control/interact with. (They accept win/lose and lose/lose)

Some further comments on this type: According to Adler (Adler1930), people who feel inferior go about their days overcompensating through what he called "striving for superiority." The only way these inwardly uncertain people can feel happy is by making others decidedly unhappy. To Adler, this striving for superiority lies at the core of neurosis. This is also highly related to narcissism.

"The two kinds of narcissists are the grandiose (who feel super-entitled) and the vulnerable (who, underneath the bravado, feel weak and helpless). Some may argue that at their core, both types of narcissists have a weak sense of self-esteem, but the grandiose narcissist may just be better at the cover-up. In either case, when you're dealing with someone who's making you feel inferior, there's a good chance that narcissism is the culprit." (Wang 2015)

- competitive dynamics (3): They feel only good if they can show value. Therefore, they try to beat others, be better. This can be aggressive, but it can also be used as motivation such as games and competition. You can use their ambition. (They accept win/lose)
- cooperative dynamics (4): These people are looking for consent (win-win). They are happy if they make other people happy (more valuable). They want to be effective and helpful, say nice things; be positive. They have the inherent wish to grow; and that the system (which they are part of) improves.

We understand that self-worth, the connection to our inherent value, is a key driver for a positive (system wide positive) behavior. If this connection is lost or was never established, the behavior is driven by insecurities and their compensation. To give you more hints and indicators here is some help and guidance to spot insecurity (Krauss2015). Basically, you have to address all forms of narcissism.

To take the concept a step further we come back to hypothesis 1: all destructive behavior is a result of compensation and a form of winning/losing (lose/lose or lose/win).

- mistrust/fear/insecurity/paranoia
- anger/aggression/threatening/being sarcastic or cynical
- passive behavior/work to rule/blame game
- lies including false reporting, bad talk behind back, gossip, …
- bullying/mobbing (actually both parties suffer from the same low self-worth, just disguise it completely different)
- and everything else that destroys team spirit

These are exactly the dysfunctions we encounter on the organizational level.

Side note: The same holds true for the whole society, which is also a closed system (and nearly all interactions are win/lose or lose/lose).

3 The cure

If we could bet, we would bet that 80% of our readers will now ask for a technical solution. Which processes/tools/methods/techniques can we use to fix this? And we are sorry, but there is no quick fix. There is never a technical solution to a personal issue. So it is about character development and cultural development.

The basic idea is to understand that many common attitudes and behaviors seen as strength, are nothing else than compensation and in many cases narcissistic behavior. Understanding the mechanisms of insecurity will already change a lot. Because you understand in negative situations: the destructive behavior is not about me, it is the insecure core of the other person.

Now the cure: We learned that every (human) being wants love and acceptance. Everyone.

We can see it for ourselves. What we want is a beautiful life. We want to be loved and accepted. We want peace. We want to be happy. We want enough to eat. We want to be effective in our work. We want a sense of accomplishment, a sense of purpose. We want support. We want direction, when we need it. We want trust. And so on. Moreover, we (EuroAsiaSPI people) want improvement -- continuous improvement.

All this is no problem, if we are surrounded by Level 4 (win/win) type people. Therefore, the cure for all the insecurities and the lack of self-worth, which is the basis for this, is *unconditional acceptance*. Unconditional acceptance allows us to (re-)connect to our inner and inherent value. As soon as I rest in this inner value, I am satisfied and in peace with myself. Then I do not have to prove anything anymore and look after my hurt ego all the time. Therefore, I am free and have time and energy for the work that has to be done.

Now we face several problems (just an excerpt):

- This is a real soft topic; no simple engineering and no quick fix, no technical solution. It gets really personal. Something we engineers do not like so much/cannot handle so well/don't believe in.
- Unconditional acceptance (trust) is counter intuitive, because we are used to rules and control (see the three destructive key beliefs). Some people believe in processes, because they do not trust the people.
- In case of hierarchical organizations (Weinberg1986): If we have problems in our teams, then managers do something wrong. That might mean we have a (few) managers, who are not yet Level 4 personalities. How can we touch this topic without loss of face for them (and us)?

So now is the time for a paradigm switch, and we have a plan for this. You can use it step by step. It does not matter, if you are a manager (appointed Leader) or a team member. You will lead by example independent of your role. It is just work on your character. Good guidance gives the following saying, which is actually quite old:

"Watch your thoughts, they become words;
watch your words, they become actions;
watch your actions, they become habits;
watch your habits, they become character;
watch your character, for it becomes your destiny." (from the Talmud)

Now you find our proposal for some first few steps. You can start right now and very simple. No additional tool or technique needed:

1) Get your intention straight: Do we want to be cool or effective? It is not an easy way. We have to grow humble and vulnerable. It's not cool and sexy on first sight. However, to become Level 4 will change the world in the long run. This is the change from self-marketing to trust and vulnerability.

2) The first action is to look differently at people: no devaluing thoughts anymore. Allow yourself no negative thoughts and feelings towards other people. Be curios. Try to understand what drives the people around you. Separate the intensions from their behavior. (Satir1991) Sounds easy, but this practice will show you quite fast where you have your personal problems and insecurities. What are your personal challenges and where do you have to connect to your inherent not losable value.

3) Change your beliefs. The personal challenges are most likely connected to wrong (not helpful for your purpose of effectiveness) beliefs. You have to understand your beliefs and challenge them. Then you can change them.

4) Give unconditional acceptance. This is again about trust and respect. With an attitude of unconditional acceptance towards others, you will increase their value. All interactions in your relationships will be beneficial, because it is not anymore about you, but about them and a higher purpose (collective goal).

Note: There are a lot of useful concepts and trainings developed since many years, which exactly focus on the buildup of character and self-worth. One example is the "7 habits of highly effective people" (Covey1989)*, which leads gradually to growth of self-worth and personal effectiveness.*

All this will continuously stretch your comfort and learning zone (Figure 5). In addition, it will affect all the people around you. Unconditional acceptance will release more power and motivation in the people you work with or supervise (or who supervise you) than any other tool.

The buildup of self-worth will change you/them and every aspect (area) of your/their life. Because it affects all relationships, we have: work, friends, family.

Coming back to Boehm: the effect of changed human behavior addresses 12 of the 20 shown cost drivers in Figure 1; which gives a total cost factor of around 24. This is far more than we can achieve using any other measure.

4 Conclusion

The topic is not new. The oldest literature we quote is from 1930. Some basic concepts are well known over 1000 years.

Our impression is that nowadays many people are tired of technical solutions to emotional problems. As examples: Following LinkedIn pulse on Management and Leadership the demand for emotional capable leaders is high and growing. In addition, the term Emotional Intelligence became more and more popular over the last 20 years, even being around since 1964.

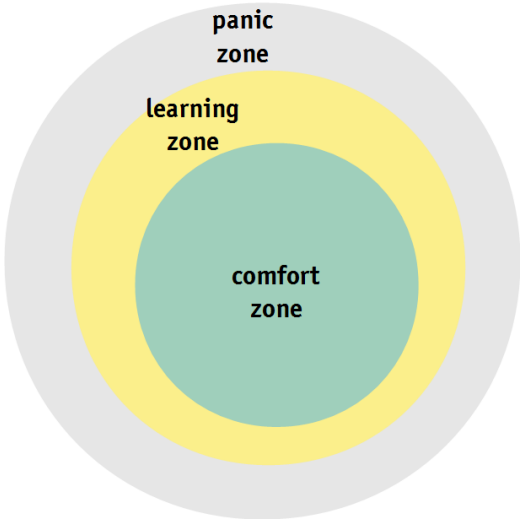


Figure 5: Growing comfort and learning zone.

The question is: has the time come for the next step in process improvement? To look at the complete system and take a holistic approach, which must include the human nature and basic needs.

This leads also to a reevaluation of our concepts of rationality. The main drivers for our behavior are not rational, and often even not conscious. Following the sketched concepts in this paper, you can see rationality just as disguised emotions. (Damasio1999) If we want to be successful in change and improvement, we have to improve the people. Help them heal and grow. This approach will lead to a new dimension of effectiveness.

Are you ready?

5 Literature

(Adler1930) Adler, A. The Education of Children. 1930. from http://psycnet.apa.org/psycinfo/1930-04004-000

(Bandura1977) Bandura, A., Self-efficacy: Toward a unifying theory of behavioral change, Psychological Review, 84, 191-215, 1977.

(Berne1964) Berne, Eric, Games People Play – The Basic Hand Book of Transactional Analysis, New York: Ballantine Books, 1964.

(Boehm1986) Understanding and controlling software costs, Barry Boehm, 1986.

(Boehm2000) Safe and Simple Software Cost Analysis, Barry Boehm, 2000.

(Brookes2015) Brookes, J., The effect of overt and covert narcissism on self-esteem and self-efficacy beyond self-esteem, Personality And Individual Differences, 85172-175, 2015.

(Covey1989) Covey, Stephen R., 7 Habits of highly effective people, Turtleback Books, 1989.

(Covey2004) Covey, Stephen R., The 8th Habit: From Effectiveness to Greatness, FreePress, 2004.

(Damasio1999) Damasio, A., The Feeling of What Happens: Body, Emotion and the Making of Consciousness, Heinemann, 1999.

(Ellis2016) Ellis, Nanice, 2016, http://wakeup-world.com/2016/02/21/escaping-the-matrix-of-depression-the-truth-about-depression-shall-set-you-free/

(Harris1976) Harris, T., I'm OK-You're OK, Avon Books, 1976.

(Holden2007) Holden, Robert, Happiness Now!: Timeless Wisdom for Feeling Good, 2007.

(Krauss2015) Krauss Whitbourne, Susan, 2015, https://www.psychologytoday.com/blog/fulfillment-any-age/201511/4-signs-someone-is-insecure

(Lencioni2002) Lencioni, P., The Five Dysfunctions of a Team, Jossey-Bass, 2002.

(Lencioni2012) Lencioni, P., The Advantage: Why Organizational Health Trumps Everything Else In Business, Jossey-Bass, 2012.

(Logan2011) Logan, D., King, J., Fischer-Wright, H., Tribal Leadership, HarperBusiness, 2011.

(Luckner1997) Luckner, J. L., Nadler, R. S., Processing the experience: Strategies to enhance and generalize learning (2nd ed.), Dubuque, 1997.

(Martin2011) Martin, Leo, Wir kriegen dich, .ARISTON α, 2011.

(Mérö1998) Mérö, László, Die Logik der Unvernunft – Spieltheorie und die Psychologie des Handelns, rororo, 1998.

(Pellicer2012) Pellicer, Joshua, The Tao of Badass, eBook.

(Satir1991) Satir, Virgnia; The Satir Model: Family Therapy and Beyond; Science and Behavior Books; 1991.

(Seltze2011) Seltze, Leon F., 2011, https://www.psychologytoday.com/blog/evolution-the-self/200809/the-path-unconditional-self-acceptance

(Spangler2003) Spangler, Brad. "Win-Win, Win-Lose, and Lose-Lose Situations." Beyond Intractability. Eds. Guy Burgess and Heidi Burgess. Conflict Information Consortium, University of Colorado, Boulder. Posted: June 2003

(Wang2015) Wang, H., Lu, C., & Siu, O., Job insecurity and job performance: The moderating role of organizational justice and the mediating role of work engagement, Journal of Applied Psychology, 100(4), 1249-1258, 2015.

(Weinberg1985) Weinberg, Gerald M., Secrets of consulting, Dorset House, 1985.

(Weinberg1986) Weinberg, Gerald M., Becoming a Technical Leader, Dorset House, 1986.

6 Author CVs

Danilo Assmann

Danilo Assmann worked now for 20 years in the domain of process modeling and coaching. He started with projects in the area of technology transfer from science to practice during his time at Fraunhofer IESE. Then he moved fully to industry, working in different companies.
In several improvement projects, he practiced how to manage change and enable people. He is still learning day by day.

Melanie Klemenz

Melanie Klemenz divides her life between Spain and Germany. She is working for over 10 years in international business, supporting and enabling people. She lives her philosophy: change happens through people.
Since 5 years she looks into bipolar and anxiety disorders to understand the mechanisms which create and may heal them. These studies started the work on this paper, and challenged both authors to bring the personal experiences in a broader context.

More Effective Sprint Retrospective with Statistical Analysis

Muhammed Emre PEKKAYA, Onur ERDOĞAN, Halime GÖK

TUBITAK-BILGEM-YTE[1]
Ankara / TURKEY
emre.pekkaya@tubitak.gov.tr, onur.erdogan@tubitak.gov.tr, halime.gok@tubitak.gov.tr

Abstract

Scrum teams aim to deliver products productively with the highest possible value and quality, so they try to deliver high priority and high value product backlog items (PBIs) first to increase their productivity without compromising product's quality. Besides, PBIs, which are going to be delivered at the end of the sprint, are determined in sprint planning meeting. Therefore, planning a sprint properly is a significant issue to optimize value. Making size estimation of PBIs correctly is one of the most prominent factors for effective sprint planning.

Sprint retrospective meetings are an opportunity for the Scrum teams to improve product quality, their productivity and estimation capability. Enhancing in those areas requires empiricism as agility requires; hence measureable indicators for quality, productivity and estimation capability should be inspected and adapted at regular intervals. Inspection and adaptation require transparency. For providing transparency, we studied how and what kind of historical data is required to be collected for monitoring, and how statistical analysis of data can be investigated for inspection and adaptation in retrospective meetings.

We have experimented that statistical results of "Correlation between Story Point and Actual Effort" and "Consistency of Relative Estimation" are very convenient for inspection and adaptation of estimation capability of the development teams in retrospective meetings. Past retrospective meetings also showed that statistical results of "Team's Actual Effort on Product", "Team Velocity", "Actual Effort for One Story Point", "Innovation Rate" and "Velocity vs. Unplanned Effort Rate" are very helpful to control and increase the productivity of the development teams. "Actual Effort Rate of Quality Activities" and "Sub-component Defect Density" statistical results also helped a great deal on enhancing the product quality.

Keywords

Scrum, Relative Estimation, Productivity, Product Quality, Sprint Retrospective, Process Improvement, Statistical Analysis

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

1 Introduction

1.1 Brief Information of the Project

In this study, we used the data generated during KAYS (Management System of Development Agencies) that is an internet based management information system project, developed in TÜBİTAK-BİLGEM-YTE. YTE is a research and development institute providing e-government software solutions with software engineering expertise. It is a middle scale institute with 150 employees, 60 of whom are software engineers. During the software development, CMMI-DEV Version 1.3 [5] high maturity practices are followed by the organization with additional support of agile frameworks, methodologies and practices approach [2].

One team, which consists of 6 team members, 1 scrum master and 1 product owner, is working on a module of the project. The team is cross functional. Each member of the team has different skills and specialties that are required for the job at hand. Hence, the team members complement one another. They pull work for themselves, don't wait for their leader to assign work and manage their work like identifying new work, updating an existing work, estimating its ideal time, entering its actual effort and allocation as a group. Up until now, the development team ran 30 sprints. Each sprint was 2 weeks long. Briefly, as shown in Fig 1, before a sprint, the product owner prioritizes PBIs and specifies its type either as story, defect, enhancement, requirement change or task. They also have to set the related sub-module of each PBI. Afterwards, the team estimates the size of PBIs by using the relative estimating method thus a sprint backlog emerges. During a sprint, the team does daily scrums. Finally, they have a retrospective meeting with the results of the statistical analyses of the sprint data such as story points, PBI type, sub-module and actual effort in order to monitor and improve quality, productivity and estimation capability.

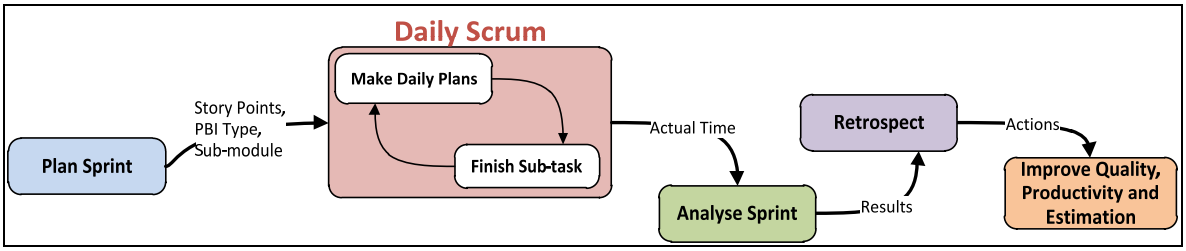


Figure 1. Workflow of the Team's Activities

1.2 Relative Estimation

During the sprint planning meeting, in order to create the sprint backlog the development team determines the size of product backlog items (PBIs) and selects the PBIs with the product owner according to their priority, scope and the development team's velocity [2]. The development team uses relative estimating method while determining the size of the PBIs. The development team compares the PBIs to each other in terms of complexity and assigns a story point to each PBI in the product backlog. Namely, it is a planning poker [3]. The team uses 0, 1, 2, 3, 5, 8, 13, 21, 34, 55 … as a story point.

1.3 Daily Planning

During daily planning the development team members identify or update sub-tasks of PBIs into sprint backlog and estimate their ideal time. Operations like identifying new sub-tasks, updating an existing

sub-task, estimating its ideal time [3], entering its actual effort and its transition from one state to another state are often performed daily via an electronic board. For example, in “Progress” phase the team members working on a PBI identify its sub-tasks [4] and estimate their ideal time [3]. After they finish each sub-task, they log its actual effort. While verifying and validating a PBI, defects and enhancements detected by the verifiers and the product owner are added as sub-defects or sub-enhancements of the PBI into the electronic board. After correcting each sub-defect of the PBI and finishing each sub-enhancement, the development team members save its actual effort into the board. Verification and test activities of a PBI are added as sub-review and sub-test into the electronic board. Thus ideal time of a PBI is calculated by sum of ideal time of its sub-tasks, sub-defects, sub-enhancements, sub-review and sub-test. Actual effort of a PBI is calculated by sum of actual effort of its sub-tasks, sub-defects, sub-enhancements, sub-review and sub-test.

1.4 Improving Estimation Capability and Productivity of the Team and Product Quality

While the team is having a sprint review meeting, the sprint is analyzed independently by Quality and Process Improvement Unit of the institute using the sprint data such as story points, ideal time, actual effort and type of PBIs originated by the development team running the sprint. The Quality and Process Improvement Unit produces results related to the development team’s productivity such as “Team’s Actual Effort on Product”, “Team Velocity”, “Actual Effort for One Story Point”, “Innovation Rate” and “Velocity vs. Unplanned Effort Rate”, its estimation capability such as “Consistency of Relative Estimation” and “Correlation between Relative Estimation and Actual Effort” and product quality such as “Actual Effort Rate of Quality Activities” and “Sub-component Defect Density” that are input for consideration in retrospective meetings. In a retrospective meeting, the team debates results and determine the corrective actions if needed to take in the following sprints for improvement [6, 7]. For instance, while the team is analyzing the “Consistency of Relative Estimation” results, they find out that relative estimation of a PBI was incorrectly determined. In such a case, the team investigates its root causes and identifies the actions to be taken, when similar PBIs will be estimated in the following sprints.

2 Materials and Method

In scrum, every sprint begins with a planning phase during which high-level user stories in the product backlog are transformed into the more detailed tasks of the sprint backlog by the scrum team. Stories, which the team needs to tackle first within the context of the sprint, are selected, negotiated and given higher priority in points using Fibonacci numbers in the corresponding sprint backlog. Within the sprints, the requirements are broken into tasks and each task’s effort is also estimated, hence everyone in the team can be aware what will be completed and included in the sprint [7].

2.1 Dataset

All PBIs and their related sub-tasks are managed using an issue management tool called Atlassian JIRA within the organization. The tool has the functionality to track, plan and follow a team’s activities. In this study, we obtained all the necessary data fields using the JQL (JIRA Query Language) such as PBI type, sprint info, status, closed date, ideal time, time spent, and story point.

2.2 Preprocessing Phase

Preprocessing takes considerable amount of effort in data analysis and it is an important preliminary step for making data ready for further analysis. Since the quality of data affects the accuracy of the results, noise reduction is applied. The first implementation is assigning the tasks into the correct sprint on which it was resolved, because JIRA automatically moves the sprint label into the next one if the issue was not resolved in the current sprint. This can be done comparing the closed date of the issue to the Sprint start/finish dates and duration interval. While doing this, history log is searched in order to

find out how much effort is planned and given to the duplicated issues taking the sub-issues' log into consideration. The second implementation is eliminating the issues with zero points. Those do not contain novel information in terms of information retrieval. The third implementation is identifying and removing the issues for which log records are not entered by the team members.

2.3 Data Analysis

Statistical methods are used to analyze the quantitative data in order to visualize the flow of process and make better decisions in future planning activities [6]. Well established metrics help the team members measure their performance and make further changes based on facts, not just on feelings. Metrics make visible the impact of any modification during the development, from the introduction of new technology to changes in process or even team composition [8]. In this study, different types of metrics are considered using parametric and nonparametric statistical methods.
Correlation between Story Point and Actual Effort: Correlation analysis is implemented in order to evaluate if there is a correlation between given story points to PBIs and actual efforts. Since data does not fit to normal distribution, Spearman correlation is interpreted considering the p-value in 95% confidence level. This tells how effective the actual effort is assigned to each point. Expected result is that there should be a correlation between story point-time spent logs.
Consistency of Relative Estimation: One of the most important analyses is identifying the outliers which are extreme values considering PBIs with story point and actual efforts spent on each PBI. These PBIs are determined using box plot outlier analyze. By doing this, false story point assignments are identified. Consequently, this gives opportunity for more accurate relative estimation in following sprint planning meetings.
Team's Actual Effort on Product: Time spent distribution based on sprints is summed and area chart is given since we expect to see similar level of actual effort in each sprint in order to see the team's concentration on the product during a sprint.
Team Velocity: In each sprint planning stage, PBIs are given points and total points tell how big the sprint is. At the end of sprint, closed PBIs are evaluated and cumulative point is calculated. This metric gives information about the team velocity. If the cumulative points are less, team is considered as slowing down; on the other hand team velocity is getting better.
Actual Effort for One Story Point: Estimated number of story points had better become similar during each plan. However, actual efforts spent on each PBI may vary. This can show the team motivation of whether time is wasted or not while solving issues. In order to detect this, total number of estimated story points is divided to actual efforts in each sprint. Hence, how much time is spent for a unit of story point may be calculated.
Innovation Rate: Business value is the ratio of PBIs whose types are story, enhancement and requirement change to the all PBIs in a sprint. The trend is given on a line chart. Then, during a retrospective meeting the team inspects the chart and if extreme situation exists in the sprint, root causes are investigated in order to take corrective actions for adaptation.
Velocity vs. Unplanned Effort Rate: Finally, line chart is given in order to show the team velocity versus the ratio of unplanned time spent within whole time spent in a sprint since hot fixes are currently added to current sprint.
Actual Effort Rate of Quality Activities: As verification and test activities are increased, quality is also enhanced for the final product. Not only fast development but also effective and proper test activities are needed. Time allocation for verification and test activities are calculated by dividing to total efforts. Expected result is to see an increasing trend.
Sub-component Defect Density: One of the informative metrics, which measure the defect density of sub components, provides the ability to take action in order to decrease defects in each sub components. Main aim is to reduce time spent for defects. This calculation is done by dividing time of working on defects to total time spent in a particular sprint. Hence, less defect working time means high quality considering density of defects.

3 Results

During software development, measurement is mostly the missing step to compare product and process quality. Statistical analysis helps the team plan the following sprints in an effective and predictable

way. Enabling agile projects to achieve the established quality and process performance objectives without losing collaboration and interactions, depends on Quantitative Management techniques for effective monitoring and controlling of the agile process [8].

From the statistical analysis results, strong correlation between the story points and the actual effort for PBIs is revealed. Spearman correlation coefficient is found as 79% with the p-value (0.00), which is smaller than 0.05 and means the correlation, is significant. The presence of strong correlation between actual effort and story points infers standardization across the team during relative estimation. This also means productivity of various sprint plans of the same team could be compared.

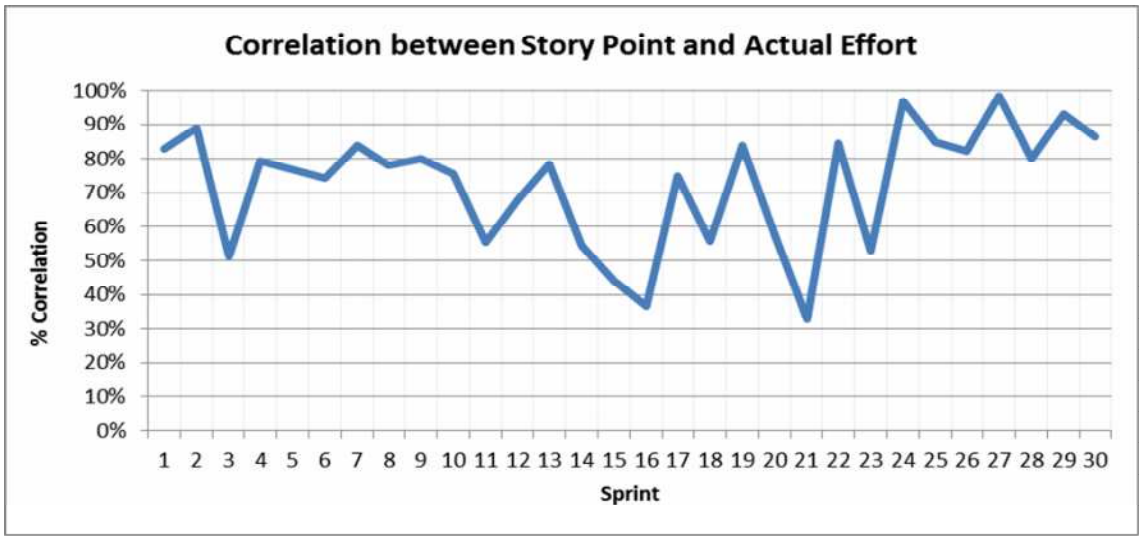


Figure 2. Correlation between Story Point and Actual Effort

Relative estimation is done for each PBI; however some of them are incorrectly predicted. Outlier detection may be useful for future sprint planning. Figure 3 shows the estimation capability of the team from first sprint to 30th sprint. Box-plot between the story points and the actual effort shows that the team mostly estimates 1-point PBIs incorrectly. In addition, the highest variation is found in 5- points and 8-points PBIs as shown in Figure 3.

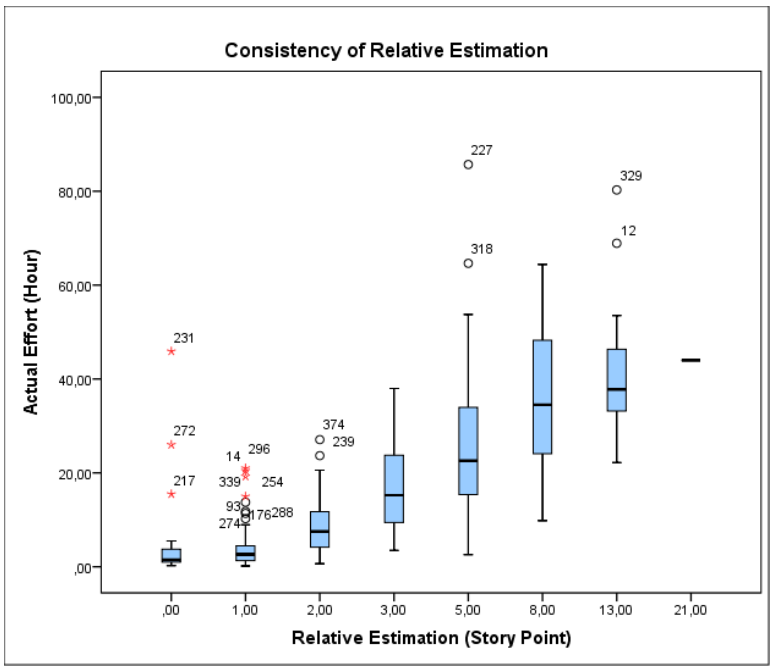


Figure 3. Consistency of Relative Estimation

"Team's Actual Effort on Product" shows the level of the team members' concentration in a sprint. In retrospective meetings, the team compares its concentration in the current sprint to the concentration levels in previous sprints. If there is an unusual decrease or increase, they elaborate on the possible root causes. In first sprint, the actual effort is less than the effort in other sprints, because the team consists of two people. After first sprint, three more people were added to the team. By 21th sprint, the team size is six people. According to

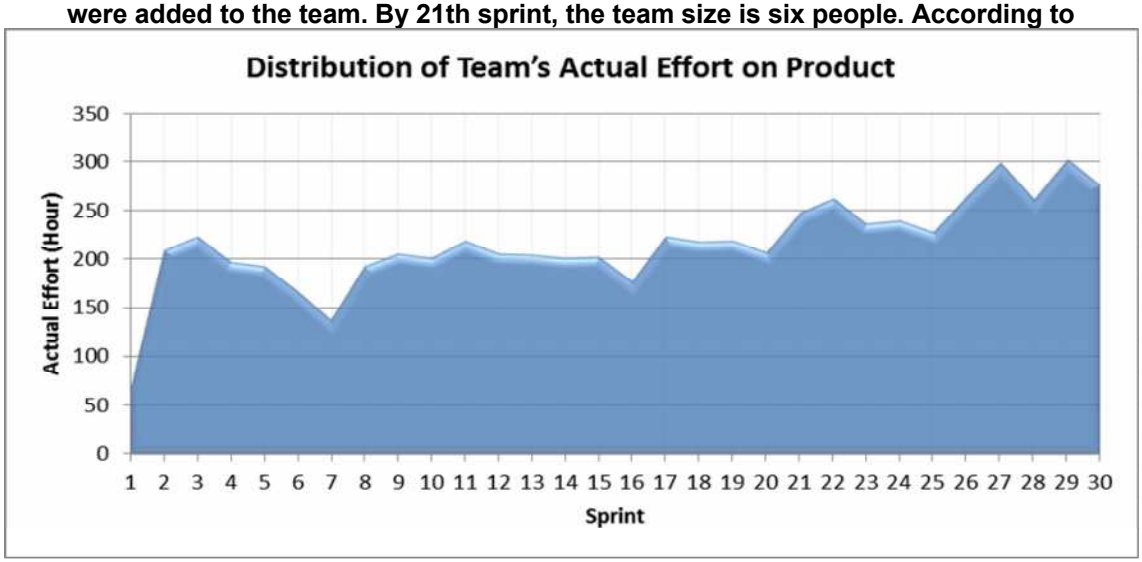


Figure 4, the team put different levels of effort for each sprint. Mostly, the variation in the team size between sprints is the main reason. However there is a descending effort from sprint 4 to sprint 7. In retrospective meetings the root cause was determined as the team members went on vacation after sprint 3 and had concentration issues from sprint 4 to sprint 7.

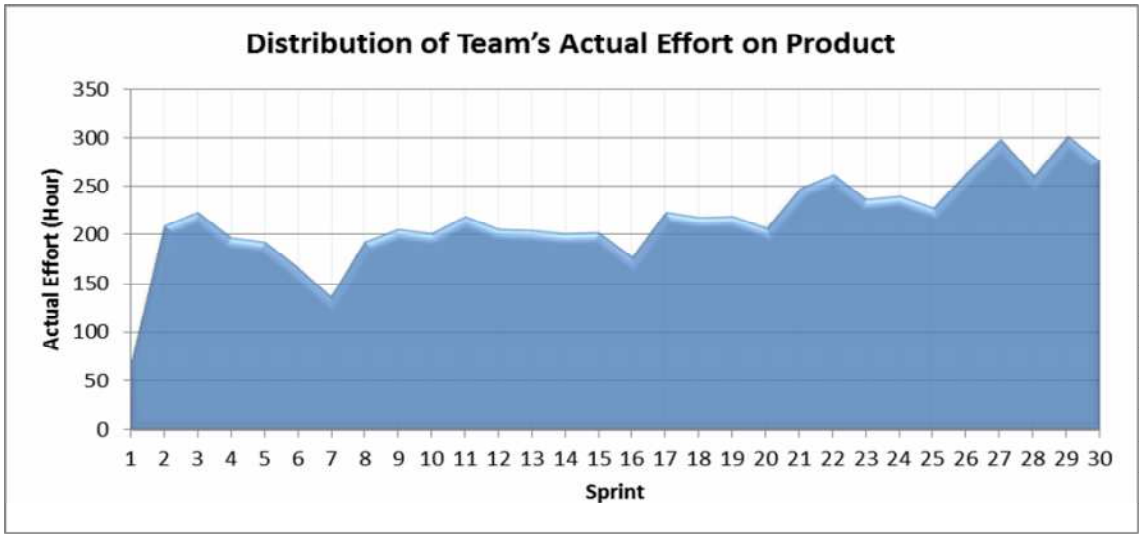


Figure 4. Team's Actual Effort on Product

According to Figure 5, the average pace is 42 story points, the lately average pace is 36.38 story points, the slowest pace is 25.33 story points and the fastest pace is 54.33 story points. Team Velocity analysis helps the team to do sprint and release planning more accurately. Hence, the team easily knows which prioritized PBIs can be delivered on the release date.

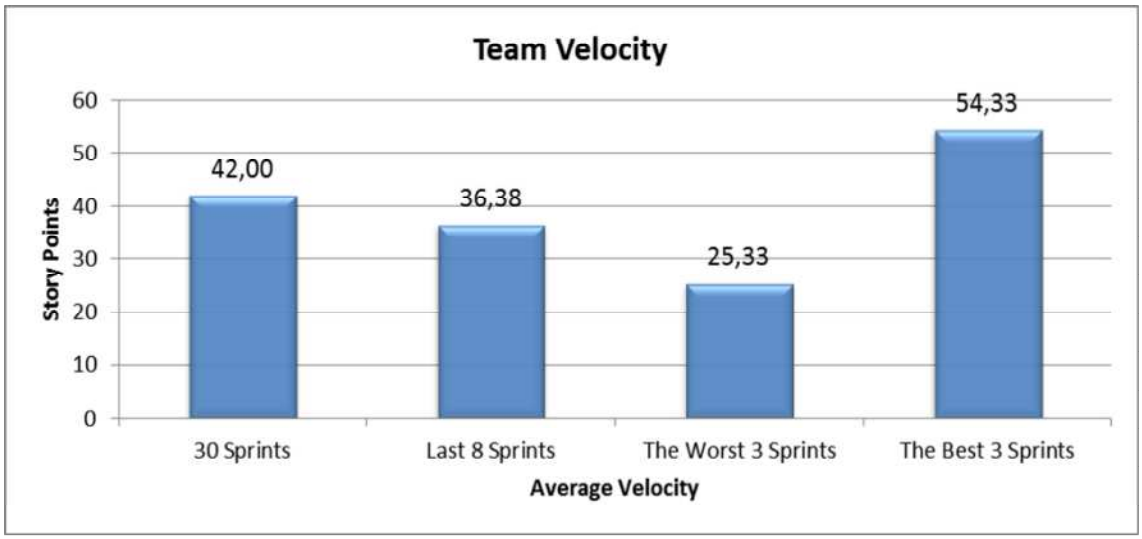


Figure 5. Team Velocity

Figure 6 shows how much average effort the team spent for one story point in each sprint. It helps the team figure out if the developed system is maintainable enough. The more features the system has, the more complex the system is. The graph below proves this situation. According to Figure 6, there is an ascending actual effort for one story point from first sprint to the last sprint. The highest effort was spent in 24th and 27th Sprint. In retrospective meetings, the team reviews the graph and if necessary, the team identifies corrective actions for increasing sustainability of the system developed.

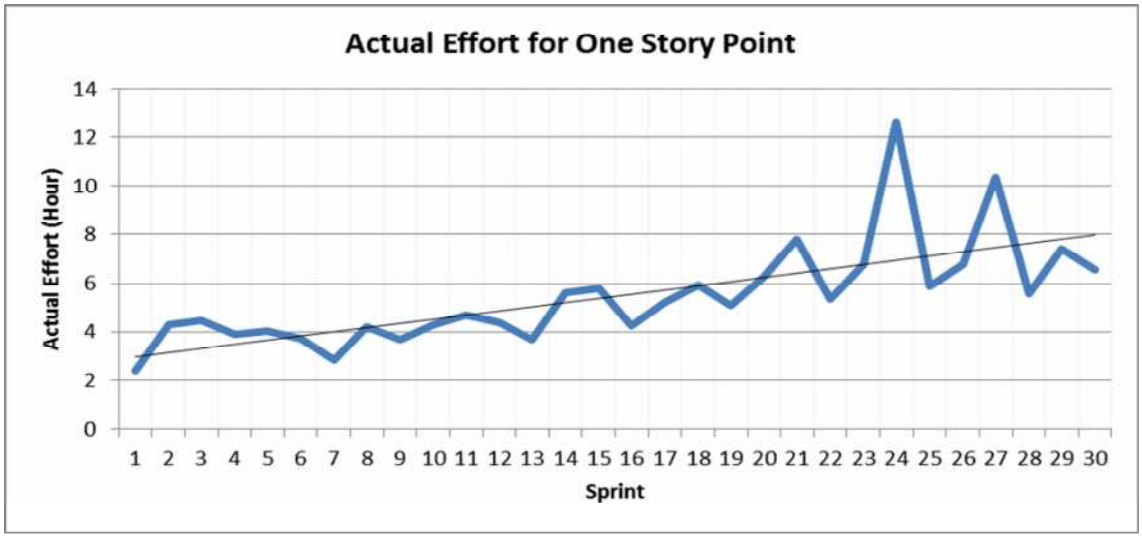


Figure 6. Actual Effort for One Story Point

As shown in the Figure 7, innovation rate is mostly more than 75%. The team aims to keep it at the level of 75% or more. The major decline in 10th sprint is due to the test activities' focus, because the product was going to be partially released for the first time. Because the product was released at the end of sprint 11, the team had to deal with the defects from customer and end user that was main reason for the ratio decrease between 12th and 17th sprints. This result was discussed in the retrospective meetings for taking actions to maximize value in future planning.

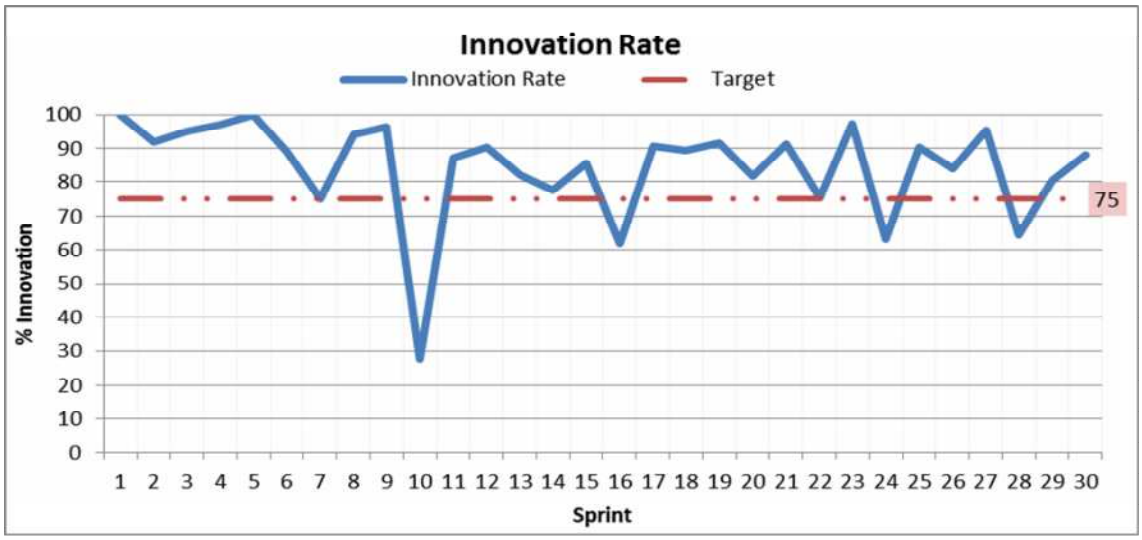


Figure 7. Innovation Rate

As shown in Figure 8, hot fixes negatively affect the sprint, the team's productivity and business value provided. In order to see how much effort is given to unplanned PBIs and whether they affect the team's velocity and how, the line chart that shows the team's velocity and unplanned effort rate is given below. Urgent customer or end user demands started to come and had to be included within the sprint after the 11th Sprint since the first release was at the end of the 11th Sprint. The highest unplanned effort is given in Sprint 14 and Sprint 19 with 8%-10%. Users started to use the main features of the product very intensively first time during 14th Sprint and a sub-module, which is related to other sub-modules, was released first time after 18th Sprint so the team had to deal with hot fixes between 14th and 19th sprints.

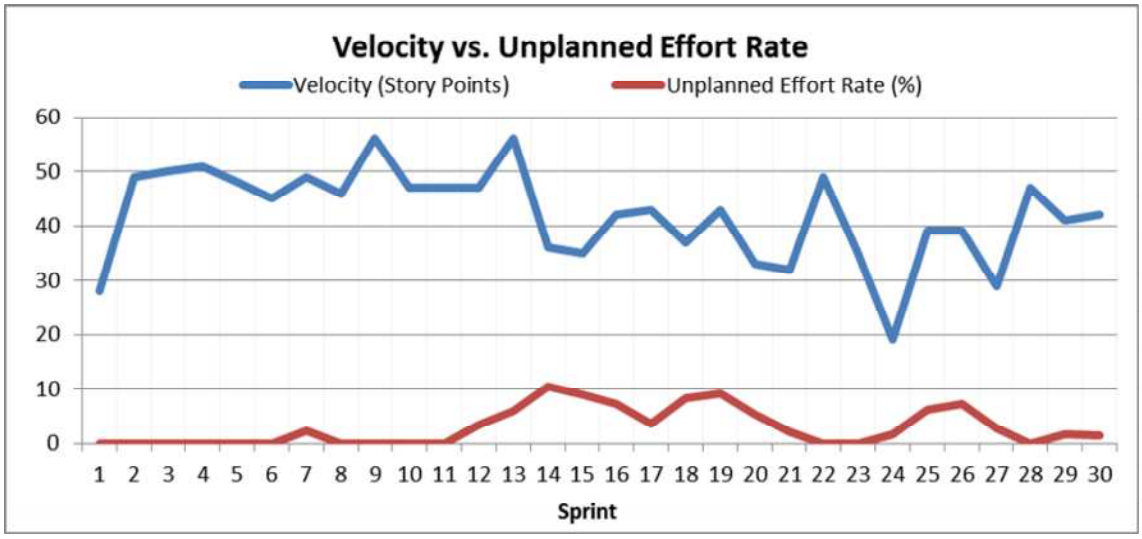


Figure 8. Velocity vs. Unplanned Effort Rate

Figure 9 shows effort rate for verification and test activities with respect to the total effort in each Sprint. In 10th Sprint, the effort rate is the highest because the team did test activities before releasing. The team discusses the quality activities effort rate in each sprint retrospective meeting.

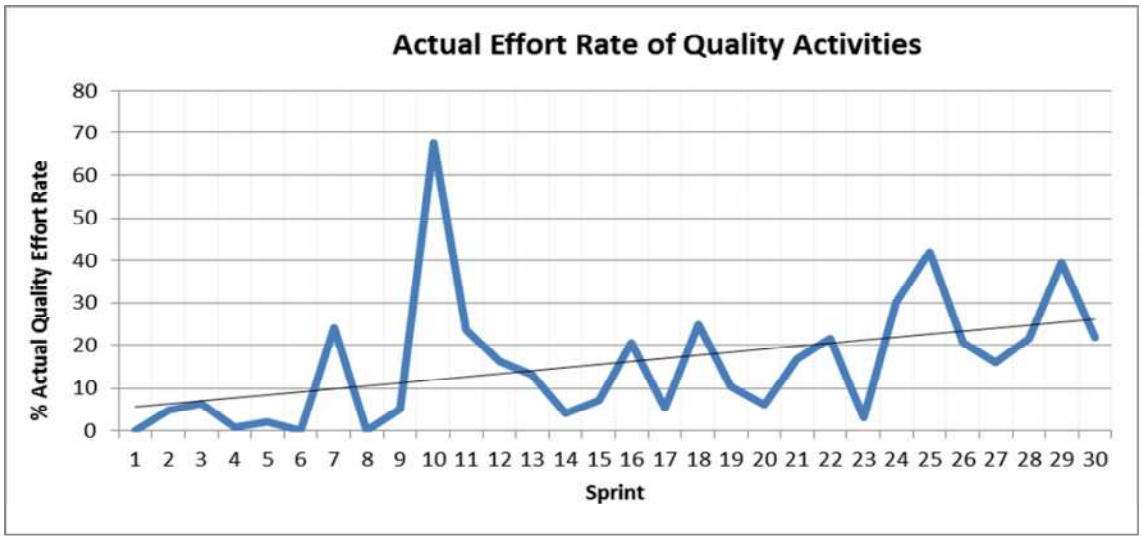


Figure 9. Actual Effort Rate of Quality Activities

The team can easily figure out which sub-components are most error-prone by inspecting Figure 10. They try to find out the root causes and make proactive decisions to decrease defect density of sub-components in retrospective meetings.

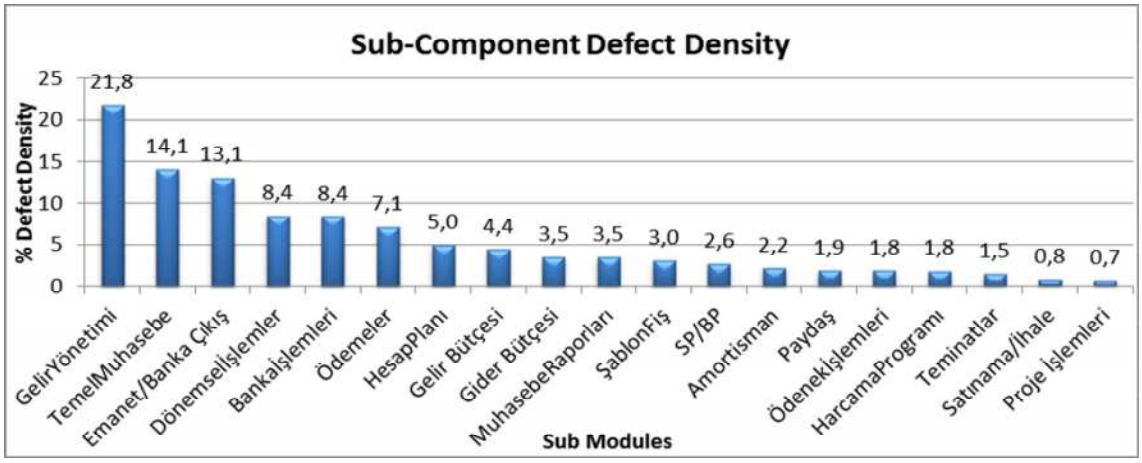


Figure 10. Sub-component Defect Density

4 Conclusion and Discussion

Scrum framework aims to deliver products productively with the highest possible value and quality [1]. In order to achieve this, sprint planning must be made accurately from the beginning of each sprint and agile teams must work productively till the end of each sprint without compromising the product's quality [9].

We observed that gathering correct data requires trust, respect, collaboration and motivation within the team. Team members should trust and respect each other and they should share successes and failures. They also should work as a team rather than as a group of individuals. Team members should be focused and interested in their work. After we are completely sure that the team has these essentials, we started to collect related data from the team. At the beginning of data collection, the team did not enter the data properly, however as they took advantages of this study considering product quality, team productivity and estimation capability in retrospective meetings, they handled entering data more precisely. Besides, measurement constructs in order to measure "estimation capability of the team"

were suggested by Scrum Master at the beginning of this study. When the team members realized that they were benefiting from the analysis, they started to participate in defining new measurement constructs for the team. Consequently, we started to measure “product quality” and “team productivity” with the team members’ suggestions. All these metrics and their analysis provided an effective and continuous improvement opportunity to the team. The challenge for the team was defining which statistical analysis can be input to retrospective meetings and how to analyze the collected data statistically for improving product quality, team productivity and estimation capability. Another challenging point of this study is to make analysis results of the sprint which is inspected in the retrospective meeting transparent to the team. At the beginning of this study, we could not analyze current sprint in a real time because of data preprocessing step. After we extracted data, we were processing and preparing data manually for further analysis. Afterwards, preprocessing step was automatized. As soon as a sprint was completed, in a couple of minutes extracted data can be easily analyzed by automatized tool and the team can inspect analysis results and take corrective actions for improvements in retrospective meetings immediately.

The results provided in Results section are investigated in every retrospective meeting by the team in order to take action in terms of increasing the possible value.

One of the most important indicators, which we discuss in the meetings to enhance the development team’s estimation capability, is “Consistency of Relative Estimation” as shown in Figure 3. This graph is derived from the actual effort and story points to show outliers that come from the false assignments and descriptive statistics such as median, quantiles and extreme values. To inspect this graph and take actions in retrospective meetings, this box-plot must be ensured of being accurate quantitatively. Therefore, we use Spearman Correlation Coefficient in order to measure the correlation between these two variables. At the end of each sprint, correlation coefficient is calculated containing the previous sprints data and results show that the correlation coefficient is strong and the value of 30th sprint is also strong and around 79%. In a retrospective meeting, consistency of relative estimation is inspected by finding out the outliers and highly skewed distribution according to coefficient of variation and then actions are taken for making more consistent relative estimation for the following sprints. For example, at the beginning of a sprint, as shown in Figure 3 PBI whose number is 14 is estimated 1 story point. Relative estimation of the PBI would be given 5 instead of 1. In retrospective meeting of the sprint the team identifies what is wrong or missing while making estimation of the PBI and creates a plan for potential improvements on PBIs that are similar to the considered PBI while making their estimation.

Another aim of the study is to increase the development team’s productivity. In this aspect, supporting metrics are “Team’s Actual Effort on Product”, “Team Velocity”, “Actual Effort for One Story Point”, “Innovation Rate” and “Velocity vs. Unplanned Effort Rate” as given in

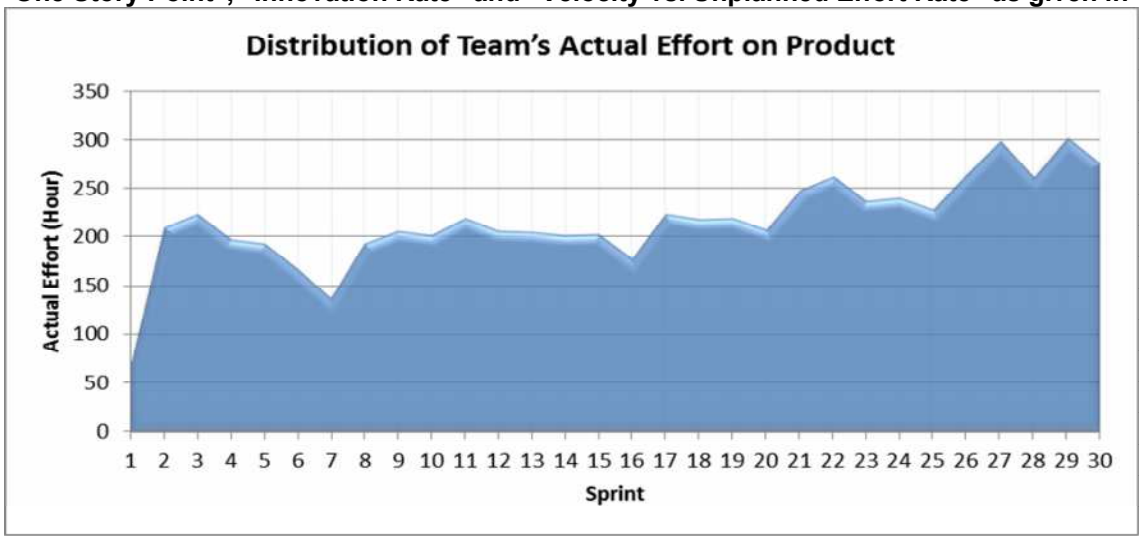


Figure 4, Figure 5, Figure 6, Figure 7 and Figure 8 respectively. If there is a decrease in actual effort on product, the team’s velocity and innovation rate and if there is an increase in actual effort for one story point and unplanned effort during a sprint, the team identifies corrective actions and creates a plan to increase the productivity by following these graphs in retrospective meetings.

Final aim of the study is not to compromise the product's quality while increasing the productivity. Many teams decrease the spent time of quality activities like reviewing, testing and validating in order to increase productivity. Thus the less product quality is, the less spent time of quality activities is. "*Actual Effort Rate of Quality Activities"* as given in Figure 9 is very useful indicator to keep the quality effort constant or increase it. Besides "*Sub-component Defect Density*" as shown in Figure 10 is very significant indicator to increase the product's quality. It shows the highest error-prone sub-components to make the team take corrective action to decrease their defect density in retrospective meetings.

This study is also shared to upper management in project review meetings and got the support and approval to spread similar analysis and process changes in other agile teams within the organization.

5 Literature

[1] K. Schwaber and J. Sutherland, "The Scrum Guide", July 2013.

[2] H. Kniberg, "Scrum and XP from Trenches", pp. 15-28, 2007.

[3] M. Cohn, "Agile Estimating and Planning", pp. 33-75, October 2005.

[4] M. Cohn, "User Stories Applied", March 2004.

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

[6] Padmanabhan, V., & Jerome, V. A. (n.d.). Maintaining Quality in Agile Environment. Retrieved April 09, 2015, from http://cmmiinstitute.com/sites/default/files/resource_asset/Maintaining Quality in Agile Environment_Jerome_SEPGNA14.pdf

[7] Axosoft. (n.d.). Scrum Success Metrics. Retrieved April 10, 2015, from http://www.scrumhub.com/wp-content/uploads/2014/10/scrumhub-implementing-scrum-part5.pdf

[8] Downey, S., & Sutherland, J. (2013). Scrum metrics for Hyperproductive Teams: How they fly like fighter aircraft. *Proceedings of the Annual Hawaii International Conference on System Sciences*, 4870–4878. doi:10.1109/HICSS.2013.471

[9] Williams, L., Brown, G., Meltzer, A., & Nagappan, N. (2011). Scrum + Engineering Practices: Experiences of Three Microsoft Teams. 2011 International Symposium on Empirical Software Engineering and Measurement, 463–471. doi:10.1109/ESEM.2011.65

6 Author CVs

Muhammed Emre PEKKAYA

Emre PEKKAYA works as an Agile Coach and Technical Lead at Software Technologies Research Institute (YTE) in Turkey. He has been developing products using agile frameworks, methods and practices since 2007. He worked in many positions like Software Developer, Business Analyst, Product Owner, Process Architect, e-Transformation Team Leader. He's got Professional Scrum Product Owner and Sun Certified Java Programmer (SCJP) certifications.

Onur ERDOGAN

Onur ERDOĞAN is currently employed in YTE (Software Technologies Research Institute) as a Measurement and Analysis Specialist in Quality and Process Management Unit. He has been working at YTE for 5 years. He got his Bachelor's degree in Department of Statistics. He is also Medical Informatics PhD Student in Middle East Technical University (METU) in Informatics Institute. His research

area is data mining, decision support systems and advanced statistical analysis.

Halime GÖK

Halime GÖK, computer engineer, works as a Quality and Process Management Unit Head in YT (Software Technologies Research Institute). She has been working at YTE for 5 years. She is particularly interested in software testing, quality management, process improvement and agile development.

Software quality measurement and evaluation framework for research and innovation projects

Marcin Wolski[1], Bartosz Walter[1,2], Patryk Promiński[1], and Szymon Kupiński[1]

[1]Poznan Supercomputing and Networking Center, Poznan, Poland
[2]Poznan University of Technology, Faculty of Computing, Poznan, Poland

Abstract (BW)

Software quality frameworks are helpful in evaluating the performance of software organizations and suggest improvement directions. In this paper we present a measurement framework supporting the quality evaluation in software products, developed and implemented within the GÉANT project. The framework is based on classical quality models, but it additionally embraces the point of view of an external funding entity, which is frequently present and plays an important role in research and innovation projects. The stakeholder's point of interest differs from that of a vendor and a user. He is usually interested in starting up the project and supporting it for a limited period of time, until it may become self-financing. The paper also provides selected results of evaluating two projects from the GÉANT ecosystem using the proposed framework.

Keywords

Software quality, measurements, quality models, evaluation, software process improvement, SPI

1 Introduction

1.1 Quality Models in Software Development

Implementing an effective approach to measurement-based quality evaluation is a significant challenge, particularly for large organizations [6]. Classical hierarchical models, like McCall's [1] or Boehm's [2], usually identify characteristics that manifest various dimensions of quality, propose instruments that collect necessary data and provide an interpretation for it. Such models are expected to provide stakeholders with adequate knowledge, reflecting both business requirements specific to the industry domain and technical details concerning the development and maintenance process. A manager expects to be able to monitor and control quality of the entire system to take corrective actions and improve processes, while a developer should be able to identify a component that seems faulty or defective. As a result, the models encompass two perspectives: *user-based*, which is focused on properties important for operating a software system, and *vendor-based*, which highlights the characteristics necessary to develop and maintain it. These perspectives interact, looking for a balance between two contradictory points of view.

However, some projects require a more complex model. For example, innovative research-oriented software projects are frequently ordered and subsidized by an external funding agency. Its perspective and objectives are different than for other participants: it is financing and stimulating the project in early stages of the system's lifecycle, and expects it would become self-reliant in a longer perspective. As a result, it should be recognized as an additional stakeholder who is not interested in achieving short-term financial or technological goals, but rather aims at verifying new concepts. Hence, the new stakeholder affects the existing interactions between the user and the supplier, introduces new ones, and enforces a change in the existing line-up.

1.2 Scope and objectives

In this paper we present a framework for measuring software quality that explicitly recognizes three stakeholders and their perspectives. This tri-party model is common for innovative, research-oriented projects development, and it significantly affects the equilibrium of forces present in such projects.
The framework is based on classical models both in terms of structure and the choice of quality characteristics and metrics. Additionally, it addresses specific features of software systems developed, operated and maintained within the GÉANT project. Availability of data and the processes implemented and present in GÉANT guided the process of selecting metrics and defining main characteristics.
The framework is an example of a hybrid approach which combines process- and product-related measurements to provide a broad perspective that enables comparing projects both internally (between various releases) and externally (among various projects).

2 Related work

The need for defining and measuring various dimensions of quality in software systems resulted in creating a number of models, aiming at evaluation, assurance and improvement of various qualitative characteristics [6]. The first public model for evaluating software quality was proposed by McCall in 1977 [1]. Its primary objective was to help developers construct better software by improving the development process and to bridge the gap between the development and users by outlining the shared points of interests. The McCall's model comprises 3 types of quality characteristics: *product revision*, understood as the system's ability to evolve *product transition* perceived as the system's ability to run in various environments, and *product operation* reflecting the way in which the system is operated. Further, the characteristics are decomposed into 11 factors and 23 quality criteria and a set of individ-

ual metrics which provide quantifiable means for measuring and controlling the properties of interest. The general concept behind the model is that a synthesis of quality factors should provide a complete perspective of the software quality.

A similar philosophy is present in the model proposed by Boehm [2]. It also defines a hierarchy of qualitative factors, evaluation criteria and metrics, but is more focused on evaluation of the subject system with a set of measurable attributes. At the top of the model there are three high level characteristics: as-is utility, maintainability and portability, which reflect main perspectives of a system's user. They are further supported by seven quality factors: portability, reliability, efficiency, usability, testability, understandability or flexibility, which constitute the expected qualities of the system. Subsequently, they are supported by a number of metrics, collecting the raw data. This structure defines each characteristic and its contribution to the overall system quality.

These two models laid a foundation for the ISO 9126 standard (or its successor, ISO 25010). It also promotes a hierarchical structure of qualitative properties and criteria.

Quamoco [9] is another hierarchical model of quality, in which product factors, like complexity, impact high-level quality aspects, e.g., maintainability. The factors are quantified by individual measures, collected with the use of instruments. Composition of data and its transition between levels is conducted with aggregation formulas (Evaluations).

Squale [5] is an example of an industry-oriented model of quality. Unlike other models, it introduces only two types of marks: (1*) low-level* measures*,* representing the plenitude of raw data acquired directly from the project, and (2) *high-level marks*, giving a more general insight to the software quality with respect to processes, components and factors. In order to produce a single grade, Squale uses the composition and aggregation approach: low-level metrics are transformed into normalized high-level marks (by composition) and are further aggregated over several components. On the top level, Squale presents the software quality three-fold: as practices, quality components and quality factors, which refer to various dimensions of quality.

An alternative approach for unifying the software quality evaluation is proposed by Lochmann at al. [4]. As the authors noticed, there exist different disciplines and process phases in software engineering developed to assure the quality of the produced software. These include checklists and guidelines, static code analysis tools, on-demand tests, ISO quality models, etc. These items are rather unconnected and not integrated. In order to provide a more complete perspective, the authors introduced a meta-model for software quality evaluation. It established common grounds for software developers, managers and other stakeholders, based on a set of activities and concepts related to software quality.

3 Overview of GÉANT ecosystem

GÉANT covers a series of EU-funded scientific projects to support the design, implementation, operation and maintenance of a pan-European high-capacity network for scientists. GÉANT projects bring together 31 National Research and Education Networks (NRENs) and 3 other organizational entities into one multi-domain community. The community constitutes cooperation of entities (organizations and individual users) that are open for collaboration and joint development of new solutions that will fit into their single-domain systems. Within the project, a number of software applications and tools have been developed, mainly focused on operating and maintaining the networking infrastructure.

The NRENs community is also actively involved in the collective software development activities, focused on delivering systems to provide network services within GÉANT. Software development in GÉANT has a number of distinctive features which have their origins in their network background, origin and target, but also in the characteristic features of software developers and users [4], which includes the following:

- *Software development commitment* – Software developers and NRENs initially declare the expected effort they are willing to contribute to the project. The effort is usually constant during the project, but it may change, depending on the internal schedule, preferences and commitments of the entities to other projects.

- *Service-orientation* – Software systems are usually used as an infrastructure for delivering network services to end-users. The recipient of the service is not always equivalent to a user of the component that supports that service.
- *Diversity of approaches to software development* – GÉANT imposes a general framework for software development, but particular teams are free to choose different styles and approaches to software development [4].
- *Expectation of high quality solutions* – Software products in GÉANT target demanding and knowledgeable users, maintainers and owners, and are based on highly developed advanced networks and systems. The operational environment, including people and existing systems, is conservative in accepting new solutions for daily use.
- *Limited end-user involvement* – End-users might not be actively involved in the software development throughout the entire lifecycle. They are often domain experts with limited time available to contribute to the project.
- *Dispersed location* – Development teams are geographically distributed [3].

4 Framework overview

The need for a unified software quality evaluation framework has been identified as the important objective for the GÉANT Service Validation and Testing process [11]. The process, which is a part of the service transition to the production phase, ensures that only quality-tested products can reach the production environment. Along with the process of maturation of the available software components in the GÉANT ecosystem, a need emerged to create a framework that would assist managers, developers and users in assessing the current state of the software quality or providing necessary security and risk-mitigation actions.

The primary objective for the framework is to provide a uniform model of tracking the progress within a single project, and between various projects across the entire GÉANT project. Additionally, the framework should be configurable in terms of selecting metrics and sources of data that is supplied to the model. It can help to strengthen the current approach to the software quality evaluation, which combines the automated testing and expert review with measurements as an objective method of control and assessment.

4.1 Structure

The framework inherits the structure and naming of some elements from other hierarchical models, e.g., McCall's [1] or Boehm's [2]. However, in recognition of the role of an external funding entity in the research and innovation projects, it extends the set of characteristics and proposes a number of metrics. The structure of the framework is composed of three tiers:

- **Tier one** defines the high-level quality characteristics that have been identified as crucial for the project's success. These characteristics are too generic to be directly measured.
- **Tier two** defines a number of intermediate quality factors that reflect different interpretations of the characteristics. The factors are specific enough to be measured.
- **Tier three** collects various metrics that measure the quality factors. Some of the metrics could also play the role of Key Performance Indicators (KPIs) used for monitoring the selected quality characteristic.

Tier one: high-level quality characteristics
These characteristics identify main quality dimensions that are important for the systems. Specifics of the GÉANT-originated projects need to be reflected by selection of high-level characteristics: some of the classical ones are absent, and the priority of some other is changed. As a result, the framework comprises five characteristics (listed along with respective quality factors):

A. Functionality: A.1. Suitability, A.2. Accuracy, A.3. Interoperability, A.4. Security

B. Maintainability: B.1. Stability, B.2. Analyzability, B.3. Changeability, B.4. Testability, B.5. Supportability
C. Marketability: C.1. Survivability, C.2. Market penetration, C.3. Market development, C.4. Market adaptability, C.5. Development velocity
D. Reliability: D.1. Maturity, D.2. Recoverability, D.3. Fault tolerance
E. Usability: E.1. Learnability, E.2. Understandability, E.3. Operability

Functionality, reliability and usability are the characteristics important for users, while maintainability is usually a concern for the vendor. Marketability is a new element, compared to the classical models. It reflects the ability of a software product or service to survive on the competitive markets, and represents the perspective of the external funding entity. It should be noted, however, that marketability is difficult to measure, and mostly it is based on subjective evaluation.

Tier two: quality factors
Quality factors provide interpretation to tier-one characteristics. They can be measured in different ways, but do not include direct measurement means by themselves. In total, the framework includes 21 quality factors for 5 characteristics.

Tier three: metrics
Tier three defines metrics that directly quantify the factors and convert them into measurable values. There are two groups of metrics: (1) external (business-oriented), and (2) internal (development-oriented). External metrics are concerned with the properties related with the use of the product and its market. They can be directly measured by the users and reflect their perception of quality. Internal metrics are related to the product's internal quality, or the development process attributes that are related to the product's quality.

4.2 Data sources

The choice of the metrics used in evaluation is guided by the availability of data. In case of GÉANT-related projects, some historical data is partially available, or it can be retrieved from archived artefacts on demand. Currently, the data is acquired from the following sources:

- **audits** – formal project reviews conducted by qualified and trained staff, following a predefined checklist;
- **code metrics** – several basic code metrics collected with by metrics calculators (e.g., SonarQube). They include both code quality metrics and repository metrics;
- **reports** of code smells and anomalies – reports generated by automated reviewing software tools, e.g., PMD and Checkstyle;
- **issue tracking data** – aggregated data from JIRA bug tracker, e.g., a number of open issues, dated issues, etc.
- **interviews** – information communicated informally by the involved stakeholders (programmers or users)

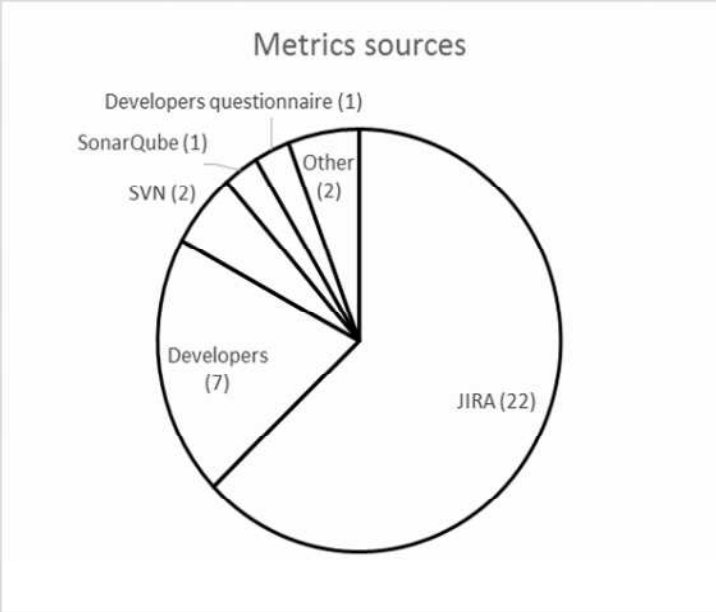


Figure 1 Metrics distribution based on of the input sources

The majority of metrics was collected from the issue tracker (JIRA) and indirectly from a developers team (interviews). The rest of metrics was gathered from past audit reports and supporting tools like code repository (SVN), code quality management platform (SonarQube).
The sources deliver data of diverse reliability, format and scale, so it needs to be transformed in order to produce usable information. Additionally, metrics can have a different scope. For some of them, the values are relatively easy to compare among various projects (like complexity), while other are project-specific and could be used to track the evolution of a single project (like size). Definitions of other need to be adjusted to every project (e.g., usability measures), which prevents them from being extrapolated or directly used in different projects.

Selected metrics play the role of Key Performance Indicators (KPIs): quantifiable measures that are correlated with the organization's objectives and help to track the progress. In most cases, they directly reflect business performance, but process-oriented KPIs are also in use. Some metrics definitions vary depending on the context. For example, the term "bug", commonly used in software development, could be mapped to a defect and a problem in ITIL-based processes. A defect describes the externally visible deviations from the normal operation, while the problem refers to the root cause for some defects. Therefore, the defect-related metrics support external, and the problem-related ones - internal KPIs.

4.3 Quantification and aggregation of metrics

As we mentioned previously, metrics are collected from various sources; their values can have different units and cannot be directly compared and interpreted. On the other hand, we need a normalized, measurement scale that could reflect the observations in a uniform way. We decided to normalize the values to range 0 to 1, based on historical data for the metric as a minimum and maximum values or the expert judgement, if the data is not available. Additionally, their monotonicity is adjusted, so that higher values represent a more positive evaluation.

Aggregation of data happens at two levels. First, a mean of normalized metrics within each quality factors is calculated. This gives a value representing the factor intensity. Next, the factors are aggregated at the Tier two level by weighting the mean with importance coefficients. They are currently proposed by experts to reflect the relative importance of particular factors within a project.

Finally, a single number from range {1; 5] expresses each of the Tier one characteristics. Plain aggregation data at Tier one could lead to negative effects, like compensation of data or information loss, so the overall quality of a system is described by a vector of values representing the characteristics. It

could be visualized by a web-chart diagram, which also facilitates comparing various revisions of a single project or various projects.

5 Validation

In the process of validating the framework two software products from the GÉANT ecosystem, codenamed System 1 and System 2, have been analysed (see Table 1). Both projects rely on the development infrastructure provided by GÉANT: SVN hosting, maven artefacts repository, continuous integration server and issues/bug tracking system.

Table 1 Description of System 1 and System 2

	System 1	System 2
Development since	June 2012	June 2013
Number of major releases	11	6
Active installations	8	3
Files	406	828
Lines of code	37126	68367
Languages	Java 88%, XML 7%	Java 85%, JSP 5% Groovy 4%
Total issues (bugs, improvements)	593	292
Open issues	192	92
Improvements	222	134
Open Improvements	71	51
Bugs	202	90
Open bugs	25	14

Table 2 includes selected results of the evaluation of System 1 and System 2. Due to limited space, we focus on two high-level characteristics: B. Maintainability and D. Reliability, along with associated sub-characteristics. The index column defines a high-level characteristics (B or D), a factor and a metric. The presented metric values have been normalized to the range 0..1, as described in Sec. 4.3, so the definitions may not fully correspond to the values.

Table 2. Normalized results of evaluation of System 1 and System 2

Index	Definition	System1	System2
B	MAINTAINABILITY	**0,33**	**0,63**
B.1.a	Ratio of reported failures before most recent maintenance activity	0,54	0,82
STABILITY [B.1]		**0,54**	**0,82**
B.2.a	Ratio of failures that have been correctly diagnosed	0,70	0,91
B.2.c	Programmer perception of analyzability	0,64	0,52
B.2.d	Relative number of defects discovered and reported during reviews/audits	0,34	-
ANALYSABILITY [B.2]		**0,56**	**0,72**
B.3.a	Mean time to implement a user-requested change	0,30	0,47
B.3.b	Code churn within last year	0,04	0,55
CHANGEABILITY [B.3]		**0,17**	**0,51**
B.4.b	Mean number of tests linked to a ticket in issue tracker	0,01	1,00
B.4.c	Ratio of re-opened tickets/issues	0,10	0,02
B.4.d	Statement coverage	0,00	0,45
TESTABILITY [B.4]		**0,04**	**0,49**
B.5.a	Number of distinct support channels (user fora, mailing lists, issue trackers)	0,20	0,60
B.5.b	Availability of guaranteed quality (SLA-based) support	0,00	0,00
B.5.c	Mean frequency of major releases (per year)	0,27	0,19

B.5.d	User perception of the quality support	-	-
SUPPORTABILITY [B.5]		**0,16**	**0,26**
D	RELIABILITY	**0,54**	**0,70**
D.1.b	Number of failures number relative to the number of test cases	0,80	0,99
D.1.c	Number of system failures reported during last year	0,00	0,33
D.1.d	Variance in the ratio of requested features to closed defects (in last year)	0,00	0,42
MATURITY [D.1]		**0,27**	**0,58**
D.2.a	Mean uptime of the system during the last year	0,66	0,66
D.2.b	Mean recovery time	0,68	0,90
D.2.c	Recent backup latency	0,00	0,00
RECOVERABILITY [D.2]		**0,45**	**0,52**
D.3.b	Mean number of distinct problems relative to the number of defects	0,99	0,99
D.3.c	Inverted ratio of highly erroneous classes (with 2 or more defects reported)	0,97	0,99
D.3.d	The ratio of mean uptime of the system relative to the number of defects	0,73	1,00
FAULT TOLERANCE [D.3]		**0,90**	**0,99**

In the Figure 2 we present aggregated results of evaluation of both analyzed systems, with respect to four high-level characteristics (excluding E. Usability). We can see the differences, in particular for B. Maintainability and D. Reliability. They indivate deficiencies of the systems and provide hints for improvement.

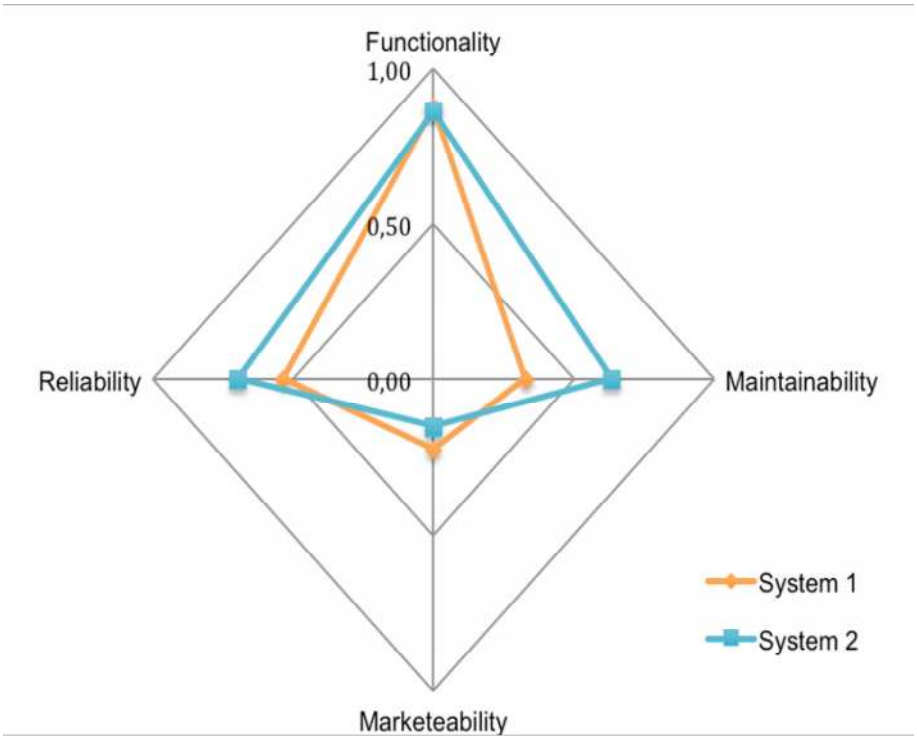


Figure 2 Aggregated results of evaluation for System 1 and System 2

6 Discussion

The presented results, albeit only partial, outlined several features specific for the applications developed in the networking community model (see Sec. 3). This, in turn, underlines the innovative character of developed solutions. Specifically, the users' involvement in the development process is relatively low, which is reflected by metrics B.3.a, B.4.c, and D.1.b. Consequently, in both projects handling improvements requests and incident reporting is an important issue, which is presented by metrics: B.2.a, B.3.a, and B.4.c.

However, a clear difference in the code coverage is visible (B.4.b and B.4.d). System 1 displays a negligible number of test cases, whereas System 2 is relatively well tested. Additionally, the variability

of the code base, reflected by the metric B.3.b, indicates that System 1 is changed less intensively than System 2. The values of metrics D.2.b and D.2.c, related to the recoverability of the system, indicate that operational demands for both systems had no major importance for the current state of their product life cycle.
It should be noted that the evaluation does not embrace the marketability. In our opinion, the collected measurements do not bring reliable and convincing data, and all of them have been acquired from interviews with programmers, which could be significantly biased. Additionally, at this stage of the systems lifecycle, the ability to survive on a market without subsidizing is not crucial.
The metrics distribution (Figure 1) confirms that the framework uses a hybrid approach to determine the software quality, combining process- and product-related measurements. We believe that such an approach offers a wider and more holistic view on software quality comparing to the existing models based on the software code metrics only [6].
Figure 2 presents the results of comparison of the two analyzed projects with respect to four high-level characteristics, excluding E. Usability. The differences between projects, visible in B. Maintainability and D. Reliability dimensions, inform about deficiencies and provide hints on the directions of improvement.
We find also that the result of evaluation relies strictly on the availability and quality of the collected data, e.g., reported issues, various surveys or results of the audits. In practice, the number of useful and valuable metrics may vary, so the framework should be configured prior to starting the evaluation process. Furthermore, defining a benchmark for software quality, based on approved standards or best practices, still constitutes a great challenge. Diversity of software products and their properties: size, lifetime, domain area, technology, etc., make the comparison across products particularly difficult.

7 Conclusions and future work

The software quality and measurement framework described in this paper is going to be introduced as a part of the GÉANT Service Validation and Testing process. It responds to the need of unifying the approach to assessment of software products across the GÉANT products. The framework supports the structured and standardized approach to quality control, and introduces more objective assessments, based on the various metrics and measurements, coming from different sources. The fact-based evaluation has been recognized as a factual and more neutral approach [7] and can extend the present evaluation based on automated tests and expert assessment [11].
The framework identifies an additional stakeholder: a funding entity, common in research and innovation projects, and highlights its potential impact on the software quality characteristics. In particular, we believe that the presence of this stakeholder also shifts the focus from users to the vendor in determining and achieving the expected software quality, and in playing an active role in the product design.
An interesting idea for a relative work would be to apply the developed framework in a similar ecosystem, where the funding entity also plays a significant role. Examples of such ecosystem are the start-ups: the world of entrepreneurs who usually base their business on innovation and carry on the business founded by external investors. However, the start-ups rely on fast growth and scalability of their business. In consequence such an approach can have a significant impact on the software quality characteristics like maintainability. Applying the model in this ecosystem would require creating new metric profiles taking into account such a tendency.
Early assessment of the framework performance shows that it is particularly useful in tracking the quality changes within individual projects, while comparison across projects brings ambiguous results. However, it helps in identifying quality areas that need improvements.
Plans for further development of the framework include several enhancements. For example, the review and prioritization of metrics could allow for defining various profiles of the framework, with metrics subsets classified to refer the project lifecycle (prototype, pilot, production), availability of the data sources (e.g. software audits) and other factors.

8 Acknowledgments

© GÉANT Limited on behalf of the GN4-1 project. The research leading to these results has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement No. 691567 (GN4-1).

Literature

[1] James McCall: Factors in Software Quality: Preliminary handbook on software quality for an acquisiton manager. Vol. 1-3. ADA049055. General Electric, Nov. 1977.

[2] Barry W. Boehm: Characteristics of software quality. Vol. 1/1978, NorthHolland Publishing Company, June 1978.

[3] Anthony R. Hendrickson et al.: "Virtual teams: technology and the workplace of the future". In: The Academy of Management Executive (1993-2005) 12.3 (1998), pp. 17-29.

[4] Klaus Lochmann et al.: "A unifying model for software quality". In: Proc. of the 8th International Workshop on Software Quality WoSQ'11. ACM, Szeged, 2011, pp. 3–10.

[5] Karine Mordal et al. Software quality metrics aggregation in industry. In: Journal of Software: Evolution and Process Vol.25/10 (2013), pp. 1117-1135.

[6] Rudolf Ferenc et al.: Evolving software systems. In: Tom Mens et al. (Eds.): Chap. Software Product Quality Models. Springer Berlin Heidelberg, 2014, pp. 65-100.

[7] Aiko Yamashita: Experiences from performing software quality evaluations via combining benchmark-based metrics analysis, software visualizations and expert assessment. Proc. of ICSME 2015. IEEE Press, 2015.

[8] Vilmos Bilicki et al.: Failure and success – how to move toward successful software development in Networking, TNC, Dublin, 2014.

[9] Stefan Wagner et al.: The Quamoco product quality modeling and assessment approach. Proc. of ICSE 2012, IEEE Press, pp. 1122-1142.

[10] Rensis Likert: The technique for the measurement of attitudes". Archives of Psychology, Vol. 140, pp. 1-55.

[11]Deliverable D8.1: "Service Validation and Testing Process". Accessed from http://www.geant.org/Projects/GEANT_Project_GN4-1/Documents/D8-1_Service-validation-and-testing-process.pdf (date)

About authors

Marcin Wolski has over 10 years of practice in database systems, scientific data management and enterprise application integration. Currently, he is in charge of the Software Quality Laboratory at PSNC. He has been involved in several national and international projects (e.g. GÉANT, Foodie) as a task leader and technical coordinator. In GEANT he is responsible for the quality management of production services, which includes the software code audits.

Bartosz Walter is an assistant professor at Poznań University of Technology. His main areas of interest include evaluation of source code quality and software refactoring. He also participated in a number of national and international research projects.

Patryk Promiński has been working on web applications development and implementation for over 5 years. During this time he has implemented several business processes for national and international companies. He has also contributed in creating educational portal for one of Polish universities. As a developer at PSNC for almost a year, he is involved in European projects (Foodie and GÉANT).

Szymon Kupiński is a system architect at PSNC. He has 10years experience in agile software development and quality assurance. He is working for PSNC since 2008 where he has gained experience also in software and usability testing. Currently he is involved in GÉANT Quality Calidation and Testing, and responsible for performance and usability testing.

Forming a European Innovation Cluster as a Think Tank and Knowledge Pool

Richard Messnarz[1], Andreas Riel[2], Gabriele Sauberer[3], Michael Reiner[4]

[1]ISCN LTD/GesmbH rmess@iscn.com
[2]EMIRAcle & ISCN Group andreas.riel@grenoble-inp.fr
[3]Termnet, gsauberer@termnet.org
[4]University of Applied Sciences Krems, Austria, michael.reiner@fh-krems.ac.at

Abstract

In the ECQA (European Certification and Qualification Association, www.ecqa.org) there are different Job Role Committee consortia which have developed training and certificates related to entrepreneurship and innovation. This paper elaborates an innovation and improvement strategy for Europe where the different consortia join forces to form a Europe wide alliance based on a pool of several modern certified job roles comprising more than 120 knowledge and training modules, 400 performance criteria and learning outcomes, as well as an online campus.

The strategy is to bring these different consortia and qualifications together and create an entrepreneurship and innovation portfolio available for universities and businesses across all the European member states. An ECQA certified terminology manager qualification approach will be used to create an ontology linking all these entrepreneurship qualifications to form a European knowledge pool.

Keywords

Innovation, Entrepreneurship, Pool of Knowledge, Terminology, ECQA

Published in: Springer Communications in Computer and Information Science (CCIS) vol. 663

Innovative Marketing in low-tech micro companies - Lessons learned from study projects

Prof. (FH) Mag. Michael Reiner, Prof. Dr. Christian Reimann, Elena Vitkauskaite M.A

Abstract

In current markets, there is a strong need for innovation (products, marketing, etc.) to set yourself apart from competitors. However, innovation is usually labor and capital intensive, and requires qualified employees with freedom for creative endeavours. Large or medium sized companies in most cases do have resources for that. High-tech micro companies (such as start-ups) appear in the market by creating innovation and therefore lack the pressure of existing companies. Low tech-micro companies on the other hand, work in established industries, have many competitors, and do not have time to do little else besides routine core activities of running their business. Therefore, these companies have a need for innovation but this need is practically impossible to fulfil, because of lack of capital, access to the required innovative technologies, the necessary experts (qualification) and general lack of time. Universities on the other hand are constantly looking for hands-on projects for students, preferring real life business problems. Moreover, universities have access to new technologies and the qualified experts. In this contribution, we are going to highlight the lessons learned from running an interdisciplinary international study project and to highlight the potential synergies of micro companies cooperating with universities to create marketing innovations. The study project presented was aiming to develop virtual reality marketing game prototypes for low-tech micro companies. Participants of the project were marketing students from Kaunas University of Technology (Lithuania), business administration students from IMC University of Applied Sciences Krems (Austria) and computer science students from University of Applied Sciences and Arts Dortmund (Germany). Oculus Rift game prototypes were successfully developed because of this project to two Lithuanian micro companies, one veterinarian practise and a tree nursery. Both micro companies now are having the opportunity to use these marketing innovations in cooperation with the universities. Project results were supported by feedback from the companies. Experts of the field as well as participating companies rated the transfer of innovation as good practise and a transferable model for innovation within low-tech micro companies.

Keywords

Innovation, VirtualReality, LessonsLearned, Projectteam, Communication

Published in: Springer Communications in Computer and Information Science (CCIS) vol. 663

User Orientation through Open Innovation and Customer Integration

Dimitrios Siakas[1], Kerstin Siakas[2], Richard Messnarz[3]

[1]Citec, Vaasa, Finland, dimitrios.siakas@citec.fi
[2]Alexander Technological Educational Institute of Thessaloniki, Department of Informatics, Greece, siaka@it.teithe.gr
[3]ISCN, Graz, Austria, rmess@iscn.com

Abstract

User participation and involvement in systems engineering, including innovation practices, are considered to be essential for value creation and competitive advantage. This study aims to make explicit the process of integrating the customer in the innovation process. Subsequently open innovation is investigated as a tool for integrating customers in the ideation stage of innovation. Open innovation, is a paradigm that assumes that companies can and should use external ideas in addition to internal ideas in order to create value. Open innovations also assume that internal ideas can be taken to market by external channels, outside the current business of the company. Online social networks are in particular suitable channels for creating value in the light of open innovation. Potential ways for gaining added business value through the use of social networking practices are investigated though an extensive conceptual literature review regarding customer integration in the innovation process and its relationship with potential value creation.

Keywords

Open Innovation, Crowdsourcing, Customer Integration, User Participation, Value Creation

Published in: Springer Communications in Computer and Information Science (CCIS) vol. 663

Proof: Maturity matters Higher maturity gives higher productivity

Jørn Johansen, Whitebox, Denmark, jj@whitebox.dk
Morten Korsaa, Whitebox, Denmark, mk@whitebox.dk

Abstract

You can discuss software development productivity from now until the sun burns out – it will be a discussion based on gut feelings and personal preferences for different solutions, **unless** you have objective measurements for functional size of the delivery and project maturity.

The basis for this paper is our experience from many years of CMMI® maturity assessments and Function Point counting for a very large number of software development projects in a large Danish bank development departments located in Denmark and India.

This paper documents how measurements can clarify and form the basis for solid conclusions, which again can bring in new extremely relevant and specific questions. These answers will bring insight, rationales and justify process improvements in the organization, and the entire industry when shared. And then we finally will know – higher maturity gives higher productivity.

Keywords

Productivity, Maturity, CMMI®, Function Point, Development Process, Process Improvement, Measurements.

1 Introduction

We have over decades heard about many kinds of second to none technologies, methods, techniques, which will bring a revolution into the product development department, and solve all problems and double the productivity.

All these good solutions are efficient and effective separately, but not holistic or profound enough to integrate with all the other initiatives in the organization – and solve the overall need for the business. In addition, how do they bring value into the development life cycle, and how much?

Management - be honest. You cannot see the “fish in the aquarium”. “The water is much too muddy” – to bring insight on productivity of the development department. If the development is shared between development centers in Denmark and India - “the water gets even muddier”.

Many enthusiastic persons have worked with productivity over time, e.g. Carol Dekkers [1] and Capers Jones [2]. But it has always been difficult to convince management in investment in the size measurements to enable productivity measurements.

We simply need to step back and agree on a set of objective measurements based on size and maturity to bring operational insight to management.

2 Context and Background

The company is a very large Danish bank with a IT development with 2000 people at several locations in Denmark and development centres offshore, e.g. in India.

The development process at the company has yearly been maturity assessed since 2007, first by external assessors, later by internal educated assessors (performed by the editors of this paper from Whitebox). Nearly all development projects, which deliver user functionality, are assessed. The number of assessed projects close to 60 per year, which is half of the total number of development projects.

The goal for these assessments is to:

1. Give management an overview of the maturity level in the development departments,
2. document the result of improvement initiatives,
3. identify process risks, and
4. identify trends in improvements and project risks.

Maturity is indirectly linked to finance – if you have a mature development organisation, statistics show that the amount of rework is lower and your level of reuse is higher than if you were a low maturity organisation. Most likely this is better business. However, the industry is lacking a more visible correlation between productivity and financial measures.

Productivity measurements have to be bases on size – in one or another way. Management in decided to use IFPUC Function Point (FP) to develop a much more directly link between business and development process capability. They used external help from consultants (the editors of this paper from Whitebox) for learning and design of the counting process. The first FP counting was performed in 2004 and was implemented for all development projects from 2007.

The combination of maturity assessment results and the FP counting, combined with other measurements, is documented in a yearly report to management addressing productivity for each department in the development organisation.

These results prove that maturity matters.

3 Problem Statement

The overall and typical problem in most of all companies is the lack of relevant and operational insight in the development process. In the 400+ assessments we have performed, the “Measurement and Analysis” process is less than 10% fulfilled.

Without these measurements and insight, it is difficult or impossible to take well-founded decisions and to follow up on these. You will never know if the decisions were beneficial or not.

Only with a certain level of insight in the development process, you are able to formulate new questions to bring more operational insight.

Example 1: As manager, you have a stomach feeling, the development productivity varies a lot across the development departments. Even worse, if the development projects includes involvement of development centre in another country (e.g. India). How would you as a manger take decisions related to resource management and budgets when the productivity varies significantly?

If you knew the productivity rate related to e.g. team size, mix of teams and competences, would it change the task you choose to outsource. Based on facts it is possible to lead the development by asking more concrete questions, which could be answered with new measurements – and the overall insight will bring considerable more substance to the decisions.

Then your sleep in the nights will be much more relaxed.

4 Approach

Relying on measurements falls natural to the company as with other financial companies, where measurements and excel sheets are part of the daily work. Many other companies find it less natural. Unfortunately.

The company has been using maturity assessments for more than 10 years. Once per year a large CMMI® assessment for all larger projects took place during a month – typical November. An assessment report including recommendations was presented for all the involved departments and top management. Based on this a number of improvement initiatives was defined for the next year.

The CMMI® score has not been progressing the last three years and is still between 2 and 3, depending on which division are assessed.

Function Point as size measurement was an important decision back in 2003, which included training in Function Point counting and development of a process for use in the organization. This work was done in collaboration with the consultants now form Whitebox.

The goal for starting the Function Point counting was to be able to evaluate the productivity across the development organization. There are several types of projects, systems and development organizations. They are distributed in two locations in Denmark and one in India. Management wanted to know how the productivity vary – why – and how can that be utilized?

Part of the Function Pointing was also registration of time used in the development projects and cost per hour. The project duration and team size was also logged.

With this information, combined with the maturity assessments the company was for the first time able to calculate e.g. the project productivity, and then relate that to maturity and to team size. It also became possible to calculate the development hours per Function Point and the price per Function Point.

4.1 Measurements

The analysis is based on about 60 projects, which are maturity assessed by the company internal assessors (educated and calibrated by external consultant) and Function Point counted by 3 internal

Function Point specialists. The measurements, which is basis for this paper, is from 2012 - 2014.

The light maturity assessments took place as a group interview with the project manager and a number of key persons in the project group. The assessor fulfilled a maturity excel based questionnaire. During the assessments, the group demonstrated a number of documents to clarify discussions. The time used for an assessment vary from three to four hours. The assessment was documented in the excel questionnaire and in notes.

The Function Point counting meeting was prepared by the project group – some questions were answered up front and documentation was ready for examination and discussions. The meeting normally took ½ a day. The Function Point counting was documented in notes together with a number of other data e.g.:

- Experience as Project Manager on a scale from one to three
- Competences in use of tools.
- Hours used in the projects.
- Number of persons in the teams.
- The duration of the project.
- Price per hour.

In the figure below shows, the price per Function Point related to the maturity. The selection of projects for maturity assessments and Function Point counting was not always overlapping. Therefor some of the projects, which is Function Point counted, does not have a maturity assessments. A presumption is, that the projects which not are assessed will spread out equally between the two groups.

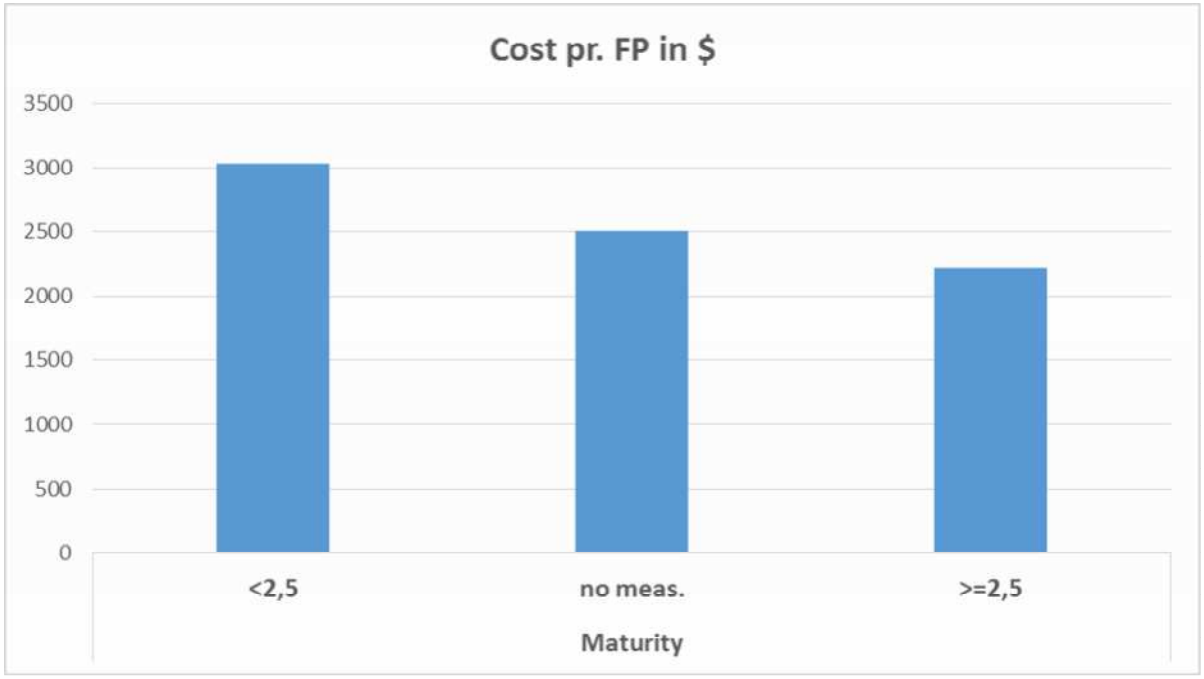


Figure 1. Price per FP in $ related to maturity

Project teams below maturity level 2,5 cost in average 27% more than project teams above 2,5. The three groups are comparable in number of projects. The previous two years the same analyse gave the respective results 24% and 23%, which demonstrate a stable result over years.

In average, the maturity is close to two for the entire development organization, which include projects in several locations, including in India.

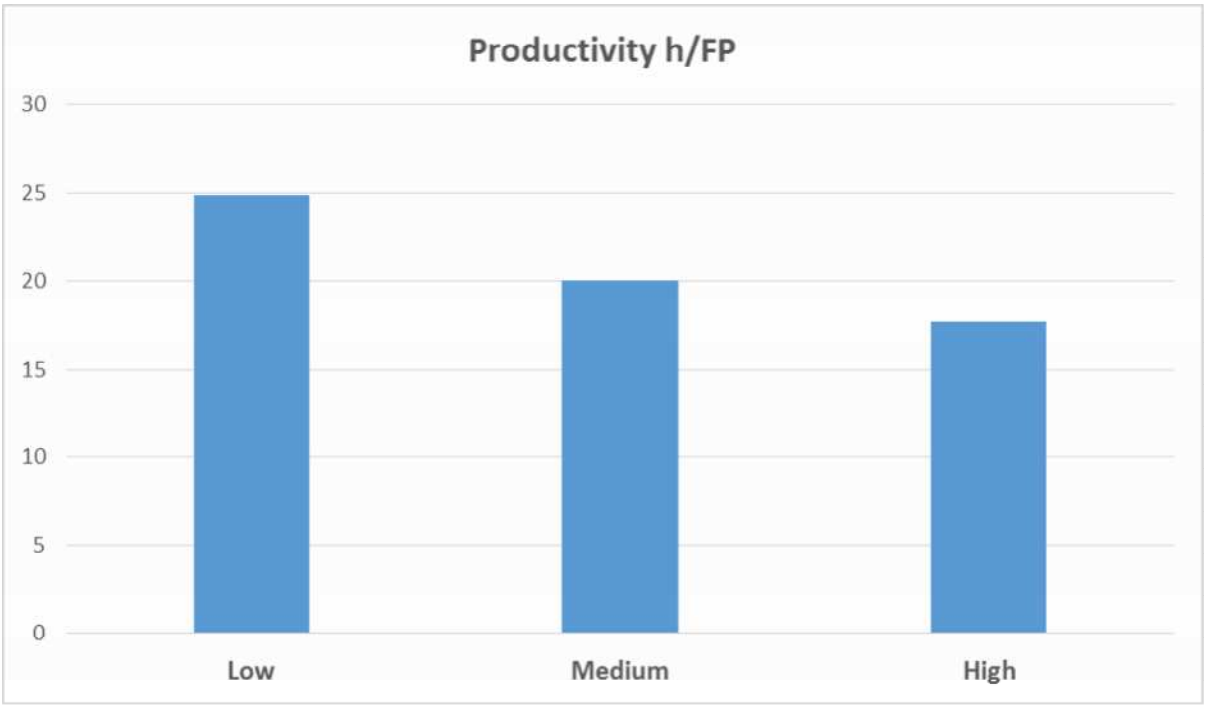


Figure 2. Productivity in Function Point per hour related to project management experience

Teams with an experienced Project Manager use 29% more hours to produce one function point than teams with a low experienced Project Manager to teams with a high experienced Project Manager. The previous two years the same analyse gave the respective results 20% and 23%, which also here demonstrate a stable result over years.

The clear conclusion is that Projects managed by most skilled Project Managers are more productive and cost efficient than projects managed by less experienced Project Managers.

The importance of having skilled project managers is probably more significant than the graph actually indicates. Some large projects in 2014 were handed over to highly skilled Project Managers when it became clear that it was necessary to secure Business releases. These projects had, at the time of replacement, already accumulated figures of low productivity and cost due to the high effort and cost at the time of rescue.

It could be interesting to correlate the project maturity with the experience in project management. We expect there to be a strong correlation.

5 Discussion and next steps

Productivity in the company is inherently linked to maturity. But there are other reasons for success or failure with a development project, e.g. the project management skills (which also correlate with maturity), tools experience, mix of team skills and so on. One clear observation is, that the more insight you get, the more you are able to define new questions to give you the next level of insight in an issue. E.g. if you find, that the productivity is less in the department in India than in departments in Denmark. Then you want to know why that is? Is it because the quality of the requirements are too low, or is it because the jobs in India is more complex, or another reason?

The focus on productivity and maturity has definitely improved the development organization. The following scatter diagram, where the cost per Function Point counted at the end of a development project from 2011 – 2014 shown as dots on a time axis, documents a significant improvement and the spread narrows in.

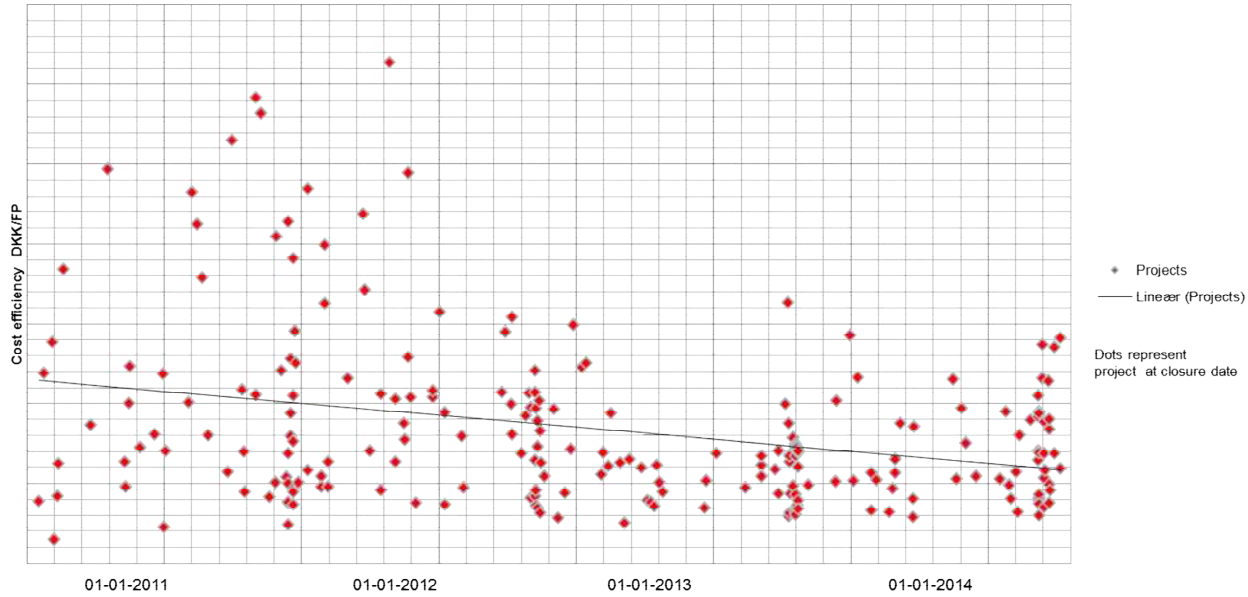


Figure 3. Cost efficiency spreads 2011 - 2014

During the time, the company has used Function Point counting and maturity assessments for the yearly productivity report, a new set of questions have included for the next years analyze.

Some of the other conclusions learned during the last 5 years in the company are:

- Productivity decreases if the team size is above seven people. The argument for this is the need for more administrative overhead and communication. Watts S. Humphrey [3] has the experience that teams between 4 and 8 people was the most efficient.
- The productivity decrease if the duration of the project increase. It seems the productivity decrease with 21%, if the projects duration is more than13 months and decrease 25% further, if the duration is more than 25 months. It is the same picture as productivity compared with size – the larger project, the lover productivity. However, the size of a project does not introduce the same risk on productivity as the duration does.
- Productivity decreases with higher percentage of participation Indian development in the projects. Consequently, it impacts the time to market. Cost efficiency

improves in most cases only when the percentage of Indian participants exceeds more than 50% as the resource model compensates for a lower productivity. The reason is to be found in decreased productivity due to the fact that dispersed teams are not as efficient as co-located teams. And many projects claim that there is too high collaboration /management overhead and still a considerate cost for support and knowledge transfer.

Many other experiences, are harvested over time in the company related to maturity assessments and Function Point counting. It is our hope that this work can and will continue and spread to other companies. It is obvious, that this work in the company has given an operational insight in productivity and maturity in the development organization. It has given management a basis for asking new information and get more insight via the analysis for the next year.

6 Conclusion

Over a time period of more than 10 years in the company has performed Function Point counting and maturity assessments.

It is clear, that the insight into these facts has given management is a strength in relation to improve the development department, and increase the productivity. It is not possible to identify which of the measures has introduced the most beneficial change. Both measures involve many persons during interview and by these interviews and discussions bring focus on improvement.

The most important experience is the development of the analysis itself. It started as two separate measures – project maturity and Function Point. Over time more and more data was included in the analysis (like internal cost per hour, number of team members …), and the reports initiated new questions and a need for more data.

The main conclusions are:

- Analysis show a clear correlation between high process maturity and high project productivity.
- Projects following defined processes are more productive and have a lower cost per function point than projects being less process mature.
- Experienced project managers have a positive influence on the productivity.

A structured approach to requirement management, planning and follow-up, risk handling, resource planning, supplier management etc. is affecting productivity with up to 30%.

Measurements give insight to management – and productivity is a measure which management can use.

Function Point measures, or other reliable size measures, is a requirement for productivity measures.

7 Literature

[1] Manfred Bundschuh and Carol Dekkers, The IT Measurement Compendium – Estimating and Benchmarking Success with Functional Size Measurements, Springer, 2008, ISBN 978-3-540-68187-8.

[2] Capers Jones, Applied Software Measurements – Global Analysis of Productivity and Quality, McGraw-Hill Companies, 2008, ISBN 978-0-07-150244-3.

[3] Watts S. Humphrey, Introduction to the Team Software Process, SEI Series in Software Engineering, 2000, ISBN 0-201-47719-X.

8 Author CV

Jørn Johansen, Whitebox, Denmark.

M.Sc.E.E. Expert in Process Improvement and Maturity Assessments. Partner & Director Process Improvement at Whitebox. Consultant in process assessments and process improvement. Has more than 35 years' experience in product development and project management.

Has worked with assessments for the last 22 years (CMMI®, ISO/IEC 15504 and ImprovAbility™) and performed more than 400 assessments.

Project manager for 3 very large Danish Reserch projects: Centre for Software Process Improvement project, Talent@IT project and SourceIT. The result of this work has e.g. been the ImprovAbility® model, which now is part of ISO/IEC 33014 - Guideline for process improvement. Also driver behind the SPI Manifesto.

Morten Korsaa, Whitebox, Denmark. Partner in Whitebox.

Mr Morten Korsaa has spent his carrier improving product development processes in a number of companies, both as employee and external consultant. He is a dedicated speaker and conveyor of professional development practices.

Mr Morten Korsaa takes pride in the holistic view on improving product development, including personal motivation, team dynamics, cultural characteristics, organisational characteristics, tools, process descriptions, change strategies, business goals, skills and competences.

Process Improving by Playing: Implementing Best Practices through Business Games

Antoni-Lluís Mesquida[1], Milos Jovanovic[1,2], Antònia Mas[1]

[1]University of the Balearic Islands, Department of Mathematics and Computer Science, Cra. de Valldemossa, km 7.5, E-07122 Palma de Mallorca, Spain
{antoni.mesquida, milos.jovanovic, antonia.mas}@uib.es
[2]University of Novi Sad, Faculty of Technical Sciences,
Trg Dositeja Obradovica 6, 21000 Novi Sad, Serbia

Abstract

This paper demonstrates the use of business games in process improvement. Research method for selecting games and their tailoring to support the activities proposed by process reference models is described. In this research two major process categories for software development companies are considered: project management and software implementation. This article focuses on project management process category (the ISO 21500 international standard for project management is taken as a reference) while software implementation is left for future research. Concrete application of one business game to the activities suggested by ISO 21500 is presented in the paper, thus showing the project management process improvement possibilities with business games.

Keywords

Process improvement; Business games; Project management; ISO 21500

Published in: Springer Communications in Computer and Information Science (CCIS) vol. 663

Infinite Demands and Constrained Methods - A Unified approach towards delivering Large Volume 'Quality' Automotive Software

AradhanaSivan, aradhana@tataelxsi.co.in
Leena Safeer, leenasafeer@tataelxsi.co.in
TataElxsi Limited
Bangalore, India

Abstract

Innovations in Automotive world in areas of comfort, safety and infotainment are increasing exponentially, leading to 'disruptive' evolutions of the functionality of vehicles. Competitiveness in the market place has led to the arrival of connected cars or 'autonomous' driving.

More than 85% of functionality in modern motor vehicles are now controlled by software, leading to a rise in number of Electronic Control Units (ECU's). While on one hand we have the growth of software size, and on the other, recalls are becoming a common practice in automotive space. Whether it is a minor or major volume recall, in general is an issue since (1) it impacts large number of vehicles and their users (2) it can lead to injury or death. These in turn negatively affects the brand and reputation of the vehicle manufacturers.

Taking into account the safety and reliability needs of Vehicle SW, it is crucial that systems and software within vehicles are designed and developed abiding with prevalent standards in the Automotive Industry vis a vis Automotive SPICE® and ISO 26262:2011. In this paper a unified approach for Software development with the integration of these as well as additional standards like Product quality ISO 25010:2011, and Software Testing ISO/IEC/IEEE 29119 is addressed.

Success of a product is largely dependent on the quality of the process used to produce it. Traditional development of Systems and Software had always consume larger cycle time however in today's world Agile practices take an upper hand due to the sheer advantage on schedule, cost and flexibility they bring in.

We at Tata Elxsi, have researched and integrated proven process improvement frameworks and international standards viz. Automotive SPICE® and ISO26262:2011, specific to automotive domain in conjunction with ISO/IEC 25010:2011and ISO/IEC/IEEE 29119 in an Agile development environment. This in itself is quite a challenge, but results are worth the effort!

The characteristics defined by ISO/IEC 25010:2011 support in developing the specification and evaluation of automotive systems and software in terms of functionality, performance, compatibility, usability, reliability, security, maintainability and portability. Application of the quality model in process areas defined by proven standards like Automotive SPICE® and ISO 26262:2011 ensures product functionality and functional safety. For each and every process

area, appropriate checks and balances aligned with ISO/IEC 25010:2011 quality model characteristics are introduced.

The paper aims to explain in detail the rationale for this integrated approach, the steps that have been implemented and associated results in terms of software quality.

Keywords

Automotive, Automotive SPICE®, ISO 26262:2011, ISO/IEC 25010:2011, ISO/IEC/IEEE 29119

Safety Analysis of a Hemodialysis Machine with S#

Johannes Leupolz, Axel Habermaier, and Wolfgang Reif
{leupolz, habermaier, reif}@isse.de

Institute for Software & Systems Engineering, University of Augsburg, Germany

Abstract

This paper reports our experiences of applying S# ("safety sharp") to model and analyze the case study "hemodialysis machine". The S# safety analysis approach focuses on the question what happens if we place a controller with correct software into an unreliable environment. To answer that question, the S# toolchain natively supports the Deductive Cause Consequence Analysis (DCCA), a fully automatic model checking-based safety analysis technique that determines all sets of component faults with the potential of causing a system hazard. To demonstrate our approach we created a model with a simplified controller of the hemodialysis machine and relevant parts of its environment and performed a safety analysis using DCCA.

Keywords

safety analysis, deductive cause consequence analysis, formal methods, model checking, embedded domain specific language, executable models, simulation, design tools and techniques, hemodialysis

1 Introduction

Classical software verification is focused on answering the question if the implementation of a piece of software conforms to a specification. This plays an important role in safety-critical domains like railway, automotive, aviation, and also medical devices. But another important aspect is to analyze what happens if a specification conforming controller is embedded into an unreliable environment. Failures of sensors or actuators may lead to situations where the state of the actual environment and the controller's internal data about the environment diverge. Such discrepancies result in degraded situations that could result in a hazard, an undesired situation with consequences like high follow-up costs or even the loss of lives.

We created S# to reveal which combinations of unexpected behavior in the environment and which faults on the component level (e.g. sensors failing) could result in a hazard. S# is an embedded Domain Specific Language (DSL) in C#, specially tailored for the analysis of safety critical systems. These models contain information about the system of interest (controllers, sensors, actuators … basically everything a system designer has direct influence on) and its environment. Furthermore, S# offers language features for modeling faults and fault effects intuitively. Compared to similar approaches, the models can be highly modularized in consequence of the adaption of C#'s object oriented concepts. Using Visual Studio, modelers can use the standard C# debugging tools to simulate their

models [10]. Our S# tool chain supports automatic analysis based on model checking with LTSMIN [11,2]. The formal analysis techniques are integrated into the tool chain and can be used without detailed knowledge of the underlying formalism, which makes it easier for engineers to focus on the models (see also [17]). Deductive Cause Consequence Analysis (DCCA), a fully automatic model checking-based safety analysis technique, can reveal detailed scenarios of how faults and unexpected behavior in the environment lead to a hazard. Furthermore, the S# tool chain can also be used in later phases of the development to validate the behavior of a real controller (implemented in C or C++) in the modeled environment [10] or used for the runtime analysis of self-organizing systems [8].

To demonstrate our approach we created a model with a simplified controller of the hemodialysis machine [5, 13] and relevant physical components like pumps and the dialyzer. To be able to adequately express the causal dependencies between different physical components it is necessary to also model the fluid flows which interconnect them. Furthermore, we explicitly modeled what happens when selected faults occur. Finally, we use the DCCA to calculate which combinations of faults can lead to the hazards “dialysis unsuccessful” and “blood entering the vein of the patient is contaminated”. S# and the complete model of the case study have been published on http://safetysharp.isse.de.

2 The Hemodialysis Machine Case Study

Due to metabolism, the human body creates metabolic waste products like urea and minerals. Usually, the kidneys are responsible for the removal of these waste products from the blood. When the kidneys fail, a hemodialysis machine can be used for this removal instead. These machines have a direct influence on the chemical composition of a patient's blood and thus form a safety critical system. This description of a hemodialysis machine here is based on a case study description written to evaluate formal languages and a training handbook for dialysis technicians [13,5].

A hemodialysis machine (see Figure 1) consists of three basic elements. The extracorporeal blood circuit (ECB), the dialyzer, and the dialyzing fluid delivery system (DFDS). Medical staff uses syringes to connect the artery and the vein of the patient to the ECB.

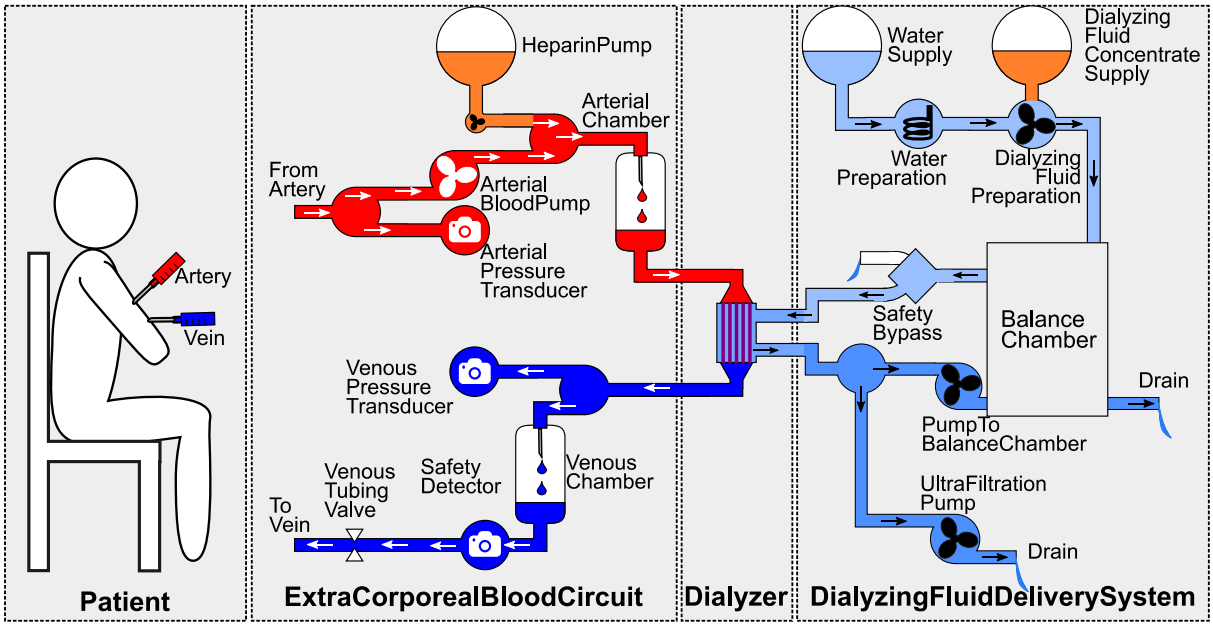


Figure 1: Hemodialysis Case Study

The main purpose of the ECB is to deliver the blood from the patient to the dialyzer and back again to the patient. A blood pump creates a suction to pump the patient's blood through the ECB. The heparin pump adds heparin into the patient's blood to prevent blood clotting. The arterial and venous pressure transducer deliver blood pressure values to allow their monitoring. The venous tubing valve enables another safety measure to prevent bad blood reentering the patient. Whenever the safety detector

detects contaminated blood or gas in the blood the venous tubing valve is closed and no blood can reenter the patient. The dialyzer itself is part of two fluid flows: A blood flow and a dialyzing fluid flow. Inside the dialyzer these two flows are separated by a semipermeable membrane. At the membrane small sized waste products go from the blood side to the dialyzing fluid side. Additionally big sized waste products can be removed from the blood side by creating a suction on the dialyzing fluid side (ultrafiltration). The incoming dialyzing fluid of the dialyzer is produced by the DFDS. The DFDS produces dialyzing fluid in several steps. The balance chamber acts as a buffer for dialyzing fluid. The safety bypass ensures that dialyzing fluid with the wrong temperature is piped to the drain instead of piping it into the dialyzer.

Figure 2 shows how the hemodialysis machine can be decomposed using the standard system modeling language SysML [7]. The complete **Specification** contains the **Patient** and the **HdMachine** (hemodialysis machine). It is necessary to include the patient in the model to be able to express hazards which concern the patient. The **HdMachine** itself consists of several parts, namely the **Dialyzer**, the **ControlSystem**, the **DialysingFluidDeliverySystem**, and the **ExtraCorporealBloodCircuit**. **DialyzingFluidDeliverySystem** and **ExtraCorporealBloodCircuit** themselves consist of several parts (e.g. **WaterSupply**). The **ControlSystem** itself only contains references to these subparts because they are *physically* not part of **ControlSystem**. The interconnection between parts is shown informally in Figure 1.

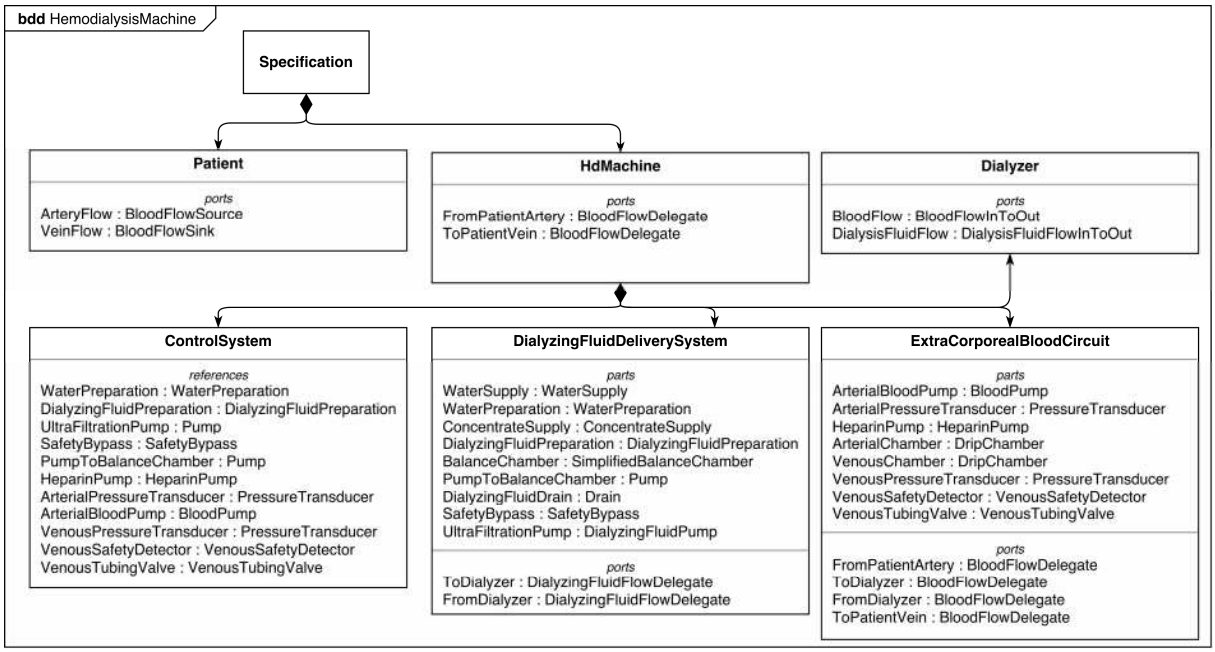


Figure 2: Structure of the complete hemodialysis machine as SysML Block Definition Diagram.

3 Controller Specification in S#

S# can be thought of as an executable, text-based extended subset of SysML. Currently, there exists no automatic translation from either language into the other. There are two established ways to model the behavior of controllers which are both directly supported by S#, namely state machines and sequential code. S# even allows to nest state machines with sequential code. The following listing shows an excerpt of our model of the control system of the hemodialysis machine.

```
public enum TherapyPhase {
   InitiationPhase,
   EndingPhase
}
public class ControlSystem : Component {
   public int TimeStepsLeft = 7; // hard code 7 time steps
   //references to components
```

```
    private readonly VenousSafetyDetector VenousSafetyDetector;
    private readonly VenousTubingValve VenousTubingValve;
    /* (other components and constructor left out for brevity) */
    public StateMachine<TherapyPhase> CurrentTherapyPhase = TherapyPhase.InitiationPhase;

    public void StepOfMainTherapy() { /* (left out for brevity) */ }
    public void ShutdownMotors() {
       VenousTubingValve.CloseValve();
       TimeStepsLeft = 0;
       ArterialBloodPump.SpeedOfMotor = 0;
       UltraFiltrationPump.PumpSpeed = 0;
       PumpToBalanceChamber.PumpSpeed = 0;
       DialyzingFluidPreparation.PumpSpeed = 0;
    }
    public override void Update() {
       CurrentTherapyPhase.Transition(
          from: TherapyPhase.InitiationPhase,
          to: TherapyPhase.InitiationPhase,
          guard: TimeStepsLeft>0 && !VenousSafetyDetector.DetectedGasOrContaminatedBlood,
          action: StepOfMainTherapy
       );
       CurrentTherapyPhase.Transition(
          from: TherapyPhase.InitiationPhase,
          to: TherapyPhase.EndingPhase,
          guard: TimeStepsLeft <= 0 || VenousSafetyDetector.DetectedGasOrContaminatedBlood,
          action: ShutdownMotors
       );
    }
}
```

First the possible therapy phases `InitiationPhase` and `EndingPhase` are declared in the enumeration `TherapyPhase`. In the following lines, the `ControlSystem` is declared. To express that the `ControlSystem` is a component it inherits from the S# class `Component`. The class contains the field `TimeStepsLeft` of type `int` with the initial value `7`. The S# DSL provides a generic state machine whose states can be determined by instantiating the generic type `StateMachine<>` with an enumeration of the desired states. One such state machine is `CurrentTherapyPhase`, whose states are determined by the enumeration `TherapyPhase`. The initial active state is set to `InitiationPhase`. The `ControlSystem` contains references to components which are not part of the `ControlSystem` itself like `VenousSafetyDetector` of type `VenousSafetyDetector`[1]. The concrete instances of the references are set in the constructor and is not shown in the excerpt for brevity. The reference is marked as `readonly` because it cannot be changed during model checking. The class `ControlSystem` also contains the methods `StepOfMainTherapy` and `ShutdownMotors`. The behavior defined in the methods use the previously declared references. Methods may be called from state machines or from other methods. Finally, the `Update` method of `ControlSystem` contains two transitions of the state machine `CurrentTherapyPhase`. The first transition is a reflexive transition from the initial state `InitiationPhase` to itself. A transition is usable when the active state of the state machine is the `from` state of the transition and the `guard` of the transition evaluates to true. Each time the `Update` method is called, the state machine selects an arbitrary usable transition, executes its `action`, and sets the target state as the next active state. Any method of the containing component can be selected as `action`.

The model of computation in S# consists of a series of system steps. In every system step the `Update`-methods of all components are executed once. This is repeated for every system step.

4 Modeling of Physical Fluid Flows in the Environment

The model of the control system would be sufficient to verify if the control system fulfills a particular specification. But to conduct a complete safety analysis we must also include non-controller components into the model on which the control system has only limited influence and which might have a

[1] When the C# compiler is able to distinguish whether an instance or a type is meant, an instance name can be equal to a previously defined type name.

negative impact on the safety of the overall system. To be able to model the non-controller components of the hemodialysis machine we need to be able to express fluid flows. Fluid flows obey complex physical laws. Fortunately, to create a qualitative model of many fluid flows it is not necessary to take account of all details of these laws. In this Section, we demonstrate how simple fluid flows can be modeled in S#. The implementation is generic and can be reused for any acyclic fluid flow where no backflow is possible i.e. the direction in which the fluids flow is fix. Finally, we present the model of the pump for dialyzing fluid used in the hemodialysis machine.

4.1 Flow Concept

The information of how flow components work and how flow components form a common flow should be separated. This increases the reuse of any flow component and might reveal problems in the model which originate from implicit assumptions about the flow. Thus, the flow between these components should only be declared by connecting these flow components. Furthermore, this property allows us to create independent flow components which could be combined arbitrarily. This corresponds to the design pattern *low coupling*.

Imagine, we want to model a fluid flow from a water supply to a drain. The exact amount of water which flows in each step is determined by a pump which sits in between the water supply and the drain. We want to treat the water supply, the pump, and the drain as separate independent flow components. The challenge is to create an adequate model (where the amount of fluid which is emitted by the water supply is determined by the pump) and adhere to the low coupling paradigm at the same time.

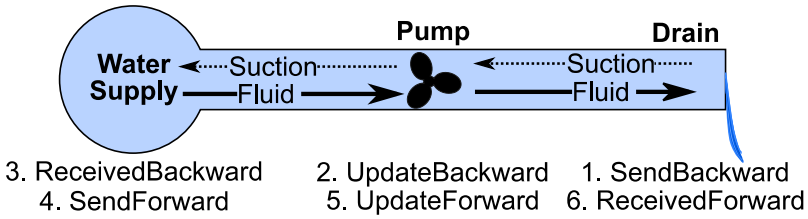


Figure 3: Example of a simple fluid flow. Fluid flows from the **WaterSupply** to the **Drain**. The amount of fluid is determined by the **Pump**.

The basic idea is to see a fluid flow as a bidirectional flow (see Figure 3). In the forward direction there is a flow of a specific amount of fluid. In the backward direction there is a suction. The suction determines the emitted fluid. In the example, the pump emits a suction on the water supply which is its predecessor in the flow. Now based on the incoming suction the water supply can determine the exact amount of fluid to emit into the direction of the pump. To calculate the amount of fluid which arrives at the drain the following sequence is executed: **1st step** The **Drain** notifies its predecessor that it is able to receive any amount of fluid (SendBackward=any amount); **2nd step** The **Pump** receives the suction information of the Drain. It ignores this information and notifies its predecessor that it wants a custom amount x of fluid (UpdateBackward=suction x); **3rd step** The **WaterSupply** receives the suction information of the **Pump** (ReceivedBackward=suction x); **4th step** The **WaterSupply** emits fluid. It knows the amount of fluid to emit from the previously received suction (SendForward=x units fluid); **5th step** The **Pump** forwards the received fluid to its successor (UpdateForward=x units fluid); **6th step** The **Drain** receives the fluid (ReceivedForward=x units fluid). Actually, the 2nd step makes the 1st step unnecessary, but adhering to this concept allows it to implement a pump in a similar way as the safety valve, drip chambers, or any other fluid component where a flow runs through.

4.2 Creating Simple Flows

The SysML Internal Block Diagram in Figure 4 depicts the structure how we realized the simple fluid flow of Figure 3 in S#. We created for each flow component a separate component declaration with its behavior. An instance of a component declaration is denoted by InstanceName:DeclarationName

where InstanceName is optional. Each component in the example contains a port with the name MainFlow which are of a different type, respectively. WaterSupply has a port of the type FlowSource with the property Outgoing; Drain a port of type FlowSink with the property Incoming; and Pump a port of type FlowInToOut with both properties. A flow is established by connecting the Outgoings with the Incomings. Connections are depicted by the dashed arrows.

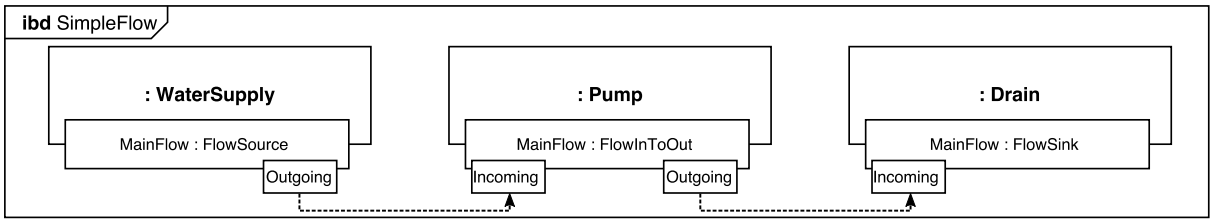


Figure 4: Simple fluid flow of Figure 3 as SysML Internal Block Diagram

The following listing presents an excerpt of the S# code of the simple fluid flow.

```
var supply = new WaterSupply();
var pump = new Pump();
var drain = new Drain();
var combinator = new DialyzingFluidFlowCombinator();
pump.PumpSpeed = 7;
combinator.ConnectOutWithIn(supply.MainFlow,pump.MainFlow);
combinator.ConnectOutWithIn(pump.MainFlow, drain.MainFlow);
combinator.CommitFlow();
```

The classes `WaterSupply`, `Pump`, and `Drain` contain generic “templates” how water supplies, pumps, and drains work respectively. These generic templates need to be instantiated to use them in a concrete flow. The flow components `supply`, `pump`, and `drain` are instantiated by calling the constructor of their corresponding classes, respectively. A `DialyzingFluidFlowCombinator` is instantiated which is used to establish the flow between the instances of the flow components.

4.3 Modeling a Pump

To conclude the modeling part we present the textual model of the Pump of the hemodialysis machine in the following listing. This pump is instantiated two times in the full model, namely as PumpToBalanceChamber and as UltraFiltrationPump (see Figure 2).

```
public class Pump : Component {
   public readonly FlowInToOut<DialyzingFluid,Suction> MainFlow;
   public int PumpSpeed = 0;
   public Pump() {
      MainFlow = new FlowInToOut<DialyzingFluid,Suction>();
      MainFlow.UpdateBackward = SetMainFlowSuction;
      MainFlow.UpdateForward = SetMainFlow;
   }

   public DialyzingFluid SetMainFlow(DialyzingFluid fromPredecessor) { return fromPredecessor; }
   public virtual Suction SetMainFlowSuction(Suction fromSuccessor) {
      Suction toPredecessor;
      toPredecessor.SuctionType = SuctionType.CustomSuction;
      toPredecessor.CustomSuctionValue = PumpSpeed;
      return toPredecessor;
   }
   public readonly Fault PumpDefect = new TransientFault();
   [FaultEffect(Fault = nameof(PumpDefect))]
   public class PumpDefectEffect : Pump {
      public override Suction SetMainFlowSuction(Suction fromSuccessor) {
         Suction toPredecessor;
         toPredecessor.SuctionType = SuctionType.CustomSuction;
         toPredecessor.CustomSuctionValue = 0;
         return toPredecessor;
```

```
        }
    }
}
```

The port `MainFlow` contains the two delegates `UpdateBackward` and `UpdateForward`, which determine the methods to call when a suction is received from the successor and a fluid element is received from the predecessor, respectively. In the constructor the methods `SetMainFlowSuction` and `SetMainFlow` are assigned to the two delegates. Every time the port receives a suction the local member `SetMainFlowSuction` is called. This method creates a suction on its predecessor in the size of `PumpSpeed`. Incoming fluids are just forwarded to the successor. Furthermore, the example shows how to declare faults and fault effects. Every fault might have several fault effects. The single fault effect `PumpDefectEffect` of the fault `PumpDefect` where the pump does not emit any suction on its predecessor overrides the original behavior of `SetMainFlowSuction`. A fault always overrides a correct method and it is even possible for more fault effects to override the same method. More information about faults can be found in our wiki on http://safetysharp.isse.de.

5 Safety Analysis of Models with nondeterministic Faults

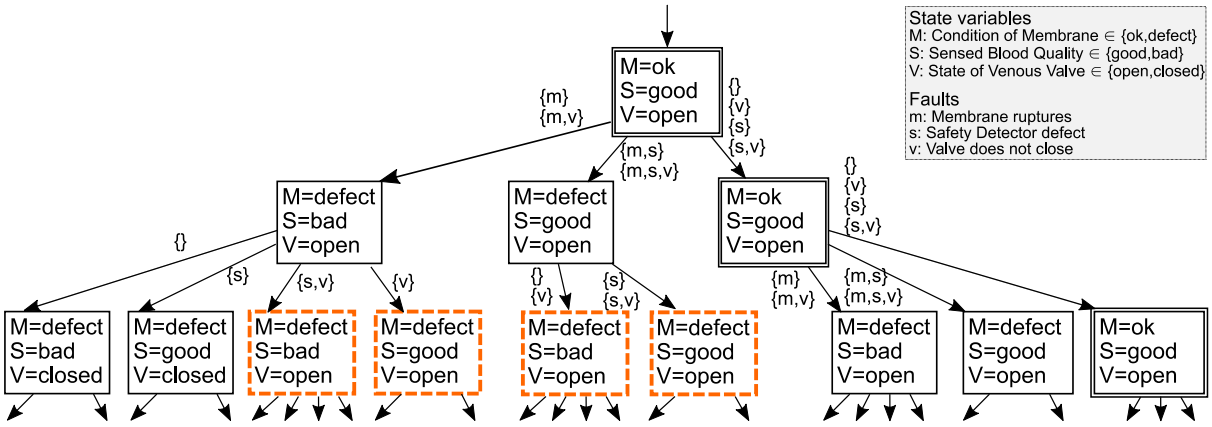


Figure 7: Simplified excerpt of the state space in the hemodialysis machine case study

The S# tool chain offers two techniques to analyze the possible behaviors of a model, namely simulation and exhaustive exploration of the state space by using model checking. The exhaustive exploration is especially useful in case studies with lots of nondeterministic behavior which is always the case with faults. An example of how a fault in one component can have an impact on the behavior of another component is shown in Figure 7. Each box represents a state consisting of the state variables denoted by the capital letters. The lower case letters represent different component faults which may occur during a transition to a successor state. It is even possible that several faults occur at the same time (e.g. {m,v}) or different combinations of faults lead to the same successor state (in these cases a transition has multiple labels). For instance, when the membrane ruptures ({m}) in the transition leaving the topmost state in figure 7, the membrane is defect in the upcoming state (M=defect). When otherwise no fault occurs ({}), the membrane in the upcoming state is intact (M=ok). An example of a good-natured trace is highlighted using double bordered boxes. In this case the membrane is not ruptured and no bad blood enters the vein of the patient. The state space also reveals reachable states where the venous valve was not closed after the membrane ruptured and thus, contaminated blood enters the patient (states indicated by dashed bordered boxes).

Even in the simplified excerpt some states have four successor states. Because of the vast number of reachable states, there is no way to inspect every possible behavior by hand. Thus, simulations and tests conducted manually can only check few traces. This fact makes model checking a useful technique to analyze the complete state space of S# models. Simulation complements model checking, because it is a valuable technique to find frequent bugs during the development in consequence of its

speed. Model checking can finally be used to find hidden bugs and gain more certainty.

6 Applying Deductive Cause Consequence Analysis

A *cut set* for a safety hazard H is a set of component faults which in combination might lead to this hazard. A *minimal cut set* is a cut set where the occurrence of every fault in the set is necessary to result in the hazard. This means that being able to prevent the occurrence of only one fault in each minimal cut set always prevents the hazard to occur. For instance, the membrane rupturing in combination with the broken valve is a minimal cut set for the hazard “blood entering the vein of the patient is contaminated”', as blood gets contaminated in the dialyzer and the valve is unable to prevent the contaminated blood from entering the vein of the patient. This minimal cut set is denoted as {DialyzerMembraneRupturesFault, ValveDoesNotClose}. If only one of these faults occurs the system is still safe. Thus, knowing the minimal cut sets also means to know the weak spots of a system. The DCCA is S#'s fully automated, model checking-based safety analysis technique which computes all minimal cut sets by individually checking all combinations of component faults, determining whether such a set does or does not have the potential to cause an occurrence of the hazard H.

For our evaluation we integrated 9 faults into the model of the hemodialysis machine:

- BloodPumpDefect: The blood pump of the extracorporeal blood circuit does not create suction.
- DialyzerMembraneRupturesFault: The membrane of the dialyzer ruptures. The dialyzing fluid inside the dialyzer gets contaminated by blood and the chemical composition of the blood inside the dialyzer gets disordered.
- DialyzingFluidPreparationPumpDefect: The pump which pumps the fresh dialyzing fluid to the balance chamber does not create any suction towards the water supply.
- SafetyBypassFault: The safety bypass cannot relay the dialyzing fluid into the drain anymore. So the bypass forwards all dialyzing fluid into the dialyzer, even if the dialyzing fluid does not meet the temperature constraints.
- WaterHeaterDefect: The water preparation does not heat the incoming water anymore.
- PumpToBalanceChamberDefect: The pump which pumps dialyzing fluid from the dialyzer back to the balance chamber is defected.
- SafetyDetectorDefect: The safety detector returns that the passing blood flow is all right even if it is contaminated.
- ValveDoesNotClose: The venous tubing valve cannot be closed anymore.
- UltrafiltrationPumpDefect: The pump for ultrafiltration is defected.

We analyzed the resulting model with the DCCA analysis for the hazards “dialysis unsuccessful” and “blood entering the vein of the patient is contaminated”. We performed S#'s DCCA analysis on the model with the integrated faults regarding both hazards. The results are summarized in the following table. The analysis has been executed on a desktop computer with a 4 core, 3.0 GHz Intel i5 and 16GB of RAM. The state space of the model consists of 362,485 states and 9,873,173 transitions. Every state has a size of 156 bytes. For every minimal cut set, a trace which leads to the hazard is automatically generated by S#, which can be replayed in a graphical visualization and therefore be validated together with domain experts.

Hazard	Minimal Cut Sets	Time
H1: Dialysis unsuccessful (blood is not cleaned and dialysis finished)	(1) {DialyzingFluidPreparationPumpDefect} (2) {WaterHeaterDefect} (3) {PumpToBalanceChamberDefect} (4) {UltrafiltrationPumpDefect} (5) {BloodPumpDefect} (6) {DialyzerMembraneRupturesFault}	2 sec
H2: Blood entering the vein of the patient is contaminated	(1) {SafetyBypassFault, WaterHeaterDefect} (2) {DialyzerMembraneRupturesFault, SafetyDetectorDefect } (3) {DialyzerMembraneRupturesFault, ValveDoesNotClose}	57 sec

7 Related Work

Other approaches also have features in their modeling language to describe faults and design alternatives. The Safety Analysis Modeling Language (SAML) is an extension of the PRISM input language (see [12]) to facilitate its application for the analysis of safety critical systems. SAML adds safety modeling features onto a state machine foundation. The SLIM language is another member of the state machine based approaches. It is derived from the standardized AADL language and also offers native fault modeling [4, 16]. By contrast, S# augments the industry standard language C# with fault modeling concepts. S# is more flexible than SLIM and SAML in the sense that it allows the behavior of a component to be modeled as either a state machine or as structured sequential code. Additionally, model structuring and composition in S# benefit from well-established object oriented design principles and language features. The inherent executability of S# models results in a unified approach for model simulation, testing, debugging, visualization as well as rigorous model checking. SYSTEMC extends C++ to make it possible to use C++ both as hardware description and high level modeling language. The approach to use a mature programming language is very similar to ours. However, S# was designed with formal analysis, failures and design exploration in mind and not so much for comprehensive hardware design [6]. MODELICA is a formalism which is very suited to model and simulate physical flows. It allows a sophisticated expression of physical laws inside the modeling language. However, it was designed rather for simulation than for model checking. It provides no native means to model faults and execute a sophisticated DCCA analysis [15].

8 Conclusion

In this paper we demonstrated how the S# tool chain can be used to model and analyze the impact of component faults in a hemodialysis machine. The possible faulty behavior in components and their interdependencies has made it essential to include the fluid flows into the model to allow an adequate analysis. It turned out that our approach is suitable for the safety analysis of this case study. The model was created in approximately 3 person weeks by an expert in S#, who was previously unfamiliar in the medical domain. Our analysis is by far not complete. With the help of domain experts, a future analysis could integrate further models of component faults and more phases of the dialysis. Furthermore, an interactive visualization could also help when a S# model needs to be validated with experts not familiar with formal methods. Our model on http://safetysharp.isse.de already contains a prototype of such a visualization. More details on S# can be found in our Wiki on http://safetysharp.isse.de.

Literature

1. Abdulla, P.A., Deneux, J., Stålmarck, G., Ågren, H., Åkerlund, O.: Designing Safe, Reliable Systems Using Scade. In: Leveraging Applications of Formal Methods, Lecture Notes in Computer Science, vol. 4313. Springer (2006)
2. Baier, C., Katoen, J.P.: Principles of Model Checking. MIT Press (2008)
3. Berry, G.: Scade: Synchronous design and validation of embedded control software. In: Next Generation Design and Verification Methodologies for Distributed Embedded Control Systems. Springer (2007)
4. Bozzano, M., Cimatti, A., Katoen, J.P., Nguyen, V.Y., Noll, T., Roveri, M.: The COMPASS Approach: Correctness, Modelling and Performability of Aerospace Systems. In: Computer Safety, Reliability, and Security. Springer (2009)
5. Curtis, J., Delaney, K., OKane, P., Roshto, B., Sweeney, J.: Hemodialysis devices. In: Core Curriculum for the Dialysis Technician: A Comprehensive Review of Hemodialysis, 4th edition. Medical Education Institute
6. Grötker, T., Liao, S., Martin, G., Swan, S.: System Design with SystemC. Springer(2002)

7. Object Management Group: OMG Systems Modeling Language (OMG SysML), Version 1.4 (2015), http://www.omg.org/spec/SysML/1.4/
8. Habermaier, A., Eberhardinger, B., Seebach, H., Leupolz, J., Reif, W.: Runtime Model-Based Safety Analysis of Self-Organizing Systems with S#. In: 2015 IEEE 9th Int. Conf. on Self-Adaptive and Self-Organizing Systems Workshops (SASOW 2015) (2015)
9. Habermaier, A., Güdemann, M., Ortmeier, F., Reif, W., Schellhorn, G.: The ForMoSA Approach to Qualitative and Quantitative Model-Based Safety Analysis. In: Railway Safety, Reliability, and Security. IGI Global (2012)
10. Habermaier, A., Leupolz, J., Reif, W.: Executable Specifications of Safety-Critical Systems with S#. In: Proc. of DCDS. IFAC (2015)
11. Kant, G., Laarman, A., Meijer, J., van de Pol, J., Blom, S., van Dijk, T.: LTSmin: High-performance language-independent model checking. In: Tools and Algorithms for the Construction and Analysis of Systems - 21st International Conference, TACAS 2015 (2015)
12. Kwiatkowska, M., Norman, G., Parker, D.: PRISM 4.0: Verification of probabilistic real-time systems. In: Proc. 23rd International Conference on Computer Aided Verification (CAV'11). LNCS, vol. 6806. Springer (2011)
13. Mashkoor, A.: The hemodialysis machine case study (2015)
14. Microsoft Corporation: C# Language Specification Version 5.0 (2015), https://msdn.microsoft.com/en-us/library/ms228593.aspx
15. Modelica Association: Modelica - A Unified Object-Oriented Language for Systems Modeling, Language Specification, Version 3.3 (2014)
16. Noll, T.: Safety, dependability and performance analysis of aerospace systems. In: Formal Techniques for Safety-Critical Systems: Third International Workshop, FTSCS 2014. Communications in Computer and Information Science, Springer (2015)
17. Visser, W., Dwyer, M., Whalen, M.: The hidden models of model checking. Software & Systems Modeling 11(4) (2012)

9 Author CVs

Dipl. Inf. Johannes Leupolz

is a PhD student at the Institute for Software and Systems Engineering, University of Augsburg, Germany. His main research interest is model based analysis of safety critical systems. He is one of the main developers of S#

M. Sc. Axel Habermaier

is a PhD student at the Institute for Software and Systems Engineering, University of Augsburg, Germany. His main research interest is model based analysis of safety critical systems. He is one of the main developers of S#

Prof. Dr. Wolfgang Reif

is director of the Institute for Software and Systems Engineering and leads the chair for Software Engineering at the University of Augsburg. Furthermore, he is vice president for technology and innovation at the University of Augsburg. His research interests are among others formal methods, safety, security, self-organizing systems, and software for mechatronics and robotics

A Preliminary Systematic Literature Review of the use of Formal Methods in Medical Software Systems[1]

Silvia Bonfanti, Department of Economics and Technology Management, Information Technology and Production, Università degli Studi di Bergamo, Italy, silvia.bonfanti@unibg.it
Angelo Gargantini, Department of Economics and Technology Management, Information Technology and Production, Università degli Studi di Bergamo, Italy, angelo.gargantini@unibg.it
Atif Mashkoor, Software Competence Center Hagenberg GmbH, Austria atif.mashkoor@scch.at

Abstract

The use of formal methods is often recommended to guarantee the provision of necessary services and to assess the correctness of critical properties, such as safety, security and reliability, in medical and healthcare systems. Several research groups have proposed and applied formal methods related techniques to the design and development of medical software and systems. However, a systematic and inclusive survey with some form of analysis is still missing in this domain. For this reason, we have collected the relevant literature on the use of formal methods to the modeling, design, development, verification and validation of medical software systems. We apply the well-known systematic literature review technique and we run several queries in order to obtain information that can be useful for people working in this area. We present some research questions and the data answering these questions. We also discuss some limitations of the adopted approach and how to address these issues in order to have a comprehensive survey.

Keywords

Systematic Literature Review, Formal Methods, Medical Software, Medical Device, Validation, Verification, Certification

[1] The research reported in this paper has been partly supported by the Austrian Ministry for Transport, Innovation and Technology, the Federal Ministry of Science, Research and Economy, and the Province of Upper Austria in the frame of the COMET center SCCH.

1 Introduction

In modern medical devices, human safety depends upon the correct operation of software controlling the device: software malfunctioning can cause injuries to, or even the death of, patients. A crucial issue is how to guarantee that the medical software has all the qualities (e.g., safety, security, liveness, and utility) expected for critical components. One way to improve and assess software quality as suggested by the literature is to use formal methods or in general rigorous methods for the design, validation, and verification of medical software. Medical standards and certification procedures, that use formal approaches, have been proposed and taken into consideration during the development, but some research questions still remain open. With this paper, we try to give a preliminary overview of the research literature in this field. The goal is twofold: 1) to provide guidance to researchers starting to work on this topic 2) to assess the state of the art which is more useful for researchers already working on this subject.

We have applied a Systematic Literature Review (SLR) process (Kitchenham, et al., 2009) to the topic of rigorous methods for designing and validation of medical software and systems. The goals of this process are (1) to gather a sufficient number of relevant articles, (2) to perform a series of analyses, and (3) to publish the results of the findings to allow researchers to browse in the collected data. This activity follows a systematic process to avoid possible biases, inclusive in order to include as much information as possible, but at the same time capable of identifying only relevant papers. In section 2, we explain the activities we performed in order to reach this first goal, the data source we use, and the technologies and tools we adopted. After that, we perform several queries over the data we collected, in order to extract useful information. The queries are driven by a series of research questions (RQ1 to RQ5). In RQ1 and RQ2, we are interested in providing some evidence of the publication trends in this field, to objectively measure the interest in the scientific community during the last 30 years. In RQ3 we are interested in knowing which are the preferred journals and conferences in these topics. In RQ4 we try to give an insight on how the community is distributed, by looking on the number of papers among all the authors. We also perform a preliminary study regarding the impact of the research in this area. Assuming impact as a measure of the number of citations, we perform several queries about the significance of the articles. RQ5 identifies the publications that have had most impact in this research area. This information can be useful, for example, for PhD students who would like to know: which are the most cited papers they must be aware of?

To the best of our knowledge, there are no systematic reviews on the literature of formal methods in this field. In (Xinxin et al., 2009) the authors review the literature on the use of formal methods on medical terminologies, and not software itself.

Although we encountered several problems and limitations of our technique, we were able to collect a great number of papers (more than 200) to make our quantitative analysis meaningful. Overall, we have found out that the research area is still growing in terms of number of publications. The presence of papers in highly ranked journal witnesses that the scientific community is aware of the importance of the work done in this field. However, the contributions seem rather extemporary, since most of the papers have no impact (in terms of citations) and most authors have published only one paper in this field.

2 The SLR Process

We apply the SLR process to rigorous methods in medical software systems following the guidelines presented in (Kitchenham, et al., 2009) with some changes to fit our goal.

Figure 1 shows the process applied. As a first step, we chose Scopus[2] to extract publications. Scopus is the largest database owned by Elsevier, it contains scientific journals, books and conference proceedings. There are more than 60 million records, over 21.500 peer-reviewed journals, over 360 trade

[2] http://www.scopus.com

publications, 7.2 million conference papers and 27 million patents. There are 5.000 articles-in-press from international publishers including Cambridge University Press, the Institute of Electrical and Electronics Engineers (IEEE), Nature Publishing Group, Springer, Wiley-Blackwell. It includes more than 113.000 books that will increase by 10.000 each year. The second step is the definition of search terms into the database. Scopus allows the user to perform different type of search, by title, by keywords, by authors or advanced search obtained by queries[3]. The research performed takes into account titles and keywords of the papers:

- TITLE(medical) AND (TITLE(software) OR TITLE(device*)) AND (TITLE(validation) OR TITLE(verification) OR TITLE(certification))
- TITLE(medical) AND (TITLE(software) OR TITLE(device*)) AND TITLE("formal methods")
- TITLE("formal method*") AND TITLE(medical)
- KEY(medical) AND (KEY(software) OR KEY(device*)) AND (KEY(validation) OR KEY(verification) OR KEY(certification)) AND KEY(formal)
- KEY(medical) AND (KEY(software) OR KEY(device*)) AND (KEY(validation) OR KEY(verification) OR KEY(certification)) AND KEY("formal method*")
- KEY(medical) AND (KEY(software) OR KEY(device*)) AND KEY("formal method*")
- KEY("formal method*") AND KEY(medical)

We obtained 238 papers[4]. We used Scopus functionality to merge the results of each search and then we downloaded the RIS[5] file containing all available papers information (e.g. title, authors and citations). After, we imported the RIS file into SciMAT[6] (Science Mapping Analysis Software Tool) (Cobo, et al., 2012). If users have more than one RIS file SciMAT allows deleting duplicate. SciMAT is open source tool and performs science mapping analysis. This tool is divided into three modules: 1. management of the knowledge base such as authors, keywords, references and citations; 2. carrying out the science mapping analysis; 3. visualization of generated results and maps. Before performing the analysis and depicting the results (see Section 3), we applied the following data pre-processing activities:

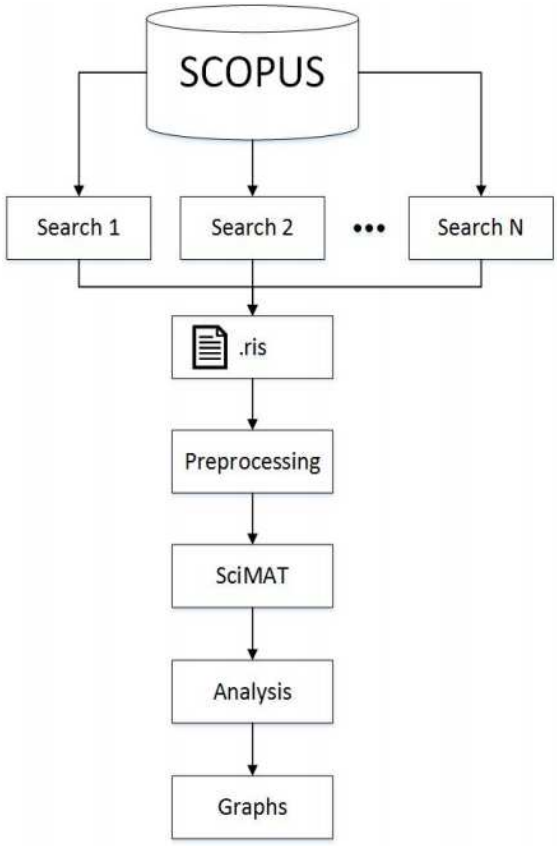


Figure 1:The applied SLR process

- We merged the authors written in a different way (i.e., with one or more names missing, extra dots or any other symbol between name and surname). SciMAT functionality finds similar authors by Levenshtein distance. The user set a number N that represents the number of deletions, insertions or substitutions required to transform a string into another one. In this set of authors, we set N equals to one and to two and we found some of duplicate authors.
- We merged the same keyword written in a different way (i.e., plurals, with symbols/spaces between words, with wrong letters inside words). SciMAT tool automatically finds and merges similar words by plurals. The search by Levenshtein distance is available to find similar words.

[3] Symbols in queries: use "quotation marks" to search for a phrase; the * symbol will replace multiple characters
[4] the list of publications is available at http://cs.unibg.it/bonfanti/EuroAsiaSPI2016SLR/ScopusResults.ris
[5] RIS is a file format developed by Research Information Systems, Incorporated to enable citation programs to exchange data
[6] http://sci2s.ugr.es/scimat/

3 Analysis and Results

In this section, we analyse the results by answering to a set of research questions (**RQ**).

RQ1: Which is the trend of publications?

As a first question, we wanted to observe the trend of publications about formal methods applied in medical field. We analysed the number of publications from 1982 (the year of the oldest publication we found) until 2015 (we did not consider 2016 since this year is not finished yet). As shown in Figure 2, until 2006 the number of paper was equal or less than five, except for 2001 and 2003. From 2006, the number of papers has started to grow until 2011. In the last four years, the number of publications has decreased (less than 20 publications per year) except in 2014 in which the number of publications has reached the maximum value over all years. The behaviour in the recent years should be taken with caution, probably because the updating of publications is not finished yet.

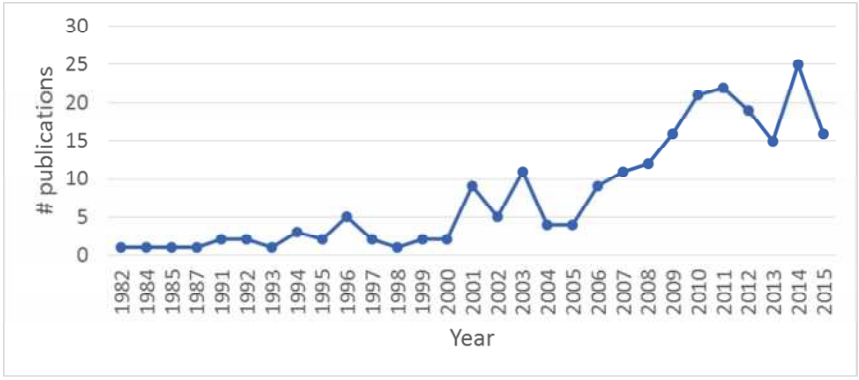


Figure 2: Publications per years

RQ2: Are there more publications in Journals or Conferences?

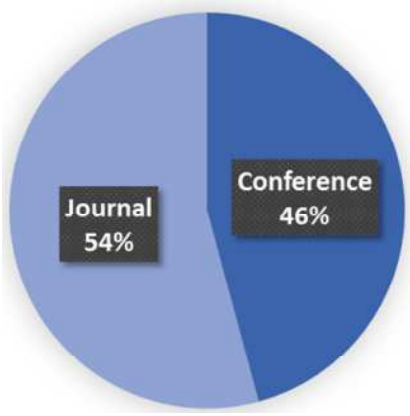


Figure 3: Journal or Conference

In Figure 3, the pie chart shows the percentage of publications in journals and in conferences. The number of publications is quite similar, but it is greater in journals (54%) compared to conferences (46%).

In Figure 4 the trend of the number of publications in journals and in conferences is depicted. For all years (except in 1984), the number of publications in journals is always greater than the number of publications in conferences and their behaviour is always the same (when the number of publications in journals grows, the number of publications in conferences grows as well).

In medical field, the number of journals is bigger than the number of conferences; this difference could be the motivation of the major number of publications in journals.

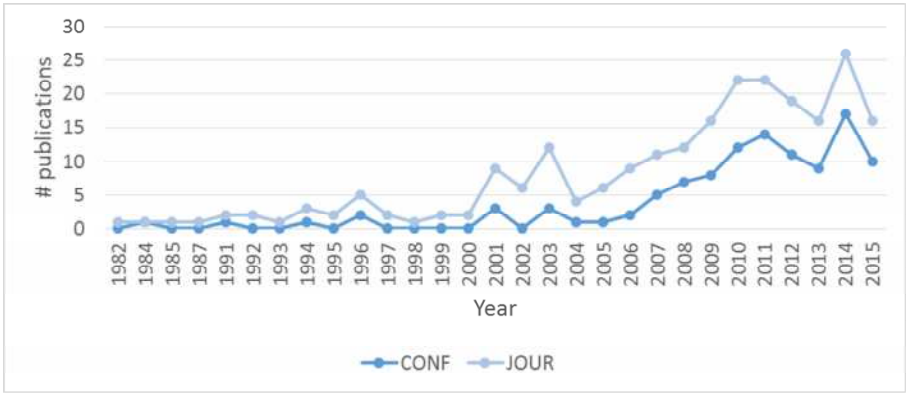


Figure 4: Publications in Journal/Conference per year

RQ3: Which are the most important journals/conferences?

Table 1 shows a classification of most important journals and conferences based on number of citations.

For each journal, we analysed the SCImago Rank (Arencibia-Jorge, et al., 2008), which measures the scientific influence of journals. This parameter assumes four values: Q1 (the highest value), Q2, Q3 and Q4 (the lower value). All journals have the highest value; this means that this topic has high importance in prestigious journals.

Name	*JOUR/ CONF*	*SCImago Rank*	*# citations*	*# publications*
Artificial Intelligence in Medicine	JOUR	Q1	255	3
International Journal of Medical Informatics	JOUR	Q1	136	2
Lecture Notes in Computer Science	CONF	N/A	95	35
Proceedings of the IEEE	JOUR	Q1	36	1
IEEE Transactions on Software Engineering	JOUR	Q1	36	1
Journal of Biomedical Informatics	JOUR	Q1	33	3
Proceedings of the 8th ACM International Conference on Embedded Software, EMSOFT'08	CONF	N/A	26	1
Computer	JOUR	Q1	26	1
Proceedings of the ACM SIGCHI Symposium on Engineering Interactive Computing Systems	CONF	N/A	26	2
Annals of Internal Medicine	JOUR	Q1	22	1
IEEE Robotics and Automation Magazine	JOUR	Q1	21	1
Biomedical Optics Express	JOUR	Q1	20	1
Journal of Pharmaceutical and Biomedical Analysis	JOUR	Q1	20	1
Joint Workshop on High Confidence Medical Devices, Software, and Systems and Medical Device Plug-and-Play Interoperability, HCMDSS/MDPnP 2007	CONF	N/A	20	2

Table 1: The list of Conferences and Journals with most citations

RQ4: How many papers about this topic have been written by the same author?

Figure 5 shows the number of publications per author. The most obvious thing is that the majority of authors (about 84%) have published only once about this topic and 10% of authors have two publications. Only 1.16% of authors have more than five publications (see Table 2). Analysing this value shows that there are many occasional contributors. Another explanation could be that this topic is new in the scientific community and authors are starting their activities in these years.

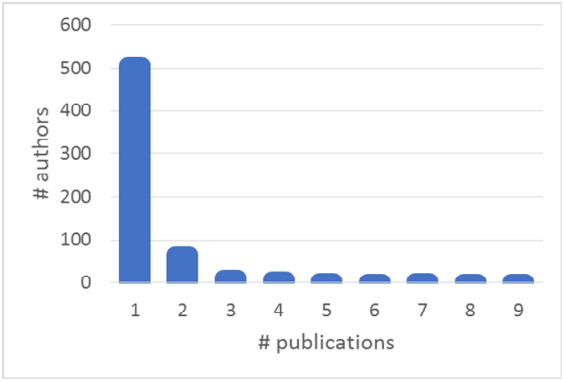


Figure 5: Publications per author

Author	*# publications*
Jones, P.L.	9
Curzon, P.	8
Mangharam, R.	8
Jiang, Z.	7
Masci, P.	7
Thimbleby, H.	7
Pajic, M.	6

Table 2: Authors with most publications

RQ5: Which are the most cited publications?

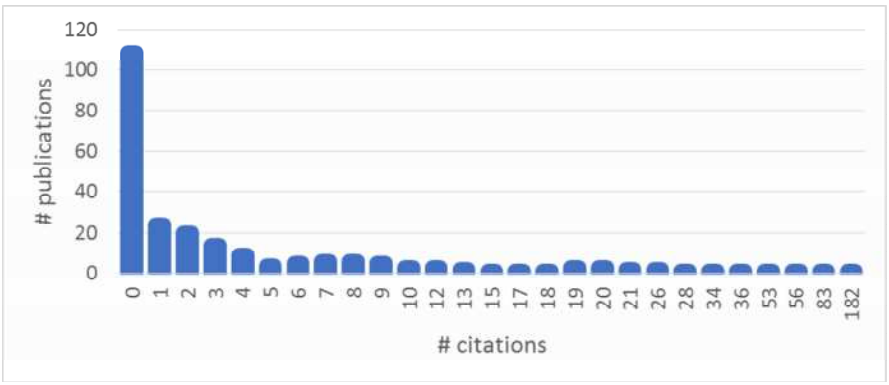


Figure 6: Citations per publications

Before introducing which are the most cited papers, we analysed the general behaviour of the number of citations[7] (see Figure 6). Overall, about 50% of publications do not have citations. About 40% of publications have less than ten citations, 6% have less than twenty citations and the same percentage have more than twenty citations. This low percentage of citations could be due to the novelty of this topic in the scientific community.

Table 3 shows the most cited publications. The publication with most citations is one of the first applications of formal methods in the medical field. It presents a formal specification language for representing medical procedures, decision, knowledge, and patient data. Paper two presents a framework for the design of a distributed and interoperable health information system. In 2006, another paper about improving medical protocols by formal methods has been written and it is one of the most cited papers (the number three in the Table). Paper four introduces a formal language developed to map different researches results into a default model. Paper number five defines a testing environment based on model-based testing and put emphasis on the lack of a formal methodology to test a medical device within the closed-loop context of patient. Paper number six is about Satisfiability Modulo Theories (SMT) solvers of embedded software. Paper number 7 applies formal methods to improve completeness and accuracy of biomedical terminologies. Paper number 8 advocates the Food and Drug Administration (FDA) defined process to evaluate the safety of medical software based on formal methods. The last paper in Table 3 is about a formal method applied to biomedical sensor networks. The authors have defined the model, have simulated the system behaviour and have applied a model checking tool to verify critical properties.

Even considering only these nine papers, it is apparent that some of them are only marginally relevant within the declared scope of our research. For instance, the paper number 7 is an interesting application of formal methods in the medical field, but has only a potential impact over the design and validation of medical software and systems. We found this the greatest limitation of the systematic approach we adopted: the use of keywords and words in titles identify also papers that fall under our criteria but

[7] Note that Scopus cannot identify self-citations in the number of citations of a given paper, so using this value as a measure of impact is not completely fair.

are not very relevant. Sometimes, we observed that papers were included only because the authors choose a wide range of keywords or because Scopus added some extra keywords. This caused the inclusion of papers that do not fit well with the goal of our SRL.

N°	*Publications*	*# citations*
1	Fox, J., Johns, N., & Rahmanzadeh, A. (1998). Disseminating medical knowledge: the PROforma approach. *Artificial intelligence in medicine*, *14*(1), 157-182.	182
2	Lopez, D. M., & Blobel, B. G. (2009). A development framework for semantically interoperable health information systems. *International journal of medical informatics*, *78*(2), 83-103.	83
3	Ten Teije, A., Marcos, M., Balser, M., van Croonenborg, J., Duelli, C., van Harmelen, F., ... & Seyfang, A. (2006). Improving medical protocols by formal methods. *Artificial intelligence in medicine*, *36*(3), 193-209.	56
4	Maldonado, J. A., Moner, D., Boscá, D., Fernández-Breis, J. T., Angulo, C., & Robles, M. (2009). LinkEHR-Ed: A multi-reference model archetype editor based on formal semantics. *International journal of medical informatics*, *78*(8), 559-570.	53
5	Jiang, Z., Pajic, M., & Mangharam, R. (2012). Cyber–physical modeling of implantable cardiac medical devices. *Proceedings of the IEEE*, *100*(1), 122-137.	36
6	Cordeiro, L., Fischer, B., & Marques-Silva, J. (2012). SMT-based bounded model checking for embedded ANSI-C software. *Software Engineering, IEEE Transactions on*, *38*(4), 957-974.	34
7	Zhu, X., Fan, J. W., Baorto, D. M., Weng, C., & Cimino, J. J. (2009). A review of auditing methods applied to the content of controlled biomedical terminologies. *Journal of biomedical informatics*, *42*(3), 413-425.	28
8	Jetley, R., Iyer, S. P., & Jones, P. L. (2006). A formal methods approach to medical device review. *IEEE Computer*, *39*(4), 61-67.	26
9	Tschirner, S., Xuedong, L., & Yi, W. (2008, October). Model-based validation of QoS properties of biomedical sensor networks. In *Proceedings of the 8th ACM international conference on Embedded software* (pp. 69-78). ACM.	26

Table 3: Publications with most citations

4 Limitations and Future Work

During our research activity we were able to identify several limitations and threats to validity of our results. We have been able to solve some of these issues by adapting our strategies, but for some of them we can only indicate our plans for the future in order to address them.

First, we have used only one source (Scopus) which we believe provides a very good mix between the number of papers included in the repository and values of the venues in which the papers have been published. For the future, we plan to consider other sources like ISI Web of Science (ISI-WoS), ACM digital library, IEEE explore, Springer Online Library, NLM's MEDLINE, Wiley Inter Science, Google Scholar, and others. After a preliminary analysis we have noticed that not all the available sources provide a good "advanced search" feature as Scopus and this can limit the introduction of a new source because we cannot easily extract information we are interested in. For example, some sources do not provide a specific language for queries. Furthermore, other sources, like Google Scholar, contain large quantity of documents and it is difficult to select those important (for example to include only those peer reviewed).

The use of words in titles and in keywords has allowed us to automatically select the papers of interest. However, we found that this makes our results very sensitive to authors' choices in terms of title words they used and of keywords they selected. Sometimes titles and keywords were matching, but the content of the paper was not in the scope of our research. On the other hand, we may have

missed interesting papers because the authors had selected particular words we did not include in our queries (for example the name of a tool or of a case study). As a future work we are planning to understand why this happens and how we can include these papers by adjusting our SLR process. One solution will be to extend the research to other sources that allow more general semantic queries. Another solution will be to manually check whether interesting papers cited in our selected papers are already included in our collection and if not, find queries to include them. With this process we will include also papers that have used different keywords to express the same concept of our SLR objective.
In general, being a preliminary analysis, we were able only to perform analysis that require a low degree of human interaction and were mainly based on the use of fields in the bibliographic entries (like year, type of publication, affiliation, citations, and so on). This has limited the results of the current analysis. To address this problem, we plan to extend the research questions with new ones that require a deeper analysis of paper contents. Examples of analysis we are interested in, are:

- What is the goal of the use of formal methods in medical field?
- Which are the notations used?
- Which are the tools used?
- Which are the methodologies applied?
- Which are the typical case studies?

We have used very simple metrics to measure impact like the number of citations and h-index. There is a general agreement on the significance of such metrics; however, some readers may find this too simplistic. We will introduce new metrics like the measure for citations using individual h-index, which normalizes the number of citations for each paper by dividing the number of citations by the number of authors for that paper, and then calculate the h-index of the normalized citation counts.
Another limitation we found in this preliminary analysis is that some journal papers are extended versions of conference papers and these should probably not contribute to the number of publications per authors. To solve this, we will manually analyze the papers with same (or similar) authors and same (or similar) titles and we will group them. After that we will keep only one paper for each group, in this way we will remove papers with the same content.

5 Conclusions

In this paper, we presented a systematic literature review about formal methods applied to medical devices. We ran several complex queries on Scopus, combined the results, and we obtained 238 publications. We performed a set of analysis (see Section 3) to provide information that can help researchers working within this domain. The number of publications per year is still growing and the researchers publish more in journals than conferences (although the difference is not big). Considering the cited papers for each journal/conference, the journal papers have more citations than conference papers. In addition, authors published in journals with high SCImago Rank (measurement of scientific influence of journals). There are a lot of authors that have published only once, and only a few authors have published more than two papers. While analysing the most cited papers (see RQ5), we found some marginally relevant papers. After a further investigation, we noticed that Scopus adds some extra keywords that do not fit well the content of the paper. As a future work, we plan to analyse in details the keywords and consider only those inserted by the authors. This allows excluding the publications that do not fit our research topic. The analysis presented in this paper is a result of a preliminary investigation.

6 References

Falagas, M. E., Kouranos, . V. D., Arencibia-Jorge, R., & Karageorgopoulos, D. E., 2008. Comparison of SCImago journal rank indicator with journal impact factor. *The FASEB Journal,* 22(8), pp. 2623-2628.

Cobo, M., López-Herrera, A., Herrera-Viedma, E., & Herrera, F., 2012. Scimat: A new science mapping analysis software tool. *Journal of the Association for Information Science and Technology,* 63(8), pp. 1609-1630.

Kitchenham, B. Brereton, O. P., Budgen, D., Turner, M., Bailey, J., & Linkman, S., 2009. Systematic literature reviews in software engineering? A systematic literature review. *Information and Software Technology,* 51(1), pp. 7-15.

Zhu, X., Fan, J., Baorto, D., Weng, C., & Cimino, J., 2009. A review of auditing methods applied to the content of controlled biomedical terminologies, Journal of Biomedical Informatics, 42(3), pp. 413-425.

Integrating HARA and TARA – How does this fit with Assumptions of the SAE J3061

Georg Macher[1], Andreas Riel[2], Christian Kreiner[3]

[1]AVL List GmbH, Hans-List-Platz 1, Graz, Austria
georg.macher@avl.com
[2]EMIRAcle c/o Grenoble Alpes University. 46 av. Félix Viallet, 38031 Grenoble, France
andreas.riel@emiracle.eu
[3]Graz University of Technology, Inffeldgasse 16/1, Graz, Austria
christian.kreiner@tugraz.at

Abstract

With the increasing replacement of classical mechanical systems by safety-critical embedded systems, car manufacturers have been raising the awareness of safety attributes and system-wide safety thinking, which culminated in the release of the ISO 26262 functional safety standard for road vehicles. In contrast to this, security topics have been seen as attacks of mechanical nature affecting single vehicles only (e.g. door lock and immobilizer related). In recent years, in-vehicle networks as well as networked vehicles have been enabling exciting new opportunities (such as advanced driver assistance systems, fleet management systems and autonomous driving). This connectivity to the cyber-physical world drives the need for built-in security solutions and architectural designs to mitigate emerging security threats.

Thus, cyber-security joins reliability and safety as a cornerstone for success in the automotive industry. As vehicle providers gear up for the cyber security challenges, they can capitalize on experiences from many other domains, but nevertheless have to face several unique challenges. The recently released SAE J3061 guidebook for cyber-physical vehicle systems provides information and high-level principles for automotive organizations to identify and assess cyber-security threats and design cyber-security aware systems.

In this paper, a review of a combined approach to a safety and security threat analysis method (SAHARA) and the recommendations of the SAE J3061 guidebook regarding threat analysis and risk assessment method (TARA) is given. Therefore, this work examines the integration of HARA and TARA and how this fits to the ISO 26262 and SAE J3061 context.

Keywords

Functional safety, cybersecurity, ISO 26262, SAE J3061

Published in: ASQ Software Quality Professional Magazin, Volume September 2016.

Automotive Security: Challenges, Standards and Solutions

Alexander Much

Elektrobit Automotive GmbH, Am Wolfsmantel 46, Erlangen, Germany
alexander.much@elektrobit.com

Abstract

Systems and software engineering for vehicles has become more complex, in-line with the complexity of the systems that are built today. Rooted in product and process quality the organizations have included functional safety aspects and are now facing the need to include automotive security in their development processes as well as into the products they build.

Keywords

Automotive SPICE, Automotive Security, Cybersecurity, Functional Safety

Published in: ASQ Software Quality Professional, Volume September 2016.

Merging FMEA and FTA for safety analysis of sensors for automotive applications

S. Mergen[1], W.J. Schreiber-Prillwitz[1], P. Schmidt-Weber[2]
[1]TDK-EPC AG & Co. KG, Stahnsdorf, Germany
[2]TDK-EPC AG & Co. KG, Berlin, Germany

Abstract

FMEA and FTA are two classical methods for safety analysis. Depending on the product developed only one of these methods is usually applied during product design phase. However, with increasing complexity of electronic systems in the automotive industry there is an increasing need to conduct both methods in order to gain thorough in-depth safety analysis of the system. This equates to reduction of faults early on in the product design. Although both FMEA and FTA have their strengths it has been shown that by applying both FMEA and FTA in an integrated approach further benefits can be gained such as discovery of additional failures modes. These failure modes are likely to remain unnoticed if FMEA and FTA are applied individually. Despite the numerous methods developed on system level, these methods do not find much application in the development of hardware components. Thus, at the beginning of this study was the question whether these methods are adequate form of analysis at the hardware component level and if so, could they be applied here. Based on the results of the study practical procedure is presented that integrates both FMEA and FTA outcomes in one single analysis method so that any additional diagnostics and design optimization can be done in just one analysis. The result of the study is a practical procedure of FMEA and FTA integrated analysis that can be applied during the components design and development.

Keywords

FMEA, FTA, safety analysis

1 Introduction

A Failure Mode and Effect Analysis (FMEA) is an established bottom-up risk analysis method which is applied in great majority of product development projects, and especially in the developments according to ISO 26262 standard where FMEA is a firm requirement for all ASIL levels (ISO 26262). For ASIL levels C and D a top-down approach analysis such as Fault Tree Analysis (FTA) is additionally recommended. The reason of conducting both analyses within the same product development is that one analysis alone might not be sufficient for systems with high safety requirements as both methods have their strengths and weaknesses. The FMEA focuses on single-point failures, investigates the failure effects and causes and rates them according to their severity, occurrence and detectability. The functions and failures of the system can thus be rated and prioritized for optimization according to their risk rating. However, dual-point and multiple-point faults are hard to tackle with the FMEA alone.

The FTA on the other hand starts with top event failure for which all possible faults are deduced and graphically presented in a fault tree. The leaves of the tree model the component failure, with the gates modelling the failure propagation. In this form of analysis both single-point and multiple-point faults can be expressed. The FTA enables qualitative and quantitative analysis. As a result of qualitative analysis the minimal cut sets are obtained which point out events (or a single event) that might lead to total failure of the system. Quantitative analysis, on the other hand, enables reliability calculations of the product.

In terms of the direction of the analysis, the FMEA starts with the failure causes of the components and deduces resulting failure effects on a higher level system. The FTA, on the other hand, starts with the failure of the higher level system and is developed down to the component level (Wada 2000). For a component manufacturer, the highest level of analysis is the interface to the system in which the component, e.g. a sensor, is built in. This direction of analysis has traditionally led to applying FTAs for analysis of large systems in wide range of industries while the FMEA is equally adequate for system and component analysis. Thus, the FTA is less likely to be done for components. However, by applying both methods at the component level, the component design can be analysed from a different angle and can potentially give insight into design imperfections that would have been difficult to discover with the FMEA alone.

Both methods are therefore beneficial. However, applied individually, there are certain short-comings and there has been great effort to combine the two methods in one integrated analysis. Especially in the software domain a variety of integrative approaches has been developed for instance for software safety or security analysis (Kim 2013). Although some reduction in resources is likely, the main benefit of performing integrated analysis was the improved quality of the analysis. This, in turn, resulted in advanced software architecture as shown by Lutz et al. (1997) and Hong et al. (2009). Further, the outcome of such analysis provided inputs for the component testing, integration testing and system testing in order to achieve reliability of the software (Khaiyum 2014).

The question at the beginning of this study was whether a method of FMEA - FTA integration, can be applied at the component level to provide more efficient and thorough analysis of a sensor design. Ideally, such method would encompass the discovery and the presentation of single-point and multiple-point faults and thus include the advantages of qualitative FMEA as well as the benefits from the ‘point of view of the system’ as applied during the FTA.

The aim of this study was to develop one comprehensive and thorough analysis method based on the integration of FTA and FMEA as described on case studies in the literature, e.g. in the work done in the software domain or on a system level. However, a thorough analysis on component level for safety relevant automotive application has different objectives and a different complexity which will be described in more detail in the following Sections. There is also a lack of case studies in the literature on component level, e.g. in sensor design. As a result of this work a practical procedure is developed for safety related sensor analysis which integrates some of the procedures described in the literature. The main advantage of such procedure is the improved quality of analysis with some minor reduction of effort compared to doing both analysis methods separately. Thus, the strength of the integrated analysis presented in this paper is the systematic method showing how to tackle various challenges, such as dual-point faults, safety requirements and risk rating in one analysis.

2 Literature Survey

It has been shown by Lutz et al. (1997) that both FMEA and FTA are complementary. Large amount of failures remain undetected if both FMEA and FTA have been done separately. Improvement and further work on integration of both methods has been conducted by various research teams (Hong 2009, Shaoping 2000, Kim 2013, Nicodemos 2013, Fatima 2013) showing increased quality of analysis. By discovering more failure causes and especially such that would have not been detected by doing FMEA and FTA on their own, these teams were able to define additional requirements at the start of the product design which, in turn, yielded improved software architecture.

The integrative methods can be classified in three different approaches: forward integration (generate FTA with help of previously done FMEA), backward integration (conduct FMEA after FTA has been

done) and the bi-directional methods such as Bi-directional Analysis (Lutz 1997) or Bouncing Failure Analysis (Bluvband 2005). The latter methods enable switching or bouncing of the analysis direction from FMEA to FTA and vica-versa. Leaning on the two later methods, some approaches are based on combining fragments from FTA or FMEA methodology at different points throughout the analysis (Kim 2013, Yang 2009). In more detail, the strength of forward integration is the identification of latent failure modes. The strength of the backward analysis is identifying coincident circumstances that allow the hypothesized failure mode to occur (Lutz1999). For some applications, e.g. in security analysis, the backward analysis was considered more efficient.

A method of combining FMEA and FTA was also briefly suggested in ISO 26262:2011 Part 10. As a top event a violation of safety goal is chosen and the related faults are derived to a sufficient depth. Once the component level is reached, the FMEA for components can be linked to the FTA. It may seem that such approach while suitable for analysis at a system level, does not generate much gain for analysis at the component level.

Most of the methods of safety analysis integration in the literature are based on the software development (Su 2014), embedded systems (Khaiyum 2014), system safety (Nicodemos 2013, Swarup 2014) and security (Kim 2013). In general, these methods have in common that they start with the top event and analyse the propagating faults down to the level of fault cause. Once the level of single fault cause has been reached, each failure mode is analysed for its cause and the effect. The failure mode is then rated in terms of the risk of failure occurrence and detection.

All of the integrative methods mentioned in the previous paragraphs apply FMEA in form of a table or formsheet regardless of the direction (forward or backward integration) of the analysis. However, applying FMEA in table form for a hardware component for automotive application is outdated as the 5-steps systematic approach defined by VDA (German Association of Automotive Industry) 4.2 FMEA (VDA 2006) has been shown to provide many benefits. The FMEA after the VDA 4.2 method consists of generation of the system structure tree based on the hardware parts, defining functions and failures for each part and connecting these in the function and failure nets for more thorough analysis. Only after the failure nets have been connected preventive and detection actions can be defined, rated and the design can be optimized. Thus, none of the integrative methods developed for software can be applied to analysis of a hardware component without further modifications. From all evaluated procedures the most potential for application have the Bi-directional Analysis by Lutz et al. (Lutz 1997) and the forward integration method described by Pickard et al. (2006).

3 Overview Simple Sensor Model

A Simple Sensor Model (SSM) has been developed in order to study the effects of integrated safety analysis method. The whole analysis was based on this model to illustrate benefits and disadvantages of some methods proposed in the literature and how these can be applied at the hardware component level. As a result of the study a practical procedure of integrated analysis was obtained which can be applied during the components design and development.

The sensor model consists of four major elements which can be found in one form or another in most sensor transmitters. These are a sensing element or unit, signal processing unit, interface to the system and the package for the sensor transmitter (Fig. 1). The sensing element is the sensor cell that converts the stimulus (e.g. pressure) into an electrical signal, e.g. gas pressure is detected by the deflection of a membrane and transduced into proportional electrical signal. This signal is then processed by the signal processing unit and transmitted via an interface to the high level system. It must be noted that this high level system is not in the focus of this study and it can be any system or item developed for automotive application. Although this model is very general it is yet adequate for testing how the failure modes can be transferred from one form of the analysis into the other.

The SSM covers basic functionality of different types of sensors (e.g. pressure sensor, motion sensor) by the same model. Such model has been generated with the APIS FMEA Software (Fig. 2) after the five step procedure described by the VDA (VDA 2006). The main functions of the sensor transmitter are to provide the required correct signal on time for the duration of the system life time (expressed as Safety Goal 1) and to fulfil all non-safety relevant system requirements (1.c in Fig. 3). Some diagnostic

functions and error management have been included in the sensor which detect faults in the sensor output and signal the type of the fault to the high level system.

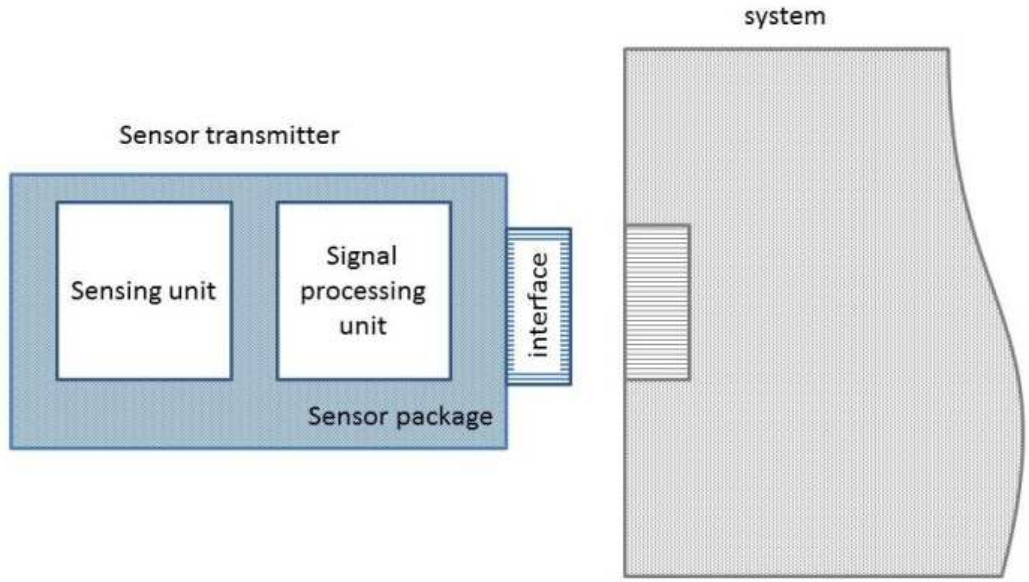


Fig. 1: Schematic drawing of a simple sensor model

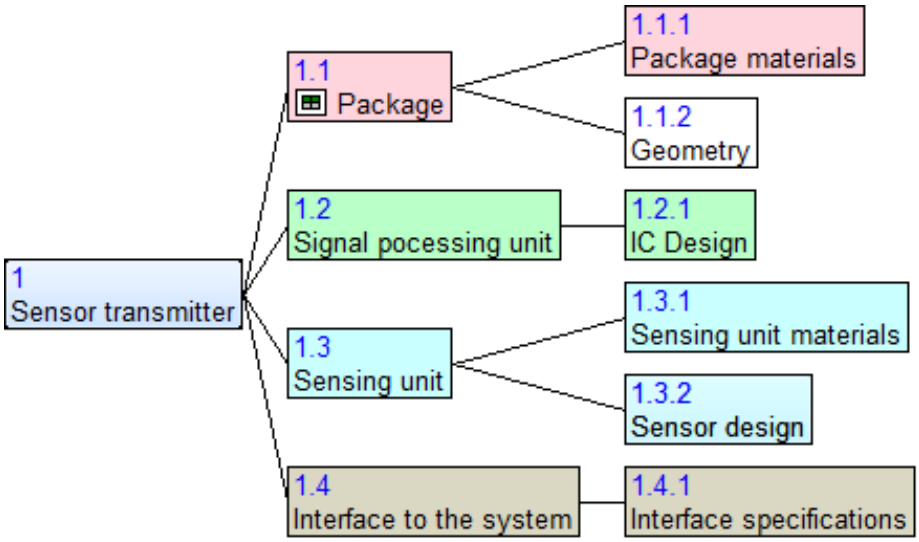


Figure 2: FMEA tree structure of the simple sensor model created with the APIS software

Two safety goals have been formulated for the sensor transmitter based on the system requirements. The Safety Goal 1 (SG1) is to provide the required correct signal on time for the duration of the system life time. The violation of this SG1 can be caused by two failures only, such when there is no signal and such that there is a signal but it is not apparent that this signal is wrong. Whether the failure is perceived at the system level or by the driver depends on the system itself and the diagnostics of the system.

Safety Goal 2 (SG2) is another measure to ensure that diagnostic features are available which will prevent some of the violation of the SG1 (wrong signal) and to signal failure of the sensor to the system.

For each element of the sensor transmitter some functions and failures have been created in a similar manner. Once the functions and failures were defined the function and failure nets can be connected.

Sensor transmitter Functions	**Sensor transmitter Failures**
1.a) Safety Goal 1: provide correct signal to the system	No signal
	Wrong signal
1.b) Safety Goal 2: Diagnostics – signal failure to the system, if failure occurs	Self test wrong
	Selft test not comprehensive
	Self test missing
1.c) Fulfil remaining system requirements	System requirements not fulfilled

Figure 3: Function and failure descriptions for the sensor transmitter

4 Integrated FMEA and FTA analysis

Step 1: Do Design FMEA

At the beginning of the analysis function and failure nets have been connected as usual. This step is analogous to the step 1 of the bi-directional analysis (BDA) developed by Lutz et al. (Lutz 1999). Two failures (‘no signal’ or ‘wrong signal’) violate the Safety Goal 1.

Step 2: Create the FTA top-level

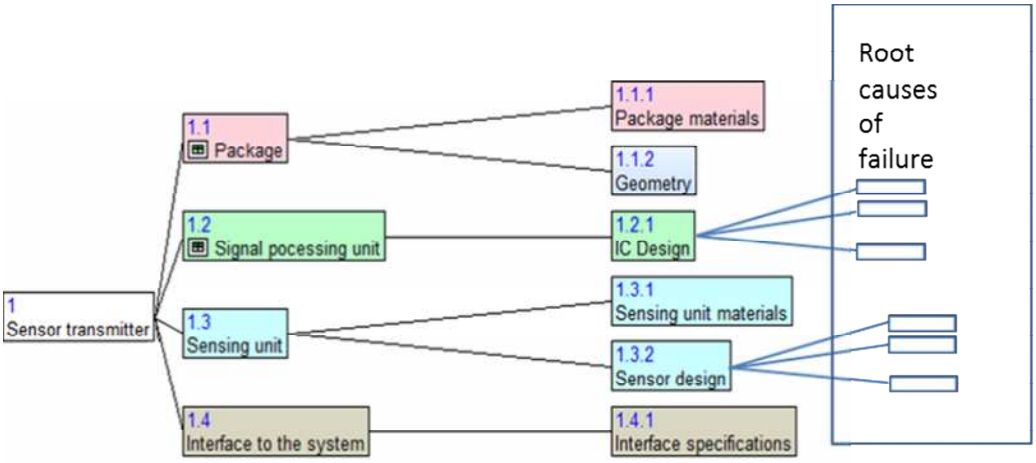
Once the FMEA has been done a top event has been chosen for the FTA, analogous to the step 2 of the BDA and the method suggested in the ISO26262. In this step of the analysis the same violation of the SG1 was chosen as the top event.

Some failures have been deduced based on the simple sensor model (Fig. 4). New failure modes included e.g. violation of a SG due to failed communication between the sensing element and the processing unit as well as failed communication between the processing unit and the interface. One of the advantages of the FTA is to include process-related failures and any pertinent causal events that can lead to failure, including software and hardware errors, human errors, and operational or environmental events (Wada 2000).

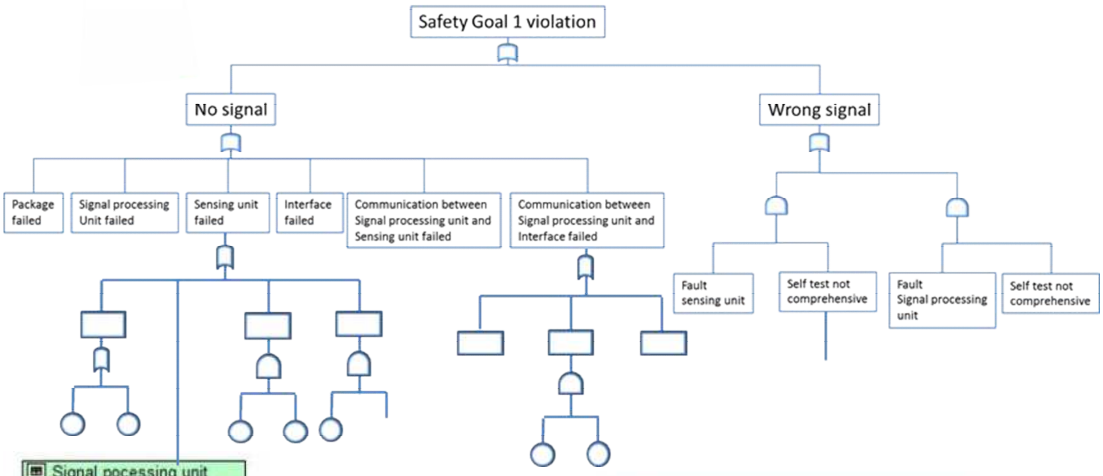
Figure 4: FTA of the top event 'Violation of the Safety Goal 1'

The FTA is completed when all further failure modes have been added, those resulted from the FMEA plus the newly identified ones. This is schematically shown in Fig. 5. where addition of newly identified failures is depicted in Fig.5 a) and the addition of failures from the FMEA is depicted in Fig. 5.b).

Based on previous experience, and as shown by (Lutz 1997) and (Hong 2009), such analysis showed the strength of the FTA to include additional faults which could have been easily missed in the FMEA. Some of these faults proved to be very valuable. It was found that the faults which happen at the interface between two sensor parts needed special attention. To illustrate, it was found that the materials properties in terms of thermal expansion must be matched for the package, the sensing unit and the bonding of the sensing unit to the package. If the thermal expansion between two materials is within specifications it is still possible that the thermal expansion between several materials at one interface can lead to failure if not matched. While this failure cause might seem trivial in reality such failures can be easily overlooked during the FMEA.


Root causes of failure
1 Sensor transmitter
1.1 Package
1.2 Signal pocessing unit
1.3 Sensing unit
1.4 Interface to the system
1.1.1 Package materials
1.1.2 Geometry
1.2.1 IC Design
1.3.1 Sensing unit materials
1.3.2 Sensor design
1.4.1 Interface specifications

a)


Safety Goal 1 violation
No signal
Wrong signal
Package failed
Signal processing Unit failed
Sensing unit failed
Interface failed
Communication between Signal processing unit and Sensing unit failed
Communication between Signal processing unit and Interface failed
Fault sensing unit
Self test not comprehensive
Fault Signal processing unit
Self test not comprehensive

b)

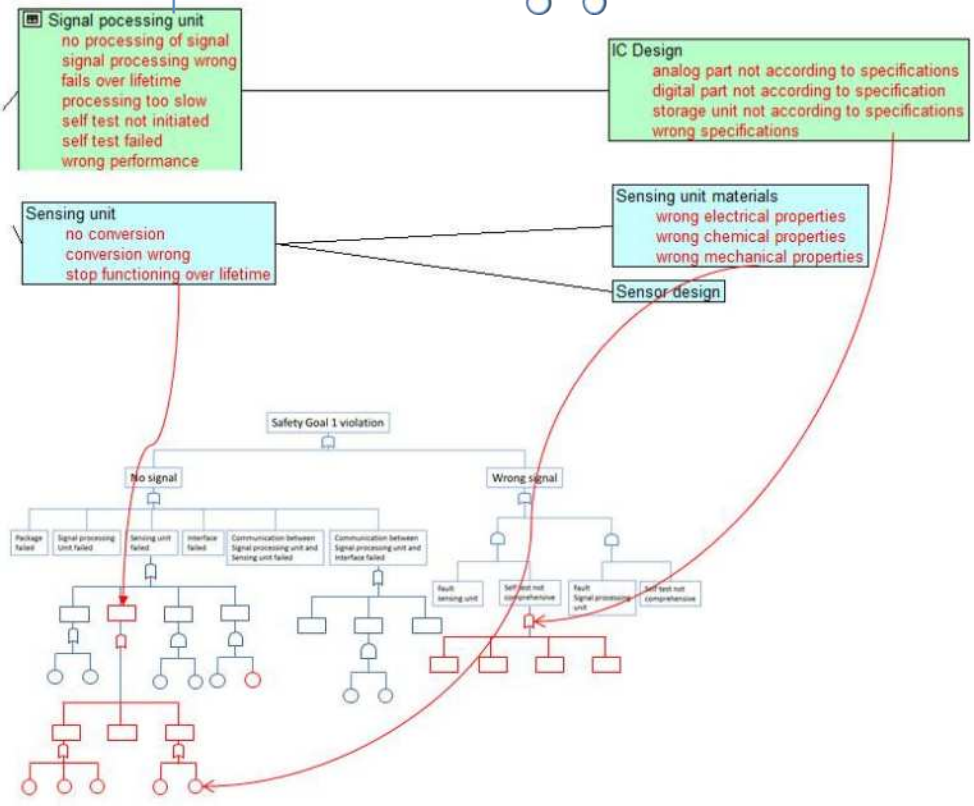

Signal pocessing unit
no processing of signal
signal processing wrong
fails over lifetime
processing too slow
self test not initiated
self test failed
wrong performance
IC Design
analog part not according to specifications
digital part not according to specification
storage unit not according to specifications
wrong specifications
Sensing unit
no conversion
conversion wrong
stop functioning over lifetime
Sensing unit materials
wrong electrical properties
wrong chemical properties
wrong mechanical properties
Sensor design
Safety Goal 1 violation
No signal
Wrong signal

Figure 5: Schematics of fault tree analysis with support of the input from the FMEA. a) Newly identified faults are marked in blue. b) Addition of failures causes from the FMEA.

Step 3: Deduce dual-point faults from the FMEA and include them in the FTA

This step addresses identification of multiple-point faults after the Bouncing Failure Analysis (BFA) method described by (Bluvband 2005). The purpose of this method is to generate systematically the FTA from faults from the FMEA. The analysis focuses on faults that can cause a failure with other non-obvious fault causes.

At first, one failure effect is chosen, e.g. 'no signal'. All causes that can lead to failure effect (no signal) directly are excluded from the analysis. There remain all failures that do not contribute to the chosen failure effect, e.g. 'wrong signal processing' in the processing unit or 'wrong protocol' at the interface. While either of the failure causes will not lead to the failure effect 'no signal' individually, these failures occurring at the same time will most likely lead to 'no signal'.

Step 4: Transfer dual-point faults back into the FMEA

Once all dual-point and multiple-point faults have been identified for the fault tree analysis and the FTA is so far completed, the newly added failures can be transferred back to the FMEA to define preventive and detective measures and to assign a risk rating to the failure effects. Thereby transferring the multiple point faults presents a challenge since the FMEA does not have the gate 'AND' to link two or more failure causes together.

To alleviate the task, an interface function can be defined. The interface function is represented by the usual failure element assigned to the relevant function, however instead of noting one failure cause per failure element all failure causes have been explicitly written out and connected by 'AND' (Fig. 6). This ensures that the relationship of these failures is not lost in the transfer between the FTA to the FMEA. The long name also makes it somewhat easier to spot the multiple-point faults. However, care must be taken when defining preventive and detection action. These should address both causes but must be described in one place only and assigned to one of the failure causes.

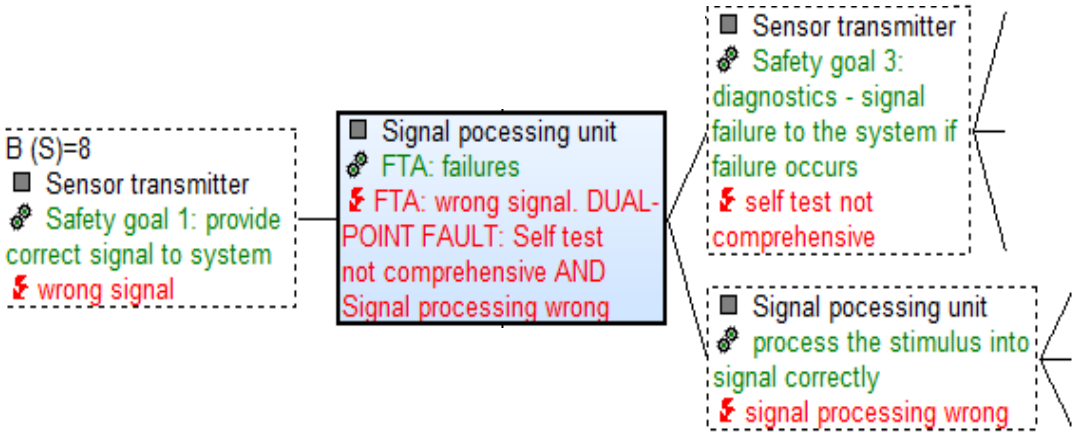


Figure 6: Example of interface function representing dual-point fault

Step 5: FMEA optimization

After completing the FTA transfer back into the FMEA the failures can be rated and preventive and

detective measured can be defined. Diagnosis measures can be implemented or failure metrics calculated. To calculate the Risk Priority Number (RPN) for dual-point faults the method of Pickard et al. (2005) can be applied. This method encompasses calculation of dual-point failures via a 3-dimensional matrix based on the ppm-occurrence of the single faults. Thus, the severity, occurrence and detectability can be looked up in the matrix and the RPN is calculated by multiplying these values.

5 Conclusions

In summary, the whole process of merging the FMEA and FTA is shown in a diagram in Fig. 7. It was found that a better analysis is obtained if FMEA was conducted first since FMEA enables greater level of detail right from the start. For the same reason, the FMEA is chosen to be the end-result of this study and to include all insights from the FTA in the FMEA.

After initial completion of the FMEA the FTA could be easily constructed. As large amount of failures was identified during the FMEA the FTA was not as elaborate and lot of failure modes could be just transferred. By switching the analysis between the FMEA and FTA new failures have been discovered. New single-point failures were then easily added back to the FMEA to the corresponding hardware parts.

It was however, more complicated to identify and include the dual-point faults in the FMEA. Here, the systematic Bouncing Failure Analysis was the key. At first, the potential dual-point faults from the FMEA were identified and then included in the FTA where they can be easily represented by the gate ‘AND’. Once the BFA was completed, the dual-point faults were transferred back into the FMEA. To transfer the dual-point failures into the FMEA an interface function was created for each dual-point fault. The nature of such faults was highlighted by explicitly writing down the full description of the fault. For the risk rating of such failures the method developed by Pickard et al. (2005) was used.

As a result, we obtain one FMEA analysis with all benefits of an FTA. In addition, by choice of software for this analysis the FTA is also available so that other calculations (e.g. reliability calculations) can be done.

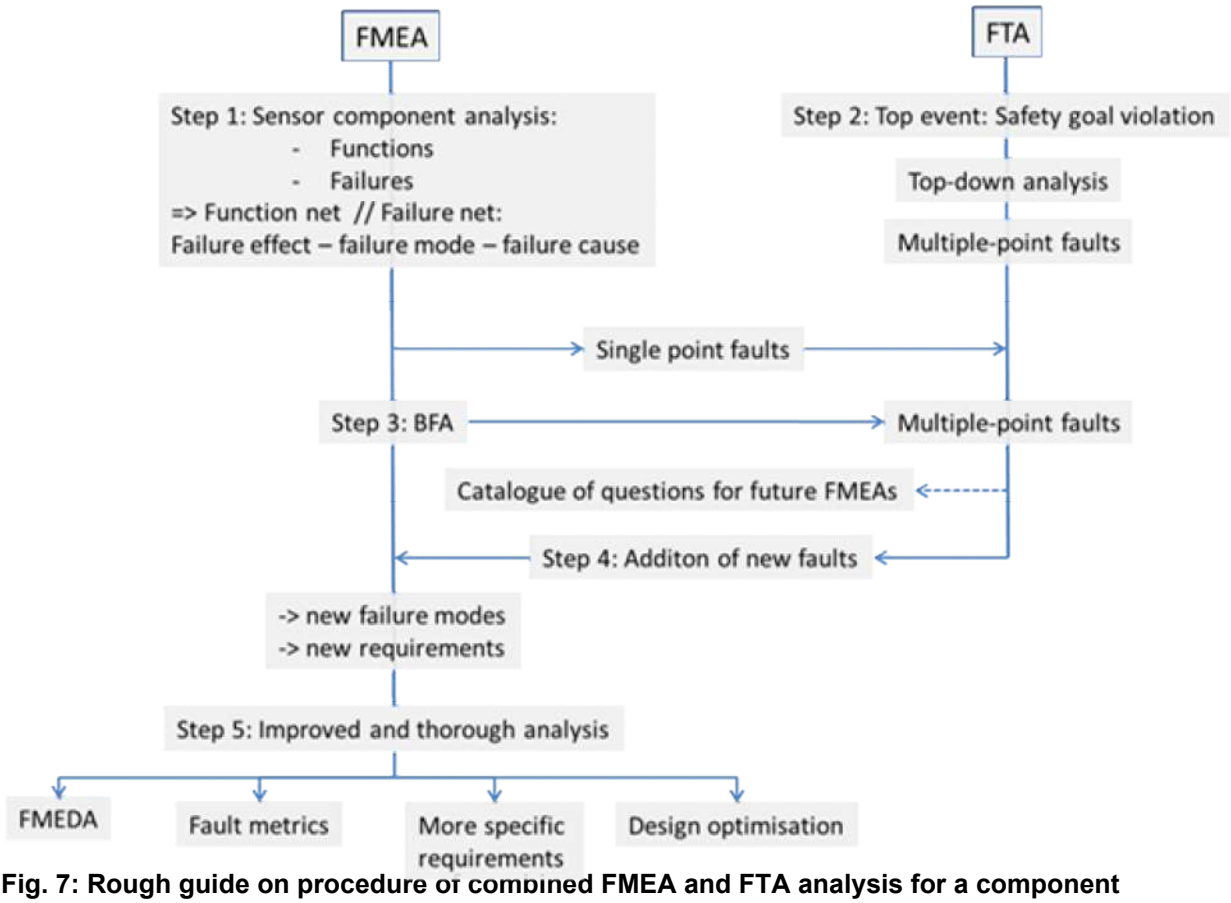


Fig. 7: Rough guide on procedure of combined FMEA and FTA analysis for a component

1 Literature

Bluvband, Z, Polak, R & Grabov, P 2005 'Bouncing Failure Analysis (BFA): The Unified FTA-FMEA Methodology', Proceeding of Annual Reliability and Maintainability Symposium, pp.463 – 467

Hong, & Liu, B. 2009, 'Integrated Analysis of Software FMEA and FTA', Information Technology and Computer Science, ITCS 2009. International Conference on , vol.2, no., pp.184-187

International Standard 2011, Road vehicles – Functional Safety, ISO 26262:2011, International Standard Organisation

Khaiyum, S & Kumaraswamy, YS 2014, 'An Effective Method for the Identification of Potential Failure Modes of a System by Integrating FTA and FMEA, ICT and Critical Infrastructure: Proceedings of the 48th Annual Convention of Computer Society of India- Vol I Volume 248 of the series Advances in Intelligent Systems and Computing, pp 679-686

Kim, MH, Toyib, W, Park, MG 2013, 'An Integrative Method of FTA and FMEA for Software Security Analysis of a Smart Phone', KIPS Tr. Comp. and Comm. Sys. Vol.2, No.12, pp.541-552

Lutz, R & Woodhouse, RR 1997, 'Requirements Analysis Using Forward and Backward Search', Annals of Software Engineering, vol. 3, pp.459-475

Lutz, RR & Woodhouse, RM 1999, 'Bi-directional Analysis for Certication of Safety-Critical Software, Proceedings, ISACC'99, International Software Assurance Certication Conference

Nicodemos, FG, Lahoz, CHN, Abdala, MAD, Saotome, O 2012, 'Using Combined SFTA and SFMEA Techniques for Space Critical Software', Proceedings of the 5th IAASS Conference A Safer Space for Safer World, pp.12

Pickard, K, Müller, P & Bertsche, B 2005, ' Multiple Failure Mode and Effects Analysis – An Approach to Risk Assessment of Multiple Failures with FMEA', Proceedings of the Annual Reliability and Maintainability Symposium, pp. 457

Su, G, Huang, L & Fu, X 2014, 'Synthetic Safety Analysis: A Systematic Approach in Combination of Fault Tree Analysis and Fuzzy Failure Modes and Effect Analysis, Proceedings of the 4th International Conference on Computer Engineering and Networks, pp 389-398

Swarup, MB & Amaravathi, K 2014, 'Safety-Critical Failure Analysis of Industrial Automotive Airbag System using FMEA and FTA Techniques, International Journal of Advanced Research in Computer Science, Volume 5, No. 5, pp.70

VDA-QMC Publications 2006, Verband der Automobilindustrie-Quality Management Centre, Band 4 Kapitel: Produkt- und Prozess-FMEA

Wada, H 2000, 'Safety analysis methods and applications at the design stage of new product development - Introducing the FMFEA and S-H Matrix Method', Environmental Testing Information for Technical Personnel, ESPEC TECHNOLOGY REPORT No. 10

Yang, S, Lu, M, Liu & B, Hao, B 2009, 'A Fault Diagnosis Model for Embedded Software Based on FMEA/FTA and Bayesian Network', Reliability, Maintainability and Safety, 2009. ICRMS 2009. 8th International Conference on , pp.778-782

2 Author CVs

Silvana Mergen

Dr Silvana Mergen is a development engineer at TDK-EPC and responsible for matters relevant to Functional Safety after the ISO 26262 in the Sensors Business Group. Silvana holds a materials science degree, a PhD in Piezoelectrics from Cranfield University and a TÜV-certificate in quality management. Before joining TDK-EPC she was a project leader at The HEARing CRC as well as a research fellow at the University of Melbourne, working in the field of Cochlear implants.

Wolfgang Schreiber-Prillwitz

Dr Wolfgang Schreiber-Prillwitz was heading the development of Inertial Sensor Systems for functional safety relevant applications in automotive applications at EPCOS AG. Before joining EPCOS, he worked in the field of MEMS system design and MEMS technology since more than 25 years in different companies at different functions. Focus of his work is design, technology and application of pressure sensors, current sensors and inertial sensors.

Philipp Schmidt-Weber

Dr Schmidt-Weber is Vice President Quality Management of TDK Sensors Business Group. After studying Chemistry he made his Dr in Physics at Fritz-Haber-Institut of the Max-Planck-Society. He then joined the start up company Sulfurcell later named Soltecture) where he worked in development for rapid thermal processing of CIGS solar cells. After the insolvency of Soltecture he started at EPCOS / TDK with a focus on project quality management and functional safety. He works as head of QM since end of 2015.

Formal Methods & Functional Safety

Micheal Mac an Airchinnigh
ISCN LTD/Gesmb, Ireland, mmaa@iscn.com

Abstract

Functional Safety is always a featured topic of the EuroSPI conferences? Hence it seems to be a given to consider all those related technical aspects. In our culture the car seems to reign supreme. The car is now endowed with all sorts of technologies. Perhaps one of the key technologies is the breaking system? Another important aspect of the modern car is the air bag, and the seat belt. Both of these have been designed and tested. There is a cultural aspect to the car, the shape and look. Once cars were basically functional. Now they (all) have a certain well-designed aesthetic. But there is another foundational aspect to this: the formal method of design, of shape. [For cyclists, bikers, etc., there are corresponding function design concerns..

Keywords

Aesthetic, cultural, formal, provable, testable, Wikipedia

Published in: Springer Communications in Computer and Information Science (CCIS) vol. 663

A Virtual Glucose Homeostasis Model for Verification, Simulation and Clinical Trials

Neeraj Kumar Singh, INPT-ENSEEIHT / IRIT
University of Toulouse, France, nsingh@enseeiht.fr

Abstract

The behaviour of a medical device is driven by the desired functionalities of the operating environment. Most of the time, any medical device fails due to lack of understanding of the system requirements, functional and non-functional requirements. In this paper, we propose the formal development of a virtual environment model, simulation framework and hardware implementation, which can be used in the development process of medical devices, particularly for analysing the system requirements, system verification and validation, behaviour simulation, system testing, and finally, for clinical trials of the medical devices. For developing the proposed concepts, we use the glucose homeostasis (GH) environment model during the formal development, simulation and implementation that can be used to analyze the patient specific medical devices like an Insulin Infusion Pump (IIP).

Keywords

Virtual environment, modeling, simulation, verification, test bench, glucose homeostasis (GH), clinical trials.

1 Introduction

Since software plays a vital role for better controllability, operability and safety in medical domains, the device manufactures require to validate any software used in their device, to test rigorously the functional behaviour of the device for detecting critical flaws and security vulnerabilities. To address the critical flaws and security vulnerabilities including device development process, the regulators provide guidelines to develop the safe and dependable critical medical systems. These guidelines play a major role to comply with certification standards. Due to increasing recalls and complexities in medical devices, industries and regulators, such as the US Food and Drug Administration (FDA), are looking for new methods and tools for improving the engineering-based review strategy that could provide the safety assurance to the developing systems.

Insulin Infusion Pump (IIP) is a complex medical device that is used by millions of people to regulate normal level of glucose for controlling diabetes. This device is used to deliver insulin doses in a control manner in order to maintain an appropriate level of glucose. Over the past few years, IIPs have been used successfully to cure diabetes. However, the failure rates of the IIPs have increased tremendously. These failures cause several deaths and serious illnesses. The FDA reported 17,000 adverse-events from 2006 to 2009 including 47 deaths due to IIP's malfunctions (Chen et al., 2013). The root causes of these device failures are considered as product design and engineering flaws, which are identified by the FDA officials during investigation of the reported deaths and illnesses related to the IIPs.

As far as we know, there is no environment model for IIPs that can be used for validating the system requirements, simulation and for critical trials of IIPs. This paper proposes the development of the glucose homeostasis (GH) virtual environment model, including simulation and hardware implementation for diabetes patients to analyze the patient specific medical devices like IIPs.

The proposed virtual environment model describes normal and diabetic conditions, by using α-cells and β-cells, and rising or dropping plasma glucose level to model pancreatic behaviour, and blood test levels for diagnosis of diabetes/pre-diabetes. The key feature of this model is to consider both normal and abnormal (hyperglycemia or hypoglycemia) behaviours that can be used to characterize a patient model. The prime use of this environment model is to assist in construction, clarification, and validation of the given system requirements by developing a closed-loop model using formal methods in the early stage of the development life cycle. Moreover, the same formal model can be used for developing the interactive simulation, to automate the process of test case generation for testing the device software, and for implementation on the hardware platform that can be used for clinical trials of medical devices. This model can also assist to medical practitioner to understand the better organ behaviour, and it can be used to generate infinite patient specific conditions that can be used for device validation. We have envisioned several benefits of using the virtual GH environment model in the development life cycle of IIP that are given as follows:

- To identify gaps or inconsistencies in the IIP requirements.
- Developing a closed-loop model for verification and validation of IIP.
- Analysing system interaction between the GH environment model and IIP at a very abstract level.
- Developing a simulation model from the formal virtual GH environment model.
- Traceability of undiscovered behaviour and validation of the IIP assumptions.
- Developing a test bench using the virtual GH environment model for clinical trials of IIPs.
- Generating test cases to test the functional correctness of IIP software.
- Developing patient specific model at various level of system development.
- The virtual environment model can be used by medical industries during the product development.
- The virtual GH environment model can be used by regulators for validating and certifying the medical devices, such as IIP.

The structure of the article is as follows. In Section 2, we present related work. Section 3 presents basic concepts of the virtual environment modeling. Section 4 presents the GH system and the formal definition of the GH. Section 5 presents design and development of GH, and Section 6 discuss the usability of the environment model. Finally, in Section 7, we conclude the paper with future work.

2 Related Work

Biological system is one of the complex, dynamic and infinite systems, which are not fully discovered yet. The medical practitioners and engineers use physical and mathematical model to characterize the biological behaviour. From a decade, several models exist for describing the glucose homeostasis and diabetes. These models are clinical and non-clinical. The clinical models are used for identifying and predicting the diagnostics, control, progression, complication, etc. of diabetes. The non-clinical models are used for modeling the insulin-glucose, hepatic glucose, glucagon, and insulin receptor dynamics, beta-cell insulin release, and brain glucose homeostasis. The first mathematical model based on differential equations to model the glucose and insulin concentration, illustrating the dynamics of insulin-glucose for diagnostic purpose and evaluating several parameters of the diabetic and pre-diabetic conditions in (Bolie, 1961). The proposed model was very useful for describing the dynamics of insulin and glucose and their concentration level. An integrated insulin-glucose model for analysing the diabetic condition using a bidirectional insulin-glucose feedback mechanism presented in (Silber et al., 2007). The theoretical treatment of the effect of external potassium on oscillations in the pancreatic β-cells presented in (Chay and Keizer, 1985). This model was able to demonstrate that insulin infusion may be useful for mimicking pancreatic insulin secretion. There are several models produced by aca-

demia and industries that incorporate different physiological processes associated with insulin-glucose dynamics and different variations (Ajmera et al., 2013).

The existing environment models, based on higher-order complex mathematics, are not easy to express in first order logic, and thus make it difficult to use for verification purpose. Moreover the existing models are developed for specific purposes that cannot support desired global behaviours. We want to describe the complete system by introducing the abstract notions of possible features that can be later extended for any particular use. The concept of environment modeling for GH system is motivated by our previous work on heart modeling (Singh, 2013) that represents an abstract behaviour of the heart using electrocardiogram. We have adopted the same methodology to design an efficient and optimum environment model for the GH system based on abstract notions of pancreatic behaviour. To our knowledge, there does not exist any virtual environment model for GH that can be used for system validation and verification, simulation and clinical trials during the device development. The model is defined through analysing the glucose regulation mechanism. The virtual environment model is described abstractly using first-order logic considering various safety properties at each incremental step, and normal and abnormal behaviours (hyperglycemia, hypoglycemia or diabetic complications).

3 Virtual Environment Modeling

To use an environment model for developing the medical devices, we need to focus on the expressiveness of the selected modeling language to describe the complex and realistic environment using various levels of abstractions. The modeling language should have well-defined syntax and semantics for the tools to analyze desired behaviours. The language should allow refinement based modeling including temporal, functional, reactive and non-deterministic behaviours for specifying the required components of the virtual environment. Environment modeling requires higher level of abstractions to describe non-deterministic behaviour compared to the system behaviour.

The environment model provides essential information about environment components, basic characteristics of each component, and structural relationship between components. The structural relationship between two components shows a common interface channel or medium to exchange the information. All these components together form an environment that simulates virtual operating environment for medical systems. Following, we discuss further in detail various guidelines for modeling a virtual environment model for medical devices.

- **Environment Components.** To design an environment model, we need to include all the essential components to describe the required system behaviours. These essential components can be expressed in both abstract and concrete as per the granularity level of system component definition. All the models from abstract to concrete levels can be used for simulation purpose that can satisfy the required behaviour of the system.
- **Relationships between Components.** The environment components are linked to each other through physical or logical relationships. The relationships preserve architectural definition of the system components. The given relationships allow all the environment components to communicate or to exchange the information.
- **Components Properties.** To design a correct environment model, we need to provide a set of properties related to functional behaviour. Model properties are indispensable to characterize the environment components to satisfy the required behaviour.
- **Modeling the System.**To use the effectiveness of environment modeling, it is important to include a system model itself for analyzing the overall system behaviours. For instance, to analyze an IIP, we need to model both the IIP and GH models.
- **Simulation of Environment Models.** An environment model developed using any methodology requires simulation and dynamic execution for understanding the correctness of biological behaviour. The simulation allows to check the temporal behaviours, interactions between the system components, and new changes evolve within the time. In fact, it must meet all the required behaviour for a real biological system.

4 Glucose Homeostasis System

Glucose is the primary source of energy of the human body. It must be maintained in the body, and to keep active all the biological organs and to do the regular normal functionalities, we need a regular supply of glucose to the body. Failure of the glucose level causes many diseases like diabetes mellitus, galactosemia and glycogen storage diseases (Ajmera et al., 2013). The normal glucose homeostasis (GH) system[1] is depicted in Fig. 1 that shows basic pattern of hormonal flows between different organs that actively participate for regulating the glucose level. It is vital for the body to maintain an appropriate level of glucose in blood, because both the low and high glucose levels are dangerous and it could lead to life threatening problems. The pancreas and liver play an important role to regulate the desired level of glucose in the body. The pancreas produces *insulin* and *glucagon* hormones to control the GH system. The available glucose is used by the body cells that can be received by the body through infusion or hepatic function.

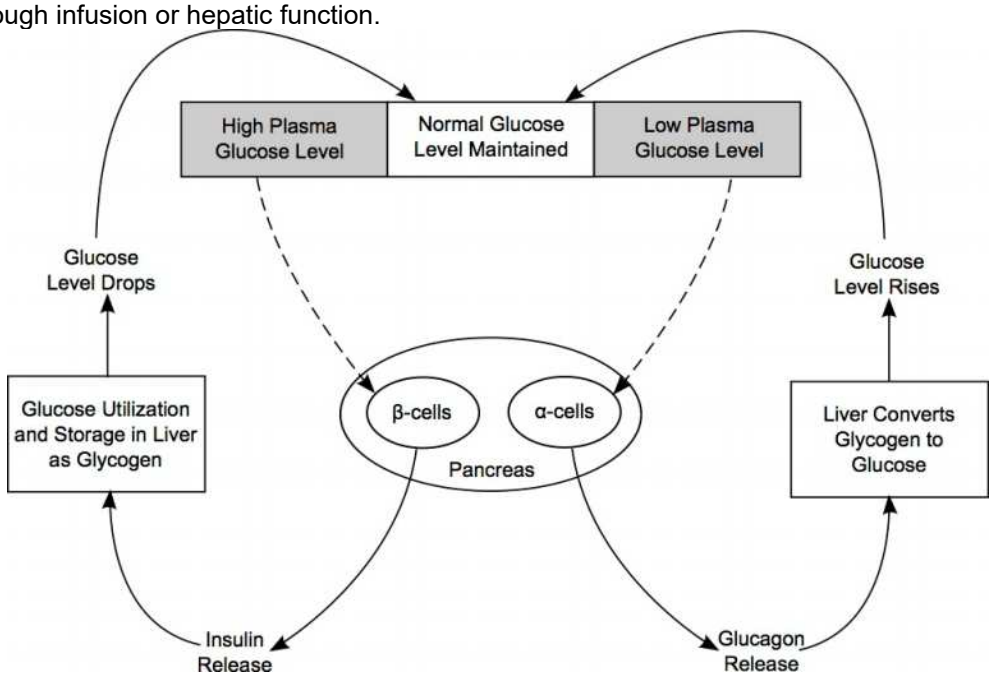


Figure 1: The Glucose Homeostasis

There are two main biological responses, low and high glucose levels, that allow to the body to adjust an appropriate level of glucose through releasing hormones. When the glucose level becomes low, then the glucagon is released by the α-cells, which helps to the liver to convert the glycogen into glucose. Similarly, when the glucose level rises, then the insulin is released by the β-cells, which helps to store the excessive glucose into glycogen (Ajmera et al., 2013) to maintain the glucose level as normal. The liver plays the central role for regulating the glucose and glycogen, and it allows to behave as a distributor of nutrients through blood to other tissues.

4.1 Normal Homeostasis System

The main components of GH are depicted in Fig. 1, in which the GH comprises the low, high and normal levels of glucose in the body and biological organs. In order to define it formally, we use eight landmarks nodes (*Hi, No, Lo, Ac, Bc, Li, St, Tr*) shown in Fig. 2 corresponding to the functional behaviour of GH. These landmark nodes are identified through a literature survey (Ajmera et al., 2013; Bolie, 1961; Li et al., 2006; Silber et al., 2007) and discussion with medical practitioners. Further the identified landmark nodes are used to describe a very abstract functionality of GH. We introduce the required elements to define GH as follows:

[1] The 'normal GH system' is when the GH system functions as it should, i.e., there are no abnormal behaviours exhibited by the system.

Definition 1 (The GH System). *Given a set of nodes N, a transition T, is a pair (i, j), with i, j ∈ N. A transition is denoted as i→j. The GH system is a tuple GHS = (N, T, N_0) where:*

- *N = { Hi, No, Lo, Ac, Bc, Li, St, Tr } is a finite set of landmark nodes in the GH network;*
- *T ⊆ N × N = { No □ Hi, Hi □ No, No □ Lo, Lo □ No, Hi □ Hi, No □ No, Lo □ Lo, Hi □ Bc, Lo □ Ac, Bc □ Li, Ac □ Li, Li □ St, Li □ Tr, St □ No, Tr □ No, St □ Hi, Tr □ Lo, Tr □ Hi }, is a set of transitions to present data flow between two landmark nodes. The last three transitions denoted in dash lines in Fig. 2 represent the case of failure of GH;*
- *N_0 = No is the initial landmark node (normal glucose level);*

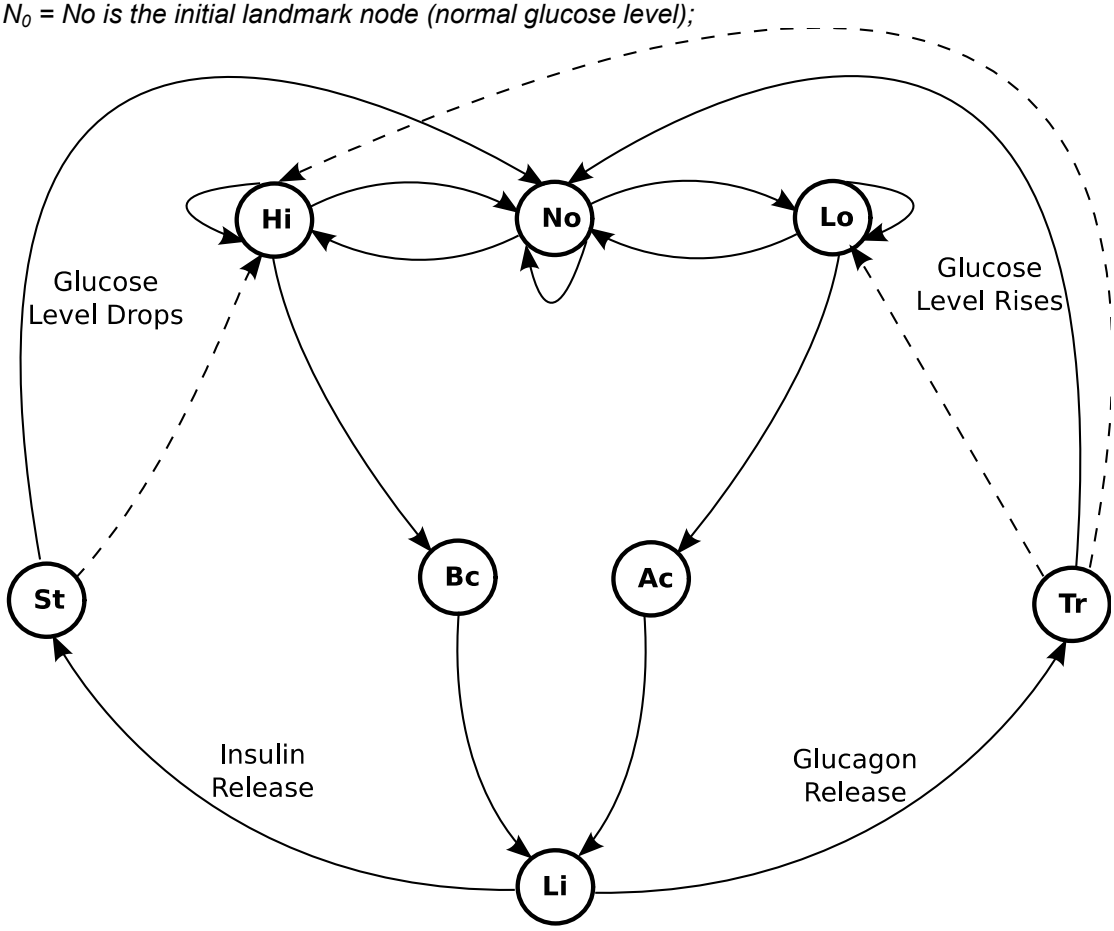


Figure 2: The Glucose Homeostasis Automata

4.2 Abnormal Homeostasis System

The abnormal homeostasis system of GH is depicted in Fig. 3. In particular, there are two types of diabetes: *insulin-dependent diabetes* (also know as *type 1 diabetes*) and *non insulin-dependent diabetes* (also know as *type 2 diabetes*). Insulin-dependent diabetes may be caused by no insulin secreted or insufficient insulin, which can be occurred due to defect in β-cells. However, in case of non insulin-dependent diabetes, the β-cells release insulin but the insulin receptors do not work due to insulin resistance in the cells, so insulin has no effect. In fact, in both cases, the glucose concentration level increases in the body, so glucose level becomes high in the body. Low glucose level is caused by defect in α-cells, abnormal release of glucagon, excess insulin or excess glucagon secretion in the body. Sometime excess glucagon secretion and defects in β-cells indicates the persistent high glucose level in the blood that can be classified as hyperglycemia-induced diabetes complications (Ajmera et al., 2013).

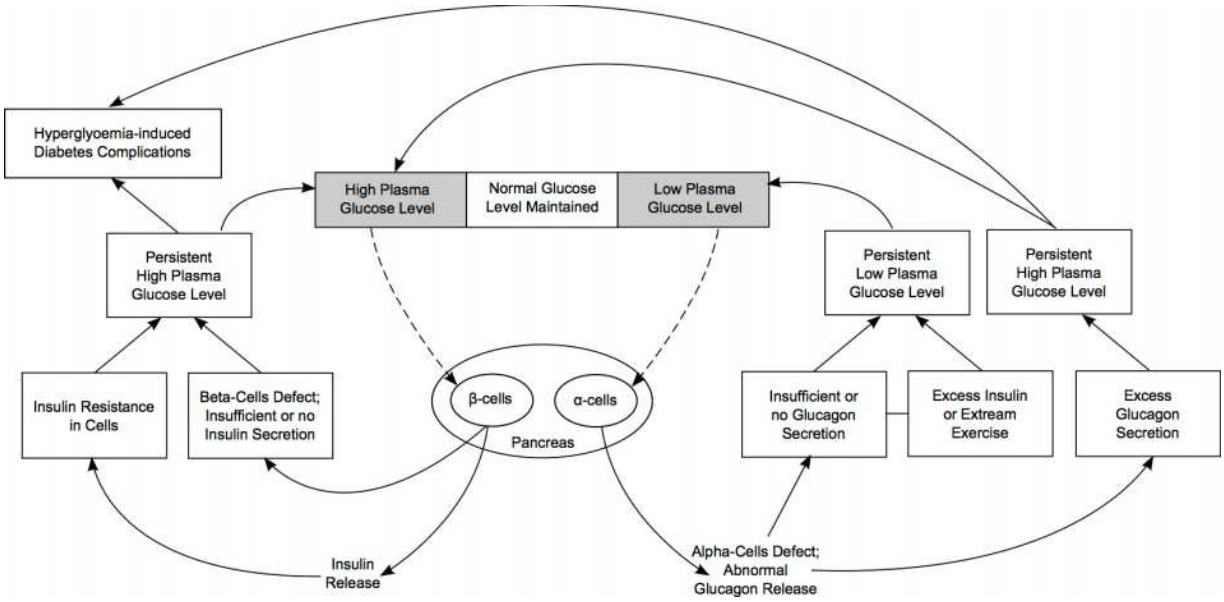


Figure 3: Abnormal GH System

4.3 Glucose Level in Blood

The glucose level in blood can be measured through analysing an amount of glucose in the blood. There are two types of well-known tests used to detect abnormality in the blood: Fasting Plasma Glucose (FPG) test and Oral Glucose Tolerance Test (OGTT) (Siperstein MD, 1975).

Property 1 (Blood Glucose Level). *The blood glucose level defines different stages, such as hyperglycemia, hypoglycemia and normal. The glucose level is low (hypoglycemia) if FPG* $\in$ *0 .. 69 or OGTT* $\in$ *0 .. 69, and the glucose level is high (hyperglycemia) if FPG* $\geq$ *126 or OGTT* $\geq$ *200, and the glucose level is normal if FPG* $\in$ *70 .. 99 or OGTT* $\in$ *70 .. 139. We classify pre-diabetes to be the range where FPG* $\in$ *100 .. 125 or OGTT* $\in$ *140 .. 199.*

5 Design and Development of GH

This section presents formalization, simulation and implementation approaches for designing and developing the GH virtual environment model. The developed formal models can be used for various purposes in the development life-cycle of IIP, such as requirement analysis, closed-loop modeling, simulation development and hardware implementation. Similarly, the implemented GH virtual environment can be used as a test bench for clinical trials of IIPs. Formal development approach, design and development are given as follows:

5.1 Formalization of GH

To develop the formal specification of GH, we use the Event-B modeling language (Abrial, 2010) that supports an approach based on incremental refinement to design the whole system in several layers. The developing specification contains an abstract model and a set of refined models. The initial model captures the basic behaviour and biological requirements of the GH in an abstract way, while the subsequent refinements are used to introduce α-cells and β-cells of the pancreas, functional behaviour of liver to convert and to store the glucose, abnormal conditions of the pancreas, diabetic conditions, and diabetes complications, and blood sugar concentration for assessing diabetes. The developed system results the dynamic behaviours of virtual GH biological environment that covers the both normal and abnormal behaviours (hyperglycemia, hypoglycemia or diabetic complications). A list of safety proper-

ties are defined at each incremental level to guarantee the correctness of designed virtual biological environment model for GH. The interested reader can check the stepwise development, including safety properties and proof details in (Neeraj et al., 2014). It should be noted that this developed formal model is our base model that can be used for different purposes later. For example, it is very useful at early stage of the system development specially for developing the closed-loop model for analysing the system requirements of an IIP. More benefits and usability of this formal model are described in the later sections.

5.2 Development of GH Simulator

Fig. 4 depicts a GH simulation framework. The model of physical processes is defined through the physiology of GH, physiology of the pancreas, physiology of the liver, insulin-glucose dynamics, glucagon-glucose dynamics, GH abnormality, and physiology of pancreatic α-cells and β-cells represented in the round rectangular box. This physical process model is derived from the formal specification of GH, modeling tools and computation tools, and required parameters. The formal model contains an abstract model and set of refined models that can be further enriched though the introduction of complex expressions. Moreover, we can select any level of refinement as per our simulation requirements. There are several existing modeling tools, such as finite elements, finite differences, lumped elements, that can be used during the simulation development. The computation tools provide a library for numerical methods that can be used for complex computation, such as dynamics calculation for the selected components. Mainly, these tools include ordinary differential equation (ODE) solvers, and linear and non-linear algebraic solvers. The simulation model is designed and supervised by the simulation kernel, which arbitrates their communication between the components. The user interface and visualization of GH component provides an interface for basic user input through the haptic interface tools and basic output though visualization tools.

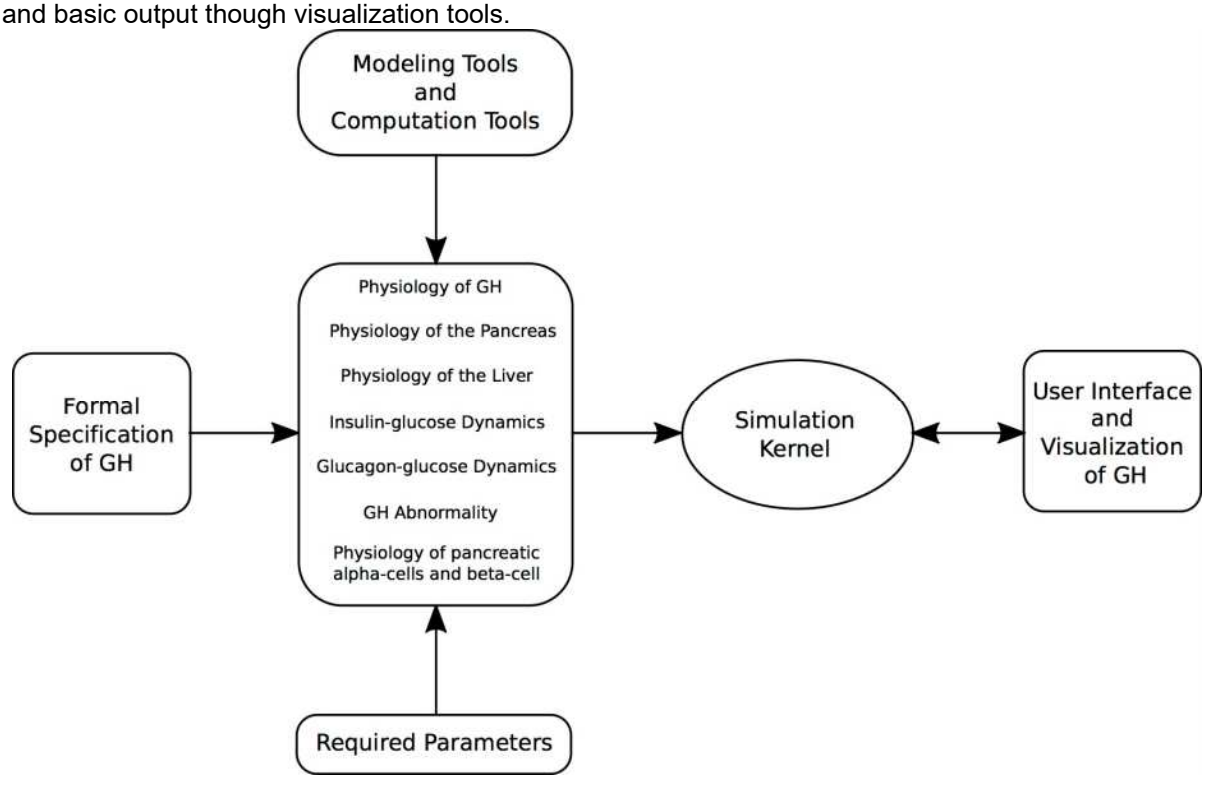


Figure 4: GH Simulator Framework

5.3 Implementation of GH System

Fig. 5 depicts an implementation of GH on the hardware platform. The modeling and implementation of GH model is represented in the round rectangular box, which is derived from both the formal speci-

fication of GH and GH simulation, which contain the formal and simulated models of the physical processes in form of discrete and continuous behaviour. The formal models contain an abstract model and a set of refined models that can be further enriched though the introduction of complex expressions. The GH simulation contains the complex mathematical equations in form of linear equation, non-linear equation and ODE of system dynamics for each and every component. The modeling and implementation blocks use existing tools like Matlab or Labview to implement the GH system, and then further it can be embedded on the FPGA hardware platform. This block also communicates with FPGA board, and the block of user interface and simulation. The user interface and visualization of GH component provides an interface for basic user input through the haptic interface tools and basic output though visualization tools according to the embedded GH model on the FPGA board.

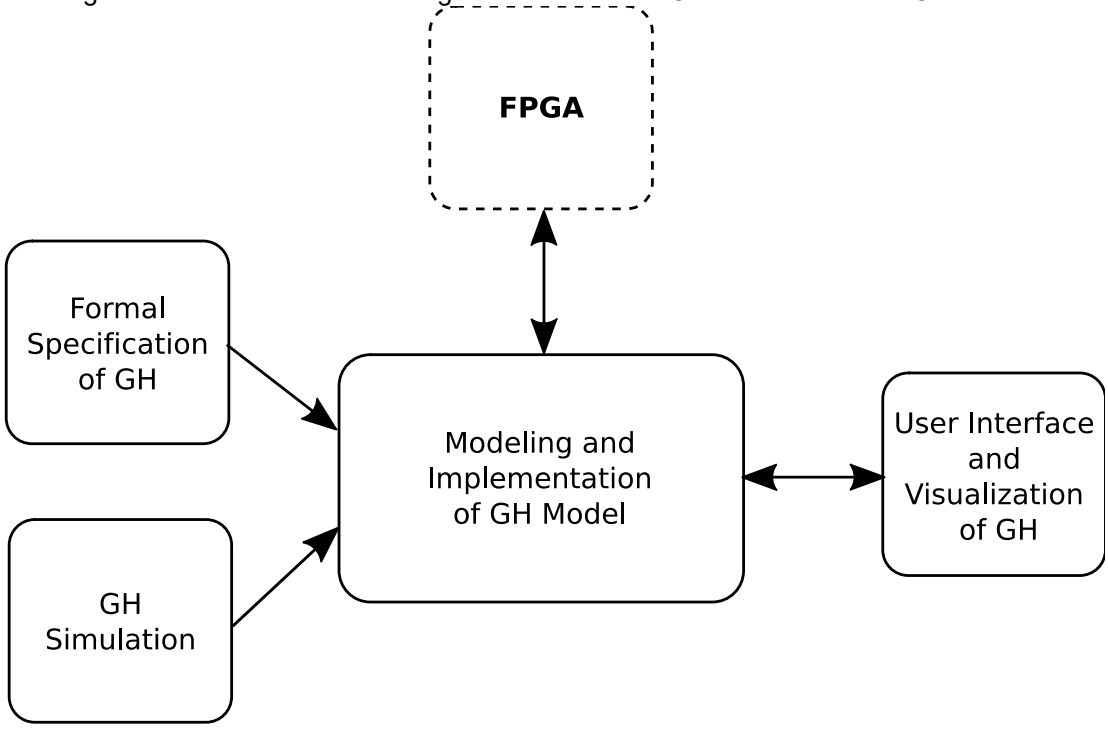


Figure 5: Hardware Implementation of GH

6 Usability of Environment Model

This section presents the usability of the virtual environment modeling for developing the IIP devices as follows:

- **Behavioural Requirements.** The closed-loop model is an integration of formal model of IIP and GH, which can be developed using refinement approach in the early stage of the system development. The closed-loop system exposes several conditions for normal and abnormal diabetic represented in Fig. 3 that occur due to several types of malfunctions.
- **To Discover Essential Safety Properties.** The closed-loop model provides high-assurance for safety and security. The environment based closed-loop modeling approach offers to finding ambiguities and inconsistencies in the IIP specification. In addition, the environment model can be reused with other similar systems, wherever a system requires environment modeling for verification and validation.
- **Patient Safety in Closed-loop.** The closed-loop system allow to monitor continue to the system requirements as per the physiological needs. Each patient has specific needs according to the diabetic symptoms, which can be diagnosed by using an IIP. In fact, an IIP has configuration parameters, which allow to configure the device for each patient by doctors by analyzing the chronicle conditions.

- **To Generate Automatic Test Cases.** Testing plays most significant role in the process of software development for checking the correctness of system implementation. The formal model of GH model can be used to automate the process of test case generation. The generated test cases can be used for testing the codes for implemented virtual environment model before embedding to the FPGA board. In fact, we do not require to prepare any test cases separately, and we can derived it from our developed formal models.
- **Test Bench for Clinical Trials of IIPs.** A test bench is a virtual environment that simulates a desired behaviour of a physical system that can be used to verify the correctness and soundness of the developing devices. The developed model on hardware platform can be used for clinical trials by the medical industries to meet all the explicit and implicit requirements of IIPs. Moreover, the developed platform can be beneficial to analyze patient specific scenarios by modifying the GH parameters. In addition, the same hardware platform can also assist to regulators, such as the FDA, to perform clinical trials based on different criteria and to check all the required functionalities of the devices before certifying these devices.

7 Conclusion and Future Challenges

In this paper, we have presented the formal development of a GH virtual environment model, simulation framework and hardware implementation. The developed formal model can be used for validating the system requirements, finding missing requirements, validating assumptions and strengthening the existing requirements during the process of requirement engineering. The simulation framework can be used to develop a simulation model of GH based on complex expressions using linear and non-linear equation and the developed formal model. The hardware implementation architecture can be used to implement the GH system on the FPGA hardware board using the developed formal and simulation models. Finally, the GH virtual environment model embedded on the FPGA board can be used as a test bench for IIPs that can be used for clinical trials. Moreover, the formal model can also used for generating test cases that can be used to test the implemented codes. This is the first computational model based on logical concepts to simulate the GH behaviour in order to analyze the normal and diabetic conditions. This is a promising simulated biological environment model that can be used during the development of the product life cycle. Moreover, this developed virtual environment model can be used to obtain certification for the medical devices related to the homeostasis system, such as IIPs. This environment model can also be used as a diagnostic tool to diagnose or understand the patient requirements.
So far, in the direction of completion of the proposed work, we have developed only the formal model of GH, which is ready to use for different purposes during the development life cycle. Our ultimate long-term goal is to develop the simulation using linear, non-linear and ODE equations, and finally implement it on the FPGA board, so that we can use it as a test bench for developing IIPs. So, we plan to integrate the developed formal model of IIP with GH virtual environment model to model the closed-loop system for verifying the desired behaviour under the relevant safety properties to guarantee the correctness of the functional behaviour of an IIP in chaotic environment. Further, we intend to investigate the simulation approach to develop a simulation model using the proposed simulation framework, and finally, we intend to implement and embed the virtual environment model, deriving from simulation and formal models, on the FPGA board for developing a test bench for IIPs.

8 Literature

Abrial, J.-R., 2010. Modeling in Event-B: System and Software Engineering. Cambridge University Publication.

Ajmera, I., Swat, M., Laibe, C., Le Novère, N., Chelliah, V., 2013. The impact of mathematical modeling on the understanding of diabetes and related complications. CPT Pharmacomet. Syst. Pharmacol. 2, e54.

Bolie, V.W., 1961. Coefficients of normal blood glucose regulation. J. Appl. Physiol. 16, 783–788.

Chay, T.R., Keizer, J., 1985. Theory of the effect of extracellular potassium on oscillations in the pancreatic beta-cell. Biophys. J. 48, 815 – 827. doi:http://dx.doi.org/10.1016/S0006-3495(85)83840-6

Chen, Y., Lawford, M., Wang, H., Wassyng, A., 2013. Insulin Pump Software Certification, in: Foundations of Health Informatics Engineering and Systems, Lecture Notes in Computer Science. Springer Berlin Heidelberg.

Li, J., Kuang, Y., Mason, C.C., 2006. Modeling the glucose–insulin regulatory system and ultradian insulin secretory oscillations with two explicit time delays. J. Theor. Biol. 242, 722 – 735. doi:http://dx.doi.org/10.1016/j.jtbi.2006.04.002

Silber, H.E., Jauslin, P.M., Frey, N., Gieschke, R., Simonsson, U.S.H., Karlsson, M.O., 2007. An Integrated Model for Glucose and Insulin Regulation in Healthy Volunteers and Type 2 Diabetic Patients Following Intravenous Glucose Provocations. J. Clin. Pharmacol. 47, 1159–1171. doi:10.1177/0091270007304457

Singh, N.K., 2013. Using Event-B for Critical Device Software Systems. Springer-Verlag GmbH.

Singh, N.K., Wang, H., Lawford, M., Maibaum, T.S.E., Wassyng, A., 2014. Formalizing the Glucose Homeostasis Mechanism, in: Digital Human Modeling. Applications in Health, Safety, Ergonomics and Risk Management - 5th International Conference, DHM 2014, Heraklion, Crete, Greece, June 22-27, 2014. Proceedings. pp. 460 – 471. doi:10.1007/978-3-319-07725-3_46

Siperstein MD, 1975. The glucose tolerance test: a pitfall in the diagnosis of Diabetes Mellitus. Adv Intern Med 20, 297 – 323.

9 Author CV

Neeraj Kumar Singh

Dr. Neeraj Kumar Singh is an Associate Professor in ACADIE research team at INPT-ENSEEIHT/IRIT, University of Toulouse, France since September 2015. He holds PhD in computer science from University of Lorraine, France (2011). From 2012 to 2013, he was a research associate in the Computer Science Department of University of York, UK, working on the EPSRC funded project. From 2013 to August 2015, he was a research fellow and team leader in the Centre for Software Certification (McSCert) at McMaster University, Canada, working on ORF-RE and APC funded projects. He leads his research in the area of theory and practice of rigorous software engineering and formal methods to design and implementation of safe, secure and dependable critical systems related to automotive, medical, avionic and nuclear domains.

Model-based offline and online testing for medical software[1]

Paolo Arcaini[1], Elvinia Riccobene[2], Angelo Gargantini[3]

[1]Charles University in Prague, Faculty of Mathematics and Physics, Czech Republic, arcaini@d3s.mff.cuni.cz

[2]Dipartimento di Informatica, Università degli Studi di Milano, Italy, elvinia.riccobene@unimi.it

[3]Department of Economics and Technology Management, Information Technology and Production, Università degli Studi di Bergamo, Italy, angelo.gargantini@unibg.it

Abstract

Software controlling medical devices is safety-critical since human safety depends upon its correct operation. Although different standards exist for the certification of medical device software, they are quite general and do not indicate which methods and techniques must be adopted to guarantee system safety and reliability. In this paper, we present a rigorous development approach, based on the Abstract State Machine formal method, which helps the designer in formalizing the requirements and mapping them to the implementation. In particular, we show how the proposed process permits to check that the implementation behaves as expected; for this purpose, we present two model-based testing techniques, working offline and online. We show the application of these two techniques to the implementation of a medical device component used to measure stereoacuity.

Keywords

Medical software, certification, formal methods, model-based testing, runtime verification

[1] The research reported in this article has been partially supported by Charles University research funds PRVOUK.

1 Introduction

Medical devices are the most widespread examples of safety-critical systems since malfunctions of the software controlling a medical device can lead to injuries or even death for humans. Therefore, methods and techniques to assure medical software safety and reliability are highly demanded.
In the last years, several standards have been proposed for the validation of medical devices: ISO 13485 (ISO, 2003), ISO 14971 (ISO, 2007), IEC 60601-1 (IEC, 2005), EU Directive 2007/47/EC (EU, 2007), etc. They are mainly related to the correct operation of electrical components of the device, and impose constraints on physical aspects of the hardware parts. These standards are not adequate to regulate the design and deployment of the software component.
Currently, the main references concerning regulation of medical software development are two: the standard IEC 62304 (IEC, 2006) from the International Electrotechnical Commission, and the "General Principles of Software Validation" (FDA, 2002) from the Food and Drug Administration (FDA). Both documents promote the adoption of well-known software engineering activities for medical software design and deployment, although they do not provide any indication regarding life cycle models, or methods and techniques to assure safety and reliability. However, both IEC standard and FDA principles aim for the use of rigorous approaches, based on the use of formal methods (IEC, 2006), (Jetley, et al., 2006). Indeed, formal models permit the designers to specify what the software is intended to do in a rigorous and precise way, avoiding ambiguities and misunderstandings. They can be used, already at the early stages of the software development, to prove that safety-critical properties are satisfied. Moreover, formal models can be exploited to derive test cases in a software independent way and to guarantee conformance of device software to behavioral models specifying safe device operation (since, most of the times, software for medical devices is not developed from scratch).
In the context of the project 3D4amb, we are applying a rigorous approach (Arcaini, et al., 2015b) to the development of software applications that must be used by optometrists and ophthalmologists to detect visual problems. Such approach is based on the use of Abstract State Machines (Börger & Stärk, 2003) (ASMs), which are an extension of Finite State Machines (FSM). Although its rigorous mathematical foundation, practitioners need no special training to use the ASM method since ASMs can be correctly understood as pseudo-code (or virtual machines) working over abstract data structures. The ASM-based design process follows an incremental life cycle model based on ASM model refinement, and embraces the main software engineering activities: requirements specification, validation (simulation and testing), and verifications (model checking and runtime monitoring). These activities can be applied at any desired level of detail. The process can guide the development of software and embedded systems seamlessly from requirements capture to their implementation, and this has been shown by numerous and successful case studies (Börger & Stärk, 2003).
In this paper, we present our experience in applying the ASM method to the development of an application used to detect amblyopia, a visual problem often known as *lazy eyes*. We briefly present the entire process and then we focus on the activities of model-based offline and online testing, that are used to check the correctness of the software behavior w.r.t. the intended requirements.

2 Stereoscopic Acuity Measurer

The aim of the project 3D4amb[2] is developing applications that can be used by optometrists and ophthalmologists to detect visual problems. One of these applications, 3DSAT (Gargantini, et al., 2014), permits to measure the stereoacuity of young patients and to detect amblyopia; the application is available for PC and smartphone (see Figure 1), and it is currently under evaluation in an Italian hospital. SAM (Stereoscopic Acuity Measurer) is the main component of 3DSAT: it decides the stereo depth of the image to be shown to the patient and when to stop the test, and provides the final exam result (stereoacuity certification). We report the informal requirements concerning the operation of SAM.

[2] http://3d4amb.unibg.it/

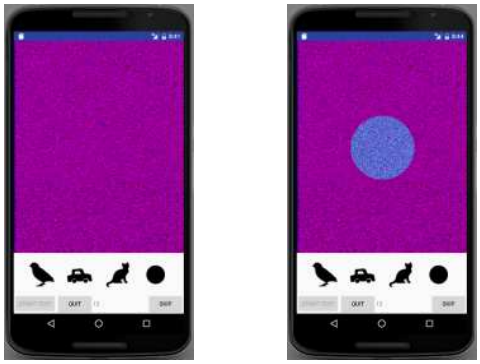


Figure 1: SAM - Stereo Acuity Test Application

The patient's stereoacuity is certified by showing different stereo random dot images (bird, car, cat, or circle) using a 3D anaglyph technology in a 3D stereo monitor (see right part of Figure 1) at six difficulty levels. The image contains some dots that are sent only to one eye (either right or left) while common dots are sent to both eyes. The test starts at the easiest level (level 6). When the patient identifies the displayed image, the *level decreases* (i.e., the test becomes more difficult); this process is iterated until the most difficult level (level 1) is reached. A patient is certified at a given level if (s)he recognizes three times the images displayed at that level. If the patient answers incorrectly, (s)he can try another time at the same level. If (s)he fails again, the *level increases* (i.e., the test becomes easier): now (s)he can only try to be certified at the upper level. If the patient fails twice the 6th level, the test stops with no certified level. At any time, a user can QUIT the test or SKIP an image. A SKIP answer is considered as a wrong answer (i.e., if the patient skips twice at the same level, the level increases). The SAM component can stop in two ways: (a) by certifying the patient at level *i,* if the patient has identified three images at level *i*, but has failed at level *i*-1 (if any); (b) without certification, if the patient has failed in completing the test or the doctor has quit the exam.

3 ASM-based development process

The ASM design formal method is based on the use of Abstract State Machines. They are transition systems that extend the Finite State Machines by replacing unstructured control states by algebraic structures, i.e., domains of objects with functions and predicates defined on them. A *state* represents the instantaneous configuration of the system and transition rules describe the change of state. A *run* is a (finite or infinite) sequence of states s_0, s_1, …, s_n, …, where each s_i is obtained by applying the transition rules at s_{i-1}. There exists a classification of functions; in this paper, we only consider *controlled* and *monitored* functions. Controlled functions can only be updated by transition rules and represent the internal *memory* of the ASM. Monitored functions, instead, cannot be updated by transition rules, but only by the *environment*; they represent the inputs of the machine.

The ASM formal method allows an iterative design process, shown in Figure 2, based on model refinement. Validation and verification (V&V) are fully integrated into the process, and are possible at any level of abstraction. Tools supporting the process are part of the ASMETA (ASM mETAmodeling) framework[3] (Arcaini, et al., 2011).

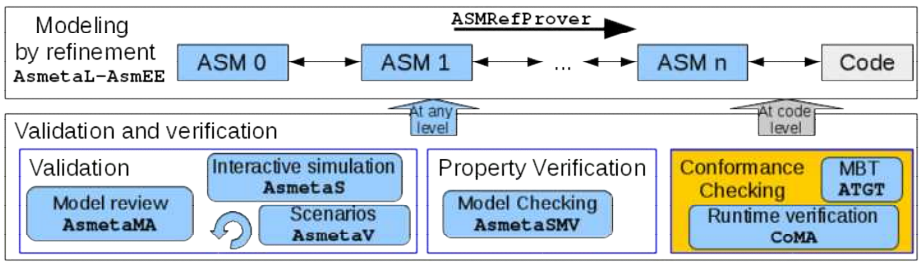


Figure 2: ASM-based process

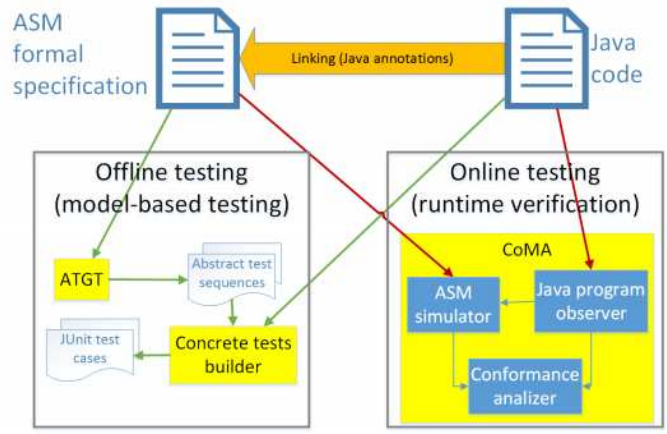
The process starts by *requirement elicitation*, namely the activity of developing a high-level model, called *ground model* (ASM 0 in Figure 2), which expresses the intended system behavior according to the informal requirements, generally given as a text in natural language. The ground model is expressed by using terms of the application domain, and its specification possibly involves all the stakeholders. This model should *correctly* reflect the intended requirements and should be *consistent*, i.e.,

[3] http://asmeta.sourceforge.net

without possible ambiguities of initial requirements. It does not need to be *complete*, i.e., it may not specify some given requirements. In the ASMETA framework, the tool **AsmEE** is available for ASM models editing in terms of the concrete syntax **AsmetaL** (Gargantini, et al., 2008).
Starting from the ground model, through a sequence of step-wise refined models, further functional requirements can be specified till a complete description of the system. The refinement process permits to tackle the complexity of the system, and allows to bridge, in a seamless manner, specification to code. At each refinement step, the refined models should be proved to be correct w.r.t. the abstract one. Such refinement correctness proof can be done by hand or, for a particular kind of refinement called *stuttering refinement*, by using the tool **ASMRefProver** (Arcaini, et al., 2016b).
The modeling activity can be complemented with different V&V activities. They are applicable at each level of refinement, starting from the ground model. Model validation helps to ensure that the specification really reflects the intended requirements, and to detect faults and inconsistencies as early as possible with limited effort. Model verification requires the use of more expensive and accurate methods, and should be applied when the developer has enough confidence that the model under development is correct. ASM model validation is possible by means of model simulation (by using **AsmetaS** (Gargantini, et al., 2008)) and by construction of critical scenarios (by the validator **AsmetaV** (Carioni, et al., 2008)). A further validation technique can be used to determine if a model has sufficient *quality* attributes, as minimality, completeness, consistency. It is a form of static analysis and is called *model review*. The tool **AsmetaMA** (Arcaini, et al., 2010b) allows automatic ASM model review. Formal verification of ASMs is possible by means of the model checker **AsmetaSMV** (Arcaini, et al., 2010a), that supports both *Computation Tree Logic* (CTL) and *Linear Temporal Logic* (LTL) formulas.
If an implementation is available, either derived from the model (as last low-level refinement step) or externally provided, also conformance checking is possible. Both model-based testing (MBT) and runtime verification can be applied to check if the implementation conforms to its specification (Arcaini, et al., 2014a). We support conformance checking w.r.t. Java code. The tool **ATGT** (Gargantini, et al., 2003) can be used to automatically generate tests from ASM models and, therefore, to check the conformance offline; **CoMA** (Arcaini, et al., 2012), instead, can be used to perform runtime verification, i.e., to check the conformance online. The application of the validation and verification activities has been already shown in (Arcaini, et al., 2015a) for the case study, and in (Arcaini, et al., 2016a) for the requirements specification of a hemodialysis device. Instead in the next sections we focus on the application of the conformance checking activities to the case study.

4 Model-based offline and online testing

We here describe how to perform conformance checking between an implementation and its formal specification (the last step of the ASM-based development process described in Sect. 3). We support two approaches for conformance checking: (a) offline testing (also known as model-based testing), (b) online testing (or runtime verification).



The two approaches are depicted in Figure 3 and described in Sect. 4.2 and 4.3. In order to show their application, we report in Table 1 an excerpt of the last refined ASM formal specification of the case study (on the left), and the corresponding Java implementation (on the right). Both conformance checking techniques require a way to link elements of the code with corresponding elements in the model; such a linking is described in Sect. 4.1.

Figure 3: Model-based offline and online testing

```
asm certifierRaff5

signature:
    enum domain UserAnswers = {SKIP | QUIT |
                               CONTINUE}
    enum domain Shapes = {BIRD|CAR|CAT|CIRCLE}
    enum domain OutMessage = {CERTIFICATE |
                              NOTCERTIFICATE}
    ...
    dynamic controlled test: Boolean
    dynamic controlled outMessage: OutMessage
    dynamic monitored displayedShape: Shapes
    dynamic monitored userAnswer: UserAnswers
    dynamic monitored userSelectedShape: Shapes
    ...
definitions:
    ...
    macro rule r_test =
        switch(userAnswer)
            case QUIT:
                r_goOut[]
            case SKIP:
                r_skip[]
            case CONTINUE:
                r_checkAnswer[]
        endswitch

    main rule r_Main =
        if test then
            r_test[]
        endif

default init s0:
    function test = true
    …
```

```
@Asm(asmFile = "models/certifierRaff5.asm")
public class SAM {
    @FieldToFunction(func = "outMessage")
    public OutMessage message;
    private boolean testRunning;
    @Monitored(func = "displayedShape")
    public ShapeImage shownShape;
    @Monitored(func = "userAnswer")
    public UserAction userAnswer;
    @Monitored(func = "userSelectedShape")
    public ShapeImage userSelectedShape;
    …

    @StartMonitoring
    public SAM() { … }

    @RunStep
    public void testShapeRecognition() {
        …
    }

    @MethodToFunction(func = "test")
    public boolean isTestRunning() {
        return testRunning;
    }

    @MethodToFunction(func = "levelTest")
    public int levelCertificate() {
        return certifier.getCurrentDepth();
    }
    …
}
```

Table 1: SAM formal specification (left) and Java implementation (right)

4.1 Establishing conformance between specification and code

For both offline and online testing, we need to define the notion of *conformance* between an implementation and its formal specification. In (Arcaini, et al., 2013), we proposed an approach to define the conformance relation by using Java annotations[4] originally introduced in (Arcaini, et al., 2012). These annotations are used to link a Java class with its corresponding ASM model. The annotations used in our case study are listed in the following (we refer to (Arcaini, et al., 2015b) for a complete description of the annotations).

- Annotation used to link a Java class to its ASM model:
  - `@Asm`: it is a class annotation and permits to identify the formal specification (specified using the string attribute `asmFile`) that corresponds to the annotated class. In the case study, the Java class `SAM.java` is linked to the ASM specification `certifierRaff5.asm`.
- Annotations used to link the class data with the ASM signature:
  - `@FieldToFunction`: it annotates fields and permits to link a Java field with an ASM function. The annotation has a mandatory attribute `func` for specifying the function name. In the case study, the Java field `message` is linked to the ASM function `outMessage`.
  - `@MethodToFunction`: it is similar to `@FieldToFunction`, but it annotates methods. The annotated methods are required to be *pure*[5]. In the case study, method `isTestRunning()` is linked with the ASM function `test`, and method `levelCertificate()` with the ASM function `levelTest`.

[4] A Java annotation is a meta-data tag that permits to add information to code elements (class declarations, method declarations, etc). Annotations are defined similarly as classes.
[5] A method is *pure* if it is side effect free with respect to the program state: it returns a value but does not assigns values to fields.

  - `@Monitored`: it has the same structure and applicability of `@FieldToFunction`, but it must be used to annotate fields that take their values from the environment (e.g., the user input or streams as files, sockets, etc.), and are not updated by a class method (differently from fields annotated with `@FieldToFunction`). Fields annotated with `@Monitored` can only be linked to ASM monitored functions that, indeed, represent the part of the ASM state determined by the environment. In our case study, fields `userAnswer` and `userSelectedShape` (representing the patient's choice) are linked to ASM functions having the same name, and field `shownShape` (representing the image shown on the screen) is linked with the ASM function `displayedShape`.
- Annotations used to link the execution of the Java program with a run of the corresponding model:
  - `@StartMonitoring`: it annotates constructors. An annotated constructor identifies a Java initial state from which the conformance checking is required. In our case study, the unique constructor is selected for conformance checking.
  - `@RunStep`: it annotates methods (called *changing methods*) that modify the observed state, i.e., the values of the linked fields (those annotated with `@FieldToFunction`) and the return values of the linked pure methods (those annotated with `@MethodToFunction`). In our case study, the only changing method is `testShapeRecognition()`.

The linking obtained by the use of the annotations allows to give the following *definitions of conformance* between an instance `obj` of the Java class and the corresponding ASM model `spec`:

- **State conformance**: *the Java state of `obj` and the ASM state of `spec` are conformant* if the values of the fields annotated with `@FieldToFunction` and the values returned by the methods annotated with `@MethodToFunction` are equal to the values of the corresponding ASM functions.
- **Step conformance**: *`obj` is step conformant with `spec`* if their states are conformant before and after the execution of a Java changing method of `obj` and the execution of a step of `spec`.
- **Run conformance:** *`obj` is run conformant with `spec`* if the following conditions hold: (a) the initial state of `obj` is state conformant with one and only one initial state of `spec`; (b) *`obj`* is step conformant with `spec` upon the execution of each changing method of `obj`.

4.2 Offline testing

The first approach for checking the conformance between the implementation and the ASM formal specification is *model-based testing* (Utting & Legeard, 2006), in which some tests are derived from the specification according to some coverage criteria and are then executed over the implementation. Since such activity is performed *before* the program deployment, we call it *offline testing*.

Our model-based testing approach is depicted in the left side of Figure 3:

- The tool ATGT (Gargantini, et al., 2003) derives from the ASM formal specification some *abstract test sequences* with the aim of achieving some testing criteria (Gargantini & Riccobene, 2001). For example, a test suite satisfies the *basic rule coverage* (BRC) criterion if every rule *r* is fired in at least one state of a test sequence, and there exists a (possibly different) test sequence in which *r* does not fire in some state.
- The abstract test sequences are then *concretized* into *JUnit test cases* for the implementation (Arcaini, et al., 2013). The translation takes advantage of the linking previously described.

We here explain how we have applied the offline testing approach to the SAM application.

Each coverage criterion produces a set of *test predicates* representing the testing goals that must be covered by the generated tests. We generated 48 test predicates for the specification: 16 have been generated by the BRC criterion, and 32 by the UR criterion that requires that all the updates are covered. One of the test predicates generated by BRC is

`test and userAnswer=CONTINUE and userSelectedShape=displayedShape and levelTest<=1 and rightAnswer>1`

that requires to visit the rule in which the user recognizes the displayed image for the third time and certifies the last level.
For each test predicate *tp*, we generated an abstract test sequence (i.e., an ASM run) in which *tp* is satisfied in some state of the sequence. Since a test sequence can cover more than one test predicate, we were able to cover all the test predicates with only 9 tests.
Then, we translated the abstract test sequences in JUnit tests. For example, a generated abstract test sequence is shown in the left part of Table 2, and its corresponding JUnit test case in the right part.

---- state 0 -----
-- controlled --
test = true
levelTest = 6
outMessage = UNDEF
-- monitored –
userAnswer = CONTINUE
userSelectedShape = CIRCLE
displayedShape = CAT
---- state 1 -----
-- controlled --
test = true
levelTest = 6
outMessage = UNDEF
 	
....
 	
---- state 3 -----
-- monitored –
userAnswer = CONTINUE
userSelectedShape = CAT
shownShape = CAT
---- state 4 -----
test = false
levelTest = 1
outMessage = CERTIFICATE
	`@Test`
	`public void test() {`
	`  SAM sut = new SAM();`
	`  // conformance checking`
	`  assertEquals(true, sut.isTestRunning());`
	`  assertEquals(6, sut.levelCertificate());`
	`  assertEquals(OutMessage.UNDEF, sut.message);`
	
	`  // set inputs`
	`  sut.userAnswer = UserAction.CONTINUE;`
	`  sut.userSelectedShape = ShapeImage.CIRCLE;`
	`  sut.shownShape = ShapeImage.CAT;`
	`  // perform step`
	`  sut.testShapeRecognition();`
	`  // conformance checking`
	`  assertEquals(true, sut.isTestRunning());`
	`  assertEquals(6, sut.levelCertificate());`
	`  assertEquals(OutMessage.UNDEF, sut.message);`
	`  …`
	`  // set inputs`
	`  sut.userAnswer = UserAction.CONTINUE;`
	`  sut.userSelectedShape = ShapeImage.CAT;`
	`  sut.shownShape = ShapeImage.CAT;`
	`  // perform step`
	`  sut.testShapeRecognition();`
	`  // conformance checking`
	`  assertEquals(false, sut.isTestRunning());`
	`  assertEquals(1, sut.levelCertificate());`
	`  assertEquals(OutMessage.CERTIFICATE, sut.message);`
	`}`

Table 2: Abstract test sequence (left) and JUnit test case (right)

The concretization works as follows:

- an instance of the class is built using the constructor annotated with `@StartMonitoring`, and it is associated with the reference variable *sut*;
- the conformance of the initial state is checked (see below for more details);
- then, for each ASM step, the following instructions are added to the test:
  - fields annotated with `@Monitored` are updated with the value of the corresponding ASM function; in the case study, fields *userAnswer*, *userSelectedShape*, and *shownShape* are updated with the values of the ASM functions *userAnswer*, *userSelectedShape*, and *displayedShape*;
  - the changing method is called; in the case study, method *testShapeRecognition*() is called;
  - the conformance checking is executed:
    - for each field *f* annotated with `@FieldToFunction`, the following assertion is built
      assertEquals(*v, sut.f*);
      which states that the value of *sut.f* must be equal to *v*, where *v* is the value of the corre-

sponding ASM function in the abstract test sequence; in the case study, field *message* is compared with the value of the corresponding ASM function *outMessage*.

- for each method *m* annotated with `@MethodToFunction`, the following assertion is built:

  assertEquals(*v, sut.m*());

  which states that the value returned by *sut.m*() must be equal to *v* (where *v* is defined as before); in the case study, methods *isTestRunning*() and *levelCertificate*() are compared with the values of the corresponding ASM functions *test* and *levelTest*.

We repeatedly executed the generated tests over the different versions of the implementation, and we were able to find different faults. For example, we found a fault due to a misunderstanding of the requirements in the implementation. Requirements state that a patient is certified at level *i* if (s)he recognizes three images at level *i*, *without moving to another level in between*. The faulty Java implementation, instead, certified the patient also if the three recognitions were not consecutive (e.g., (s)he recognized an image at level 2, moved to level 1, failed twice at level 1, and then recognized two images at level 2).

The generated tests cover 70.2% of the program instructions and 41.6% of the branches; the mutation score (computed with PIT[6]) is 45.7%. These data show that more testing is needed.

4.3 Online testing

The offline testing approach described in Sect. 4.2 assures that the implementation is conformant with the specification over the produced tests (i.e., that some specification behaviors are reproducible also in the implementation). However, the tests usually do not cover all the possible implementation behaviors (neither all the specification behaviors) and, therefore, some implementation faults may be undetected by offline testing. Thus, for safety-critical systems, one may want to continue checking the conformance also after the program deployment.

In (Arcaini, et al., 2012), we proposed a *runtime verification* approach (CoMA) for checking the conformance of a Java code w.r.t. the ASM specification at runtime. That approach can be thought as a kind of *online testing*; it is described in the right part of Figure 3:

- the *Java program observer* monitors the execution of the Java program and, whenever it detects that a changing method (i.e., a method annotated with `@RunStep`) is executed, it triggers the execution of simulation of the ASM specification with the ASM simulator;
- the *conformance analyzer* checks the step conformance between the Java step and the ASM step by comparing the values of the linked functions/methods (those annotated with `@FieldToFunction`/`@MethodToFunction`) with those of the corresponding ASM functions. If a violation of conformance is detected, an error is reported.

The runtime framework is also able to check the conformance of nondeterministic systems in which there are multiple states that can be obtained by executing a changing method. For this kind of systems, the ASM simulator must find if there exists a *next* state that is conformant with the obtained Java state. This can be done by explicitly computing all the next states (Arcaini, et al., 2013), or by symbolically representing the next states using an SMT solver (Arcaini, et al., 2014b).

The 3DSAT software is now under evaluation in an Italian hospital. Since (as seen in Sect. 4.2) offline testing does not provide enough confidence on the program correctness, we need to continue testing the application also after the deployment, and so we have integrated the runtime framework in 3DSAT. The framework will record any conformance violation in a log file, so that we will be able to analyze failures that may occur during the doctors' evaluation.

[6] http://pitest.org/

5 Conclusions

The paper has presented an approach to link an implementation with its formal specification by means of code annotations, and has described two techniques for checking whether the implementation is conformant with the specification: an *offline* technique based on model-based testing, and an *online* technique based on runtime verification. The software of a medical device (SAM) to measure stereo-acuity has been used as case study.

Although the offline testing is a good starting point for evaluating the correctness of the implementation, it cannot provide any guarantee: in our experiments, 30% of the program instructions were uncovered by the generated tests that achieved only 45.7% mutation score. Therefore, for safety-critical systems, offline testing cannot be sufficient. Classical approaches based on theorem proving and formal verification could provide further guarantees, but they may be difficult to apply and require stronger mathematical skills. We have here shown a (runtime) verification approach that permits to check the conformance online (i.e., during program execution after deployment). The approach has the advantage that it reuses all the theoretical framework of offline testing and, therefore, does not add additional burden to the developer. The on-line testing technique permits to verify *any* program execution (also in the presence of nondeterministic systems), in contrast to the offline approach that only checks some selected program executions. The online verification approach is currently integrated in SAM that is under evaluation in an Italian hospital: this allows us to detect any possible conformance violation that may occur during the use of the program.

6 Literature

Arcaini, P., Bonfanti, S., Gargantini, A., Mashkoor, A. & Riccobene, 2015a. *Formal validation and verification of a medical software critical component.* In *Proceedings of MEMOCODE 2015*, IEEE, pp. 80-89.

Arcaini, P., Bonfanti, S., Gargantini, A. & Riccobene, E., 2016a. How to assure correctness and safety of medical software: the Hemodialysis Machine Case Study. In *Proceedings of the 5th International Conference on Abstract State Machines, Alloy, B, TLA, VDM, and Z (ABZ 2016),* Springer International Publishing, pp. 344-359.

Arcaini, P., Gargantini, A. & Riccobene, E., 2010a. *AsmetaSMV: a way to link high-level ASM models to low-level NuSMV specifications.* In *Proceedings of the 2nd International Conference on Abstract State Machines, B and Z (ABZ 2010)*, Springer, pp. 61-74.

Arcaini, P., Gargantini, A. & Riccobene, E., 2010b. *Automatic Review of Abstract State Machines by Meta Property Verification.* In *Proceedings of NFM 2010*, pp. 4-13.

Arcaini, P., Gargantini, A. & Riccobene, E., 2012. CoMA: Conformance Monitoring of Java Programs by Abstract State Machines. In *Proceedings of Runtime Verification 2011,* Springer, pp. 223-238.

Arcaini, P., Gargantini, A. & Riccobene, E., 2013. *Combining Model-Based Testing and Runtime Monitoring for Program Testing in the Presence of Nondeterminism.* In *Proceedings of AMOST 2013*, IEEE, pp. 178-187.

Arcaini, P., Gargantini, A. & Riccobene, E., 2014a. Offline Model-Based Testing and Runtime Monitoring of the Sensor Voting Module. In *Proceedings of ABZ 2014: The Landing Gear Case Study.* Springer International Publishing, pp. 95-109.

Arcaini, P., Gargantini, A. & Riccobene, E., 2014b. Using SMT for dealing with nondeterminism in ASM-based runtime verification. *ECEASST,* Volume 70.

Arcaini, P., Gargantini, A. & Riccobene, E., 2015b. Rigorous development process of a safety-critical system: from ASM models to Java code. *International Journal on Software Tools for Technology Transfer,* pp. 1-23.

Arcaini, P., Gargantini, A. & Riccobene, E., 2016b. SMT-based automatic proof of ASM model refinement. In *Proceedings of the 14th International Conference on Software Engineering and Formal Methods (SEFM 2016), Vienna, Austria, July 4-8, 2016,* Springer International Publishing.

Arcaini, P., Gargantini, A., Riccobene, E. & Scandurra, P., 2011. A model-driven process for engineering a toolset for a formal method. *Software: Practice and Experience,* Volume 41, pp. 155-166.

Börger, E. & Stärk, R., 2003. *Abstract State Machines: A Method for High-Level System Design and Analysis.* Springer Verlag.

Carioni, A., Gargantini, A., Riccobene, E. & Scandurra, P., 2008. A Scenario-Based Validation Language for ASMs. In *Proceedings of the 1st International Conference on Abstract State Machines, B and Z (ABZ 2008).* Springer-Verlag, pp. 71-84.

EU, 2007. *Directive 2007/47/EC of the European Parliament and of the Council.*

FDA, 2002. *General Principles of Software Validation; Final Guidance for Industry and FDA Staff, Version 2.0.*

Gargantini, A., Facoetti, G. & Vitali, A., 2014. *A Random Dot Stereoacuity Test Based on 3D Technology.* In *Proceedings of REHAB14 - 2nd ICTs for improving Patient Rehabilitation Research Techniques Workshop*, pp. 358-361.

Gargantini, A. & Riccobene, E., 2001. ASM-Based Testing: Coverage Criteria and Automatic Test Sequence Generation. *Journal of Universal Computer Science,* Volume 7, pp. 262-265.

Gargantini, A., Riccobene, E. & Rinzivillo, S., 2003. *Using Spin to Generate Tests from ASM Specifications.* In *Proceedings of Abstract State Machines 2003: Advances in Theory and Practice 10th International Workshop, ASM 2003*, Springer Berlin Heidelberg, pp. 263-277.

Gargantini, A., Riccobene, E. & Scandurra, P., 2008. A Metamodel-based Language and a Simulation Engine for Abstract State Machines. *Journal of Universal Computer Science,* 14(12), pp. 1949-1983.

IEC, 2005. *IEC 60601-1:2005 Medical electrical equipment – Part 1: General requirements for basic safety and essential performance.*

IEC, 2006. *IEC 62304 - Medical device software - Software lifecycle processes.*

ISO, 2003. *ISO 13485:2003 Medical devices - Quality management systems - Requirements for regulatory purposes.*

ISO, 2007. *ISO 14971:2007 Medical devices - Application of risk management to medical devices.*

Jetley, R., Iyer, P. S. & Jones, P. L., 2006. A Formal Methods Approach to Medical Device Review. *Computer, n*39(4), pp. 61-67.

Utting, M. & Legeard, B., 2006. *Practical Model-Based Testing: A Tools Approach.* Morgan-Kaufmann.

Modelling bio-compatible and bio-integrative medical devices

Didier Fass *didier.fass@loria.fr*
Dominique Méry, *dominique.mery@loria.fr*

Abstract

Developing and producing medical devices and healthcare systems is a crucial issue, both for the economy and for providing safe advances in healthcare delivery. We propose a taxonomy of medical human machine systems and we define classes of healthcare applications for identifying a number of approaches and to overcome difficulties of bio-compatibility and bio-integration. Our aim is to demonstrate how medical devices design, and more generally human-machine system concepts and epistemology, depend on our skills to think and conceptualize generally human system integration. We claim that it is necessary to reclaim these concepts for ensuring correct by construction medical devices bio-compatibility and bio-integrative properties from the early stage of the design pro*cess.*

Keywords

Medical devices, modelling, system engineering, human system integration, bio-CPS, bio-compatibility, bio-integration, correctness by construction

1 Introduction

Developing and producing medical devices and healthcare systems is a crucial issue, both for the economy and for providing safe advances in healthcare delivery. Medical devices become smaller in physical terms (see for instance the new pacemaker implanted inside the ventricular) but larger in software-based elements, the design, testing, validation, and eventual authority device approval is becoming expensive for medical device manufacturers both in terms of time and cost. The increase of alarming rate leads to recalling failing devices. Recalling failing devices is increasing cost and gives a very negative feeling to patients as well as physicians; it leads to consider both forensic and medical issues. Questions on costs should not hide safety and security problems that are now stated in this new generation of software systems. Moreover, safety problems are related to requirements that are either related to technical elements or related to human body - both anatomical and physiological, features. As claims by Dines Bjoerner [5,6,7] “Before software can be developed its requirements must be stated. Before requirements can be expressed the application domain must be understood”. Since main healthcare system domain is life and health (biology and its two sub-domains: anatomy and physiology), medical devices require integration within or on the body – living system, it leads *to modeling bio-compatible and bio-integrative medical devices.* That needs to take into account information that is not necessarily explicit in the list of requirements and in the definition of domains such as nature of interactions and time representation. Systems under investigation are often called *cyber-physical systems*, for short CPS, and necessitate medical human-machine systems design integration, validation and certification. We propose a taxonomy of medical human-machine systems and define classes of healthcare applications for identifying a number of approaches and to overcome difficulties of bio-compatibility and bio-integration. Using the pacemaker case study, we sketch our Event-B-based methodology, which did not consider bio-compatibility and bio-integration and we show how we intend to take into account these new ideas in a correct-by-construction approach.

2 . Categorisation of medical devices

Our aim is to focus primarily on the application of software and systems engineering to software-based medical devices used for patients. However, we classify medical devices or healthcare systems into categories corresponding to the use and to the bio-integration of these systems in the life of a patient.

A first class of medical devices is defined as devices which are permanently implanted in the body of a patient, as pacemakers, artificial heart. It requires a clear and as complete as possible statement for the environment. The challenge is to develop models for environment as well as tools. Moreover, when considering the artificial heart, we are facing questions of bio-compatibility and bio-integration.

A second class of medical devices is defined as devices which are permanently used by patients but not implanted in the body of a patient, as the hemodialysis. The model for environment is yet a challenge because the global system is integrating several organs and regulations rules for glucose. It is then clear that models for environment necessitates to express flows of fluid in the body of the patient and rate of sugars in the blood. The system is used on a regular and periodic period.

A third class of medical devices is defined as devices which are temporary used for helping somebody to reach a given target state in intensive care, as for instance the extracorporeal membrane oxygenation (ECMO) [1]. It may happen after a heart attack, when the patient’s

heart requires a minimal activity, or / and during a pulmonary insufficiency (i.e. severe state with H1N1 flu) [16]. The system supports one or more functionalities and assists the patient. Still in this case we have to integrate bio-compatibility and bio-integration.

Currently, implantable or wearable medical devices are designed by abstraction with a reductionist and functionalist approach. Thus living systems are reduced to their physical properties and logical models. Therefore developed and implemented algorithms result from functional analysis methods - abstraction and digitalization, and skills (knowledge and understanding) of the designers.

Mockups and prototypes are validated firstly by technical tests for ensuring technological regulatory standards and secondly by experimental and clinical test before their wide distribution and marketing. However, certification of medical devices , as in aerospace and nuclear industry, remains a challenge.

Medical devices require original, sound and reliable validation methods [14,4,5] based on scientifically validated modeling methods and tools (model-based engineering) and grounded on theoretical biology, medical knowledge and clinical evidences, and physicians expertise (evidence-based medicine).

We show how medical devices design, and more generally human-machine system concepts and epistemology, depend on our skills to think and conceptualize generally human system integration. We claim that it is necessary reclaiming these concepts for ensuring correctness by construction medical devices bio-compatibility and bio-integrative properties from the early stage of the design pro*cess grounding on both theoretical integrative physiology principles and trustable formal method. Next, we describe the correct-by-construction approach that was used by Méry and Singh* [18], *when considering the pacemaker challenge using the Event-B modeling language.*

3 Model-based Design of Medical Devices

Formal methods have emerged as a complementary approach to ensuring quality and correctness of high-confidence medical systems, overcoming limitations of traditional validation techniques such as *simulation* and *testing*. In [18], authors propose a new methodology to obtain certification assurance for complex medical systems design, based on the use of formal methods. The methodology consists of five main phases: first, informal requirements, resulting in a structured version of the requirements, where each fragment is classified according to a fixed taxonomy. In the second phase, informal requirements are represented in formal modelling language, with a precise semantics, and enriched with invariants and temporal constraints. The third phase consists of refinement-based formal verification to test the internal consistency and correctness of the specifications. The fourth phase is the process of determining the degree to which a formal model is an accurate representation of the real world from the perspective of the intended uses of the model using model-checker. Last phase provides an animation framework for the formal model with real-time data set instead of toy-data, and offers a simple way for specifiers to build a domain specific visualization that can be used by domain experts to check whether a formal specification corresponds to their expectations. Fig. 1 sketches the methodology based on the refinement process providing possible feedbacks in the design of a system following a correct by construction process and producing a trustable formal model from the informal requirements. As we have mentioned in the previous lines, the animation phase is a way to integrate the domain experts in the design process. We have to warn the reader that the model of the pacemaker is a formal expression in a formal modelling language based on set theory and on the notion of generalized substitution. The link between knowledge of experts and the formal expressions are very critical to elaborate. Our point of view is that we are facing the question of validation

of the formal model with respect to the informal requirements and the integration of physiological information or expert knowledge. A possible solution is to exploit domain ontologies combined with formal modelling language as proposed by Aït-Ameur and Mery [4]. However, it remains to consider how to model physiological features and to validate the resulting model.

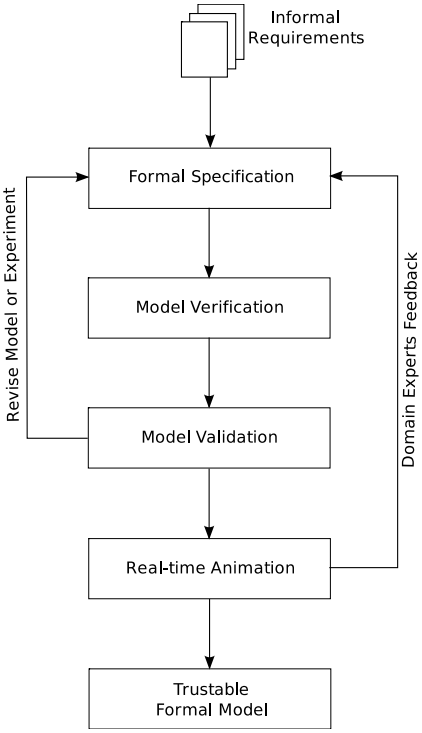


Figure 1: Design of Correct by Construction Medical Devices

We give a short introduction on Event-B [3,8] to identify what we can model with Event-B. Event-B is a formal method for system-level modelling and analysis using set theory. In Fig.1, this phase also gives the feedback to the formalization phase in case of unexpected behaviours of the system. The feedback approach is allowed to modify the formal model and verify it using any theorem prover tool and finally validate it using a model checker tool. The verification, validation and real-time animation processes are applied continue until not to find the correct formal model according to the expected system behaviour. In this phase of the formal development, most of errors are discovered by the domain experts. The real time animation is a key tool for helping the communication with domain experts.

An Event-B model expresses a state property called invariant which is defining the set of possible states of the model. A state is a mapping relating each variable to a given value. The value of a variable is in a given domain which can be either codable in a programming language or member of a domain which is representing possible values. It means that an Event-B model can have variables and state variables modifiable by a finite list of events which are modelling the possible modifications of variables. An event of an Event-B model is not executed but observed, when its guard is true. Only one event may be observed at any time. An Event-B model is discrete and does not express any liveness or fairness property, even if there some extensions of the original Event-B models. Moreover, an Event-B model is based on a logical theory based on set-theoretical or predicate calculus notations. It means that the design of an Event-B model requires specific checking of well typing or verification conditions validating the invariant property. Questions are related to what properties over medical devices can we express and how tractable are these properties? In Event-B, we can express, first, invariant properties defining a stability condition of the system under consideration, or safety properties stating that nothing bad will happen. For instance, the

pacemaker should not pace in the red zone and can pace in the green zone. Fig. 2 is containing the annotated ECG as well as the text of the invariant.

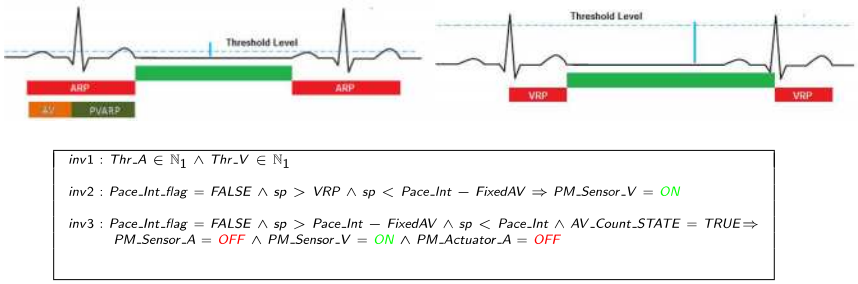


Figure 2: Example of a relation between an invariant and an ECG-based state

When validating an Event-B model, one should prove that each event maintain invariant and safety properties and this operation is done by the underlying proof assistant. The underlying proof process is adding trust to the resulting model and guide the designer when following the incremental refinement-based process for constructing formal models from informal requirements. The formality of Event-B models is helping to get a trustable model but we need to have modalities for communicating to the domain experts.

The choice of the Event-B modelling language is mainly guided by the simplicity of the basic concepts and the existence of a toolset namely Rodin [2]. Others modelling techniques can be used for developing models and the most important feature is the methodology of correct by construction development using the refinement. The refinement allows us to play with levels of abstraction. In next section, we sketch the methodology used for developing a closed-loop model for the pacemaker and we will give more details on the example of the pacemaker. We illustrate what authors were able to handle using Event-B and what we have validated and then we state questions on bio-compatibility and bio-integration in medical design devices.

4 Example of the Pacemaker

The pacemaker is a medical device which is implanted in the body of the patient. It is related to the heart by leads and it interacts with the heart according to modes defined by the physician. It is an example of a device which is supposed to remain for a long time in the body. Moreover, it augments the functionalities of the heart by helping it and the heart is evolving with the presence of the pacemaker. Consequently, the function of pumping is supported not only by the heart itself but also by the pacemaker. The heart is, after a while, working only with the pacemaker. In previous works, Mery and Singh [19] developed a discrete model of the pacemaker and the heart was modelled implicitly by records of ECG. We obtain a validation by off-line records.

In a second step, we should test the model with respect to a heart model called the electrical heart model developed by physicians. We summarize the electrical model. The heart consists of four chambers: right atrial, right ventricle, left atrial and left ventricle, which contract and relax periodically. The natural heart's system requires an electrical stimulus, which is generated by the small mass of specialized tissue located in the right atrium called the sinus node. This electrical stimulus travels down through the conduction pathways and causes the heart's chambers to contract and pump out blood. Each contraction of the ventricles represents one heartbeat. The atrial contract for a fraction of a second before the ventricles, so their blood empties into the ventricles before the ventricles contract. The electrical current

flows progressively in the heart muscle using special conduction cells. In previous works, we developed a discrete model based on the electrical conduction flow according to the points A, B, C, D, E, F, G ,H.

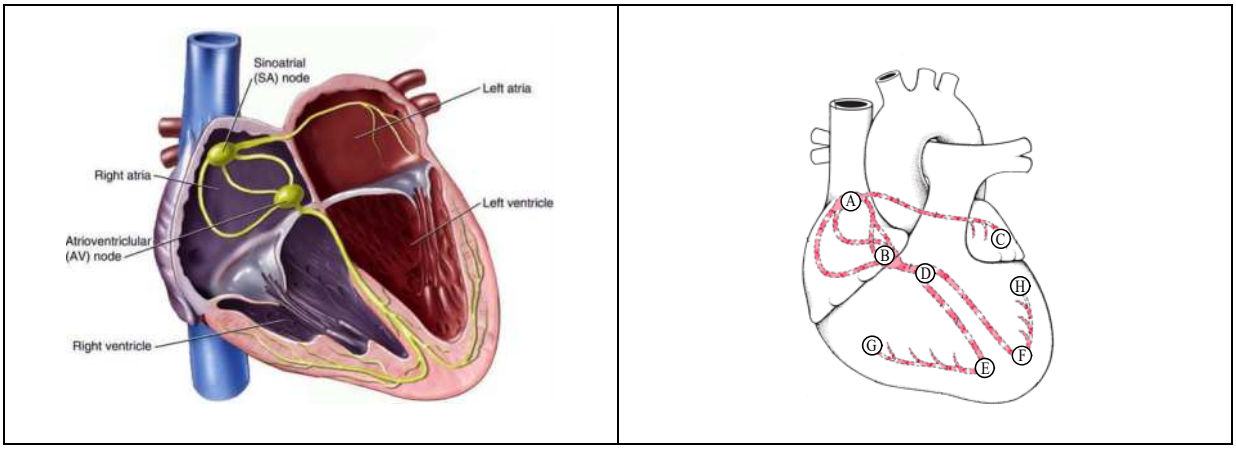


Figure 3: Different elements of the heart and the points defining the electrical conduction model of the heart.

We obtain a closed-loop model of the pumping function, since the heart was no more considered as the environment of the pacemaker but as a part of the model itself. The closed-loop model is used for studying the behaviour of the pacemaker in situ and it takes into account possible dysfunction of heart. Real time animation of formal models is a possible mean for communicating with physicians for instance in clinical experiences. Figure 4 describes the platform for animating Event-B models using a database of ECGs and plugins as Brama for feeding models. The platform is based on the Rodin tool which is offering functionalities for designing, verifying and animating models. An extra work has been done by N. Singh for making visual communications possible.

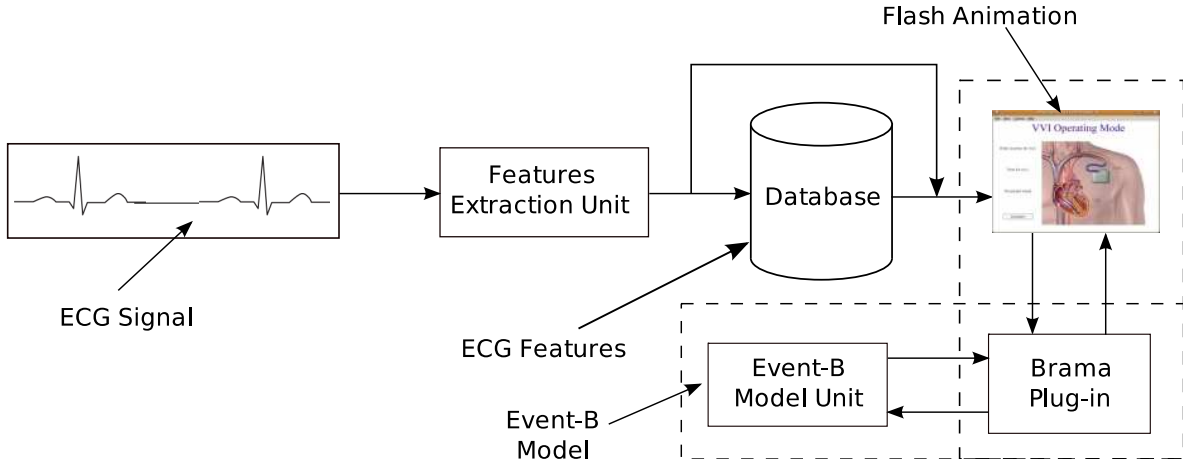


Figure 4 : Implementation of proposed functional architecture on the single electrode cardiac pacemaker case study

In the last step, we have got a more trustable model for the pacemaker. However, we have not completely dealt with questions of bio-compatibility and bio-integration. These two extra-points can be considered with the proposed methodology. However, they require to extend the methodology by considering expertise domains and by improving communication between physicians, designers and patients. Expertise domains require to integrate biology models. Finally, we summarize our position on the two concepts bio-compatibility and bio-integration in the final section.

5 Concluding Remarks and Future Directions

Medical devices belong to the class of cyber-physical systems, since they must be integrated within or on the human body generating integrated wholes, human machine systems. They are made up of two main categories of systems. These two kinds of systems differ in their nature: their fundamental organization, complexity and behaviour. The first category, the traditional one, includes *technical* or *artifactual* systems that could be engineered. The second category includes *biological* systems: the human that could not be engineered. Thus, integrating human and cyber-physical systems in design is to couple and integrate in a structural (anatomical) and dynamical or functional (physiological) coherent way, a biological system (the human) with a technical and artifactual system in the same isomorphic framework. So medical devices engineering needs to model the organ or the part of the human body and its behaviour on the one hand, the cyber-physical system physical and computational components and its behaviour on the other hand, as well as their dynamical relation (interaction or coupling) to test and validate human machine reliability and human systems integration [12].

5.1 Bio - Cyber-Physical Systems (Bio-CPS)

Traditional medical device design methods rely on analytic and reductionist concepts based on a mechanical or computational metaphor. The aim of making human and medical CPS "working" together is to assist a dysfunctional organ, transiently or permanently, to maintain the whole living system (a person) in stable condition or in a recovered "normal" condition. We call bio-CPS the whole and functional "living human artefact system" resulting from human medical CPS coupling and integration. The challenge is the design of medical CPS that ensure the correct integrative coupling by construction in spite of their different nature of interactions and the time representation.cf. table 1.

	Biological	Cyber	Physical
Symmetricity of interactions	Never	Both symmetric and non-symmetric	Always
Locality of interactions	Mainly non-local	Mainly local	Both local and non-local
Time representation	Continuous (Functional level) Discrete (Structual level)	Discrete or Event basd	Continuous

Table 1 : ***Classification of systems considering the nature of the interactions and the time representation [13]***

5.2 Medical CPS domain engineering

Using Bjoerner's framework, we can mention the issue of medical devices domain to highlight its fundamental properties, which could satisfy human systems integration requirements. Bjoener's framework [7] based on the triptych: D, S -> R, where D is the domain of the problem and where requirements R are satisfied by the relation ->, which intends to mean *entailment*; so, S is a model of our system built or expressed from D. The domain provides a way to express properties and facts of the environment of the system under construction.

Bio CPS must help organ and the body (a person) to recover a stable state (invariant) com-

patible with survivability in intensive care or a physiological normal quality of life for implantable system like heart pacemaker despite pathology or illness.

Consequently, Bio CPS specifications or model (S) must satisfy human system integration and integrative physiology requirements (R) [12] and theoretical principles [9,10,11]. This especially concerns the artificial biological interface system and the behavioral monitoring and assisting computerized system and its algorithms. Their domain engineering (D) is bio-integrative engineering. By consequence we can say that an "artificial" medical CPS that satisfied human system integration properties must present two fundamental properties: bio-compatibility and "bio-integrability".

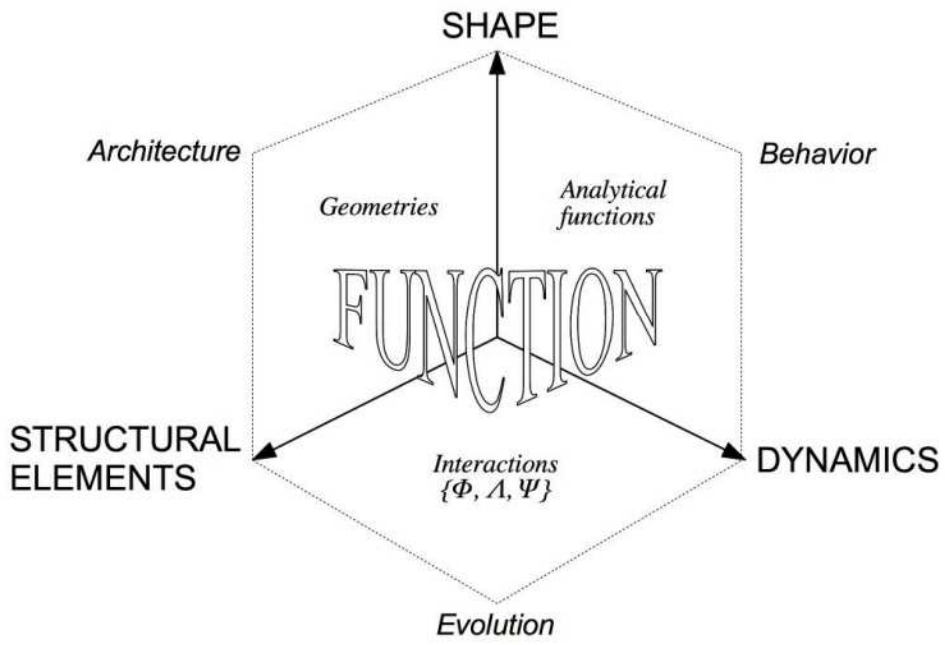


Figure 5: Our isomorphic epistemic framework for human-machine systems integrative design and organization. Function results from its correctness-by-construction and the wholeness stability of Bio-CPS elements integration [12].

For modelling biological system and biocompatible and biointegrable artificial systems - medical devices, we propose an isomorphic framework (Figure 5). This conceptual framework describes three categories of required main system dimension: structural elements, shapes or forms and dynamics. Taking into account two by two this main classes of system variables, one can describe three specification plan: architecture (structural elements to shape or form specify geometrical structure or system architecture), behaviour (shape or form to dynamics specify analytical functions or functionally analysed) and evolution (structural elements to dynamics specify three main types of functional interactions: physical Φ, logical Λ, biological Ψ). If we assume that a function does not exist by itself but is the emerging result of integrative organization, this framework grounds our bio-integrative model based Bio-CPS engineering. The schema extends clearly our methodology (Figure 6).

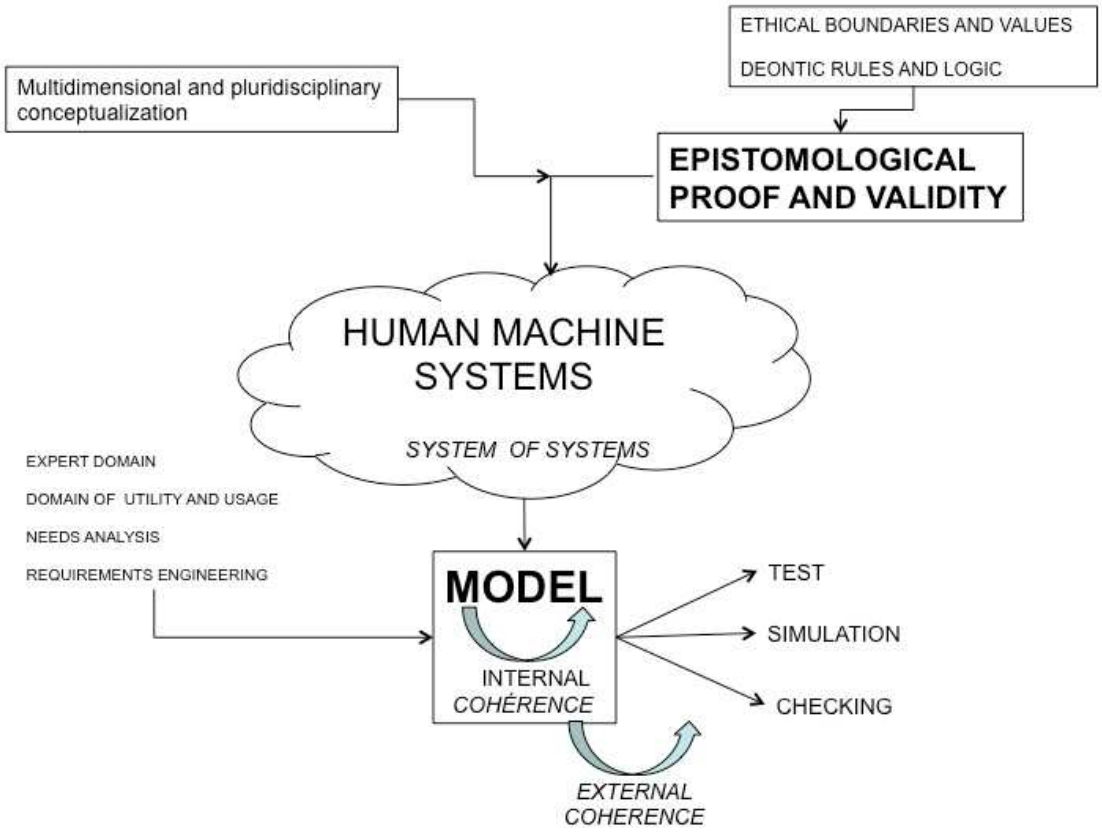


Figure 5: Challenging human-machine system - biocompatible and bio-integrative medical devices, design and organization is modeling an heterogeneous system of systems (different by nature). That requires a proven and validate epistemic framework fitted to hybrid system and specific domain engineering, and challenging the question of human machine system nature, correctness-by-construction and ensuring human systems integration reliability.

5.3 Bio CPS Methodology

We have first described a methodology based on a formal notation namely Event-B, which has been used for developing a closed-loop model for the pacemaker. The methodology is described by Fig 1 and is not restricted to Event-B. We have to address questions on the extension of the expressivity of the assertion language that should be able to state properties of the medical domain. Moreover, modelling CPS using Event-B or another formal method is a real challenge because we have to be able to manage hybrid medical models. Hybrid medical models are mathematical objects integrating discrete and continuous features of medical system under consideration.

6 Literature

[1] Darryl Abrams, Alain Combes, and Daniel Brodie. What's new in extra- corporeal membrane oxygenation for cardiac failure and cardiac arrest in adults? Intensive Care Medicine, 40(4):609–612, 2014. ISSN 1432-1238. URL http://dx.doi.org/10.1007/s00134-014-3212-0.

[2] Jean-Raymond Abrial, Michael J. Butler, Stefan Hallerstede, Thai Son Hoang, Farhad Mehta, and Laurent Voisin. Rodin: an open toolset for modelling and reasoning in event-b.

STTT, 12(6):447–466, 2010. URL http://dx.doi.org/10.1007/s10009-010-0145-y.

[3] Jean-Raymond Abrial. Modeling in Event-B - System and Software En- gineering. Cambridge University Press, 2010. ISBN 978-0-521-89556-9. I-XXVI, 1-586 pp.

[4] Yamine Aït-Ameur and Dominique Méry. Making explicit domain knowl- edge in formal system development. Sci. Comput. Program., 121:100–127, 2016. URL http://dx.doi.org/10.1016/j.scico.2015.12.004.

[5] Dines Bjørner. Domain engineering: A software engineering discipline in need of research. In SOFSEM 2000: Theory and Practice of Informatics, 27th Conference on Current Trends in Theory and Practice of Informatics, Milovy, Czech Republic, November 25 - December 2, 2000, Proceedings, pages 1–17, 2000. URL http://dx.doi.org/10.1007/3-540-44411-4_1.

[6] Dines Bjørner. SOFSEM 2000: Theory and Practice of Informatics: 27th Conference on Current Trends in Theory and Practice of Informat- ics Milovy, Czech Republic, November 25 – December 2, 2000 Proceedings, chapter Domain Engineering: A Software Engineering Discipline in Need of Research, pages 1–17. Springer Berlin Heidelberg, Berlin, Heidelberg, 2000. ISBN 978-3-540-44411-4. URL http://dx.doi.org/10.1007/3- 540-44411-4_1.

[7] Dines Bjørner. Domain Engineering - Technology Management, Research and Engineering, volume 4 of COE Research Monograph Series. JAIST, 2009. ISBN 978-4-903092-17-1.

[8] Dominique Cansell and Dominique Méry. The Event-B Modelling Method: Concepts and Case Studies, pages 33–140. Springer, 2007. See [150].

[9] Gilbert A. Chauvet. Hierarchical functional organization of formal biological systems: a dynamical approach. i. the increase of complexity by self- association increases the domain of stability of a biological system. Philo- sophical Transactions of the Royal Society of London B: Biological Sciences, 339(1290):425–444, 1993.

[10] Gilbert A. Chauvet. Hierarchical functional organization of formal biological sys- tems: a dynamical approach. ii. the concept of non-symmetry leads to a criterion of evolution deduced from an optimum principle of the (o-fbs) sub-system. Philosophical Transactions of the Royal Society of London B: Biological Sciences, 339(1290):445–461, 1993.

[11] Gilbert A. Chauvet. Hierarchical functional organization of formal biological sys- tems: a dynamical approach. iii. the concept of non-locality leads to a field theory describing the dynamics at each level of organization of the (d-fbs) sub-system. Philosophical Transactions of the Royal Society of London B: Biological Sciences, 339(1290):463–481, 1993.

[12] Didier Fass. Augmented Human Engineering: A Theoretical and Experimental Approach to Human Systems Integration. INTECH Open Access Publisher, 2012. ISBN 9789535103226. URL https://books.google.fr/books?id= 5ujboAEACAAJ.

[13] Didier Fass and Franck Gechter. Towards a theory for bio-cyber physical systems modelling. In Digital Human Modeling. Applications in Health, Safety, Ergonomics and Risk Management: Human Modeling, pages 245– 255. Springer, 2015.

[14] R. Jetley, S. Purushothaman Iyer, and P. Jones. A formal method approach to medical device review. Computer, 39(4):61–67, April 2006. ISSN 0018-9162.

[15] Cliff B. Jones, Peter W. O'Hearn, and Jim Woodcock. Verified software: A grand challenge. IEEE Computer, 39(4):93–95, 2006. URL http://dx. doi.org/10.1109/MC.2006.145.

[16] Antoine Kimmoun, Fabrice Vanhuyse, and Bruno Levy. Improving blood oxygenation during venovenous ecmo for hearts. Intensive Care Med, 39: 1161–1162, 2013.

[17] I. Lee, G. J. Pappas, R. Cleaveland, J. Hatcliff, B. H. Krogh, P. Lee, H. Ru- bin, and L. Sha. High confidence medical device software and systems. Computer, 39(4):33–38, April 2006. ISSN 0018-9162.

[18] Dominique Méry and Neeraj Kumar Singh. Trustable formal specification for software certification. In Leveraging Applications of Formal Methods, Verification, and Validation - 4th International Symposium on Leveraging Applications, ISoLA 2010, Heraklion, Crete, Greece, October 18-21, 2010, Proceedings, Part II, pages 312–326, 2010. URL http://dx.doi.org/10.1007/978-3-642-16561-0_31.

[19] Dominique Méry and Neeraj Kumar Singh. Formalization of heart models based on the conduction of electrical impulses and cellular automata. In Zhiming Liu and Alan Wassyng, editors, FHIES, volume 7151 of Lecture Notes in Computer Science, pages 140–159. Springer, 2011. ISBN 978-3- 642-32354-6.

7 Author CVs

Didier FASS

FASS Didier, Associate, Associate Professor at ICN Business School and researcher at LORIA in MOSEL team**,** responsible for Artem "Augmented human" project. Professional Skills: Expert for ANR, IEEE, Human and Systems, Health ethics National and International Projects: 6FP EU Craft DRIVESAFE (2004-2007); PPF Fibrous Material "Pôle de Compétitivité" Natural Fibers Grand Est (2005-2008); PAUSA Aerospace Valley and DGAC (2006-2008) Collaborations: NASA Ames Research Center Human System Integration Division, ONERA Salon de Provence, Georgia Institute of Technology, Intensive care unit CHU Nancy Brabois, Academic Title: PhD in Neurosciences on the 23. December 2002 under supervision of Professor Jean-Paul Haton (Artificial Intelligence) and Professor Francis Lestienne (Neurophysiology); Doctor of Dental Surgery on the 6. June 1991 on Knowledge based Diagnosis modeling and Medical imaging (TDM and RMI) under supervision of Professor Daniel Rozencweig (Occlusodontics) and Professor Augusta Tréheux (Medical Imaging); Master in Management Nancy 2 University (1992); Master in Cognitive Sciences, LIMSI, Orsay Paris XI (1994); Degree in Psychophysiology (1995) and Post-graduate in Forensic Expertise (2010) Medical School Nancy University. Price and Professional achievements: Fellowship French Occlusodontics College (1991), Member of the Board – Forum advisory committee TELECOM International Telecommunication Union (ITU) (2007-2012).

Dominique MERY

MERY Dominique, Full Professor of Computing Science at Université de Lorraine since the first of September 1993, Head of the Research Group (Formal Methods and Applications) at the LORIA Laboratory. Head of the PhD School IAEM Lorraine. Professional Skills: Expert for NSF, Enterprise Ireland, ANR, AERES National and International Projects: ANR SETIN RIMEL (2007-2010); RNRT EQUAST (2003-2005): ACI Sécurité DESIRS (2003-2006); CTI CNET 1995-1998 Academic Titles PhD in Computer Science on the 31. May 1983 under the supervision of Professor Patrick Cousot and Thèse d'état on the 26. February 1993 in Mathematics. Prices and Professional Achievements: Member of l'Institut Universitaire de France [1995-2000]. Grant for Scientific Excellence Grad A (2009- 2013)). Member of IFIP WG 1.3 Foundations of System Specification.