LUND UNIVERSITY

Software Risk Management in the Safety-critical Medical Device Domain - Involving a User Perspective

Lindholm, Christin

2015

Link to publication

Citation for published version (APA):
Lindholm, C. (2015). *Software Risk Management in the Safety-critical Medical Device Domain - Involving a User Perspective*. [Doctoral Thesis (compilation), Department of Computer Science].

Total number of authors:
1

General rights
Unless other specific re-use rights are stated the following general rights apply:
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.
• Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
• You may not further distribute the material or use it for any profit-making activity or commercial gain
• You may freely distribute the URL identifying the publication in the public portal

Read more about Creative commons licenses: https://creativecommons.org/licenses/

Take down policy
If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

LUND UNIVERSITY

PO Box 117
221 00 Lund
+46 46-222 00 00

Software Risk Management in the Safety-Critical Medical Device Domain - involving a user perspective

Christin Lindholm

Doctoral Dissertation, 2015

Department of Computer Science
Lund University

Dissertation 45, 2015
LU-CS-DISS: 2015-01

ISBN 978-91-7623-219-4 (printed version)
ISBN 978-91-7623-220-0 (electronic version)
ISSN 1404-1219

Department of Computer Science
Faculty of Engineering
Lund University
Box 118
SE-221 00 Lund
Sweden

Email: christin.lindholm@cs.lth.se

Printed in Sweden by Tryckeriet i E-huset, Lund, 2015

© Christin Lindholm

To Henrik, Niklas and Johan

Abstract

There is a thin line between life and death. In the medical domain, risk management can be an instrument that helps the development organisations to develop safer medical devices. A medical device that fails can bring harm to both patients and medical staff. The medical device domain is a complex field where there are several characteristics contributing to the complexity. Many of the functions performed by medical devices and systems are affecting human lives, directly when the devices are used in treatment and indirectly when the devices are used in monitoring.

In the risk management process a major challenge is to assure safety and prevent the patients and the medical staff from harm. The process is a dynamic process and it is necessary to manage risk throughout the whole lifecycle of the medical device in order to avoid potential hazardous situations over time.

The main goals of the research effort in this thesis are to integrate users and user perspective in the software risk management process in the medical device domain, and to develop a new risk management process involving a user perspective.

This thesis is based on empirical research with both qualitative and quantitative approaches. The research contains a survey presenting the characteristics of the state of practice of software development in the context of the medical devices and systems. One part of the survey focuses on quality assurance of software, risk management, and the developers' conception of safety criticality of software. The conception of risk was further investigated in two controlled experiments. The identified challenges and experiences from the survey and the experiments were utilized after that in three case studies.

A new software risk management process, RiskUse, was derived from the experiences and conclusions gained from two of the three case studies. In the first case study was the risk management process studied and in the second case study the introduction of usability testing included the risk management process. The aim of RiskUse is to support software risk management activities in the medical device domain and to bring in an emphasised user perspective into the risk management process. Finally, the first version of RiskUse was empirically evaluated in the third last case study. The research was conducted as action research with the aim to evaluate the user perspective parts of the new risk management process.

In conclusion RiskUse, is found, in the studied cases, to support the practitioners in their work with user risks and risk management.

Contents

INCLUDED PAPERS

Paper III: Different Conception in Software Project Risk Assessment

Paper IV: A Case Study on Software Risk Analysis and Planning in Medical Device Development

LIST OF PUBLICATIONS

This dissertation consists of two parts. The first part is an introductory part presenting the context of the research, the research methodology and a summery of the research results and future research. The second part consists of the six research papers on which the conclusions of the first part are based on.

PUBLICATIONS INCLUDED IN THE DISSERTATION

I. **A Survey of Software Engineering Techniques in Medical Device Development.**
Feldmann, R.L., Shull, F., Denger, C., Höst, M. & Lindholm, C. (2007). In *Workshop on High Confidence Medical Devices, Software and Systems (HCMDSS) and Medical Device Plug-and Play (MD PnP*), pp. 46-54.

II. **Risk Identification by Physicians and Developers - Differences Investigated in a Controlled Experiment**
Lindholm, C. & Höst, M. (2009). In *Proceeding of the Workshop on Software Engineering in Health Care (SEHC09), at ICSE 2009*, pp. 53-61.

III. **Different Conceptions in Software Project Risk Assessment.**
Höst, M. & Lindholm, C. (2007). In *proceedings of the Software Engineering Track at the 22:nd Annual ACM Symposium on Applied Computing (SAC),* pp. 1422-1426.

IV. **A Case Study on Software Risk Analysis and Planning in Medical Device Development.**
Lindholm, C., Pedersen Notander, J. & Höst, M. (2014). *Software Quality Journal*, 22(3), pp. 469-497.

V. **Introducing Usability Testing in the Risk Management Process in Software Development.**
Lindholm, C. & Höst, M. (2013). In *Proceeding of the Workshop on Software Engineering in Health Care (SEHC13), at ICSE 2013*, pp. 5-11.

VI. **Validation of a Software Risk Management Process, Involving User Perspective**
Lindholm, C. (2014). Submitted to a journal.

RELATED PUBLICATIONS

VII. **IESE-Report No. 071.07/E, Fraunhofer institute for experimental software engineering.**
Denger, C., Feldman, R.L., Höst, M., Lindholm, C. & Shull, F. (2007).

VIII. **A Snapshot of the State of Practice in Software Development for Medical Devices.**
Denger, C., Feldman, R.L., Höst, M., Lindholm, C. & Forrest, Shull. (2007). In *proceedings of International Symposium on Empirical Software Engineering and Measurement (ESEM),* pp. 485-487.

IX. **Development of Software for Safety Critical Medical Devices – an Interview-based Survey of State of Practice.**
Lindholm, C. & Höst, M. (2008). In *Proceeding of the 8th conference on software engineering research in and practice in Sweden (SERPS 08),* pp. 1-10.

X. **Software Risk Analysis in Medical Device Development**
Lindholm, C., Pedersen Notander, J. & Höst, M. (2012). In *Proceedings of the 37th EUROMICRO conference on* Software Engineering and Advanced Applications (SEAA), pp. 362-365.

XI. **A Case Study on Software Risk Analysis in Medical Device Development**
Lindholm, C., Pedersen Notander, J. & Höst, M. (2012). In *Proceedings of Software Quality: 4th International conference (SWQD 2012)*, pp. 143-158.

CONTRIBUTION STATEMENT

The author of this dissertation is the main author of four of the included papers in the second part of this thesis. She is the main author of Paper II, Paper IV, Paper V and Paper VI, and as such responsible for running the research, dividing the work between co-researchers and performing most of the writing.

Paper I were produced in cooperation with another university. The researchers at Lund University were responsible for the design, analysis and results regarding the parts of the survey concerning quality assurance, safety criticality, and risks. The Fraunhofer Institute performed the execution of the survey. More detailed results from the survey are presented in a technical report (Related publications VII). The design and analysis in Paper II and III were made in cooperation between the authors of the papers and the author of this dissertation also executed the experiments. All the authors participated in the observations, discussions and writing of Paper IV. The main author performed the interviews, the second part of the observations, and most of the analysis as well. Paper V and Paper VI were performed mainly by the main author of the papers, who designed and conducted most of the work as well as reported in the studies. Paper VI describes the risk management process, RiskUse, designed by the main author.

Acknowledgements

It has been a long journey, and I would like to thank everyone that has made my journey possible. I am very grateful for all I have learned and experienced during this journey.

First of all I would thank my supervisor Prof. Martin Höst for his guidance, support and comments on my work and thanks also to my assisting supervisor Prof. Per Runeson.

Many thanks to my present and former colleges at LTH Ingenjörshögskolan, Campus Helsingborg, the Software Engineering Research Group and the Department of Computer science. To Lise Jensen and Ylva Oscarsson, special thanks for your support and understanding. I would also like to thank Prof. Boris Magnusson for providing the contact with the medical device organisation, making a major part of this research possible and Jesper Pedersen Notander as co-author. Thanks, to all participants in the conducted studies in this thesis, and a special thank you to Jimmy Johansson who made many of them possible.

Special thanks, to my family and friends for being there and supporting me, and to my children Henrik and Niklas for bringing me much joy in life. I am also grateful to my parents, no longer with us, for all your love and support. Dad, you were with me most of the journey. I know your deepest wish was to attend my dissertation defends, but you had to move on. However, I know you will still be with me.

From my heart I would also like to thank Stoika for her inspiration, our interesting discussions, our long walks, our laughs and for always standing by my side, both in good and bad times. Thank you also Hanna and Birgitta for your support and encouragement, Jan, for your good advices and Edvin for lighting up the world with your smile.

Thank you all!

“ It is dangerous to live, one can even die”
Stoika Hristova

Människa,
Utveckling,
Teknik,
men störst av allt är Kärleken

Popular Science Summery In Swedish

Ett fel i en medicinsk apparat kan vara skillnaden mellan Liv & Död

Picture is available at www. Flicr.com under Creative Common License

Av Christin Lindholm
Institutionen för Datavetenskap
Lunds universitet

En medvetslös man ligger på gatan i Malmö. Mannen hade bråttom på morgonen på väg till sitt arbete. Sirener hörs i bakgrunden. Några personer står lite på avstånd och ser handfallna ut. Föraren till bilen som körde på mannen är i chock. Han vet inte riktigt vad han ska göra med varken sig själv eller med mannen som ligger på marken. Till hans lättnad är ambulansen framme på plats och sjukvårdarna och tar hand om situationen. Medan de undersöker mannen, som slagit huvudet hårt i gatan, slutar mannen att andas och hans hjärta stannar. Nu hänger mannens liv på sekunder. Mannens hjärta måste börja slå igen. Ambulanssjukvårdarna förbereder snabbt och effektivt *hjärtstartaren* och påbörjar återupplivningen. I detta akuta läge måste hjärtstartaren fungera felfritt och från sjukvårdarnas sida ska det inte råda minsta tvekan över hur apparaten ska användas. Mjukvaran (datorprogram) i hjärtstartaren kan orsaka fel och sjukvårdaren kan använda apparaten fel. – Hur minskar vi riskerna för det? Det är en fråga som en forskargrupp vid Lunds Tekniska högskola ställer sig.

Antalet medicinska apparater ökar på våra sjukhus men även i

våra hem och på olika platser ute i samhället. Hjärtstartare som till exempel sjukvårdarna använde, kan vi finna på fler och fler offentliga platser såsom bibliotek, köpcentra och idrottsanläggningar Dessa hjärtstartare ska kunna användas av vem som helst oavsett tidigare kunskaper. Det ställer speciella krav på utformningen av apparaterna. De som ska använda apparaterna måste på ett enkelt och snabbt sätt förstå hur de ska använda den. Då många medicinska apparater i dagens läge innehåller mjukvara och mängden mjukvara har stadigt ökat under årens lopp, påverkar det också kraven på apparaterna och utvecklingen av dem. Fördelarna med mjukvara är att den tillåter apparater att utföra mer och mer avancerade saker samtidigt som storleken på apparaterna minskar. Nackdelen är att mjukvaran är svårare att kontrollera och testa för alla fel som kan uppkomma.

Mjukvara består av skrivna instruktioner, data och kommentarer, kallad kod. Mängden mjukvara i en apparat räknas i rader kod (liknar skrivna rader i ett dokument). Även en liten apparat kan innehålla många rader kod. En *pacemaker* som opereras in kroppen för att hjälpa en persons hjärta att fungera, är en liten apparat på omkring 5 centimeter, den innehåller så mycket som cirka en halv miljon rader kod. Självklart måste vi kunna lita på att all denna kod fungerar som den ska och att den skapats på rätt sätt.

Picture is available at www. Flicr.com under Creative Common License.

En hjärtstartare, en apparat som används vid hjärtstopp för att få ingång hjärtverksamheten igen

När man utvecklar mjukvara använder man olika metoder och verktyg för att designa, skapa (programmera) och testa mjukvaran. De företag som utvecklar medicinska apparater måste följa olika lagar, regler och standarder. Vilka lagar och regler som företaget måste följa beror ofta på i vilket land företaget ska sälja sin produkt.

De företag som utvecklar medicinska apparater måste följa olika lagar, regler och standarder. Vilka lagar och regler som företaget måste följa beror ofta på

i vilket land företaget ska sälja sin produkt.

Picture is available at www. Flicr.com under Creative Common License.

En pacemaker, en liten apparat på 5 cm som hjälper en persons hjärta att fungera

Det kan vara livsavgörande för patienterna att risker med den medicinska apparaten upptäcks i tid. Det är också viktigt att bedöma hur allvarliga riskerna är och att sedan åtgärda dem på bästa sätt. Det står tydligt skrivet i lagar och standarder att företagen ska hitta och åtgärda risker men inte exakt hur de ska göra. Bra, konkreta och lättanvända metoder som beskriver hur man ska arbeta med risker som rör mjukvara och medicinska apparater behöver därför utvecklas. Detta är något som just nu en forskargrupp på Lunds Tekniska högskola arbetar med. En av forskarna har utvecklat en metod, RiskUse som involverar även användarna till apparaterna i identifieringen och hanteringen av risker. Forskargruppens forskning inriktar sig som helhet på utveckling av mjukvara, uppdelat på olika inriktningar såsom hur man arbetar med krav, testar mjukvara och förbättrar kvalitén på olika sätt på mjukvaran som utvecklas. Forskarna som deltar i forskningen som beskrivs i denna artikel, deras forskning är inriktad mot hantering av risker. En av forskarna är också speciellt inriktad mot medicinska apparater och kombinerar därmed sina kompetenser i både mjukvara och medicin.

Användarna är minst lika viktiga som apparaterna anser forskarna. Som exempel visar studier att nästan 90 procent av alla tillbud och olyckor som inträffar med övervaknings-apparater beror på den mänskliga faktorn, en människa som råkar göra fel. Varför gör vi fel? Det kan bero på många olika saker, som att vi är stressade, trötta, nervösa eller befinner oss i situationer där många saker påkallar vår uppmärksamhet. Även nya miljöer, nya apparater, tids-press, otillräckligt med information, instruktioner eller träning påverkar. Som ambulanssjukvårdarna tidigare, de måste veta exakt hur de ska använda utrustningen när de kommer till platsen så ingen tid går förlorad. Människor gör fel, det kan vi

aldrig undvika helt, men vi kan minska riskerna, också på områden där användning av medicinska apparater finns i stor omfattning såsom på våra sjukhus. Hur gör vi detta? Inom forskargruppen anser vi att det är viktigt att låta de som ska använda apparaterna, vara med i utvecklingen av dem. Ta tillvara användarnas unika kunskap och utnyttja den i de metoder som används vid utvecklingen.

Vi går tillbaka till händelsen i Malmö. Mannens hjärta började slå igen, sjukvårdarna fick igång hans hjärta och förde honom till sjukhus. Eftersom han hade allvarliga hjärnskador vårdades han på intensiven och hans tillstånd övervakandes dygnet runt med hjälp av flera övervakningsutrustningar.
Just en av övervakningsapparaterna, den som övervakar blodflödet i hjärnan har forskarna på Lunds Tekniska Högskola varit med i utvecklingen av. Vårdpersonalen har tillsammans med kvalitetsansvariga och forskarna deltagit i arbetet med risker kring övervakningsapparaten och även i testningen av själva apparaten. De har varit med och hittat och bedömt hur allvarliga riskerna är och i arbetet med att planera åtgärder för att ta bort riskerna eller göra följderna mindre allvarliga om en risk skulle inträffa.
Användarna har också testat övervakningsapparaten i så kallade *användartestning* där man registrerar exakt vad användarna gör när de an-vänder apparaten, vilka problem de har och även hur de tänker och upplever apparaten.
Övervakningsapparaten som utvecklats är en apparat som direkt och kontinuerligt mäter blodflödet i hjärnan på patienter som drabbats av stroke eller allvarliga skallskador. Vid skallskador gäller det att hela tiden se till att patienten har rätt flöde i hjärnan. Blir flödet för högt kan hjärnan svullna och blir det för lågt kan det blir syrebrist i hjärnan. Båda tillstånden kan leda till allvarliga bestående skador och i värsta fall döden. Blodflödet i hjärnan varierar betydligt mer över tid än man tidigare trott. Idag görs andra typer av undersökningar än den som testas i Lund. Dessa ger endast information om blodflödet just vid under-sökningstillfället och de är osäkra, tidsödande och dyra jämfört med den nya metoden.

Övervakningsapparaten som används till att övervaka patientens blodflöde i hjärnan.

Resultatet av studien visar att genom att kombinera användartestning och arbete med risker så hittar man fler problem och potentiella risker än man gör annars. Det visade sig att cirka 58 procent av användarnas problem som man hittade i användartesterna inte identifierats som risker. Det var speciellt två typer av problem som man inte sett som risker när man bara arbetade med riskerna. Den ena typen av problem är när användaren och utvecklarna har olika uppfattningar om hur saker är eller ska fungera. Ett exempel är användare som tror att de sparat text som de skrivit in men det har de inte gjort. Användaren har inte tryckt på den knapp som utvecklaren förutsatte att användaren skulle trycka på. Självklart för utvecklaren men inte för användaren. Den andra typen av problem visade sig vara funktioner som fanns på apparaten men de var för svåra att hitta, användaren hittade dem helt enkelt inte.

Bland de problem som både fanns med bland riskerna och de som hittades i användartestningen, kunde man se att vissa risker var undervärderade och vissa övervärderade. Vissa risker som man hade räknat med inte skulle ställa till några problem vållade många användare stora problem och tvärtom.

Om man undervärderar en risk så vidtar utvecklarna inga åtgärder för att ta bort eller minska risken vilket kan skapa onödig fara. Om man å andra sidan övervärderar en risk, kanske utvecklarna går in och gör onödiga ändringar på apparaten vilket kostar tid, arbete och pengar. Det kan också medföra att man inför nya risker som inte fanns tidigare. Slutsatsen man kan dra av studien är att användartestning är ett bra komplement till hela riskarbetet där också användarna är med.

Mannen med skallskadan är nu helt återställd men det finns exempel på där det inte slutar lika lyckligt. Två personer dog när

inställningarna i deras inopererade pacemakers ändrades på grund av strålning från andra apparater. Tre personer dog och flera skadades allvarligt vid strålbehandling av cancer. Personalen insåg inte att de gav 100 gånger starkare strålning än tänkt (Therac 25). Vi har också händelser som när en *infusionspump* levererade max-värdet istället för värdet personalen ställt in och övervakningsutrustningen som var kopplad till flera patienter samtidigt men sparade upp-gifterna för fel patient.

Nästa steg är nu att fortsätta utvärdera och förbättra riskhanteringsmetoden RiskUse. Målet är att metoden ska leda till apparater som är mer anpassade till användarna, som i sin tur kan leda till att användarna gör mindre fel och därmed ökas säkerheten för patienterna.

En första utvärdering har redan skett av RiskUse. Det gjordes i ett projekt som utvecklar medicinsk utrustning för vård av patienter i deras hem. Utvärderingen föll väl ut och metoden upplevdes som enkel att använda. Vissa delar av metoden kan förbättras ytterligare och efter det kommer nya utvärderingar att ske.

Ett fel i en medicinteknisk apparat kan vara skillnaden mellan liv och död. RiskUse kommer att fortsätta att utvecklas med förhoppningen att bidra till en tryggare och säkrare miljö för både patienter och sjukvårdspersonal. Människoliv och lidande kan inte värderas i pengar men minskas antalet fel och risker minskas även kostnaderna för vården.

Hjärtstartare: defibrillator är en apparat som används för att ge elektriska stötar till en person som drabbats av hjärtstillestånd.
Pacemaker: är en liten elektrisk apparat som opereras in under huden på bröstkorgen och elektroderna som tillhör apparaten placeras i hjärtat. Pacemakern känner av personens hjärtslag och skickar impulser för att skapa en jämn och regelbunden hjärtrytm.
Användartestning: är en metod som används för att utvärdera en produkt. Produkten testas av de som är tilltänka att använda produkten. Syftet är att testa produkten inte användarna.
Infusionspump: en elektrisk pump som kontrollerar tillförseln av vätska, läkemedel eller näring till en patient.

THESIS INTRODUCTION

INTRODUCTION

1 Research context

When a small slip, fault or mistake is made in our daily life, it might not be so severe, but in the health care domain the smallest mistake in development can make the difference between life and death. Medical devices can be safety-critical devices, which means that they have the potential of causing harm to people or the environment (ISO 2012*)*. It is essential to show that safety-critical devices are safe and of high quality. Quality and the concept of quality is an important part of health care. The history of quality in health care goes back to the 1860s and Florence Nightingale, who strongly advocated the need for a uniform system to collect and evaluate hospital statistics. She showed with statistics for example, that the mortality varied significantly between one hospital and another and she was one of the first to use statistics to persuade people of the need for change. Her efforts play an important role in laying the foundation for health care quality assurance programs (Small 1998).

Many functions provided by medical devices affect human lives, either directly when the medical devices are used in the treatment or indirectly when the devices are used in monitoring. A wide variety of these functions rely heavily on software. Most of these capabilities could not be offered without the underlying integrated software solutions. Software is becoming more and more important and widespread because of the introduction of new IT-systems, e.g., patient journal systems and administration systems, and the increasing amount of software in medical devices, such as defibrillators, cardiac rhythm management devices, and patient monitoring systems. Important quality attributes of software include, for example, inclusion of correct functionality, reliability with respect to fault content, usability for all users, and maintainability. Software is easier to change later in the development life

cycle than many other entities, which gives flexibility during development, but it also puts high requirements on quality assurance during development. Software is also of very high complexity and it is hard to develop fault free software in general (Vogel 2006). Several characteristics of the medical device domain itself contribute further to the complexity. The majority of stakeholders are non-technical professionals, e.g., physicians, nurses, and administrators and they work in an environment where they are often interrupted and are required to handle unexpected situations when they occur. It is impossible to categorise patients in the same way as products since treated patients have an unlimited set of characteristics that constantly change and interact (Garde & Knaup 2006). Other characteristics, contributing to the complexity, are the multitude of medical terminology and medical standards and laws to address specific issues within the medical device domain. There are various standards, laws and recommendations regulating the development of medical devices and medical device software. However, in many cases the standards are vague regarding the concrete software engineering techniques that should be used in the development life-cycle. In practice this gives the development organisations a high degree of freedom in instantiating the processes.

Traceability, safety, and risk are three important, highly intertwined concepts to consider in the software development process for medical devices. To comply with the regulatory requirements of the medical device domain it is essential to have traceability from requirements throughout the entire development and maintenance process. Traceability is also essential from risks to requirements and further within the risk management process. When it comes to safety, a safe medical system can be described as a system not causing a high degree of risk to property, equipment or people (Knight 2002). More specifically, medical device safety is concerned with malfunctions or failures that introduce hazards and is expressed with respect to the level of risk.

Risk and risk management is the focus of the research presented in this thesis and risk management is an important part of a development process for safety critical systems (Leveson 2011; Sommerville 2007). The thesis approaches problems with software risk management in the medical device domain and is focusing on integrating the user perspective into the medical device software risk management process.

A person is more inclined to take greater risks if the risks are voluntary and not forced upon the person, and a person perceives less

risk if he or she has trust in the source of the risk (Reason 1990). Trust is often defined in terms of risk and uncertainty and is defined to be an interaction between value, attitudes and emotions. In medical care situations, trust in both humans and medical equipment is crucial. If the expectations of the involved parties are not fully filled, trust is undermined and leads to insecurity and greater risks in the care situations. Errors are costly in terms of trust in health care systems and diminished satisfaction by both patients and health care professionals. Human errors and system failures can never be completely eliminated, however, it is possible to lower the risk of humans handling medical devices in an incorrect way. The majority of medical errors do not result from carelessness or the actions of a special group of users. Systems, processes, and conditions are leading people to make mistakes or hamper people to prevent them (Kohn et al 2000).

To be able to adjust and develop technology and to perform sufficient risk management, it is important to understand how the human mind and body works and how different factors affect people's actions. All humans age and their sensitivity to sound, light and colours degrade with age. In a working environment, actors often represent a mixture of different ages and when designing, for example, different user interfaces, instructions, notes, warnings and alarms, aging and other biological and human factors have to be considered

Risk management includes the identification of risks, analysis and evaluation of risks, risk control and monitoring risks over time. When covering these steps it is important to understand the complexity of the product and also the usage of the product. This means that it is necessary to consider medical device-related and usage-related factors as well as involving several different roles in the development process, such as users, developers and process experts. The research presented in this thesis has concluded that multiple roles, and thereby different experiences, will affect the risk identification process. By involving multiple roles, for example users and developers, in the risk identification process, it will result in a more complete set of identified risks than if only one role is included in the process. It was also shown that people are more and less risk seeking and by having a risk management group with multiple participants, preferable with different roles, the group will probably consist of both risk seeking and risk adverse participants. In order to support practitioners, mainly risk managers, RiskUse, a user perspective based software risk management process has been developed. The

concept of user perspective has been brought into the process by the use of predefined use cases in the risk assessment phase, the users attending the risk meetings and the use of usability testing as part of the process. The RiskUse process also supports traceability between requirements, use cases, hazards including risks and usability tests.

The first part of this thesis is an introductory part, summarising the research work and the second part includes a collection of six papers supporting the main contributions. The outline of the introductory part is as follows. Section 2 provides background information and related work of the research presented in this thesis. Section 3 presents research focus and the investigated research questions, followed by Section 4 presenting the research methodologies used to answering the research questions. Section 5 reports the synthesis of the research results and the conclusions and main contributions of the research. Finally, in Section 6 future research directions are outlined.

2 Background and related work

2.1 Medical device domain

A wide range of the functions provided by todays medical devices rely heavily on software. Research indicates an increasing importance and use of software and embedded systems, controlled and managed by software in the medical device industry (Allen 2014; Bovee et al. 2001; Chunxiao et al. 2013; Lindberg 1993; McCaffery et al. 2005; Méry & Kumar Singh 2010).

In the Medical Device Directive (MDD) 93/42/EEC (European Council 1993) the term "medical device" is defined as: "Medical device means any instrument, apparatus, appliance, material or other article, whether used alone or in combination, including the software necessary for its proper application intended by the manufacturer to be used for human beings for the purpose of:

- Diagnosis, prevention, monitoring, treatment or alleviation of disease.
- Diagnosis, monitoring, treatment, alleviation of or compensation for an injury or handicap, investigation, replacement or modification of the anatomy or of a physiological process.

- Control of conception (birth control, solve infertility, miscarriage etc.)."

It is important to notice that it is the manufacturer's purpose and the operation of the product that decides if the product is classified as a medical device, not the designer or the user.

Medical devices can be safety-critical devices, which means that they have the potential of causing harm to people or the environment. In the standard IEC 62304 (IEC 2006a) safety is defined as "freedom of acceptable risk" and according to Bowen and Stavridou (1993) safety can be defined as "freedom from exposure of danger or exemption from injury or loss". Further, in a safety-critical system functionality handling safety has to be designed into the system during the design phase and not later in the development process. In health care, there are many different safety critical systems, for example, defibrillators, dialysis machines, surgical devices and pacemakers. It is therefore essential to demonstrate that the safety-critical devices are safe and have high quality. This can be done through the application of a structured development process that is compliant with a safety standard. Examples of safety standards are IEC 61508 (IEC 2010a), which is a safety standard for electrical, electronic, and programmable electronic safety-related systems, and IEC 61511 (IEC 2003), which covers integration of components developed according to IEC 61508 (Gall 2008). Companies must comply with the regulatory requirements of the country in which they wish to market their medical devices. How strict and detailed the manufacturer's processes have to be depends on the safety classification of the product. The requirements for medical devices are defined in Europe in the Medical Device Directive (European Council 1993) and amendment MDD 2007/47/EC (European Council 2007) and in the US, the Food and Drug Administration, FDA, (FDA 2006) is responsible for the medical device regulation and compliance. Standalone software can, according to the amendment MDD 2007/47/EC (European Council 2007), be classified as an active medical device in its own rights. Every member state in the EU must adopt and publish laws, regulations and administrative provisions to implement the directive. There are some variations in national requirements; most of these concerns the need to notify the Competent Authorities, for example, in Sweden the Medical Products Agency (MPA), when medical devices are placed on the market in their countries. Duplication of registration procedures for a medical device placed on different markets is needed, even if it is the same

medical device. For example, a medical device placed on both the US and the European market, needs duplicate registration depending on the various prevailing laws and regulations. In order to market a medical device in Australia the device must be approved and registered by the Therapeutic Goods Administration (TGA), in China the approval must be obtained by the State of Food and Drug Administration (SFDA) and there are similar regulatory bodies in other countries throughout the world, for example, South Korea, Japan, Brazil and Mexico. In the work of harmonising regulations and standards, the International Medical Device Regulators Forum (IMDRF), is working towards global harmonisation in medical device regulations. IMDRF has permanently replaced Global Harmonization Task Force (GTHF) and consists of voluntary representatives from national medical device regulatory authorities. GTHF consisted of both voluntary representatives from national medical device regulatory authorities and from medical device industry. The goal was standardisation of medical device regulation across the world, the same goal as IMDRF, who also aim to accelerate towards harmonisation and convergence. IMDRF has more member countries than GTHF and they have the World Health Organisation (WHO) as an official observer.

Medical devices in the EU are divided according to the Medical Device Directive (European Council 1993) into different classes according to risk level, as presented in Table 1 and also with examples of medical devices in the respective class. All medical devices on the European market are classified in one of these classes based on the level of control necessary to assure safety and effectiveness.

Table 1. Medical device classification

Class	**Risk potential**	**Example**
Class I	Low	Syringe (non active)
Class Is (supplied sterile)	Low	Bandage (non active)
Class Im (measurement function)	Low	Thermometer
Class IIa	Moderate	Patient monitor system
Class IIb	High	Ventilators
Class III	Very high	Pacemakers

The manufacturers themselves classify the medical device. For medical devices classified in Class I the manufacturers themselves assess if they fulfil laws and regulations. The manufacturing process however, shall be controlled by a third part, often a Notified Body (NB). For medical devices in Class IIa a limited third part assessment is required where certain aspects are assessed. For the medical devices with the high risk potential classified in Class IIb and Class III it is required a full third part assessment. The classification is built upon the risks, which the human body can be exposed to due to the design, the use or the mode of manufacture of the medical device.

Medical information systems are systems, handling medical information such as information about the patient, images, diagnosis, medication, planned and completed treatment and so on, these systems are also classified. For example, transportation and storage of information (without affecting the information) are classified in Class I, imaging (CT, x-ray) in Class IIa, and control of treating radiological equipment in Class IIb.

The classification in the US differs from the European classification. They have three different classes, based on the level of control, necessary to assure safety and effectiveness. A medical device is assigned to one of these three regulatory classes and the three FDA classes are:

- FDA Class I require General Controls,
- FDA Class II require General controls and Special Controls
- FDA Class III require General Controls and Premarket Approval (PMA)

General controls are the baseline requirements of the Federal Food, Drug, and Cosmetic Act (FDA 2006) that applies to all medical devices. The manufacturer has to register their establishment and their device with FDA, comply with the labelling regulation, design and produce devices under good manufacturing practices (GMP), and submit a premarket notification [510(k)](FDA 1995) to FDA. The premarket notification [510(k)] is submitted to demonstrate that the device be market is safe and effective. FDA Class III is the most stringent regulatory category and usually contains devices that support or sustain human life and medical devices classified in Class III must have a premarket approval (PMA) from the FDA.

Concerning software, medical device software is regarded as a medical device when the manufacturer has specified the use of the software to be intended for one or several medical purposes defined above. Medical

device software can be a part of a medical device, a stand-alone software/IT-system or accessory to a medical device.

An analysis of medical device recalls by the FDA in 1996 (Wallance and Kuhn 2001) found that the software was increasingly responsible for product recalls. A subsequently made analysis showed that between 2006 and 2011, 5294 recalls were reported to the FDA and nearly 23 % of them were due to computer related failures. According to fault classes and risk levels, there is a dominance of software-related failures, but looking at the total number of devices, hardware-related recalls have a larger impact than software (Alemzadeh et al. 2013). To address such issues, various standards, laws and recommendations regulate the development of medical device software. In general, these standards describe software life-cycle models that shall be implemented by manufacturers. For example IEC 62304 (IEC 2006a) a key standard for medical device software development, covering the software life-cycle processes, ISO 13485 (ISO 2003) specifying requirements for medical device quality management system, EN 60601-1 (EN 2006) medical electrical equipment general requirements for basic safety and essential performance. EN 60601-1 (EN 2006) is the main standard containing the other standards, 60601-1-* and 60601-2-* covering, for example radiological equipment, EMC, alarm, electrosurgical equipment and electrocardiographs. Manufacturers are obliged, according to IEC 62304 (IEC 2006a) to assign safety classes to the software. The software at system level shall be assigned a safety class based on the most patient critical functions in the system. Parts in the software can be assigned a lower risk level than the whole system but not higher. The software safety classes are assigned according to the possible software hazard effects on patients, medical staff or other people resulting from a hazard to which the software can contribute. The classes are assigned based on severity as follows (IEC 2006a):

Class A – no injury or damage to health is possible.

Class B – non-serious injury is possible.

Class C – death or serious injury is possible.

Serious injury means life-threatening injury, permanent injury or when treatment is needed to prevent permanent injury.

ISO 13485 (ISO 2003), mandates that the medical device organisation's risk management process is documented and the standards IEC 62304 (IEC 2006a) and EN 60601-1 (EN 2006) specify basic risk management process actives. In practice, there is a high degree of

freedom in instantiating the processes.

The regulatory requirements do not specify the use of any particular development process when developing medical device software. However the standard IEC 61508 (IEC 2010a), a safety standard for electrical, electronic and programmable electronic safety-related systems recommends the use of the V-model to, for instance, achieve traceability (Smith and Simpson 2011). In the medical domain, it is shown that developers often use plan-driven software process models such as the waterfall model or the V-model (Lindholm & Höst 2008; McCaffery et al. 2012; McHugh et al. 2013). Though the use of agile practices within software development is increasing (Conboy & Fitzgerald 2010; Gary et al. 2011) the rate of adapting to agile practices within medical software development is slow (McHugh et al. 2013). However, McHugh et al. (2014) has found that there are no existing external barriers to adopt agile practices within the medical domain, on the other hand there are perceived barriers against adopting these practices. For example, agile practices are perceived to be contradictory to regulatory requirements and have insufficient coverage of risk management activities (McHugh et al. 2014). To show that it is possible to adopt agile practices to the development of regulatory compliant software, the Association for the advancement of medical instrumentation (AAMI) successfully mapped suitable agile practices to the stages of development in IEC 62304 (IEC 2006a) and presented this in a technical report (AAMI TIR 45:2012). Gary at al (2011) are also arguing that agile practices can contribute to safety critical software development and that they allow including activities related to risk reduction such as fault-tree analysis (FTA) and failure mode effects analysis (FMEA). Rottier and Rodrigues (2008) showed that adapting Scrum and map it to current process in a medical devices company and still satisfy standards and regulation is possible. It is also possible to combine participatory design with agile methods, even if this is not straightforward work (Abelein et al. 2013). When the agile process is tailored to meet the need of regulated environments and appropriate tools support the process, the agile approach is highly suitable in a regulated environment according to Fitzgerald et al. (2013).

To comply with the regulatory requirements of the medical device domain, it is essential to have traceability from requirements, including risks, throughout the whole development and maintenance process (Casey & McCaffery 2013). The requirement should be documented prior to development and this can be perceived as a barrier for adapting

agile practices (McHugh et al. 2014), however McHugh et al. (2014) have concluded that the FDA General principle of software validation, accept iterative software development models and that they thereby enables for the use of agile practices.

2.2 Risk management in the medical device domain

A challenge an organisation developing medical software has to meet is to identify a relevant set of risks for their products. Given potential of harm, inadequate medical device software can cause, has to be successfully addressed in the work with safety and risk management. Companies are required to have expertise in effective risk management practices, to be familiar with software safety and to be able to adopt a risk management mind-set. The medical device development organisations must also address different risks regarding patients, users, the environment, and third parties, for example, service technicians (Ratkin 2006). The research presented in this thesis is focusing on users and user risk. Users can involve different groups of users, where patients and third parties sometimes are part of the user groups and are using the medical device.

The risk management process is an important part of the development process for safety critical systems (Leveson 2011; Sommerville 2007). The term risk can be defined in different ways, risk is according to Fairley (2005) ‘‘the probability of incurring a loss or enduring a negative impact’’ and according to Leveson (1986) ”a function of the probability of a hazardous state occurring in a system, the probability of the hazardous state leading to mishap, and the perceived severity of the worst potential mishap that could result from the hazard”. Leveson’s definition is more in line with the definition in the standard for application of risk management to medical devices, ISO 14971 (ISO 2012) where risk is defined as “combination of the probability of occurrence of harm and the severity of that harm”. The standard refers to harm instead of mishaps and harm meaning “physical injury or damage to the health of people, or damage to property or the environment” (ISO 14971 2012). Risks can be classified into three classes according to acceptance Sommerville (2007); a) intolerable, when the system is designed so the risk will never rise or if it rises, it will not result in an accident, b) as low as reasonable practical (ALARP), the system is designed so the probability of hazard is minimized, and c) acceptable when the design has reduced the probability of an acceptable hazard

without increasing costs or time. According to mitigation of risks, the ALARP principle considers that any mitigation can result, new risks as well (Bianco 2011).

Risk management (Boehm 1991; Hall 1998; Crouhy et al. 2006) typically includes identification of risks, analysis and prioritisation of risks, and handling and monitoring of risks. Relevant people identify risks during the risk identification and then the risks are prioritised with respect to the probability of the risk actually occurring and the potential effect they will have if they occur. According to Pfleeger (1999) the prioritisation of risks is often carried out through discussions where participants see risks in different ways and valuate them differently.

A well-defined risk management process must be applied throughout a product's whole life-cycle process, from inception until the product is no longer in use. The risk management process presented by Hall (1998) consists of five essential elements and the risk management process presented in ISO 14971 (ISO 2012) consists of four essential elements, both processes are presented in Figure 1. How the elements in the two processes correspond to each other are indicated with arrows in the figure.

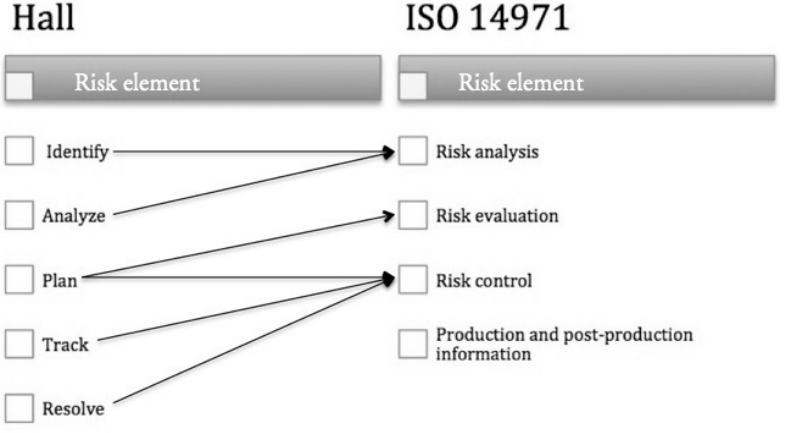

Figure 1. Essential elements of the risk management process

The two compared risk management processes use different terminology in their descriptions. Hall (1998) is referring to risk, defined as "a measure of the probability and consequence of an unsatisfactory outcome", (e.g. similar to the risk definition in ISO 14971 presented above) and ISO 14971 (ISO 2012) is referring to hazard. Hazard, hazardous situations and harm are the key concepts in risk management

within the medical device domain. According to Leveson (2011) there is a problem with the definition of hazard as "potential source of harm" (ISO 2012), since all system states have the potential to cause harm.

In a typical risk management process, the manufacturer of a medical device shall identify the hazards associated with the medical device, estimate and evaluate the associated risks, control these risks and monitor the effectiveness of the control. The risk management processes by Hall (1998) and the risk management process in the standard (ISO 1497 2012) are similar in content. The first step, risk analysis in ISO 14971, includes the two first steps, identify and analyse in the process by Hall (see Figure 1). During the risk analysis process are hazards identified and the risk(s) is estimated for each hazardous situation (e.g. assessment of severity of harm and probability of occurrence).

In the risk evaluation step shall for each identified hazardous situation, be decided if risk reduction is required or not. The decisions are based on predefined criteria. The risk evaluation step in ISO 14971 is part of the planning step in the process by Hall. The step called risk control in ISO 14971 covers the activities in part of the planning step, the tracking step and the resolving step in the process by Hall. During the risk control phase risk control measures are decided and implemented for all hazardous situations. Risks arising from risk control measures shall also be handled and residual risk evaluation performed where it shall be determined if the implemented measures have made the risk acceptable, if not must additional risk control measures be implemented. The risk control phase in ISO 14971 contains risk/benefit analysis, not included in the risk management process by Hall; neither is the production and post-production. For some occasions where the risk is greater than the criteria for acceptable risk is the manufacturer obliged to do a risk/benefit analysis to show that the benefit outweighs the risk and the users need to be informed about the remaining risks (residual risks). Information important for the production and post-production phase that are gathered and documented during the risk meetings, for example, introduction of warnings in the graphical user interface, labelling and special training, shall be documented in the risk management report. Post-production problems reported by users, personnel who install the device, service personnel and product instructors should be discussed at a risk meeting and if so decided, the problems shall be incorporated in the risk management process.

Risk reduction can be implemented at three levels (ISO 14971 2012), at the first level, by inherent safety by design, at the second level by using proactive measures in the medical device or in the manufacturing process and at the third level, by informing the user. The risk reduction shall be introduced in this order but the three levels can also be used in combination. The most effective way to reduce defects and avoid serious consequences is, to design in safety in the software product during the development process (Ratkin 2006; Cooper & Pauley 2006). The risk management process described in ISO 14971 and the risk management process described by Hall has been carefully studied during the development of RiskUse.

The specific standard regarding risk management for medical devices is, as mentioned, ISO 14971 (ISO 2012) and the risk management standard for IT-networks incorporating medical devices is IEC 80001-1 (IEC 2010b). The responsible organisation has to coordinate a high-level risk management process of its IT-networks and the manufacturer needs to supply information about residual risks connected to their network products. Risks connected to IT-networks could for example be incorrect access, corrupt or incorrect data, and lack of traceability. Some examples of typical medical device hazard can be, incorrect measures, loss of function, incorrect output, memory failure, and use errors. It is also important to notice that there is a difference between wilful or reckless misuse of a medical device and misuse of a device because the user uses the device in other ways than intended by the manufacturer. The later misuse might be the result from, for example, misunderstanding of the use instructions, which means that it is important to consider the user instructions in the risk management process. When analysing hazards, to determine if the device can be used in ways that deviate from the intended use, is found to be difficult. (McRoberts 2005).

To provide high-level guidance in achieving regulatory compliance in the risk management field, there are guidance document published. IEC/TR 80002-1:2009 (IEC/TR 2009) provides guidance on the application of ISO 14971 (ISO 2012) to medical device software. Two other examples of guidance documents are, Do it by design (FDA 1996), introducing human factors in medical devices development and Medical device use-safety (FDA 2000), providing guidance on how to incorporate human factors engineering into the risk management process. However the authorities do not provide any real detailed guidance or specific methods demonstrating how regulatory compliance shall be achieved,

even if they require a demonstration of regulatory compliance from organisations developing medical devices.

Several approaches and strategies are used in order to address risk management within the medical device domain. To trace risks in medical devices software or systems, Fault Tree Analysis (FTA) (Krasich 2000; Hyman 2002; IEC 2006b) or Failure Modes and Effects Analysis (FMEA) (IEC 2006c; Chiozza & Ponzetti 2009; Jain et al. 2010; Xiuxu & Xiaoli 2010) are often used. In a survey (Paper I) the findings indicated that FMEA is the most frequently applied method and FTA seems to be of lower importance. FTA (IEC 2006b) is a top–down analysis method where undesirable end events are identified and then all contributing factors, determine which failures are most critical. The fault tree analysis begins at system level and starts with a top event, that is a failure or an undesired event, then systematically identifying factors or events at lower levels contributing to the top event. The lower levels of events are combined in series by the use of Boolean logic and results in a graphical presentation of cause and effect. However the fault tree can expand widely and generate a need for tool support. FTA can be combined with other methods, for example, FMEA providing a bottom-up analysis and thereby contributing to a more comprehensive analysis (IEC 2006b). FTA is a practical method for causal analysis of the undesirable events and can be used for both single and multiple failure modes (IEC 2006b).

The main purpose of FMEA is to early in the design process identify potential problems that can affect safety and performance and take action to eliminate or minimize them (Kamm 2005)

As mentioned before, FMEA (IEC 2006c) is a bottom-up analysis method and it is used to identify each potential failure mode for all the parts in the system and trace negative effects through the system. The analysis starts with the lowest level of components and proceeds up until the effect of the system is identified. A failure effect at a lower level can become a failure mode of an item at the next higher level. The FMEA process can also provide measures according to severity, occurrence and detection and risk priority number can be calculated as a product of these three measures (Xiuxu & Xiaoli 2010). Advantages with FMEA are that it can be tailored to meet specified industry and product needs (IEC 2006c) and by using FMEA every component of the system is systematically examined (Jain et al. 2010). A limitation however, is that it can only be used for single failure modes (Jain et al. 2010).

Failure Modes and Effects Criticality Analysis (FMECA) is an extension to FMEA where severity ranking of the failure modes are made and allows prioritisation of countermeasures. FMECA investigates how the system detects and recovers from failures and for each failure mode is effects criticality and description documented (Becker and Flick 1997).

Another failure mode method is Healthcare Failure Mode and Effect Analysis (HFMEA™) developed by the United States Department of Veterans Affairs' National Center for Patient Safety, it is based on multidisciplinary teams identifying possible failure modes using graphical described health care processes (Habraken et al. 2009). The objective of HFMEA is to systematically identify and analysis potential failure modes of healthcare processes and for those failure modes requiring further analysis make decision tree analysis. HFMEA also includes action planning and after that evaluation of the planned actions (Trucco & Cavallin 2006). Two drawbacks with HFMEA identified by Habraken et al. (2009), the method is very times consuming and the risk assessment part is difficult to carry out.

Hazard and Operability Studies (HAZOP), is a qualitative method, for identifying hazards and operational problems with the use of guide words (more, less etc.). Emphasis is put on the meetings where deviations in every information flow of the design is identified, analysed and documented, as in an iterative process (McDermid et al. 1995; Paper I). HAZOP can be used early in the system and software design to reduces the amount of design changes later in the process (Jain et al. 2010) and the method should preferable be used at higher levels of complex systems to remain cost effective (McDermid et al. 1995).

The medical device regulatory requirements require production and postproduction monitoring of the medical device for discovering additional or unexpected sever risks. The Corrective Preventive Action (CAPA) system is used in some cases, to collect, organises and trace failures. Information about problems and issues is collected from, for example, internal reviews or user complaints and the problems are evaluated for risk, severity and necessary action. Necessary actions are then taken to correct the problem and prevent their reoccurrence (Bills & Tartal 2008; Lozier 2010).

Several researchers have reported on risk management on software development in general, e.g. Boehm (1991), Hall (1998), Charette (1989), and Jones (1994). In the medical domain, the published research discusses risk management from a high-level perspective; often the overall

risk management process is described without detail descriptions of each process step. McCaffery et al. (2009, 2010) who have developed and tested a software process improvement risk management model (Risk Management Capability Model) that integrates regulatory medical device risk management requirements with the goals and practices of the Capability Maturity Model Integration (CMMI) describes one example. Schmuland (2005) also investigates the whole risk management process, although he focuses on residual risks, i.e. the remaining risks after the risks have been handled, and how to assess the overall residual risk of a product. It is based on the identification of all the important scenarios. Hegde (2011) presents a case study of risk management based on ISO 14971 (ISO 2012) and concludes that the standard as guideline can ensure a safe product with an acceptable level of risk. Then, there are several studies presenting specific methods, for example, the use of FMEA in the risk management process (Chiozza and Ponzetti 2009; Xiuxu and Xiaoli 2010; Habraken et al. 2009) and also different frameworks (Barateiro and Borbinha 2012; Iversen et al. 2004); Padayachee 2002). Benet (2011) suggests a risk driven approach to medical device testing as a way of handling the risk verification process and assure overall safety of the medical device. There are some researchers that focus on one of the steps in the risk management process. In the medical domain, for example, Sayre et al. (2001) in particular studied the risk analysis step. They described an analytical tool for risk analysis of medical device systems, a safety model based on Markov's theory and argue that this safety model presents significant opportunities for quantitative analysis of several aspects of system safety. Dey et al. (2007) have identified the need for analysing risk management issues in software development from the developers' perspective with the involvement of the stakeholders.

The different laws and regulations, standards, guidelines and methods described in this thesis have been studied in-depth during the development of the new risk management, RiskUse. The terminology used in RiskUse is adapted to the terminology used by regulatory bodies and the requirements within the medical device domain.

To summarise, the discussed standards, guidelines and methods in this section, regarding risk management in the medical devices domain are presented in Table 2.

Table 2. Risk management standards, guidelines and methods

Standards	ISO 14971 IEC 80001-1
Guidelines	Do it by design Medical device use-safety
Methods	Fault-tree analysis (FTA) Failure mode effects analysis (FMEA) Failure Modes and Effects Criticality Analysis (FMECA) Healthcare Failure Mode and Effect Analysis (HFMEA™) Hazard and Operability Studies (HAZOP) Corrective Preventive Action (CAPA)

During the development of RiskUse all the standards, guidelines and related work were closely studied and considered.

2.3 Human error and usability

The research in this thesis focuses on involving users in various ways in the risk management process. The user is a key player in the usability field and in the medical device domain is the user defined as "any human that might handle, operate and otherwise interact with a medical device though the device user interface" (IEC 2007). To achieve successful development of software systems it is essential to have users participating in the software development process (Abelein & Paech 2014). User participation and user involvement are often interchangeable concepts, but Abelein and Paech (2014) use two separate definitions. User involvement concerns the psychological state of the user, how important and personally relevant a system is to the user and user participation on the other hand is focused on the behaviours and activities the user perform during the system development process. Direct communication between users and developers is a specific form of user participation and in large scales IT projects this communication is limited, and there are no commonly used methods to achieve that (Abelein & Paech 2014). However a review study by Abelein et al. (2012) showed that various aspects of user involvement and especially user participation have a positive effect on system success. Integrating users in the risk management process, as in RiskUse, is one way to increase direct communication between users and developers.

Doerr et al. (2008) argue for the need to consider usability and user acceptance issues early in the development of medical products and present an approach where overall user satisfaction is measured. If long development cycles are used the end user do not feel integrated in the project (Abelein & Paech 2013). By involving users in the risk management process from the beginning, users can feel more involved in the project and the thoughts on usability are introduced early. Usability is the "weakest link in the security chain" in many systems and in many cases there is a trade-off between usability and security (Jørsang et al. 2007). Usability problems that could threaten the security could, for example, be that the users do not understand what action that are required of them or the system does not provide sufficient information to the user to take corrective action (Jørsang et al. 2007).

Where there are users, there are also human errors and historically, the earliest documented report of human errors in medical device usage, can be traced back to 1849 when an error in the administration of anaesthetic resulted in death (Dhillon 2008). Today, human errors in health care are the eighth leading cause of death in US (Dhillon 2008); the costs are high, and more than 50% of technical medical equipment-related problems are caused by operator errors (Dhillon 2000). Walsh and Beatty (2002) refer to a wide range of studies showing that 87% of critical incidents connected to patient monitoring is due to human factor errors. Other medical devices with high incidence of human errors are according to Dhillon (2008), for example, glucose meters, balloon catheters, orthodontic bracket aligners, and administration kit for peritoneal dialysis. System processes that lead the users to making mistakes cause more common errors in health care systems. Between 44 000 and 98 000 patients die in hospitals, throughout the world, from medical errors that could have been prevented (Kohn et al. 2000). Since users still cause many errors, user errors have to be reduced. Users involved in identifying and evaluating risks are one way, and usability testing contributes to the users actual behaviour.

The concept human errors include all the occasions when a planned sequence of mental or physical activities does not lead to the intended result and when the failure cannot be related to chance (Reason 1990). It is important to be aware of the various factors that influence people to do wrong and why they make these errors, in order to be able to assess the risks and be able to choose the correct countermeasures. Two important factors connected to human errors are cognition and perception. where

cognition embraces mental processes such as memory, reasoning, thoughts, decision-making and problem solving. Perception on the other hand is how people perceive the world around them through their senses (e.g., eyesight, hearing, smell, etc.) and how people perceive expectations and control from the outside incoming data. A person's prior knowledge of a situation is the foundation for the perception in similar situations (Reason 1990).

Human errors can be divided into three different primary error types; mistakes, lapses and slips and they can occur in the different cognitive stages of planning, storage, and execution (Reason 1990). Mistakes can further be divided into mistakes made by an expert or mistakes made by a non-expert. The expert has a large collection of problem-solving routines, can see things on an abstract level and is able to work with more extensive problems than the novice. If an expert and novices are given the same problem to solve, the expert way of thinking based on professional experience makes the expert's error more predictable. However, if an expert runs out of acceptable problem-solving routines the expert's performance approaches the level of the novice (Reason 1990). Experiments have shown that people stick to accustomed solutions even if there are solutions that are smarter and simpler, people stick to their rules. A rule that is proven to be useful in a specific situation, defied as a "good rule" by Reason (1990) and the first time this "good rule" does not work, a strong-but-now-wrong rule is used, which results in the development of variant rules for use in different situations. There are many factors that affect people's behaviour and way of thinking. If the information does not fit the individual's conception of the world and if an individual is overloaded with information, then only a part of the information is processed by the individual. Rules that are used many times with success are strong rules and only a partial match is needed for it to be used. The use of rules is influenced by the individual's inherent cognitive conservatism and to illustrate that is the quotation "for a person with a hammer, every problem looks like a nail" a good example (Reason 1990).

Humans make various types of errors and they can be classified according Dhillon (2008) in seven different classes, presented in Table 3 accompained with examples of reasons that can cause the errors.

Table 3. Classification of human errors

Human errors	Example cause
Assembly errors	Poor illumination Poorly designed work layout Poor communication if related information
Design errors	Failure to implement human needs in the design Failure to ensure the man-machine interaction effectiveness Failure to assign inappropriate functions to humans
Handling errors	Due to poor transportation or storage facilities
Inspection errors	Rejecting and accepting in-tolerance and out-of-tolerance parts, respectively
Installation errors	Failure to install equipment according to the manufacturer specification
Maintenance errors	Repairing the failed equipment incorrectly Calibrating equipment incorrectly
Operator errors	Complex tasks Operator carelessness Poor training

Changes are also a source of human errors, when will establish routines are left. People are mentally prepared to change and have rules to cope with changes, but even if the change is expected, the person is now a novice in the situation and has no old routines to fall back on. Mistakes can for example be seen when there is a change in design or instructions (Reason 1990). A further source for human errors can be interruptions in performing activities. It is common that the medical staff is interrupted when performing different activities. The staff's attention is captured in a critical phase and then a stronger routine takes command over their action or they miss out doing parts that they were supposed to do (Reason 1990). When individuals are exposed to high pressure, stress and demands increase the risk for faults and errors. To lower risks or increase productivity without increasing the risks are the common activities, education and training, selection, improve human-

machine systems, improve the working environment and improve the management and the psychosocial environment (Reason 1997). Medical staff is exposed to stress and stressful situations and stress affect the individual as well as the environment around him or her. Stress is a cognitive state when the individual perceives that the demands exceed her coping resources and that the she cannot handle the situation or the demands. Increased pressure on the individual increases the risk of making the wrong actions even moderate pressure can cause stress. Stress also affects the way of making decisions since the individual only consider the most distinguished parameters and that gives a limited rationality. In limited rationality provide the decision maker herself with a simplified model of reality and acts rationally according to the model, for example, value time and cost short-sighted instead of quality long-sighted. Limited rationality gives increased risk for accidents and that risk is also increased by late decisions (Reason 1997).

To lower the risk for human errors and also to improve work performance, there are important qualities to consider when designing devices and system. These qualities are visibility, things humans perceive with all senses and this can be accomplished by showing the right information by grouping information, colours, icons and text, affordance when the artefact leads us how to use it, mapping when design or placement of controllers or information carriers mirror how to be used, feedback, when the user is given feedback according to what has happened or happens as consequence of her behaviour, and last but not least, usability according to relevance, efficiency, attitude and learnability (Reason 1997).

Potential user-related hazards are best identified and addressed by human factors engineering (HFE). HFE is defined as the application of knowledge about human capabilities and limitations to design and development of devices, systems, tools, organisations and environments (ANSI/AAMI 2009). The process of HFE extends to all medical devices and has an impact on both the risk management process and life-cycle process. The concept of human factors is described by the FDA as “a discipline that seeks to improve human performance in the use of equipment by means of hardware and software design that is compatible with the abilities of the user population” (FDA 1996). In the risk management work, in order to get safe and effective medical devices, human factors regarding use environment, user and device have to be considered (FDA 2000) and human factors must be considered in the

design and the safety assessment process of the system (Cacciabue & Vella 2008).

When it comes to medical device risk assessment focusing on users, there are critical factors to consider both according to the medical device itself and to the usage of the device (Dhillon 2008). These critical medical device-related and usage-related factors are presented in Figure 2 and it can be noticed that human factors are critical factors to the medical device as well as the usage of the device. It is also important to bear in mind that use patterns are clearly different from the same medical device, for example, used in day surgery than the use in a helicopter-based medical rescue service (Wilkins & Holly 1998).

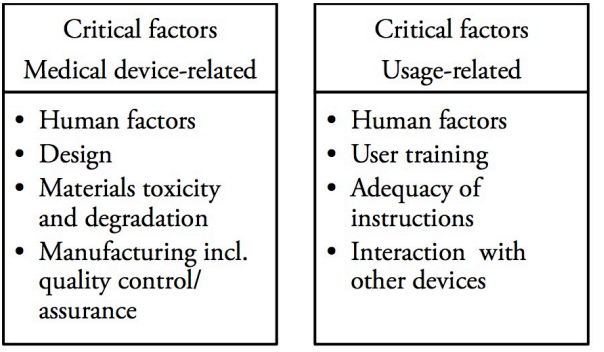

Figure 2. Critical factors

In Europe, IEC 62366 (IEC 2007) is the standard for application of usability to medical devices. In the standard the term usability engineering is used. The terms human factors engineering and usability engineering are often used interchangeably for the process of achieving highly usable devices. To minimise user errors and understand user-related risks, it is important to have a complete understanding of how a device will be used and the goal with incorporating users in the risk management process is to minimise usage-related hazards so the intended users can safely use the medical device. User errors can occur in normal use and is an act or omission. Such use error results in a medical device response that differs from the response expected by the user or the response intended by the manufacturer (IEC 2007).

There are several standards involving usability, the ANSI/AAMI HE 74-2001 (ANSI/AAMI 2001), ANSI/AAMI HE 75-2009, (ANSI/AAMI 2009) and the third edition of the medical electrical equipment standard

EN 60601-1 (EN 2006) where usability is an integrated part of the standard. To also provide some high-level guidance in achieving regulatory compliance there are guidance document like, Do it by design, an introduction to human factors in medical devices (FDA 1996) and Medical device use-safety, incorporating human factors engineering into risk management (FDA 2000).

Usability and usability engineering is getting more and more important in the medical device domain (Hrgarek 2012). Usability is according to a working environment often broken down to at least six goals (Rogers et al. 2011);

1. Effective to use – if the users can carry out their work efficiently
2. Efficient to use – when the user has learned to use the product can the user sustain high level of productivity. How well the product supports he user.
3. Safe to use – product safe to use, protecting the user from damage an undesirable situations, but also in the medical device domain the patient and sometime the environment.
4. Having good utility – provide an appropriate set of functions. Provide the right functions so the users can do what they need and wants to do.
5. Easy to learn – how easy it is to learn to use the system. People do not like to spend time on learning how to use a product.
6. Easy to remember how to use – Once learned, how easy to remember how to use the product.

It agrees well with the definition in the European standard IEC 62366 (IEC 2007) where usability is defined as "characteristics, if the user interface that establish effectiveness, efficiency, ease of user learning and the user satisfaction". The standard is focused on how to find and identify user hazards where user hazard is a situations connected to the use of the device that can harm the patient or the staff.

The usability engineering process, whose primary goal is to make the medical device safer, more effective and easier to use, needs to be incorporated in the overall development process (Gosbee & Ritchie 1977). Usability tests, interviews and questionnaires are commonly used methods for capturing user perspectives (Shah & Robinson 2006) and can also be used in the risk management process. Usability testing is used as a part of the process in RiskUse.

The users expect the user interface to follow their logic and the product to serve them (Merrill and Feldman 2004) and even if the users

has no primary responsibility they are the key to product success and it is important to collect details about the end users (Anderson et al. 2001). There are different ways to gather and document information about the users, for example the use of user matrix (Merrill & Feldman 2004) and personas (Anderson et al. 2001). The users can be divided into user groups and in the medical domain they are often divided into the following groups, a) decision makers such as physicians and specialists, b) care providers such as nurses and health care specialists, and c) care receivers such as patients and patient family (Yang et al. 2003). Depending on the device that is developed is determined which user groups are interesting to involve in the process. This decision is taken in the preparation phase in RiskUse.

For evaluating usability, usability testing is considered one of the most powerful ways (Daniels et al. 2007) and perhaps the most powerful one (Kushniruk 2002). Usability testing was therefore chosen as an integrated part in RiskUse. Kushniruk et al. (2005) has used usability testing to study the relation between usability and errors and found that different types of usability problems can be associated with specific types of errors, so there might be a possibility in using usability engineering to predict medical errors. According to Wiklund et al. (2011) ”usability tests are like snowflakes meaning that each is unique” and need to be designed according to existing circumstances (Wiklund et al. 2011). When the usability test is performed it is recommended to use a direct method and the one most in favour is, where the test user is thinking aloud or in combinations with the facilitator asking questions (Daniels et al. 2007; Holzinger 2005; Velsen et al. 2007). The test user is encouraged to verbalise their thoughts during the test, describe what they are doing, what they are thinking and so on. However there is a problem because people have a tendency to be quiet when they meet a problem and for many people it feels unnatural to talk all the time. Question asking is therefore a recommended complement to thinking aloud where the facilitator is asking questions like; what do you think now?, what are you doing? and so on. To perform a usability test it is needed around five participants and that results in that around 80% of the real problems will surface (Nielsen 1994; Virzi 1992). The facilitator gives the test user specific predefined tasks to perform and then the test users actions are logged, either written down or videotaped. Then all the material is analysed and reported (Garmer et al. 2002; Wiklund et al. 2011). Madrigal and McClain (2010) provide practical guidance including a list

of “do’s and don’ts” to consider when performing usability testing. They also point out that “usability testing is not one of the most glamorous but most important aspects of user experience research”.

Bartoo and Bogucki (2013) have established that user errors are a significant source of harm to patients; especially where users pressing the wrong button or the users are not requested to confirm important actions. Obradovich and Woods (1996) have in earlier studies found that common human-computer action problems are poor feedback about device state and behaviour, complex and ambiguous sequences, ambiguous alarms and the users getting lost in multiple displays. In the standard (IEC 2007) is use errors defined as “act or omission of an act that results in a different medical device response than intended by the manufacturer or expected by the user”. However, it is often difficult to anticipate problems with device usage that could result in hazards because users interact with devices in many different ways and a device used safely by a group of user might not be used safely by another (FDA 2000). This makes it very important to consider different factors regarding the user environment; the user and the device itself according to FDA (2000), see Figure 3, inspired by FDA (2000). When the dynamics of user interaction results in harm caused by use errors, it is related to safety and should be part of risk management. Interaction between human factors considerations can result in either a safe and effective use or in an unsafe and ineffective use. Examples of device user factors to consider are knowledge and expectations, regarding device user environment factors, light, distraction and workload, and device user interface factors could be, operational requirements, device complexity and specific user interface characteristics.

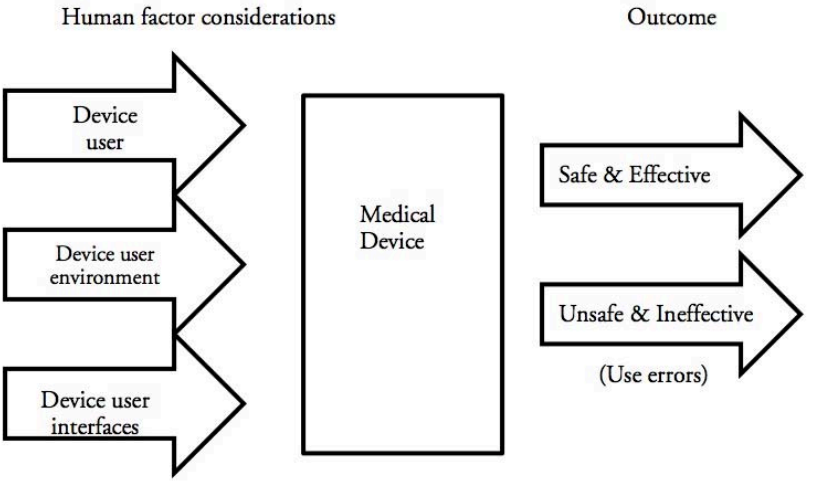

Figure 3. Interaction of human factor consideration

3 Research focus

The main goal of the research effort in this thesis is to integrate users and user perspective in the software risk management process in the medical device domain, and to develop a new risk management process including a user perspective. The risk management process RiskUse have been developed and evaluated.

This thesis is based on empirical research with both qualitative and quantitative approaches. To strengthen the validity of empirical research, triangulation is an important concept (Runeson et al. 2012) and there are four different ways to apply triangulation (Stake 1995). Triangulation can be applied by using more than one data source or collecting the same data at different occasions (data triangulation), by using more than one observer in the study (observer triangulation), by using alternative theories (theory triangulation) or by combining different data collection methods (methodological triangulation). In this thesis a combination of qualitative and quantitative methods is used. Data triangulation in the papers presented in this thesis has been used by combining multiple data sources, e.g., observations and interviews in the case studies (Paper IV and VI) and by using the same usability test method in two case studies (Paper V and VI). To achieve observer

triangulation and to lower the risk of researcher bias, more than one researcher are involved in the studies.

3.1 Research questions and research papers

The main research questions (RQ) investigated in this thesis are as follows:

- RQ1: What is the state of practice of medical devices with respect to the software development and software quality assurance including risk management*?*
- RQ2: What differences can be identified between the users of a system and developers of a system with respect to risk identification?
- RQ3: How can different people's risk tendency be defined in an adequate way, with respect to conception of risk, in order to support the risk management process?
- RQ4: How can users be integrated in the risk management process?
- RQ5: How can usability evaluation methods, especially usability testing contribute to the risk management process?
- RQ6: How can a software risk management process, including user perspective be designed to be appropriate for a medical device development organisation?

The six research questions and in which paper in this thesis they are addressed in, are listed in Table 4. Different methodological approaches have been used in the research. RQ1 was investigated in a survey, RQ2 and RQ3 in controlled experiments and RQ4, RQ5 and RQ6 in case studies. The research methodology is further described in Section 4.

The relationship between the different research questions and papers are presented in Figure 4. The research in Paper I, Paper II and Paper III aimed to gain empirical insight into factors at play in the medical device domain and especially practices and challenge in risk management. From the two case studies in real-life contexts, presented in Paper IV and Paper V, experiences and empirical data were gathered to validate the relevance of the research questions and give valuable input to the process of developing a new risk management process. The research presented in Paper IV and Paper V contributes to answering RQ4, RQ5 and RQ6 and experiences and lessons learned were further used to develop the new risk management process, RiskUse, presented and evaluated in Paper VI.

Table 4. Research questions cover by this thesis an in which paper.

Research question		Addressed in
RQ1:	*What is the state of practice of medical devices with respect to the software development and software quality assurance including risk management?*	Paper I
RQ2:	*What differences can be identified between the users of a system and developers of a system with respect to risk identification?*	Paper II Paper III
RQ3:	*How can different people's risk tendency be defined in an adequate way with respect to conception of risk, in order to support the risk management process?*	Paper III
RQ4:	*How can users be integrated in the risk management process?*	Paper IV Paper V Paper VI
RQ5:	*How can usability evaluation methods, especially usability testing contribute to the risk management process?*	Paper V Paper VI
RQ6:	*How can a software risk management process, including user perspective be designed to be appropriate for a medical device development organisation?*	Paper IV Paper VI

RiskUse was evaluated in a real-life context and empirical data was gathered for further improvement of RiskUse and the research gave also more extended answers to RQ4, RQ5 and RQ6. The transfer process from knowledge to practice is presented more in detail in Section 4.3.

The research contains a survey in Paper I presenting the characteristics of the state of practice of software development in the context of the medical devices and systems. A part of the survey focuses on quality assurance of software, risk management and developers conception of safety criticality of software. Findings indicate that FMEA is the most frequently applied method and FTA seems to be of lower importance.

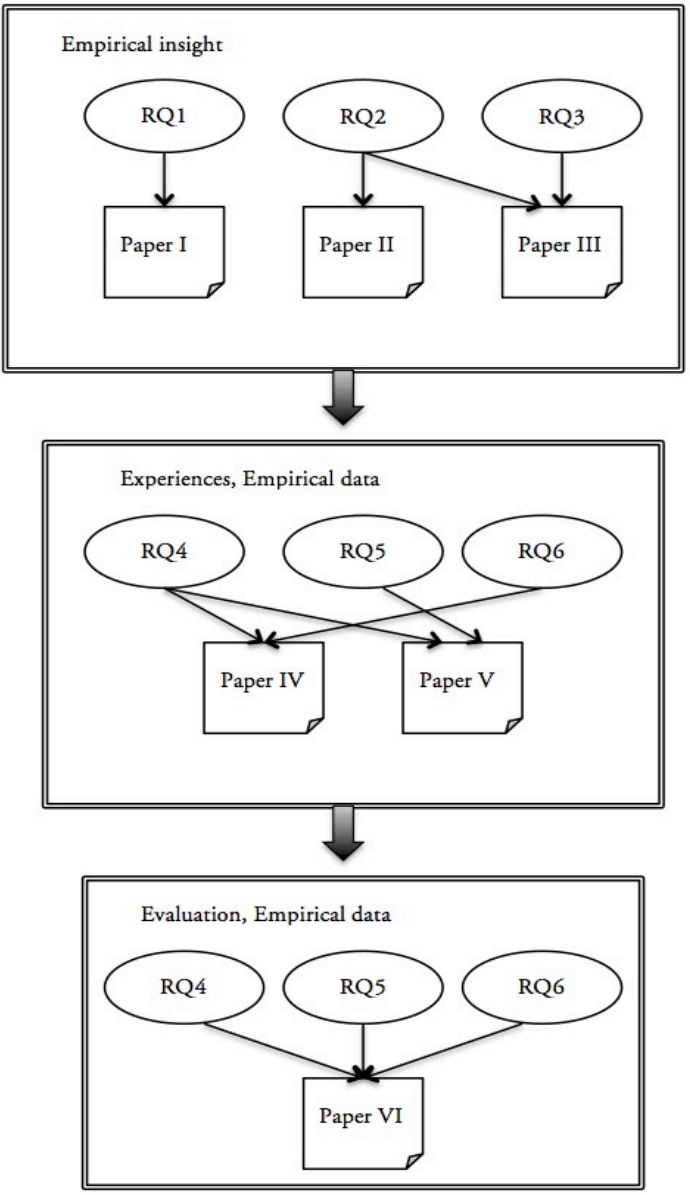

Figure 4. The relationship between the different research questions and research papers in this thesis.

The conception of risk was further investigated in two experiments in Paper II and Paper III. The differences between physicians, software developers' and medical software developers' views of risks were investigated in a controlled experiment, presented in Paper II. Risk tendency was investigated in another controlled experiment (Paper III). The identified challenges and experiences from the survey and experiments were utilized after that in three case studies.

Paper IV presents experiences and lesson learned from the first case study focusing on user risks in the three first steps of the risk management process. In the second case study (Paper V) usability testing was introduced as a part of the risk management process. This was also established in Paper VI.

A new software risk management process, RiskUse was derived from the experiences and conclusions gained from the two case studies in Paper IV and V and from document studies (e.g. standards). The aim of the new process is to support software risk management activities in the medical device domain and to bring in the user perspectives into the risk management process. The last paper in this thesis, Paper VI presents the case study where the new software risk management process, RiskUse was empirically evaluated within the medical domain.

4 Research methodology

For the research presented in this thesis, empirical research methods have been used. Empirical research is seeking to explore, describe and explain different phenomena through, collecting and using evidence based on observations or experiences. The evidence is obtained through for example interviews, surveys or experimentation (Wohlin et al. 2000; Sjøberg et al. 2007). Guidelines for empirical research in software engineering was introduced by Kitchenham et al. (2002) and empirical studies have today gained acceptance and is an important part of software engineering research (Shull et al. 2008; Runeson & Höst 2009) According to Seaman (1999) most empirical software engineering studies combine qualitative and quantitative methods and data. In this thesis a variety of research designs, research methods and data collection methods have been used to deal with the research challenges and to provide answers to the research questions. The research is focusing on practical, useful knowledge defined as practical knowledge by Easterbrook et al. (2008) and is characterised by a pragmatic approach. Practical knowledge

is valued and adopts an engineering approach to the research (Easterbrook et al. 2008).

Design science improves the environment in which it is applied and throughout the design science process knowledge is interchanged between the knowledge base and the environment (Hevner 2007; Gregor & Hevner 2013). The design science process includes six steps; problem identification and motivation, definition of objectives for a solution, design and development, demonstration, evaluation, and the final step communication (Hevner & Chatterjee) 2010). The research within the process iterates between the design and development, the evaluation of the artefact and its refinement based on feedback (Hevner 2007; Gregor & Hevner 2013). The research presented in this thesis includes parts of such iterations where the risk management process, RiskUse is developed and refined based on feedback but also based on knowledge seeking.

4.1 Methodological approach

Research can according to Robson (2002) have two main types of research design, fixed or flexible. Fixed designs, also called quantitative designs relying on quantitative data, are either descriptive or experimental, and are highly pre-specified and prepared. Fixed designs are often concerned with comparing two or more groups, and a theory is required in order to define what to search for (Robson 2002). Flexible designs, also called qualitative designs, are concerned with studying objects in their natural setting and describe issues of the real world. It allows changes to the research design based on new information during the study process, e.g. change of research questions and data sources, and the design is intended to evolve over time as the researchers gain more knowledge.

A design cannot be both fixed and flexible at the same time, but a design could have flexible phases followed by fixed phases; the other way around is very rare. Flexible designs can include the collection of quantitative data (Robson 2002), for example, allowing free-text answers in and experiment, and vice versa.

The research in this thesis uses four types of research methods; surveys, experiments, case studies and action research and the data collection and analysis method used are questionnaires, observations, interviews, content analysis and statistical analysis. Action research closely relates to case studies according to Runeson et al. (2012). In this

thesis the action researcher part has been active observations. In the performed case studies the active observations have been an integrated part. The results in this thesis are reached through the use of the research methods presented in the sections below. The design science research approach where the research iterates between design and development has influenced the design and development of RiskUse. Throughout the process, knowledge was interchanged both with the medical device organisation and the knowledge base (further presented in Section 4.3). For each paper, research questions (RQ), type of research, research design and data collection method are presented in Figure 5. Further details on the different research studies can be found in respective paper.

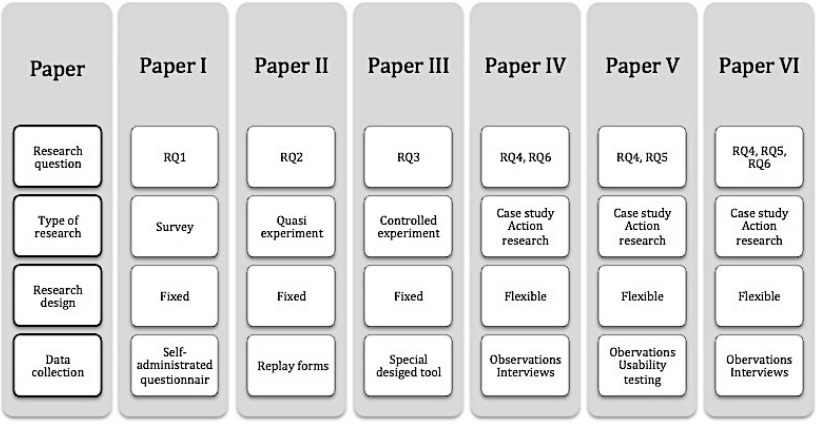

Figure 5. The relationship research papers, research questions, type of research, research design and data collection method.

4.1.1 Survey

The purpose of a survey is "to understand, describe, explain or explore the population" (Wohlin et al. 2000). It is difficult to give a concise definition of survey research, but a survey has often three typical central features according to Robson (2002):

1. Fixed, quantitative design is used.
2. From a relatively large number of subjects a small amount of data in a standardised form is collected.
3. A representative sample of individuals from known populations is selected.

These three central features capture a large part of surveys, but there are surveys where considerable amounts of data are collected from each

individual, but the individual does not represent himself or herself but rather a company or organisation.

Representative samples selected from a well-defined population distinguish survey research (Easterbrook et al. 2008) and the strategy can be used to identify characteristics of a broad population of individuals (Rea & Parker 2005; Easterbrook et al. 2008). A clear research question, focusing on the nature of the target population is a precondition for conducting survey research (Easterbrook et al. 2008).

However, even if surveys often are referred to as a fixed design, Robson (2002) also argues that surveys can be based on either flexible or fixed design depending on the degree of pre-specification. In typical fixed design the data collection is made by self-administrated questionnaires with closed questions and in typical flexible design the data collection is made through interviews with open-end questions. Structured recorded reviews and structured observations are other survey instruments that can be used (Fink 2003).

The research in Paper I, about the state of practice in the medical device domain is based on a survey. The research design is fixed and carried out through a web-based questionnaire sent out to medical device companies from Europe and the US. The fixed design with the use of a web-based questionnaire was chosen because it is an easy way to retrieve information from a large set of people in different countries. It allows anonymity and can provide a large amount of information to a low effort in a short period of time. The design was typically fixed with the closed questions, where it is possible to know that the questions mean the same to the different respondents.

4.1.2 Experiment

Experiments are conducted when the researcher wants control over the situation with systematic manipulation of the behaviour of the studied phenomena (Wohlin et al. 2000), and in software engineering experiments are mostly dependent on human subjects performing some task (Easterbrook et al. 2008). Experiments (Robson 2002) are of fixed design type and the studies are focused with a few variables to handle. Before the main data collection begins, the design details are fully pre-specified.

The experimentation is a research strategy that involves manipulation of one or more independent variables by the researcher and the effects of

the manipulations are measured. The measured effect is then statically analysed to confirm the significance of the effect (Wohlin et al. 2000). Experiments reduce complexity by only allowing a few variables to vary (Easterbrook et al. 2008). Jedlitschka et al. (2008) provide a guideline on how to report software engineering experiments. The guideline unifies and extends other existing guidelines by various authors.

The experiment in Paper II is an experiment in real-life context and is a quasi experiment. Quasi experiments are experiments when units are non-randomly assigned to experimental groups (Höst et al. 2005; Easterbrook et al. 2008). Kampenes et al. (2009) conclude that quasi-experimentation is useful in many settings in software engineering. Quasi experiments according to Basili (1996) tend to involve qualitative analysis components and they can easily be done in real-life context with experts working in large projects. The subjects used in the quasi experiment described in Paper II are subject in three different categories of professional practitioners; software developers, medical device developers and physicians. The subjects were non-randomly selected.

Paper III is based on an experiment where students acted as subjects. The use of students as subjects can be questioned. Höst et al. (2000) conclude in a comparative study of students and professionals in lead-time impact assessment that "there are only minor differences between the conception of students and professionals and there is no significant difference between the correctness of students and professionals". Even if the intention was that the subject in the experiment should be representative of engineers working with this type estimation in real-life experiment, it cannot be concluded with large validity that the students that participated in the experiment presented in Paper III are representative of professional practitioners.

Another factor regarding the participants in controlled experiments is the incentives for participants in the experiment. Höst et al. (2005) argue that the validity of a study is affected by the motivation of the participants and they introduce a way of trying to capture the motivation by looking at the experimental situation where the subjects are participants. In the experiment described in Paper III the intention was to take that into count and motivate the students as subjects, by having a seminar about risks and by designing the experiment to be representative for engineering work. The experiment was part of a project course the students attended at that time.

4.1.3 Case study

According to Yin (2003), ‘‘a case study is an empirical inquiry that investigates a contemporary phenomenon within its real-life context, especially when the boundaries between the phenomenon and context are not clearly evident’’. The study of software engineering often is of a complex nature and cannot be studied in isolation. The complexity arises from the role of human behaviour in software development (Seaman 1999) and software engineering in general, which is a complex social phenomenon. A real-life context and the study of contemporary phenomenon are conditions that are valid for many research studies in software engineering and make a case study approach feasible (Runeson et al. 2012).

The research design of a case study is flexible where the research strategy develops during the data collection and analysis (Robson 2002). In a case study the selection of a case is rather purposive than random (Easterbrook et al. 2008) and the research questions are allowed to evolve during the study. Case study methodology typically involves multiple data collection methods such as observations, interviews, and documentary analysis (Robson 2002) and due to the flexible nature of case studies that data collection will often continue even after the analysis begun. There is a wide range of ways to analyse the data, from qualitative analysis where data from, for example, interviews are coded and categorised to the use of statistical methods (Runeson et al. 2012).

In a case study there are confounding factors, which are not entirely known and cannot be controlled, and these factors might affect the result. Consequently, the researchers do not have the same level of control in a case study as in an experiment. The results of a case study are also more difficult to interpret and generalise than results of experiments (Wohlin et al. 2000).

The three case studies presented in this thesis have been performed in real-life settings, to gain insight in risk management in medical software development, to evaluate the contribution of usability testing to risk management and to evaluate the developed risk management process RiskUse. The objective of the first case study (Paper IV) was to collect and summarise experiences from conducting risk management with in an organisation developing medical devices and the overall objective of the second case study (Paper V) was to investigate how usability testing can contribute to a software risk management in the medical device domain. Finally, in the last case study (Paper VI) the new software risk

management process was evaluated in a medical devices development project, in order to study strengths and weaknesses of the developed risk management process, RiskUse. A mixed data collection approach was used in all the three case studies, using a combination of observations and interviews with the aim to gather different kinds of data (Runeson et al. 2012). The use of more then one data collection method also allows different perspectives and enables the use of triangulation (Robson 2012).

4.1.4 Action research

In action research, there is collaboration between researchers and those who are the focus of the research (Robson 2002) and the purpose is to influence or change some aspects in the studied environment. Action research is closely related to case studies (Runeson et al. 2012). The aim, according to Robson (2002) is to improve practice, the understanding of practitioners, and the situation in which practices take place (Robson 2002). The application of action research emphasises "more on what practitioners do than on what they say they do" (Avison et al. 1999). A large part of software engineering research is using an action research strategy (Easterbrook et al. 2008). In software engineering research it is a common scenario that trying out the idea in a real industrial context develops the originally idea. The use of action research in the information systems (IS) domain has also become more frequent (Davison et al. 2004). According to Sjøberg et al. (2007) action research is regarded as "the most realistic research setting found", because the setting of the study is the same as the setting in which the result will be applied for a given organisation, apart from the presence of the researcher. A researcher coming from outside looking at the studied phenomena with fresh eyes and different angles is beneficial. However, when implementing improvements, commitments within the organisation are needed and this is best achieved by involving people inside the organisation (Runeson et al. 2012). Since each action research project to some extent, is unique, it is difficult to draft general rules about how to carry out such projects. However there are general guidelines presented by, for example, Avison et al. (2001) and Sagor (2011).

The action research process is composed of a four-stage procedure (Robson 2002; Sagor 2011), starting out with the planning stage. The four stages are shown in Figure 6. The process is a cyclical or a spiral

process, based on iterations where the researcher in real situations wants to try out theories with practitioners, gain experience and feedback, modify the theories and try again.

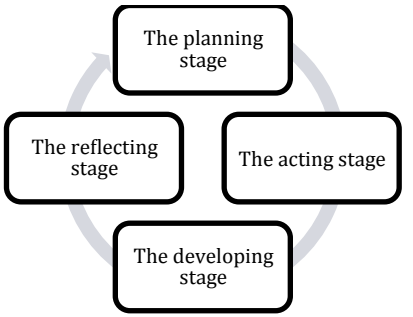

Figure 6. Action research four-stage procedure.

Action research has been used in as part of the case studies in Paper IV, V and VI. It has been active observations, allowing the researchers to influence the outcome of the observed activity. In Paper IV, the aim was to observe how the activities are performed in their context, not to actually perform the activities. However, during the activities, it was natural for the researchers to give input and support to the development organisation. The aim was also to get information about aspects of the activities by asking questions and giving advice on relevant topics. In Paper V and VI the researcher had a more active role when taking part in the usability testing (e.g. as facilitator) and in the risk management process, (e.g. as participant in the risk management team), where the purposed risk management process was evaluated.

4.2 Data collection and analysis

An important part of the research methodology is to choose data collection methods (Robson 2002; Lethbridge et al. 2005; Easterbrook et al. 2008). Data collection can be done with several different methods depending on the type of empirical study. The collected data in empirical studies could be quantitative as numbers or qualitative as words and pictures (Seaman 1999). There are many aspects to consider when choosing data collection methods; one of the most common aspects is

the degree of involvement of the software researcher (Lethbridge et al. 2005). According to Lethbridge et al. (2005) data collection methods can be divided into three degrees. The first degree includes direct methods were the researcher is in direct contact with for example interviewees and collect data in real time. Interviews, focus groups and observations with "think aloud" are examples of first degree methods. Indirect methods such as automatic monitoring of usage of software engineering tools are methods of the second degree and the researcher makes the data collection without interacting with participants. Third degree methods are methods where the researcher independently analyse already available work artefacts, such as failure reports and requirement specifications. First degree methods have been used in Paper IV – Paper VI and second degree used in Paper I – Paper III. First and second degree methods have the advantage that it is easier for researchers to control which data is collected, how it is collected and so on (Runeson et al. 2012). Proper data collection and analysis methods need to be used to get the right and valuable information.

The data collection and analysis methods used in the papers presented in this thesis are presented in the following five sections.

4.2.1 Questionnaire

Questionnaires are a data collection tool with sets of questions in a written format. The questions can be closed or open (Robson 2002). Closed questions aim to test the respondents' preferences and it might give insight to what the respondent believes or feels and they provide the respondent with predefined answers. Open questions let the respondent describe phenomena as they see them, and are useful when issues still are unknown. A problem with open questions it that the responses could be difficult to interpret by the researcher (Robson 2002). To assure valid results, wording and ordering of the questions and the layout of the forms is important to consider in the design of the questionnaires (Lethbridge et al. 2005).

Questionnaires allow collection of large amounts of data in a time and cost effective way (Lethbridge et al. 2005) and if web-based questionnaires are used, it allows data collection from diverse geographical locations as in Paper I where questionnaires were distributed both in the US and in Europe. A self-administrated questionnaire is a questionnaire that the respondent answers on site or is

sent out by mail (Fink 2003). When web-based questionnaires are used, is the time to deliver and get the answered questionnaires in return significantly reduced according to Punter et al. (2003).

When conducting a survey, questionnaires are one of the possible data collection methods (Robson 2002; Easterbrook et al. 2008) and it is suitable for software engineering research (Lethbridge et al. 2005; Kithenham & Pfleeger 2008). Kithenham and Pfleeger (2008) also provide guidelines for designing questionnaires.

Low response rate is a common disadvantage of questionnaires and Lethbridge et al. (2005) reported a 5% response rate for web-based software engineering surveys. The survey in Paper I had a response rate of 16%. The questionnaire in Paper I was a web-based self-administrated (Lethbridge et al. 2005) questionnaire, meaning that the respondents filled in the answers themselves. A mail containing a link to the URL was sent out to potential participants encouraging them to answer the questionnaire. The identified data set from the answered questionnaires was then analysed with descriptive statistics and relationships between variables (Wohlin et al. 2000; Robson 2002).

4.2.2 Observations

The idea with observations "is to capture first hand behaviour and interaction that might not be noticed otherwise" (Seaman 1999). Observations used for data collection are typically used for gathering data about what is going on in a certain situation, what terminology is used and how people behave and interact. The advantage of observations is that they provide a deep understanding of the phenomena under study (Runeson et al. 2012).

Observations of a more participatory type are observations where the researcher participates in the situation under study (Bell 2005; Robson 2002). When using observational methods it is important for the researcher to aware of the risk of not being objective. There is a risk that the researcher overlook aspects but on the other hand the researcher will gain good understanding of the existing procedures in the observed organisation (Robson 2002).

Observers must take measures to ensure that the subjects observed not constantly think about being observed (Seaman 1999). According to Runeson et al. (2012) observations can be divided into four categories depending on the degree of interaction by the researcher and the

awareness of the subjects of being observed. In the action research study presented in Paper IV, the observations had a high degree of interaction by the researchers, and the subjects had a high awareness of being observed. Due to that the researcher is seen only as a researcher and the data was collected during the usability testing with the help of "think aloud", the degree of interaction by the researcher was low and the subjects awareness of being observed was high in Paper V. Finally in Paper VI, the subjects' awareness of being observed was low and the researcher more seen as a "normal participant" during the risk meetings."

The data collection in Paper IV and VI was made through two different sources: interviews and observations. All collected data were treated confidentially in order to protect the participants of the study and to ensure that the participants felt free to speak during data collection. Data collection during the observations (e.g. risk meetings) was conducted through active observations by the researchers. The purpose of the interaction with the researchers in Paper IV was to capture interesting aspects as well as advantages and disadvantages regarding the process. The researchers asked direct questions during the risk meetings, for example, if something was vague regarding the process. In Paper VI, the researcher took part in the risk meeting as risk manager and the observations were used to evaluate the proposed risk management process, RiskUse.

During the risk meetings, the observations were documented on paper. These notes contained both direct observations and the researchers' own reflections. The notes, as well as personal reflections, were in most cases discussed by the researchers directly or shortly after the meetings. The notes were compiled into a list of statements, which were recorded in the case study protocol. Each statement was then coded, grouped, and interpreted (Seaman 1999; Robson 2002; Runeson et al. 2012). The data collection from the usability tests in Paper V was made at the usability test sessions where the observer logged all the actions. All observations were written down during the sessions and then transcribed and interpreted.

4.2.3 Interviews

The purpose behind the use of interviews in empirical studies is often to collect data about phenomena not suitable for quantitative measures (Hove & Anda 2005). It is a commonly used method for collecting

qualitative data (Seaman 1999), and in case studies one of the most frequently used and most important data sources in software engineering (Runeson et al. 2012). According to Lethbridge et al. (2005) interviews are the most straightforward instrument for data collection.

Interviews can be classified into different types depending on how structured they are and it is the situation and the research questions that determine which one to use. The three types of interviews are: fully structured, semi-structured and unstructured (Robson 2002).

- Fully structured – all questions are planned in detail in advance and they are asked in the same order as planned. Often closed questions
- Semi-structured - the questions are planned, but can change in wording and order. Often a mixture of open-ended and closed questions, designed to also elicit unexpected information.
- Unstructured – the interviewer has a general area of interest, but the conversation interviewer and the interviewee are allowed to develop and can be completely informal.

During the interview session interview questions can be asked, according to three different principles, the funnel model, the pyramid model and the timeglass model (Runeson et al. 2012).

- Funnel model – starts with open questions and move towards closed.
- Pyramid model – starts with closed questions and opens up during the interview session.
- Timeglass model – starts with open questions move towards closed questions in the middle of the interview and the questions opens up again in the end of the interview.

When using interviews for qualitative research, it in important that the design of the interview is flexible enough so it allows unforeseen types of information to be collected (Seaman 1999). The advantage of flexible design is that it gives the researcher the possibility to follow up answers, interpret feelings, body language and intonations during the interview. On the other hand, there is the disadvantage, that interviews are rather time consuming (Robson 2002).

A semi-structure interview approach (Robson 2002) was used for all the performed interviews in Papers IV and VI. Interviews were used as data collection method together with observations. According to Lethbridge et al. (2005) interviews are a good method to gain opinions

about a process or a product. The questions were predefined and open-ended, and the interviews were conducted as an open dialog between the researcher and the interviewees in a timeglass model way. The respondents were allowed to talk freely after each question and in some cases follow-up questions were posed. Interview guides (Seaman 1999) were used as a support to the researcher in the interview process. All the interviews were conducted face-to-face and recorded by the same researcher with the intention to make the interviewees feel as comfortable as possible during the interview. Interviewees feeling comfortable are more willing to share their experiences (Hove & Anda 2005). The recordings were later transcribed, coded and analysed according to the guidelines by Seaman (1999), Robson (2002) and Runeson et al. (2012).

4.2.4 Content analysis

Content analysis is a method of data collection in review of written documents (Robson 2002) focusing on gathering information and generating findings. The method is based on existing document; a third degree method according to Lethbridge et al. (2005) and Runeson et al. (2012) and classified as an "unobtrusive measure" meaning that the data collection does not affect the documents (Robson 2002). Content analysis can also include analysis of the content of interviews and observations where the data are collected directly for the purpose of the research (Robson 2002) or as a useful method to be used when the goal of the research is to gather or propose a set of metrics (Lethbridge et al. 2005). Content analysis has been used in all papers presented in this thesis, except in Paper III where statistical analysis (Wohlin et al. 2000) was used. Paper II includes both content analysis and statistical analysis.

According to Fink (2003) content analysis can be based on either inductive or deductive analysis. Deductive analysis has been used in the research in this thesis. The researchers have preselected the themes and categories that were likely to occur before the data were collected. When inductive analysis is used instead, the researchers look for dominant themes and categories in the collected data.

4.2.5 Statistical analysis

After collecting experimental data, conclusions shall be drawn from this data. The quantitative interpretation of the data may be carried out in three steps (Wohlin et al. 2000; Rosenberg 2005)

1. Use descriptive statistics, for example, measures central tendency, dispersion and dependency to describe and present the collected data. This can be graphically presented by, for example, scatter plots and box plots (used in Paper II).
2. Reduce the data set by excluding abnormal and false data points.
3. Data is analysed by hypothesis testing. The tests can be classified as parametric or non-parametric tests. Parametric tests are based on a specific distribution and in most cases; it is assumed that some of the parameters are normally distributed. Non-parametric tests are more general.

In the controlled experiments presented in Paper II and Paper III statistical analyses are made and both parametric and non-parametric tests were used.

4.3 From theory to practice

To successfully transfer knowledge and technology from research to practice, close cooperation and collaboration between researchers and practitioners is needed (Seaman 1999). Technology is in the context of software engineering according to Pfleeger (1999) "any method, technique, tool, procedure or paradigm used in software development or maintenance". However, it is generally challenging to transfer research results into industrial practice (Zhang & Xie 2013) and the process of transfer is complex, involving many roles and phases (Buxton & Malcolm 1991).

The risk management process, RiskUse is developed in close contact with the organisation and the purpose of the risk management process is to provide practitioners, mainly risk managers with a software risk management process. The process has a defined user perspective, and including hands-on recommendations on how to use the process. Usability testing was chosen as an integrated part of the risk management process because usability testing is considered as one of the most powerful ways (Daniels et al. 2007) and perhaps the most powerful one (Kushniruk 2002) to evaluate usability. To adjust to practice, the terminology used in RiskUse is adapted to the terminology used by regulatory bodies and requirement within the medical device domain.

The risk management process is developed based on contributions from each of the five first papers in this thesis. More specifically based on empirical knowledge about the state of practice gained from the survey

presented in Paper I, knowledge regarding human factors from different angles from the experiments in Paper II and III and the case study results on the risk management process in Paper IV and usability testing in Paper V, complemented within depth studies of risk management standards and guidelines within the medical device domain. The empirical data is collected according to Table 5.

In order to go from theory to practice, the research covered in this thesis started with the generation of empirical knowledge and theory. The technology transfer process used for this research is inspired by the process described by Gorschek et al. (2006), but adapted to the prevailing circumstances. The used technology transfer process is presented in Figure 7.

Table 5. Collected empirical data

Paper	**Type of research**	**Source**
Paper I	Survey	Software developers from different companies in the US and Europe.
Paper II	Experiment	Physicians from one clinic at a hospital. Medical software developers and software developers from different companies in Sweden.
Paper III	Experiment	Students taking a project course
Paper IV	Case study	Software developers and risk managing team from a medical device development organisation. A medical device development project.
Paper V	Case study	Usability testing of the same project as in Paper IV but at a later stage in the project.
Paper IV	Case study	The same development organisation as in Paper IV and Paper V but another risk management team and mostly other software developers. Another medical device development project than in Paper IV and Paper V.

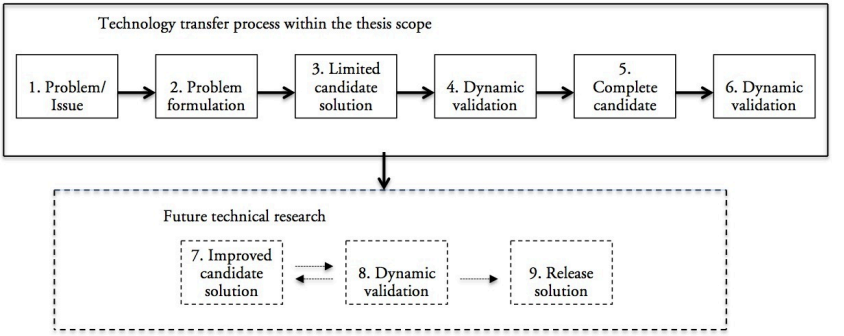

Figure 7. The technology transfer process.

The different steps are:

1. **Problem/issue**. The aim with the first step is to observe the real world before the research questions are formulated and identify potential areas of improvement. The aim with the survey in Paper I was to get empirical knowledge about the state of practice and to understand goals and concerns of companies within the medical device domain. The survey results showed that it is unclear in practice exactly how software risk management should be performed and that the used risk management process is not user-centered. In 2000 FDA highlighted the need of incorporating human factors in the risk management process (FDA 2000). However, guidelines and standards are vague regarding the concrete implementation and give practice a high degree of freedom in instantiating the process. The experiments in Paper II and III were used to look deeper into human factors from different angles.
2. **Problem formulation.** In the second step contact was established between the researchers and an organisation, which develops and maintains medical devices. The development organisation has extensive experience in developing and maintaining medical devices, but not with devices including software. The organisation had previous collaboration with researchers, but contacted the researchers specifically to ask for help with their software risk management process. The problem formulation was made in close cooperation and it was decided to design a limited candidate solution covering the three first steps of the risk management process, risk identification, risk analysis and

risk planning. The state-of-the-art in the risk management field was also taken into account.

3. **Limited candidate solution.** The limited candidate solution was developed in cooperation between the researcher and the organisation, presented in Paper IV. The state-of-art were further studied and accounted for, with special focus on laws, regulations and guidelines with in the field.
4. **Dynamic validation.** The limited solution was validated in the two case studies presented in Paper IV and V. All the empirical data was coded and grouped (Seaman 1999), identifying patterns between these codes and incorporating these patterns in the evolving risk management process.
5. **Complete candidate solution.** The risk management process presented in Paper VI was developed based on the results in Paper IV and V and in depth studies of risk management standards and guidelines within the medical device domain.
6. **Dynamic validation.** Finally, the risk management process RiskUse has been evaluated within the organisation and the results are presented in Paper VI.

However the risk management process RiskUse will be further improved (see Section 6) and further evaluation is needed before releasing the solution. RiskUse may need to be tailored for different organisations since organisations can have different definitions, vocabulary, relative to product and development (Gorschek et al. 2006).

4.4 Validity

It is important to address the validity in research. The results should always be evaluated and questioned and the researcher is obliged to describe the used research methodology to a sufficient level of detail and also how the researcher has reached the conclusions. Validity threats must be addressed already during the design of the research study, so the validity threats can be reduced as much as possible, waiting to the analysis phase is to late (Runeson et al. 2012).

The classification of validity threats in literature varies, but in general there are four types described for empirical research (Wohlin et al. 2000; Robson 2002; Yin 2003; Runeson et al. 2012). In Paper II and III the classification of validity described by Wohlin et al. (2000) is used and in

the rest of the papers the classification of Yin (2003). The classifications are very similar; three of four types have the same meaning and are also called the same. The fourth type denoted, conclusion validity by Wohlin et al. (2000) and reliability by Yin (2003) are both concerned with the ability to draw accurate conclusions and to replicate the study.

Validity is classified according to Yin (2003) in construct validity, internal validity, external validity and reliability:

> **Construct validity** is affected by how well the collected operational measures represent the concept studied by the researcher. To increase the construct validity, multiple sources of evidence can be used and also establish a chain of evidence. In case studies construct validity can be increased by letting people with key information, review the preliminary case study report. To improve construct validity in interview studies, the researchers has to ensure that the questions are interpreted in the same way by both researchers and interviewees (Runeson et al. 2012).
>
> **Internal validity** is affected by factors that are outside the control of the researcher but affect the measures and reflects whether a particular treatment really has caused a certain outcome. In experiments it is critical to address internal validity so the researcher does not draw incorrect conclusions about the effect of the treatment on the outcome without knowing if there in fact is a third factor affecting the outcome (Wohlin et al. 2000). In case studies it is applicable to consider internal validity for explanatory and causal studies and not for descriptive and explanatory studies, which are not making causal claims. However, there is a threat to internal validity in case studies due to the changes in industrial environments over time (Yin 2003). Internal validity threats might be lowered by using pattern-matching or explanation-building (Yin 2003).
>
> **External validity** concerns the problem of how general findings are with respect to the subject population and beyond the immediate study. Results obtained in a specific context or with a specific group of participants may not be representative for or transferable to other environments or settings. External validity threats can be lowered by using theory in single-case studies and

replication logic in multiple-case studies. In case studies focusing on understanding and explaining phenomenon within its real-life context, the aim of the study is not to generalise (Runeson et al. 2012).

Reliability concerns how reliable a study is and depends on how well the described procedures are followed and documented, so that the study can be repeated in the same way over again. "The goal of reliability is to reduce errors and biases in the study" (Yin 2003). Threats to reliability might be unclear coding of collected data or unclear questions in interviews in questionnaires (Runeson et al. 2012). The use of use case protocol is one way to lower threats to reliability (Yin 2003; Runeson et al. 2012).

There are several possible threats to validity in studies and a main concern in the work with the contributions presented in this thesis has been to identify and reduce all the validity threats as much as possible.

Threats to construct validity can be participants' bias. There is a risk that the participants misunderstand and interpret terms, concepts or questions differently. In the studies described in this thesis, the risk of misunderstanding have been reduced by only allowing responses on such a level of detail that subjectivity from the participants and need for interpretation are minimised. During the interviews and risk meetings presented in Paper IV and Paper VI, the concepts were explained during the interviews and the definitions of terms and concepts in the risk management process were adapted to the standards in the medical device domain with the aim to avoid misunderstanding. In Paper II the instructions and risk scales were written and defined to be as clear and unambiguous as possible. Another participant bias can be that the participant gives a too positive picture of the situation. In order to lower this risk in the survey study presented in Paper I and the interview studies presented in Paper IV and Paper VI, the participants were allowed to be anonymous. However here is a risk that the participants exaggerate the negative sides in frustration over different situations and this must be taken in consideration when interpreting the results, which has also been done in the studies. To further reduce the risks of participants' bias in the case studies (Paper IV to VI), the interviews were audio recorded and triangulation (Robson 2002; Runeson et al. 2012) was applied The interview transcripts were reviewed by other researcher,

multiple sources for the data was used and member checking of the material by the medical device organisation. Reactivity (Robson 2002) is a threat to validity in flexible design where the presence of a researcher might affect the participants and thereby influence and limit the outcome. The participants might act or respond after assumed expectations or hide facts. To reduce this threat the participants were guaranteed anonymity regarding all collected data, e.g. the survey interview and observation material was not showed or used by researchers outside the studies or by other organisations.

The threats to internal validity have been minimised in the contributions in this thesis according to Paper I, Paper IV, Paper V and Paper VI, by only analysing relations between factors and not drawing any conclusions of causal direction in the studies. In the experiment presented in Paper II there is a threat to internal validity since the participant took part in the experiment under different conditions, out of the researchers control. The threat to internal validity was lowered in the experiment presented in Paper III by letting the participants carry out the assignment at the same time. The participants analysing the given scenarios in the same order, however is a threat.

External validity primarily relates to how general the results of the study are and to what extent the findings are applicable and of interest outside the investigated cases. The evaluation of the risk management process presented in Paper VI is done with a limited set of participants from two single projects. This means that the results cannot automatically be generalised to other organisations. However, probably some of the issues in the risk management process, to some extent could be general for organisations developing software in the medical device domain. The results can help to understand phenomenon and the context and also to improve and further evaluate the risk management process. To support generalisation and allow external comparison, the context and characteristics of the projects have been presented as extensively as possible under given confidentiality constraints. To be able to show generalised results the new risk management process, RiskUse, need to be independently used in a larger amount of projects.

A threat to external validity can be that the participants are not representative of the target population. In the studies in all the papers in this thesis, professionals from the software and medical sectors have been involved but in the study in Paper III students have been used as

participants in the experiment. This is a threat and this experiment should be repeated again with participants working as software engineers.

Concerning replication, controlled experiments in a laboratory can be performed as direct replications and external replications can be done where the replications is conducted in, for example, different environments and then compared to the original study. In studies of flexible nature, such as case studies, the design is more difficult to recreate since the context of the study always changes. The procedures and changes in the studies presented in this thesis have been carefully monitored and documented over time to get reliable studies that can be repeated again by other researcher after minor changes and other medical devices under study. To further increase reliability several researchers were included in the studies. The reliability was also addressed by applying triangulation to the data collection and analysis procedures and final versions of Paper VI, V and VI was reviewed and approved by participants from the development organisation. Feedback was also given from the development organisation during the research process.

Regarding conclusion validity in experiments (Wohlin et al. 2000), it relates to the possibility to draw correct conclusions. A typical threat can be the use of wrong statistical tests. With this in mind the statistical tests were chosen for the studies with great care and for example in the controlled experiment presented in Paper II the analysis was also done with non-parametric test.

The ethical principles presented by Vinson and Singer (2005) has also been regarded during the research work. That is the subjects have been informed about all relevant facts concerning the studies and have decided themselves about their participation. The researchers have also undertaken an effort to maintain confidentiality of data and sensitive information and guaranteed the subjects' anonymity.

Further details regarding the threats to validity, are discussed separately in each paper, together with applied strategies to reduce the threats.

5 Research contribution

This section presents the main results and contributions of this thesis. The discussed results and contributions are based on the conclusions from the included papers and are summarised per respective paper and addressed research question. The relationship between the different research questions and research papers are presented in Table 4 in Section 3.1.

5.1 Paper I - State of practices

RQ1: What is the state of practice of medical devices with respect to the software development and software quality assurance including risk

The aim of Paper I was to understand goals and concerns of companies within the medical device domain. Since a general lack of empirical knowledge was identified, combined with the increasing use of software in the medical device domain, it was decided to conduct an international survey. More than 100 companies from Europe and the US participated in the survey, ranging in size from small organisations with less than 20 employees, to global players with several thousand developers. The products of the companies range from devices implanted during surgery to diagnosis systems to non-invasive devices and information systems. Most of these products are classified as products of class II according to the FDA standards and class IIa/IIb according to European standards. The survey was conducted via questionnaire, which was available on a website. All participants were invited by email.

The results show that it is evident that software is an important part in many medical devices, the majority of the companies rate software as either a very important or important component of their products and that the majority of the software developers have a background other than computer science. The survey supports the assumption that software engineering methods, techniques, tools, and standards could be better integrated into the used development processes and existing standards for medical devices.

Most issues regarding software quality stem from activities involved in planning the software development, the functionality it should

accomplish, and how it will be accomplished. Activities related to the requirements phase are perceived as the major source of issues regarding software quality. Risk management activities seem to be the activities that are most frequently supported by tools. However a detailed analysis of the data shows that almost all of the companies state that they do not use commercial tools specialised to support this activity but common applications such as text editors or spreadsheets. Regarding the use of methods for risk analysis, FMEA is the most frequently applied method. Even though many developers understand that risk analysis should be performed on software and software components, it remains unclear in practice exactly how this should be performed.

More detailed results from the survey is presented in a technical report (Related publication VII) where it is concluded that there are people who are working with products that are classified as safety critical that do not perceive the products as safety critical and a shorter version in Related publication VIII).

The findings from the survey have been used to better understand issues and challenges within the medical device domain, especially regarding the risk management process and to help identifying focuses for improvements in this area. The results can also be support other researchers in their work to identify improvement areas and for goal-oriented actions to further integrate software engineering methods, techniques, tools, and standards into the medical device domain.

5.2 Paper II – Risk identification

RQ2: What differences can be identified between the users of a system and developers of a system with respect to risk identification?

The research in Paper II is based on the basic assumption that if multiple roles, and thereby different experiences, are involved in the identification activity, the resulting list of identified risks will be more complete than if only one role was included. An experiment was conducted where physicians, developers and software developers for medical devices were asked to identify risks in a given scenario, describing the procurement of a patient monitoring system. A quasi-experimental design was used, not involving random allocation of participants from different groups. Since the subjects involved in this experiment are from different categories of

professional practitioners, i.e., physicians, developers and medical device developers, is it not possible to randomise over a sample of people. All the participants were presented to the same risk scenario and a reply form was used for data collection.

The results show that there is a difference regarding the view of risks between physicians, developers and medical device developers. Looking at the number of identified risks, developers identified a larger amount of risks per person than physicians. The three risks, that the physicians gave the highest risk value were not specific medical risks. The other two groups did not identify these risks, but they easily could have. A difference in between the groups with respect to which risks they see as important as also identified.

Based on the experiment it is concluded that multiple roles, and thereby different experiences, will affect the risk identification process. Involving multiple roles, for example users and developers in the risk identification process, will result in a more complete set of identified risks than if only one role is included in the process. It can thereby be concluded that it is necessary, at least for this kind of systems, to include the users in the risk identification process in order to get a more complete risk identification. It is not sufficient to only include the developing organisation in identification of risks. Involving different roles in risk identification may probably be advantageous in several types of systems, but the advantages must, of course, be compared to the extra cost.

User involvement in the risk identification process has been a crucial part in the case studies involving a patient monitor system (Paper IV and V) and is one of the fundamentals in the developed risk management process, RiskUse (Paper VI).

5.3 Paper III – Conception of risk

RQ2: What differences can be identified between the users of a system and developers of a system with respect to risk identification?

RQ3: How can different people's risk tendency be defined in an adequate way with respect to conception of risk, in order to support the risk management process?

In Paper III, different people's opinions about the importance of identified risks are investigated in a controlled experiment through the use of utility functions. Utility functions (e.g. Wakker & Deneffe 1996) describe how different people value a property. For example, a utility function could describe how people value the expected life-duration after different alternative medical treatments. Based on the shape of the utility function it is possible to discuss whether different individuals act as risk-averse, i.e. they tend to avoid risks and choose a safer treatment with lower gain in life-duration, or risk-seeking, i.e. seeking a possible high gain in life-duration instead of a more certain lower one Engineering students participated as subjects in the experiment and the experiment was conducted as part of a software engineering project course. In every project groups the students were divided into the following roles: project leaders, technical responsibility, developers, and testers. The students were presented with two scenarios and the utility function of every student was elicited with the Trade-off method (TO-method). According to the TO-method (Wakker & Deneffe 1996) the subject is iteratively asked to compare different scenarios called "lotteries".

Based on the results from the experiment, differences were identified with respect to the perceived importance of the risks, although the experiment could not explain the differences based on undertaken role in a development course. So it is not possible to state that any role is more risk seeking than any other role.

There are some threats to validity in this study, for example according to the use of students as subjects. To lower such threats studies can be done, involving people with more experience in general and with more experience from their project-roles.

However it is possible to generalise and conclude that different people are more or less risk seeking and it also applies to participants in the software engineering projects. This is important to know in a risk management process methods for assessing the level of risk seeking are available, but in most cases it is probably enough to be aware of the differences.

As also identified in the experiment in Paper II there are differences between different people with respect to the perceived importance of a risk. Based on that it can be concluded that in order to get a more accurate risk assessment it is necessary to involve multiple persons in the risk assessment process. It is also shown that people are more or less risk

seeking and by having a risk management group with multiple participants, preferable with different roles, the group will probably consist of both risk seeking and risk averse participants. If the risk manager is aware of the differences and tries to balance the group setting it would probably lead to a more accurate risk assessment.

A risk management group consisting of multiple participants (e.g. intended users, development organisation and researchers) was used to performed the risk assessment in the project studied in Paper IV and in the developed risk management process, RiskUse (Paper VI) it is stated that the risk assessment process shall be performed by a risk management group consisting of multiple roles including users.

5.4 Paper IV – Risk analysis and risk planning

RQ4: How can users be integrated in the risk management process?

RQ6: How can a software risk management process including user perspective be designed to be appropriate for a medical device development organisation?

The main objective of Paper IV was to collect and summarise experiences from conducting risk management with an organisation developing medical devices with specific focus on the first three steps of the risk management process, i.e. risk identification, risk analysis, and risk planning. Earlier versions of the study were presented in Related publication X and XI. The motivation for the study was to get experiences from having a user perspective in these three risk management steps, with the long-term objective to design an improved version of the risk management process. The main objectives were defined based on the general interests of the researchers, and the interests of the development organisation. This was defined in informal meetings between the researchers separately and between the researchers together with the development organisation. The case study was conducted at a department at a large hospital in Sweden, which develops and maintains medical devices. The development organisation has extensive experience in developing and maintaining medical devices, but not with devices including software.

The research was conducted as action research, with the aim of analysing and giving input to the organisation's introduction of a software risk management process. The data collection was made through two different sources: interviews and observations. The defined and used process focuses on user risks, based on scenarios describing the expected use of the medical device in its target environment, in this case with a patient monitor system as the medical device. During the use of the process, different stakeholders were involved, medical physicians with competence on the monitored medical processes were involved, together with engineers with competence on the software and hardware, and personnel with competence on the required procedures in the organisation.

In the risk management process, a scenario-based identification method was used and the risks were identified through brainstorming, focusing on user interaction and user related risks. Some technical risks were also identified using the scenarios. Since technical risks are of a more general nature and not scenario-specific, there is a need for a separate risk identification regarding these risks. When using this kind of scenarios there is also a need for risk identification of external factors, for example, process and project risks. When the scenarios were discussed step by step, it could be noted that the user representatives, are the dominant part, since they possess domain knowledge regarding the target environment and medical issues. The developers had a more peripheral role and were consulted regarding technical aspects of the system. A tendency to discuss action proposals instead of risks during risk planning was also identified. A possible solution to the dominance factor and discussion focus could be to have very strict control of the meetings, and with the ambition to get the opinion from all the participants, for example specifically address each participant. This approach was successfully used at the risk meetings in the case study in Paper VI. The way of mixing action proposals that are implemented with proposals that are not introduces unnecessary confusion. To avoid this it is highly recommended that the risk analysis should be done prior to implementation.

From the results, it can be concluded that the system boundaries must be set carefully and not without considering dependencies between components. Before defining the system boundaries, it should be clear how components are coupled and components with strong coupling should be analysed together. The documented risk descriptions have an

impact on the risk planning process, because the descriptions influence the understanding of the risk context. In the studied process, risk descriptions typically only contain a very short summary of the nature of the risk. To lower the risk of misunderstanding and misinterpretation later in the process, the risk must be described in detail and placed in its context.

The order of estimation influenced the outcome of the risk analysis, thus the prescribed order of estimation, e.g. severity, probability, and detectability, should be strictly followed. The concept of detectability was not well understood and the provided scale did not give as much help, as was the case with probability and severity. After completing the risk analysis the development organisation decided, based on the encountered problems, to remove detectability from the process. Although this simplifies the process, it removes potentially important information about risks. There is a need for further research on how to define and estimate detectability of identified risks, see Section 6.

It can also be concluded that the used risk process is considered effective and easy for new personnel to adapt to according to the health personnel working with risk management. The main challenge is however to find the time and right competences for the risk analysis team. Even if it is difficult and time-consuming to produce relevant user scenarios the scenarios make the software easier to understand, which in turn improves the understanding of potential risks. With these difficulties in mind a slightly different scenario design process is used in the new risk management process, RiskUse, presented in Paper VI.

From this case study it can be concluded that the used risk management process has the potential to be used in a medical device development organisation or similar organisations. Criticism could be pronounced that, the focus might be too high on the user interface. However, since it is well known that many risks are related to the usage of a system and the user interface, e.g. Dhillon (2008) reports that 50 % of technical medical equipment-related problems are caused by operator errors, it is important that the user interface stays in focus.

In the following work with designing the new more complete risk management process, RiskUse, covering all the steps in the process, presented and evaluated in Paper VI, great emphasis has been placed on the results and findings in Paper IV.

5.5 Paper V – Usability testing in the risk management process

RQ4: How can users be integrated in the risk management process?

RQ5: How can usability evaluation methods, especially usability testing contribute to the risk management process?

The overall objective of the research presented in Paper V is to investigate how usability testing can contribute to the software risk management process in the medical device domain. Experience has been collected from both the risk management process presented in Paper IV and the usability testing of the patient monitor system is in focus for the risk management. Since risk management as well as usability are important areas in the development process of medical devices and other safety critical systems, there is a need for research to investigate how these two areas can interact in a beneficial way and to involve user in different ways in the development process.

The usability test method used was the "active intervention" (Dumas et al. 1999). However, the test person was also encouraged to think out loud (Nielsen 1992; Sharp et al. 2007) while using the system and verbalise her thoughts.

The results from the usability test show that there were two functionalities generating most of the usability problems, the commenting function and the alarm function. There were also two dominating types of usability problems found, a) problems occurring when the user interface is unclear, it does not match the test participant's mental model or her previous experience, and b) problems when the users do not see the existing entity or fails to realise that they are supposed to interact with it. The users and developers perceive things differently. Things that are obvious for the developers are not even noticed by the users, and the users see and interact with the medical device in their context and on the basis of their domain knowledge. In this case, user representatives have been part of the development process and the risk management process but there have not been representatives from the whole user spectrum. Regarding risk it was found that approximately half of the usability problems found during the usability

tests were not identified as risks in the risk management process. Several of these usability problems imply risk and should be handled in the risk management process, especially the four usability problems that were found by all the users.

From the case study it can be concluded that usability tests can indicate risks that are not identified in the risk management process and give a possibility to verify if risks with high risk value actually cause the presumed problems. It is also possible to capture "problem functionality" e.g. for functionality that is new or unknown to the user. Usability testing also catches problems that are good risk candidates, where the functionality is unclear to the users and where the developers and the users have different mental models. Timing is important when it comes to usability testing connected to the risk management process. The time must be right, so no changes are made only based on the risks, before the usability test is performed. The usability tests can, for example, verify that a risk with a high risk-value actually is a problem for the users before any changes are made. Risk values are assumptions so if they can be identified in additional ways before any action is taken, effort and time can be saved by the development organisation, due to the avoidance of unnecessary changes. Then it can be concluded that usability tests can give valuable input to the risk management process if integrated as a natural part in the development process and the development organisation can save time and effort and also receive more faultless and safer products.

Usability testing has ben integrated as a part in the new risk management process, RiskUse, and further evaluated in the research in Paper VI.

5.6 Paper VI – Evaluation of the risk management process RiskUse

RQ4: How can users be integrated in the risk management process?

RQ5: How can usability evaluation methods, especially usability testing contribute to the risk management process?

RQ6: How can a software risk management process including user perspective be designed to be appropriate for a medical device development organisation?

Paper VI introduces the first version of RiskUse, a medical devices software risk management process with an emphasised user perspective. The process was used and evaluated in a case organisation developing and maintaining a medical devise. The risk management process, RiskUse was developed over time in close contact with the organisation. The purpose of the risk management process is to provide practitioners, mainly risk managers, with a software risk management process that has a well defined user perspective, is easy to apply, and including hands-on recommendations on how to use the process. The aim is also to present a risk management process that allows the development organisation to perform adequate risk management activities that can ensure that the developed software is safe. The main goal is to integrate users and user perspective in the software risk management process and to introduce usability testing, as an integrated part in the risk management process contributing to the goal of integrating users.

RiskUse is developed based on contributions from each of the five first papers in this thesis. More specifically, RiskUse is based on empirical knowledge about the state of practice gained from the survey presented in Paper I, knowledge regarding human factors from different angels from the experiments in Paper II and III and the case study results on the risk management process in Paper IV and usability testing in Paper V, complemented with in depth studies of risk management standards, regulatory requirements and guidelines within in the medical device domain.

The evaluation was carried out using an action research approach according to the observations of risk meetings, supplemented with interviews and observations of usability testing. The goal of the case study was not only to evaluate the risk management process but also to make improvement proposals to the development organisation, based on the results from applying RiskUse in a project and to use the result to further improve RiskUse.

In conclusion, RiskUse is found to support the practitioners in their work with risks and risk management. It can also be concluded that the process has the potential to be used in a medical device organisation and bring value to the organisation. The risk management process is also found to be easy to understand and apply, according to the practitioners participating in the case study. Limitations identified through the case study (presented in Paper IV) were considered and possible solutions were implemented and evaluated. According to the results from the case

study in Paper VI it can be concluded that the solutions are worth maintaining in RiskUse and further evaluate. The findings from Paper V show that usability testing can contribute to the software risk management process was further strengthened in this case study.

Concerning further work, it includes addressing the identified improvement proposals as well as further evaluation of the risk management process. The two last phases, the monitoring phase and the completion phase and need to be evaluated and require an evaluation over time and in addition the whole risk management method. To fully understand and evaluate the possibilities and potential benefits of RiskUse, the process has to be used over time and in a complete project from the start to the end. Broad generalisations of the results can therefore not be made since this is the first evaluation of the risk management process, only involving one organisation. However the case study shows that the risk management process is applicable and the positive results provides a strong argument to continue the evaluation and to promote the risk management process, RiskUse.

5.7 Research questions synthesis

RQ1: What is the state of practice of medical devices with respect to the software development and software quality assurance including risk management?

An international survey was made to gain empirical knowledge on the state of practice of the medical devices software development and software quality assurance processes (Paper I). The results show that it is evident that software is an important part in many medical devices and that quality assurance methods could be better integrated into the development processes. The companies perceive activities related to the requirement phase as the major source of issues regarding software quality. Though risk management activities seem to be the activities that are most frequently supported by tools the results show that common applications such as text editors or spreadsheets are used instead of commercial tools. Developers understand the concept of software risk management but how it should be performed in practice is unclear.

RQ2: What differences can be identified between the users of a system and developers of a system with respect to risk identification?
Two different experiments were launched, looking at differences with respect to risk identification. An experiment was conducted where physicians, developers and medical device developers were asked to identify risks (Paper II). According the number of identified risks, developers identified a larger amount of risk per person than physicians. It was also shown that the physicians did not identify any risks that are specific medical risks. The other two groups could have identified these risks as well. Another difference identified between the groups where with respect to which risks they saw as important. In the experiment described in Paper III the results show a difference with respect to the perceived importance of the risks but it is not possible to state that any role is more risk seeking than any other role. To conclude from both experiments, there is a difference with respect to the perceived importance of the risks and Paper II indicates that different experiences, will affect the risk identification process.

RQ3: How can different people's risk tendency be defined in an adequate way with respect to conception of risk, in order to support the risk management process?
Utility functions were used in a controlled experiment to investigate different people's opinions about the importance of identified risks (Paper III). With the help of utility functions it is possible to discuss whether different individuals act as risk-averse or risk-seeking, and the experiment showed that there is a difference with respect to the perceived importance of the risks. It is possible to generalise this and conclude that different people in the software engineering process are more or less risk seeking and it can be concluded that a risk management process methods for assessing the level of risk tendency are available, but in most cases it is enough to be aware of the differences.

Based on the differences between different people with respect to the perceived importance of a risk it can be concluded that it is necessary to involve multiple persons in the risk assessment process in order to get a more accurate risk assessment. It is also shown that people are more or less risk seeking and by having a risk management group with multiple participants, preferable with different roles, the group will probably consist of both risk seeking and risk averse participants. If the risk

manager is aware of the differences and tries to balance the group setting it would lead to a more accurate risk assessment.

RQ4: How can users be integrated in the risk management process?
To comply with regulatory requirements, get an improved risk management process and incorporating usability engineering into the risk management process the user needs to be a part of the process. The users and their perspective have been incorporated in the risk management process in different ways. In this thesis users are integrated by the use of use cases as input in the risk identification process and users as participants at risk meetings (Paper IV and VI), and by performing usability testing as part of the risk management process.

The use cases (named user scenarios in Paper IV) are perceived as easy to work with and the use of them makes the risk meeting participants feel safe and secure in the discussions. The discussions are also focusing on the right things, saving effort and time.

Users attending the risk meeting also bring the user perspective into the process since users contribute to the discussions with their domain knowledge. Paper IV shows that the user representatives dominated the discussions whereas the developer representatives held a lower profile. To address this dominance, every one of the participants can explicitly be addressed, which would rule out the dominance during the discussions. This was done in the case study presented in Paper VI. However the discussion may be affected by participants' personality.

Usability testing focuses on the end users and that will bring in a new category of users into the process and new perspective on the use of the medical device. Usability tests can indicate risks that are not identified in the risk management process and give a possibility to verify if risks with high risk value actually cause the presumed problems.

RQ5: How can usability evaluation methods, especially usability testing contribute to the risk management process?
The results from the case studies in both paper V and VI show that usability tests can give valuable input to the risk management process. The usability tests indicate risks that are not identified in the risk management processes and they also catch problems that are good risk candidates, where the functionality is unclear to the users and where the developers and the users have different mental models.

The usability tests can, for example, verify that a risk with a high risk-value actually is a problem for the users before any changes are made. Risk values are assumptions, so if they can be identified in additional ways before any action is taken, effort and time can be saved by the development organisation, due to the avoidance of unnecessary changes

RQ6: How can a software risk management process including user perspective be designed to be appropriate for a medical device development organisation?
The main goal with the risk management process RiskUse is to provide practitioners, mainly risk managers, with a software risk management process that has a well defined user perspective, is easy to apply and including hands-on recommendations on how to use the process. The goal is to integrate users and user perspectives in the software risk management process and to introduce usability testing, as an integrated part in the risk management process contributing to the goal of integrating users. Three case studies (Papers IV, V and VI) examined the risk management process and how it can be tailored to incorporate users and user perspectives. The three first steps, risk identification, risk analysis, and risk planning including use cases and user participation at risk meetings were studied in Paper IV. It was concluded that the used risk process was considered effective and easy for new personnel to adapt to. The results in Paper V and VI show that usability testing contributes in a positive way to the risk management process. RiskUse was evaluated in a case study (Paper VI) and in conclusion, RiskUse was found to support the practitioners within the medical device domain in their work with risks and risk management including users and user perspective. RiskUse needs to be further evaluated and requires an evaluation over time to further identify possible improvement, fully understand and evaluate RiskUse.

5.8 Conclusion and main contributions

The overall contribution of this thesis is to bring a user perspective into the medical device software risk management process. In order to help practitioners, mainly risk managers, with a software risk management process, introducing user perspectives into the process, RiskUse has been presented. The concept of user perspectives has been brought into the process by the use of predefined use cases in the risk assessment phase,

the users attending the risk meetings, and the use of usability testing as part of the process.

RiskUse is developed based on empirical insights from the state of the practice regarding medical device software development and on human factors from different angels. The survey on state of practice was used to understand issues and challenges within the medical device domain, especially regarding the risk management process. When looking at human factors it was concluded that multiple roles and thereby different experiences, would affect the risk identification process. By involving multiple roles, for example users and developers in the risk identification process, it was shown that it would result in a more complete set of identified risks than if only one role is included in the process. It was also shown that people are more or less risk seeking and by having a risk management group with multiple participants, preferable with different roles, the group will probably consist of both risk seeking and risk adverse participants.

The concept and gradual evolution of RiskUse grew out of the collaboration between the development organisation and the researcher, with the aim at addressing the challenges identified in cooperation with the development organisation and challenges found in research. The evaluation of the first version of RiskUse shows potential but gives also information about further improvements and how the process can become more comprehensive. RiskUse is found to be of value for the practitioners in their work with risks and risk management. The process has also the potential to be used in a medical device organisation and bring value to the organisation. More over the risk management process can help to support traceability.

6 Further research

Further research based on this thesis should focus on further improvements of RiskUse. During the evaluation, different areas of improvements were identified, e.g. regarding working procedures and the risk meeting documentation. The two last phases, the monitoring phase and the completion phase should further be evaluated, which require an evaluation over time. In addition, RiskUse, the whole risk management process needs to be evaluated from start to end in real-life projects. It would also be beneficial to further evaluate the risk management process according to the use in an iterative process model and to try to adapt RiskUse to agile practices and evaluate in an industrial setting. RiskUse in focusing user interaction and user related risks and need to be supplemented with a formal way of handling the technical risks and risk regarding external factors, as for example, process, project and environmental risks.

More research is needed on the concept of detectability. Detectability is not a part of the current version of RiskUse, and although this simplifies the process, it removes potentially important information about risks. During the case study presented in Paper IV, detectability was partly used but removed due to the challenges regarding the concept. The participants at the risk meetings thought it was difficult, even impossible to assign an appropriate value to detectability. The scale was considered imprecise and did not assist the participants in the estimation effort and another problem was that the concept was not so well understood.

Further work should also focus on the role of the user in the risk management process and investigate if participants' risk tendency affects the way they regard the functionality up for risk assessment and if it affects identified risks and the assessment of these risks. In the area of usability evaluation methods, other methods might be investigated as a complement to usability testing. Interesting for further research would be to tailor the whole usability process and the risk management process together so they will be harmonised and can benefit from each other in an optimum manner. Another area for further research would be to involve the psychological side, combining multiple disciplines. Investigating factors causing human errors such as stress, change and

interrupted work and investigate if and how these factors can be considered even more specific in the risk management process.

Another possible continued work is to study the generalizability of RiskUse. Investigate how to use the process in other medical device development organisation and also if the process can be tailed and suitable for other domains developing software.

REFERENCES

AAMI (2012). AAMI TIR 45:2012 *Guidance on the use of agile practices in the development of medical device software,* http://webstore.ansi.org August 2014.

Abelein, U. & Paech, B. (2012). A proposal for enhancing user-developer communication in large IT projects. In *Proceeding of the 5th International workshop on cooperative and human aspects of software (CHASE),* pp.1-3.

Abelein, U., Sharp, H. & Paech, B. (2013). Does involving users in software development really influence system success? *IEEE Software,* 30(6), pp. 17-23.

Abelein, U. & Paech, B. (2014). State of practice of user-development communication in large-scale IT projects. Results of an Expert interview series. In *Proceeding of Requirements Engineering: Foundation for Software Quality, (REFSQ 2014),* pp. 95-111.

Alemzadeh, H., Iyer, R.K., Kalbarczyk, Z. & Raman, J. (2013). Analysis of safety-critical computer failure in medical devices. *IEEE Security & privacy,* 11(4), pp. 14-26.

Allen, S. (2014). Medical device software under the microscope. *Network Security,* 2, pp. 11-12.

Anderson J., Fleek F., Garrity K. & Drake F. (2001). Integrating usability techniques into software development. *IEEE Software*, 18, pp. 46-53.

ANSI/AAMI (2001). ANSI/AAMI HE74:2001 *Human factors design process for medical devices.* Arlington VA: Association for the advancement of medical instrumentation.

ANSI/AAMI (2009). ANSI/AAMI HE75:2009 *Human factors engineering – design of medical devices.* Arlington VA: Association for the advancement of medical instrumentation.

Avison, D., Lau, F., Myers, M. & Nielsen, P.A. (1999). Action Research, *Communication of the ACM*, 42(1), pp. 94-97.

Avison, D., Baskerville, R. & Myers, M. (2001). Controlling action research projects. *Information technology & people,* 14 (1), pp. 28-45.

Barateiro, J. & Borbinha, J. (2012). Managing risk data: From spreadsheet to information systems. In *Proceeding of the 16th IEEE Electrotechnical Conference Mediterranean (MELECON),* pp. 673-676.

Bartoo, G. & Bogucki, T. (2013). Essentials of Usability in Point-of-Care Devices. In *Proceedings of IEEE Point-of-Care Healthcare Technologies (PHT)*, pp. 184-187.

Basili, V. R. (1996). The Role of Experimentation in Software Engineering: Past, Current and Future, In *Proceedings of the 18th International conference on software engineering*, pp. 442-449.

Becker, J.C. & Flick, G. (1997). A practical approach to failure mode, effects and criticality analysis (FMECA) for computing systems. In *Proceeding of the IEEE High-Assurance Systems Engineering Workshop,* pp. 228-236.

Bell, J. (2005). *Doing your research process,* Berkshire, England: Open University press.

Benet, A.F. (2011). *Advances in Systems Safety*, Chap: A risk driven approach to testing medical device software, pp. 157-168, London: Springer.

Bianco, C. (2011). *Advances in Systems Safety*, Chap: Integrating a risk-based approach and ISO 62304 into quality system for medical devices, pp. 111-125, London: Springer.

Bills, E. & Tartal, J. (2008). Integrating Risk Management into the CAPA Process. *Biomedical instrumentation and technology,* 42(6), pp. 466-468.

Boehm, B. (1991). Software risk management: Principles and practices. *IEEE Software*, 8(1), pp. 32–41.

Bovee, M.W., Paul, D.L. & Nelson K.M. (2001). A framework for assessing the use of third-party software quality assurance standards to meet FDA medical device software process control guidelines. *IEEE Transactions on engineering management,* 48(4), pp. 465-478.

Bowen, J. & Stavridou, V. (1993). Safety-critical systems, formal methods and standards. *Software engineering software,* 8(4), pp. 189-209.

Buxton, J.N. & Malcolm, R. (1991). Software technology transfer. *Software Engineering journal,* 6(1), pp. 17-23.

Cacciabue, P.C. & Vella, G. (2008). Human factors engineering in healthcare systems: the problem of human error and accident management. *International Journal of Medical Informatics,* 79(4), pp. 1-17.

Casey, V. & McCaffery, F. (2013). A lightweight traceability assessment method for medical device software. *Journal of Software: Evolution and Process,* 25(4), pp. 363-372.

Charette, R. N. (1989). *Software engineering risk analysis and management.* McGraw-Hill Software Engineering Series, New York: McGraw-Hill.

Chiozza, M. L. & Ponzetti, C. (2009). FMEA: A model for reducing medical errors. *Clinica Chimia Acta*, 404(1), pp. 75–78.

Chunxiao, L., Raghunathan, A. & Jha, N.K. (2013). Improving the trustworthiness of medical device software with formal verification methods. *IEEE Embedded systems letters,* 5(3), pp. 50-53.

Conboy, K. & Fizgerald, B. (2010). Method and developer characteristics for effective agile method tailoring: a study of XP expert opinion. *ACM Transactions on Software Engineering and Methodology,* 20(1), pp. 1-28.

Cooper, E.S. & Pauley, K. (2006). Healthcare software assurance. In *Proceedings of AMIA Annual symposium 2006*, pp. 166-170.

Crouhy, M., Galai, D., & Mark, R. (2006). *The essentials of risk management*. Maidenherd: McGraw-Hill.

Daniels J., Fels S., Kushniruk A., Lim J. & Ansermino J.M. (2007). A framework for evaluating usability of clinical monitoring technology. *Journal of Clinical Monitoring and Computing,* 21, pp. 323-330.

Davison, R.M., Martinsons, M.G. & Kock, N. (2004). Principles of canonical action research. *Information systems journal,* 14, pp. 65-86.

Dey, P. K., Kinch, J., & Ogunlana, S. O. (2007). Managing risk in software development projects a case study. *Industrial Management and Data Systems*, 107, pp. 284–303.

Dhillon, B. S. (2000). *Medical device reliability and associated areas*. Boca Raton: CRC press Taylor & Francis Group.

Dhillon, B.S. (2008). *Reliability technology, human error and quality in health care.* Boca Raton: CRC press, Taylor & Francis Group.

Doerr, J., Kerkow, D. & Landmann, D. (2008). Supporting requirements engineering for medical products – early consideration of user-perspective quality. In *Proceedings of International conference on software engineering (ICSE 08),* pp. 10-18.

Dumas, J.S. & Redish, J.C. (1999). *A practical guide to usability testing.* Exeter: Intellect Books.

Easterbrook, S., Singer J., M.A., & Damian, D. (2008). *Guide to advanced empirical software engineering.* Chap: Selecting empirical methods for software engineering research, pp. 285-311, London: Springer-Verlag.

EN (2006). EN 60601-1, *Medical electrical equipment - Part 1: General requirements for basic safety and essential performance.* http://www.sis.se. August 2014.

European Council (1993). Council Directive 93/42/EEC Concerning medical devices. Luxembourg, Official Journal of the Communities.

European Council (2007). Council Directive 2007/47/EC (Amendment). Luxembourg, Official Journal of the European Union.

Fairley, R. E. (2005). Software risk management. *IEEE Software*, May/June, pp. 101.

FDA (1995). U.S. Food and Drug Administration, 1995. Premarket Notification [510 (k)], Regulatory Requirements for Medical Devices, HHS Publication, FDA 95-4158.

FDA (1996). Do it by design: An introduction to human factors in medical devices.

FDA (2000). Medical Devise Use-Safety: Incorporating human factors engineering into risk management.

FDA (2006). U.S. Food and Drug Administration, Federal Food, Drug and Cosmetic Act section 201(h).

Fink, A. (2003). *The survey handbook.* Thousand Oaks California: Sage Publications.

Fitzgerald, B., Stol, K-J., O'Sullivan, R. & O'Brian, D. (2013). Scaling agile methods to regulated environments: An industry case study. In *Proceedings of IEEE International conference on software engineering (ICSE 2013),* pp. 863-872.

Gall, H. (2008). Functional safety IEC 61508/IEC 61511. The impact to certification and user. In *Proceedings of IEEE International conference on computer systems and application,* pp. 1027-1031.

Gamer K., Liljegren E., Osvalder A-L. & Dahlman S. (2002). Application of usability testing to the development of medical equipment. Usability testing of a frequently used infusion pump and a new user interface for an infusion pump developed with Human Factors approach. *International Journal of Industrial Ergonomics*, 29, pp. 145-159.

Garde, S. & Knaup, P. (2006). Requirements engineering in health care: the example of chemotherapy planning in paediatric oncology. *Requirements Engineering*, 11(4), pp. 265–278.

Gary, K., Enquobahrie, A., Ibanez, L., Cheng, P., Yaniv, Z., Cleary, K., Kokoori, S., Muffih, B. & Heidenreich, J. (2011). Agile methods for open source safety-critical software. *Software: Practice and Experience,* 41(9), pp. 945-962.

Gosbee J. & Ritchie E. (1977). Human-computer interaction and medical software development. *Interactions,* 4, pp. 13-18.

Gorschek, T., Garre, P., Larsson, S. & Wohlin, C. (2006). A model for technical transfer in practice. *IEEE Software,* 23(6), pp. 88-95.

Gregor, S. & Hevner, A. R, (2013). Positioning and presenting design science research for maximum impact, *MIS Quarterly,* 37(2), pp. 337-355.

Habraken, M. M. P., Van der Schaal, T. W., Leistikow, I. P., & Reijnders-Thijssen, P. M. J. (2009). Prospective risk analysis of health care processes: A systematic evaluation of the use of HFMEA in Dutch health care. *Ergonomics*, 52, pp. 809–819.

Hall, E. M. (1998). *Managing risk: Methods for software systems development*. Reading: Addison Wesley.

Hegde, V. (2011). Case study: Risk management for medical devices. In *Proceedings of reliability and maintainability symposium (RAMS*), pp. 1-6.

Hevner, A.R. (2007). A three cycle view of design science research, *Scandinavian journal of information systems,* 19(2), pp. 87-92.

Hevner, A.R. & Chatterjee, S. (2010). *Design research in information systems: Theory and practice,* New York: Springer.

Holzinger, A. (2005). Usability engineering methods for software developers. *Communications of the ACM*, 48, pp. 71- 74.

Hove, S.E. & Anda, B. (2005). Experiences from conducting semi-structured interviews in empirical software engineering research, In *Proceedings of the 11th IEEE International software metrics symposium, pp. 23-33.*

Hrgarek N. (2012). Certification and Regulatory Challenges in Medical Device Software Development. In *Proceedings of Software Engineering in Health Care*, pp. 40-43.

Hyman, W.A (2002). A generic fault tree for medical device error. *Journal of Clinical engineering,* 27(2), pp. 134-140.

Höst, M., Regnell, B. & Wohlin, C. (2000). Using students as subjects - a comparative study of students and professionals in lead time impact assessment, *Empirical Software Engineering*, 5 (3), pp. 201- 214.

Höst, M., Wohlin, C. & Thelin, T. (2005). Experimental context classification: Incentives and experiences of subjects. In *Proceedings of the 27th International conference on software engineering*, pp. 470-478.

IEC (2003). IEC 61511, *Functional safety – Safety instrumented systems for the process industry sector.* Geneva, Switzerland. IEC.

IEC (2006a). IEC 62304:2006, *Medical device software – software life cycle processes.* http://www.iso.org. August 2014

IEC (2006b). IEC 61025, *Fault tree analysis (FTA),* http://www.iec.ch. August 2014.

IEC (2006c). IEC 60812, Analysis techniques for system reliability - Procedure for failure mode and effects analysis (FMEA), http://www.iec.ch. August 2014.

IEC (2007). IEC 62366:2007, *Medical devices – application of usability engineering to medical devices.* http://www.iso.org. August 2014.

IEC/TR (2009). IEC/TR 80002-1:2009, *Medical device software -- Part 1: Guidance on the application of ISO 14971 to medical device software.* http://www.iso.org. August 2014.

IEC (2010a) IEC 61508:2010 *Functional Safety of Electrical/Electronic/Programmable Electronic Safety-related Systems.* Geneva, Switzerland, IEC.

IEC (2010b). IEC 80001-1 *Application of risk management for IT-networks incorporating medical devices -- Part 1: Roles, responsibilities and activities,* http://www.iso.org. August 2014.

ISO (2003). ISO 13485:2003 *Medical devices -- Quality management systems -- Requirements for regulatory purposes.* http://www.iso.org. August 2014.

ISO (2012). ISO 14971:2012 *Medical devices -- Application of risk management to medical devices,* Geneva, Switzerland. ISO.

Iversen, J.H., Mathiassen, L. & Nielsen, P.A. (2004). Managing risk in software process improvement: An action research approach. *MIS Quarterly,* 28(3), pp. 395-433.

Jain, R.K., Ananthakrishnan, T.S., Mandalik, S.A. & Jindal, G.D. (2010). Risk analysis of medical instruments – case study of cardiac output monitor. In *Proceeding of 2nd International conference on reliability, safety and hazard (ICRESH),* pp. 637-641.

Jedlitschka, A., Ciolkowski M. & Pfahl D. (2008). *Guide to advanced empirical software engineering.* Chap: Reporting Experiments in Software Engineering, pp. 201-228, London: Springer-Verlag.

Jones, C. (1994). *Assessment and control of software risks*. Englewood: Prentice-Hall.

Jørsang, A., AlFayyadh, B. & Grandison, T. (2007). Security usability principles for vulnerability analysis and risk assessment. In *Proceedings of 23rd Computer Security Applications Conference (ACSAC 2007*), pp. 269-278.

Kamm, D. (2005). An introduction to risk/hazard analysis for medical devices. FDA consultant, http://www.fdaconsultant.com/cv_kamm.htm. August 2014.

Kampenes, V. B., Dybå, T., Hannay, J. E. & Sjøberg, D. I. K. (2009). A systematic review of quasi-experiment in software engineering, *Information and Software Technolog,* 51, pp. 71-82.

Kitchenham, B.A., Pfleeger, S.L., Pickard, L. M., Jones, P.W., Hoaglin, D.C., Emam, K.E. & Rosenberg J. (2002). Preliminary guidelines for empirical research in software engineering. *IEEE Transactions on Software Engineering*, 28(8), pp. 721-734.

Kitchenham, B.A. & Pfleeger, S.L. (2008). *Guide to advanced empirical software engineering*. Chap: Personal opinion surveys, pp. 63-93, London: Springer-Verlag.

Knight, J.C. (2002). Safety critical system: Challenge and directions. In *Proceeding of the 24th International conference on software engineering (ICSE),* pp. 547-550.

Kohn, L., Corrigan, J. & Donaldson, M. (2000*). To err is human: building a safer health care system*. Washington: National Academy Press.

Krasich, M. (2000). Use of fault tree analysis for evaluation of system-reliability improvements in design phase. In *Proceeding of annual reliability and maintainability symposium,* pp. 1-7.

Kushniruk, A. (2002). Evaluation in the design of health information systems: application of approaches emerging from usability engineering. *Computers in biology and medicine,* 32, pp. 141-149.

Kushniruk A.M., Triola M.M., Borycki E.M., Stein, B. & Kannry, J.L. (2005). Technology induced error and usability: The relationship between usability problems and prescription errors when using a handheld application. *International Journal of Medical Informatics*, 74, pp. 519-526.

Leveson, N.G. (1986). Software safety: why, what and how. *ACM computing surveys (CSUR),* 18(2), pp. 125-163.

Leveson, N.G. (2011). *Engineering a safer world: Systems thinking applied to safety,* London: MIT Press.

Lindholm, C. & Höst, M. (2008). Development of software for safety critical medical devices – an interview-based survey of state of practice. In *Proceeding of the 8th conference on software engineering research in and practice in Sweden (SERPS 08),* pp. 1-10.

Lethbridge, T.C., Sim, S.E. & Singer J. (2005). Studying software engineering: Data collection techniques for software field studies. *Empirical software engineering,* 10, pp. 311-341.

Lindberg, K.R. (1993). Defining the role of software quality assurance in a medical device company. In *Proceeding of the 6th Annual IEEE symposium on computer-based medical systems,* pp. 278-283.

Lozier, T. (2010). Streamline Your CAPA Process: Use Risk Assessment to Improve Quality and Compliance. *The Quality Assurance Journal,* 13(1-2), pp. 37-40.

Madrigal, D. & McClain, B. (2010). Do's and don'ts of usability testing. http://www.uxmatters.com. August 2014.

McCaffery, F., McFall, D., Donneley, P., Wilkie, F.G. & Steritt, R. (2005). A software process improvement lifecycle framework for the medical device industry. In *Proceeding of the 12th IEEE International conference and workshops of the engineering of computer-based systems (ECBS 05),* pp. 273-280.

McCaffery, F., Burton J. & Richardson I. (2009). Improving software risk management in a medical device company. In *Proceedings of the International conference on software engineering (ICSE),* pp. 152-162.

McCaffery, F., Burton, R. & Richardson, I. (2010). Risk management capability for the development of medical device software. *Software quality journal,* 18, pp. 81-107.

McCaffery, F., Casey, V., Sivakumar, M., Donneley, P. & Burton, J. (2012). Software and system traceability. Chap: Medical device software traceability, pp. 321-343, Berlin: Springer-Verlag.

McDermid, J.A., Nicholson, M., Pumfrey, D.J. & Fenelon, P. (1995). Experiences with the application of HAZOP to computer-based systems. In *Proceedings of the conference on 10th annual Computer Assurance Systems Integrity, Software Safety and Process Security* (COMPASS'95), pp. 37-48.

McHugh, M., Cawley, O., McCaffery, F., Richardson I. & Wang, X. (2013). An agile V-model for medical device software development to overcome the challenge with plan-driven lifecycles. In *Proceeding of the software engineering in healthcare workshop at the 35th International conference on software engineering (ICSE 2013)* pp. 12-19.

McHugh, M., McCaffery, F. & Casey V. (2014). Adopting agile practices when developing software for use in the medical domain. *Journal of software: evolution and process,* 26, pp. 504-512.

McRoberts, S. (2005). Risk management of product safety. In *Proceedings of IEEE Symposium on product safety engineering*, pp. 65-71.

Merrill C. & Feldman D. (2004). Rethinking the path to usability. How to design what users really want. *Computer Society*, 6, pp. 51-57.

Méry, D. & Kumar Singh, N. (2010). Trustable formal specification for software certification. In *Proceedings of 4th International Conference on Leveraging Applications of Formal Methods, Verification, and Validation. (*ISoLA'10), pp. 312-326.

Nielsen, J. (1992) The usability engineering life cycle, *Computer,* 25(3), pp. 12-22.

Nielsen J. (1994). Enhancing the Explanatory Power of Usability Heuristics. In *Proceedings of Human Factors in Computing Systems Conference*, pp. 152-158.

Obradovich, J.H. & Woods D.D. (1996). Users as Designers: How people cope with poor HCI Design in Computer-Based Medical Devices. *Human Factors*, 38, pp. 574-592.

Padayachee, K. (2002). An interpretive study of software risk management perspectives. In *Proceeding of the annual research conference South African institute of computer scientists and information technologists on Enablement through technology (SAICSIT 02),* pp. 118-127.

Pfleeger, S.L. (1999). Understanding and improving technology transfer in software engineering, *Journal of systems and software,* 47(2-3), pp. 111-124.

Punter, T., Ciolkowski, M., Freimut, B. & John, I. (2003) Conducting on-line surveys in software engineering. In *Proceeding of the International Symposium on Empirical Software Engineering, (ISESE 2003),* pp. 80-88.

Rea, L. & Parker, R. (2005). *Designing and conducting survey research: a comprehensive guide*. San Francisco CA: Jossey-Bass.

Rakitin, S. R. (2006). Coping with defective software in medical devices. *IEEE Computer*, 39(4), pp. 40–45.

Reason, J. (1990). *Human error*. Cambridge: Cambridge University Press.

Reason, J. (1997). *Managing the risks of organizational accidents*. Surrey: Ashgate Publishing Limited.

Robson, C. (2002). *Real world research* (2nd ed.). Oxford UK: Blackwell Publishers.

Rogers Y., Sharp, H. & Preece J. (2011). *Interaction design: Beyond human – computer interaction*, (3rd ed.), West Sussex, UK: Wiley.

Rosenberg, J. (2005). *Guide to advanced empirical software engineering*. Chap: Statistical methods and measurement, pp. 155-185, London: Springer-Verlag.

Rottier, P.A. & Rodrigues, V. (2008). Agile development in a medical device company. In *Proceeding of conference Agile (AGILE 08),* pp. 218-223.

Runeson, P. & Höst, M. (2009). Guidelines for conducting and reporting case study research in software engineering. *Empirical Software Engineering,* 14(2), pp. 131-164.

Runeson, P., Höst, M., Rainer, A. & Regnell, B. (2012). *Case study research in software engineering: Guidelines and example*s. Hoboken, New Jersey: Wiley.

Sagor, R. (2011). *The action research guidebook: A four-stage process for education and school.* Thousand Oaks California: Sage Publications.

Sayre K., Kenner, J. & Jones P. (2001). Safety models: an analytical tool for risk analysis of medical device systems. In *Proceedings of 14th IEEE symposium on computer-based medical systems (CMBS'01*), Maryland, US, pp. 445-451.

Schmuland, C. (2005). Value-added medical-device risk management. *IEEE Transactions on Device and Materials Reliability,* 5(3), pp. 488–493.

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

Shah, S.G.S. & Robinson I. (2006). User involvement in healthcare technology development and assessment. *International journal of health care quality assurance* 19(6), pp. 500-515.

Sharp, H., Rogers, Y. & Preece, J. (2007). *Interaction design: beyond human-computer interaction (2nd ed.)* West Sussex: John Whiley & Sons, Ltd.

Shull, F., Singer J. & Sjöberg, D. I. K. (2008). Guide to advanced empirical software engineering. Chap Introduction, pp. 1-5, London: Springer-Verlag.

Sjøberg, D. I. K., Dybå, T. & Jörgensen, M. (2007). The Future of Empirical Methods in Software Engineering Research. In *Proceeding of IEEE Future of software engineering (FOSE '07),* pp. 358-378.

Small, H. (1998). Florence Nightingale's statistical diagrams. In *Proceeding of Stat & Lamps research conference,* pp. *1-5.*

Smith, D. J. & Simpson, K.G.L (2011). *Safety Critical Systems Handbook A Straightforward Guide to Functional Safety, IEC 61508 and Related Standards, Including Process IEC 61511 and Machinery IEC 62061 and ISO 13849*, Oxford: Elsevier.

Sommerville, I. (2007). *Software engineering (8th ed.)*. Readings: Addison Wesley.

Stake, R.E. (1995). *The art of case study research.* SAGE Publication.

Trucco, P. & Cavallin, M. (2006). A quantitative approach to clinical risk assessment: The CREA method. *Safety Science,* 44(6), pp. 491-513.

Wakker, P. & Deneffe, D. (1996). Eliciting von Neumann-Morgenstern utilities when probabilities are distorted or unknown. *Management science,* 42(8), pp. 1131-1150.

Wallace, D.R. & Kuhn, R. (2001). Failure modes in medical device: an analysis of 15 years of recall data. *International journal of reliability quality and safety* 8(4), pp. 351-373.

Walsh, T. & Beatty, P. C. W. (2002). Human factors error and patient monitoring. *Physiological Measurement*, 23(3), pp. 111–132.

Velsen, L., Geest, T. & Klaassen, R. (2007). Testing the usability of a personalised system: comparing the use of interviews, questionnaires and thinking-aloud. In *Proceeding of* IEEE *International professional communication conference (IPCC 2007),* pp. 1-8.

Wiklund M., Kendler J. & Strochlic A.Y. (2011). *Usability Testing of Medical Devices,* U.S.: CRC Press.

Wilkins, R.D. & Holley, L.K. (1998). Risk management in medical equipment management. In *Proceeding of the IEEE 20th annual international conference on Engineering in Medicine and Biology Society,* 6, pp. 3343-3345.

Vinson, N.G. & Singer, J. (2005). *Guide to advanced empirical software engineering.* Chap: A practical guide to ethical research involving humans, pp. 229-257, London: Springer-Verlag.

Virzi R.A. (1992). Refining the test phase of usability evaluation: How many subjects is enough? *Human Factors,* 34(4), pp. 457-471.

Vogel, D.A. (2006). Software safety for every phase of software development. *Biomedical instrumentation & technology*, 40(4), pp. 309-314.

Wohlin, C., Runeson, P., Höst, M., Ohlsson, M.C., Regnell, B. & Wesslén A. (2000). *Experimentation in Software Engineering: An introduction.* Boston: Kluwer Academic.

Xiuxu, Z. & Xiaoli, B. (2010). The application of FMEA method in the risk management of medical devices during the lifecycle. *In Proceedings of 2nd International conference on e-business and information system security (EBISS)*, China, pp. 1-4.

Yang, L., Frize, M. & Eng, P. (2003). Incorporating Usability Design Factors into Development of Clinical Decision Support Systems. In *Proceedings of the 25th International conference of the IEEE EMBS*, Cancun, Mexico, pp. 3594-3597.

Yin, R. K. (2003). *Case study research: Design and methods* (3rd ed.). Beverly Hills: Sage.

Zhang, D. & Xie, T. (2013). Pathways to Technology Transfer and Adoption: Achievements and Challenges (Mini-Tutorial), In *Proceeding of the 35th International conference on software engineering (ICSE),* pp. 951-952.

INCLUDED PAPERS

Paper I

A Survey of Software Engineering Techniques in Medical Device Development

R. Feldmann, F. Shull, C. Denger, M. Höst, and C. Lindholm

Abstract

A wide variety of the functions provided by today's medical devices rely heavily on software. Most of these capabilities could not be offered without the underlying integrated software solutions. As a result, the medical device industry has become highly interdisciplinary. Medical device manufacturers are finding an increasing need to incorporate the research ideas and results from traditionally disconnected research areas such as medicine, software and system engineering, and mechanical engineering. In 2006, we conducted a survey with more than 100 companies from Europe and the USA to shine some light on the current status of the integration of software engineering technologies into the medical device domain. The initial results of this survey are presented in this paper. Both software engineers and the medical device industry can use these findings to better understand current challenges and future directions, to achieve a better integration of the fields.

Published in the proceedings of the workshop on High Confidence Medical Devices, Software, and Systems and Medical Device Plug-and-Play Interoperability, Boston, USA, 2007, pp. 46-54.

1 Introduction

Today, many medical devices could not fulfil their intended use without the software embedded within them, which implements a variety of functions and features. Surveys of trends in the medical device industry (e.g., AdvaMed 2004; IETA 2005; BDI 2005) indicate that software is one of the most decisive factors for producing innovative products with new capabilities, and predict that the importance of software will only further increase in the future (BMBF 2005). Studies also predict that the research and development (R&D) investment in software in this market will increase to 33% of the overall budget by 2015 (IETA 2005).

As the role of software in the medical device domain increases in importance, so do the failures due to software defects. An analysis of medical device recalls by the FDA in 1996 (Wallace & Kuhn 2001) found that software was increasingly responsible for product recalls: In 1996, 10% of product recalls were caused by software-related issues. This was up from 6% in the years 1983–1991. A German survey on medical device recalls in the medical sector indicates that software is the top cause for risks related to construction and design defects of medical device products. This analysis, from June 2006, shows that 21% of the medical device design failures are caused by software defects (BFARM 2006). This is an increasing trend, since the same figures from November 2005 show software responsible for 17% of construction and design defects.

To address such issues, the development of medical device software is regulated by various standards, laws and recommendations (e.g., ISO 2000; IEC 2000; CDRH 2002). In general, these standards describe software life-cycle models that should be implemented by manufacturers. The overall objective is the definition of general process steps and intermediate work-products. Adhering to the regulations and following the specified processes increases an organization's ability to produce safe, high quality medical device software. However, in many cases the standards are quite vague regarding the concrete software engineering techniques that should be used in different development steps. Thus, in practice there is a high degree of freedom in instantiating the processes. This may be an indicator that currently

software engineering standards are only loosely integrated into this domain.

Given this context, and the general lack of empirical knowledge about the state of the practice regarding medical device software development, we designed an international survey. Our objective was to understand the current goals and concerns of companies in this market and how they choose software practices to address the issues that they see. We were interested in understanding the extent to which the standards that have been developed have been recognized and instantiated by industry. Furthermore, we were interested in eliciting the most important challenges with respect to developing embedded software for medical purposes. These results can be used by researchers and practitioners to get an overview of the level of usage of various processes and tools in the domain as well as to understand useful targets for developing new techniques and methods aimed at further improving software quality in the medical sector.

This paper presents the basic statistical results of that survey, which was run in 2006. The questions targeted in the survey can be summarized as:

1. How can the medical device area in general be characterized regarding software engineering?
2. What are the most frequently applied software engineering techniques, methods and tools during medical device production?
3. What are the most recent challenges with respect to software engineering in medical device production?

More than 100 companies from Europe and the USA participated in the survey, ranging in size from small organizations with less than 20 employees to global players with several thousand developers. The products of the companies range from devices implanted during surgery to diagnosis systems to noninvasive devices and information systems. Most of these products are classified as products of class II (Special Controls) according to the FDA standards and class IIa/IIb according to European standards (i.e., devices with medium risk such as electro-medical devices). On this basis, the survey covers a wide cross section of the industry.

To our knowledge, this is the first survey of this size that explicitly

focuses on the topic of software engineering in the medical device domain. We perceive the results as a basis for the identification of current and future challenges with respect to software engineering and quality assurance in the field. Researchers and practitioners can use the identified challenges as input for…

- … a better integration of software engineering standards into the applied processes and standards.
- …the development of new, innovative techniques and methods to further improve software quality, tailored to the medical device domain.

The remainder of this paper is structured as follows: In section 2 we describe the design of our survey. Key objectives are discussed, as well as our sample target group and the chosen approach for collecting the data. Next, we present an overview of the survey outcomes (section 3). Insights into the original data set as well as the methods used for analysis are given. Based on the results we provide an interpretation of the findings from a software engineering point of view. We summarize our findings and conclude in section 4.

2 Survey design

This survey was designed by software engineering researchers from three institutions in the United States and Europe (the Fraunhofer Institute for Experimental Software Engineering in Germany; the Fraunhofer Center in Maryland, USA; and the Lund University, Sweden). The main objective of the survey was to characterize the state of the practice of software development in the specific context of medical devices and medical information systems, in order to understand the extent to which software engineering practices are integrated and used in the medical device domain. In all cases, our objective was to characterize the state of the practice in this context. The survey collected no information that could be used to directly evaluate the effectiveness of the practices that are being applied, nor did it collect information that would compare respondents in this survey to those in another domain.

2.1 Sample and target group

The target population for the study consists of organizations developing software for medical devices and medical information systems. This is a large population and it is impossible to carry out a study with the complete population where every organization is included. Therefore, a sample of the population has to be chosen.

The invited organizations were chosen based on the contacts that the authors had and were able to obtain in the initial steps of the survey. Email invitations were sent to all of these contacts as well as to contacts in the databases of large industry associations. These associations supported the authors in conducting the survey by mailing out the link to the questionnaire throughout their distribution lists. However, since we were not allowed to access these third party databases directly, we do not know how many invitations were sent out in total. Obviously, the chosen methodology of the survey does not necessarily result in a rigorous random sample of medical device companies. However, in the given circumstances it was the best possible design we could come up with for our research interests.

2.2 Conducting the survey

The survey was carried out through a web-based questionnaire. Potential participants were given a link to the URL and asked to fill out the questionnaire. In order to motivate people to participate they were given the possibility to register their email in order to obtain the result of the survey. As a further motivation for participants we donated $1 to the International Red Cross for each completed on-line survey.

3 Collected data

As a response to the invitations, 349 individuals looked at the starting page of the questionnaire. From these, survey responses at some level of completeness were received from 113 participants.

Since it was possible to skip questions when answering the survey, some respondents did not provide answers to all survey questions. We filtered the dataset by deciding on a minimum threshold, below which

too little information had been provided to be usefully included in the analysis. The survey ends with a set of characterization questions about the respondent's organization and the developers working there. Only the answers of those respondents who have answered at least one of these characterization questions are included in the survey. We believe that subjects, who reviewed all of the questions, even if they did not provide answers to all of them, are likely to provide the most valid data. Because of the filtering, the dataset was left with 57 valid responses, which were included in the analysis. In the following, we call this set of companies the "core data set."

The respondents come from many different types of companies. 71% of the companies participating in the survey are small and medium sized companies with 10- 250 employees. Within these companies, the size of the software development departments / teams varies. Almost 50% of the software development teams are smaller than 11 people. Only 18% of the companies have development teams with more than 50 people. Of the 57 companies from the core data set, most respondents came from Germany (38), the USA (8), and Sweden (5).

With these numbers, our survey may not be indicative of the state of the entire industry. However, a large percentage of our respondents were alike in having primary focuses on safety critical software.
Thus, our result scan shed some light on software engineering practices in this specialized area. In addition, the results seem to be consistent with the experience of some domain insiders from which we received feedback, and thereby seem to provide a fair characterization of the industry.

3.1 Analysis of data

One of the most important questions focused on how companies who produce medical devices obtain the software components (i.e., whether the software is developed in-house or obtained from outside). Of the 57 respondents, 20 companies use only software developed in-house in their products and seven companies use only third party software in their products. 29 companies do both, obtaining some software from third parties and integrating it with software developed in-house. One company stated that their products do not contain software.

This has implications for data that can be used in further analysis. We use 49 respondents for questions dealing with software development and 36 respondents for questions with respect to software from third parties.

The analyses have been carried out with two data sets. First, the analysis was carried out with the core data set consisting of the answers from the 57 people who had answered the characterization questions. Where relevant, the results from the core dataset were further tested by analysing the entire set of 113 respondents who answered the entire questionnaire. In no case were the results of this different from those for the core dataset, so we do not report these cases explicitly in the following sections.

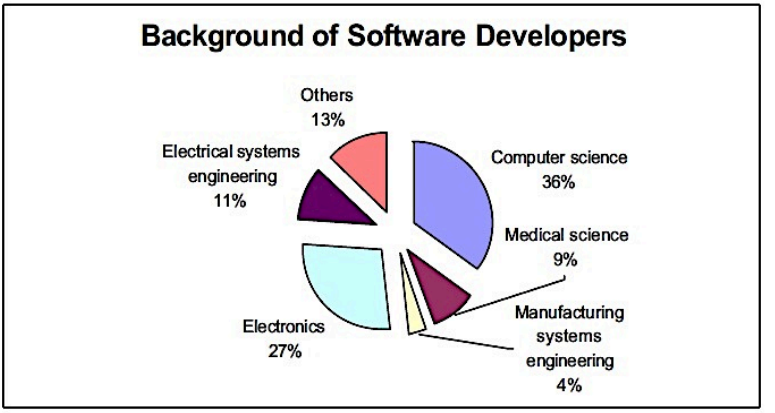

Figure 1. Educational background of software developers

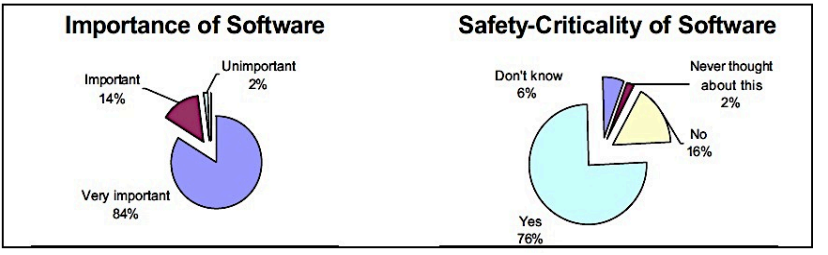

Figure 2. Importance and safety criticality of software

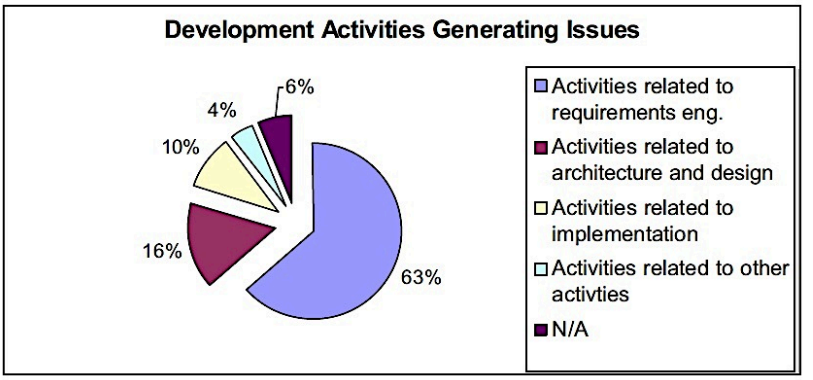

Figure 3. Development activities from which most issues originate

3.2 Characterizing software development in the organization

Participants were asked to characterize the educational background of the majority of the members of their software development team. Figure 1 shows the results: In 64% of the cases, the majority of the software developers have a background other than computer science. These developers come from disciplines such as electronics, medical sciences, or electrical systems engineering. Only in 36% of the cases is the medical device software created mainly by computer scientists. Figure 2 indicates the relevance of software in medical device products. It is evident that software is an important part in many medical devices: 98% of the companies rate software as either a very important (84%) or important (14%) component of their products. The figure also indicates that software is a safety critical element of the product in over 75% of the cases. In contrast, for only 16% of the companies does the software clearly realize non-safety-critical functionalities.

Asked about some of the perceived challenges, 64% of our participants noted that finding sufficient software developers (i.e., staff with a computer science background) is a challenge. At the same time, most
issues regarding software quality stem from activities involved in planning the software, the functionality it should accomplish, and how it will accomplish it. That is, most challenges stem from requirements

activities (63%) and architecture and design activities (16%) (see Figure 3). Actually implementing the software code is perceived as the most challenging activity by only 10% of the respondents. This indicates that investments in requirements engineering activities seem to be most promising to gain significant improvements in the software development process.

Looking in more detail at issues that cause problems for each type of activity, for requirements-related activities 86% of the companies perceive changing requirements as the main problem. Missing requirements (33%) and misinterpreting requirements (39%) are also perceived as important.

Related to architecture and design, the main challenge is missing information in the diagrams (53%) and inconsistencies between the planned architecture and the software (39%). Missing opportunities to reuse software code in a systematic way (33%) and difficulties in maintaining the software code (29%) are perceived as the main issues during implementation.

Figure 4 shows that around 50% of the companies follow a defined process to perform the activities mentioned above on a regular basis, that is, they said that they always or frequently follow such a process. (If the criteria are loosened to include companies that follow defined processes in about half of their projects, then 78% of the respondents had a defined process for implementation, 71% had one for architecture, and 69% had one for requirements).

3.3 Characterizing the challenges of using notations and tools

In order to document the results of the various activities involved in constructing software, different notations and languages can be applied. Most of our respondents were using relatively informal notations and techniques to do so. Formal languages (e.g., temporal logic, architecture description languages) describing software requirements or architectures were rarely utilized. For example, for describing software requirements as well as architecture and design, only 2% of the companies use formal languages in all of their projects. In 22% of the companies, formal languages are used frequently in the requirements phase and in 14%

formal languages are used frequently for architecture and design.

Consequently, less formal notations and languages are most popular for use in all types of activities.

Figure 5 shows the results for requirements engineering. There, natural language (e.g., English or German) is used in almost all companies in all projects. In 92% of the companies this kind of notation is used always or frequently.

A detailed analysis of the answers reveals that for 46% of the companies natural language is the one and only notation to specify requirements. Structured notations such as use cases are used by 40% of the companies on a regular basis (i.e., always or frequently used).

For architecture and design, structural diagrams are the most frequently applied notations for modelling the software. These diagram types (e.g., class diagrams, package diagrams, functional block diagrams) are used by 64% of the companies on a regular basis (always or frequently applied). Sequence and data flow diagrams seem to have a lower importance. These diagrams are frequently used by 40% and 36% of the companies, respectively. More formal notations such as state charts and/or Time Petri Nets seem to be of low importance in the medical device domain. State charts are used on a regular basis by 23% of the companies, while Time Petri Nets are not applied regularly by any respondent.

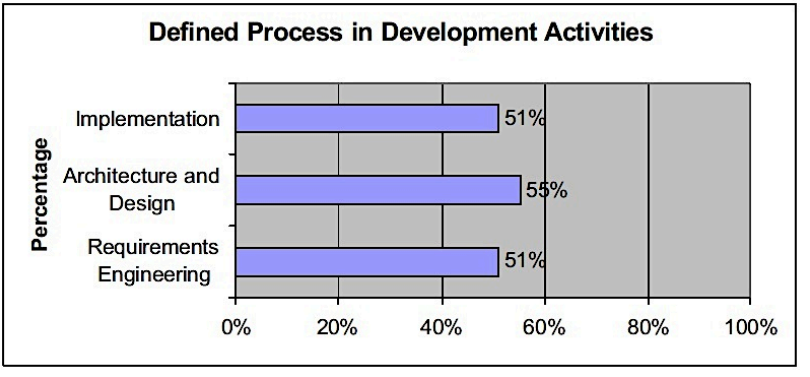

Figure 4. Defined processes for development activities

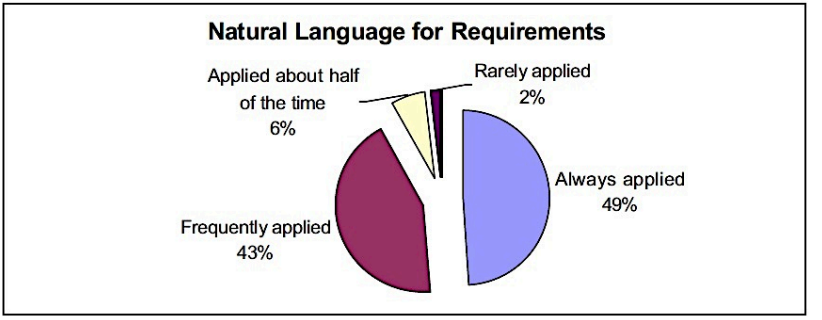

Figure 5. The usage of natural language in the requirement phase

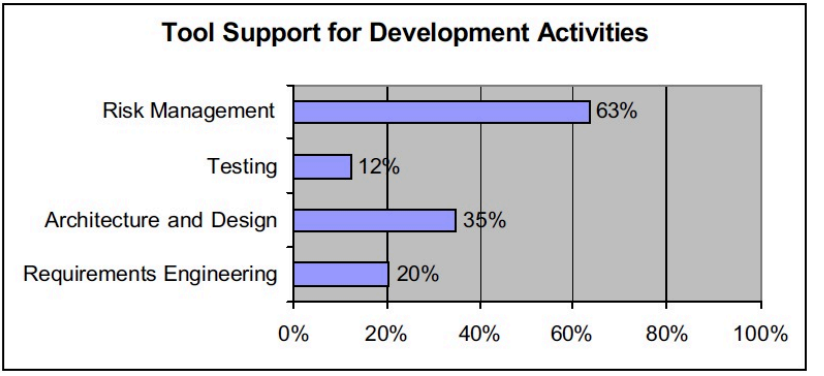

Figure 6. Usage of tools for different activities

Only 6% of the companies occasionally (i.e., less then half the time) use this notation.

Many tools exist that assist with various constructive techniques (e.g., requirements management tools, design modelling tools, test execution tools). Software engineers have long found that, in order to make effective use of a tool, it is important to have a good process definition in place. A tool cannot overcome deficiencies regarding the processes followed. Consequently, frequent tool usage may be an indication of higher process maturity in software development activities when paired with well defined processes. However, Figure 6 shows that tools are not frequently used in the medical device domain.

For example, even though testing can be easily supported by tools, less than 13% of the companies use tools on a regular basis for this purpose.

Tool support for other types of activities is also infrequent.

Specifying and managing requirements is tool-supported in 20% of the companies. (For this purpose DOORS™ (16%) and RequisitePro™ (10%) are the most frequently used tools.) Architecture and design activities are supported by tools in 35% of the companies. Here UML modelling tools (25%) had the highest importance. Tools that are typically applied in many other domains of embedded software (e.g., in the automotive industry), such as Matlab Simulink or LabView, seem to have a lower importance in the medical device domain. Matlab is applied by 8% of the companies on a regular basis, LabView by 12%.

Risk management activities seem to be the activities that are most frequently supported by tools (63% of the companies use a risk management tool on a regular basis). However, these numbers must be carefully interpreted. A detailed analysis of the data shows that almost all of the companies state that they do not use commercial tools specialized to support this activity but common applications such as text editors (like MS Word™) or tabular calculation sheets (e.g., MS Excel™).

3.4 Characterizing quality assurance for software

Quality assurance activities are an integral part of software development processes. Especially in safety critical domains such as the medical device domain, risk management activities that are used to analyze, control and monitor safety risks of the devices and the software included therein are of high importance. We found that software quality assurance activities are important in the overall medical device development process. 60% of the companies frequently plan quality assurance activities, while another 14% establish such plans for about half of their software development projects. In addition, we found that the results of software quality assurance techniques are frequently documented. 62% of the companies always or frequently document them.

However, mature software quality assurance activities are supported by activities aimed at quantitatively assessing progress and quality, such as the collection of code or design metrics or performing benchmarks of the software. It seems that these supporting activities are not applied in a wide range of medical device companies. Less than 5% of the

companies always collect code or design metrics of their software. Benchmarks are frequently used by fewer than 13% of the companies. The most frequently applied supporting activity is defect classification: 35% of the companies use a defect classification to understand better the nature of the software faults. However, it remains unclear how detailed these classifications are defined in practice. Consequently, judging their value for supporting quality assurance planning is not possible.

Among the quality assurance techniques, testing was the one most frequently used. System validation activities applied to the software (i.e., testing / validating that the software fulfils its requirements) are performed by 78% of the companies in all projects and in a further 16% of the companies frequently. Inspections and reviews (i.e., the verification of the quality of intermediate development products) are also applied quite frequently: 32% of the companies perform this activity in all of their software development projects, 24% apply the technique frequently, and another 16% perform inspections in half of their projects.

Risk analysis can be performed using different techniques such as Fault Tree Analysis (FTA) (FTA 2007) or Failure Mode and Effect Analysis (FMEA) (Kasker 2004). These techniques are recommended by various standards as suitable approaches for systematically identifying the most relevant risks during development of a medical device. Our findings indicate that FMEA is the most frequently applied technique for risk analysis. 42% of the companies always or frequently use the FMEA technique for this purpose. FTA seems to be of lower importance as only 18% of the companies always or frequently use this technique. However, only 18% of the respondents stated that they perform software FMEA on a regular basis in their development processes. Since this is rather less than the number of respondents applying FMEA in general, companies
seem to be applying the FMEA technique at the system level, rather than specifically to the software component. This finding is consistent with our practical experiences that even though many developers understand that risk analysis should be performed on software and software components, it remains unclear in practice exactly how this should be performed. Other risk analysis techniques (e.g., hazard and

operability analysis (HAZOP) (HAZOP 2007), reliability block diagrams, or event trees) seem not widely used in the medical device domain, as they were not mentioned by the respondents as alternative techniques for risk analysis.

4 Conclusion

Our survey supports the assumption that software engineering methods, techniques, tools, and standards could be better integrated into the used development processes and existing standards for medical devices.
One reason for this might be the fact that software in organizations developing medical devices is mainly created by developers with a background other than computer science. Even though we currently do not know of studies regarding the differences of computer sciences vs. non-computer sciences in developing software in specific domains, the findings for the medical device industry are similar to those in the automotive or embedded system industry. In all of these disciplines, which traditionally have more of a system engineering legacy, the number of computer scientists who are developing the software is relatively low while at the same time the issues with the produced software are relatively high.

Since the availability of computer scientists may not significantly increase in the near future, training programs tailored for software developers in this specific domain may be helpful. Such training programs need to be developed in a joint venture between medical experts, software engineering experts and (hardware) developers.

Activities related to the requirements phase are perceived as the major source of issues regarding software quality, which is thus a primary target for improvement activities. Focusing on this phase is also important given the fact that errors in this phase are often not detected until the very end of the development process. Hence, in order to fix these errors, it is necessary to go through some rework in all development phases, which is obviously more expensive then fixing errors that only affect one or maybe two development phases. In 50% of all cases requirements are managed with commercial standard applications (e.g., MS Word™ or MS Excel™). The use of natural language (instead of formal languages and notations) seems to go along

with these findings, since it is known that natural languages offer room for ambiguities (e.g., Kamsties 2001). Better integration of software engineering knowledge (i.e., methods, techniques, tools, and standards) into the requirement phases seems to be a promising way to save effort and thereby development cost without reducing the necessary quality of the developed software.

Because only about 50% of the organizations represented in this survey follow defined processes, it is not surprising that tools in general are used to a lesser extent than in other domains. Similar results hold for quality assurance activities. This is likely an indicator that verification and validation techniques commonly used in other domains, or process standards like SPICE and CMMI®, cannot simply be applied for this domain but need customization to be successfully integrated. This calls for actions on both sides, the medical device industry as well as in the field of software engineering. Besides the development of new, tailored solutions for the domain, additional training for the development teams seems to be a promising way. Integrating better guidance on how and when to use which tools could help to improve the existing standards.

Studies like the survey described in this paper can help to identify focuses for improvements and allow for goal-oriented actions to further integrate software engineering methods, techniques, tools, and standards into the medical device domain. For researchers interested in starting this work, a more complete set of survey results can be obtained by request to the authors. We are currently in the process of conducting further analysis on the collected data, which will include correlations between different variables (e.g., the developer's background and tool usage). Outcomes of these results will be made public as they become available.

Acknowledgements

We would like to thank the numerous participants of the survey for providing their feedback. The help of Andreas Schlichting from the Fraunhofer Institute for Experimental Software Engineering, Kaiserslautern, Germany, in summarizing the raw data sets is much appreciated

References

AdvaMed (2004). Future Trends in Medical Device Innovation. Advanced Medical Technologie Association, http://advamed.org. 2007.

BDI (2005) Wirtschaftsfaktor Gesundheit: Chancen und Potentziale für Deutschland. BDI-Initiative Vitale Gesellschaft, Bundesverband der Deutschen Industrie. http://www.vitalegesellschaft.de. 2007.

BFARM (2006). Bundesinstituts für Arzneimittel und Medizinprodukte. http://www.bfarm.de. 2007.

BMBF (2005). Studie zur Situation der Medizintechnik in Deutschland im internationalen Vergleich. Bundesministerium für Bildung und Forschung, http://www.bmbf.de. 2007.

CDRH (2002), General Principles of Software Validation; Final Guidance for Industry and FDA Staff. (see also http://www.fda.gov/cdrh/comp/guidance/938.html. 2007)

FTA (2007). Fault Trees And Reliability Block Diagrams. http://www.weibull.com/basics/fault-tree/index.htm. 2007.

HAZOP (2007). The HAZOP (Hazard and Operability) Method. http://www.acusafe.com/Hazard_Analysis/HAZOP_Te chnique.pdf. 2007.

IEC (2000). IEC 60601-1-4, Medical electrical equipment - Part 1-4: General requirements for safety - Collateral Standard: Programmable electrical medical systems. Ed. 1.1

ISO (2000). ISO 14971:2000(E) *Medical devices -- Application of risk management to medical devices,* Geneva, Switzerland. ISO.

ITEA (2005) 2 Blue Book, www.iteaoffice.org. 2007

Kamsties, E. (2001) *Surfacing Ambiguity in Natural Language Requirements.* PhD Thesis in Experimental Software Engineering, University of Kaiserslautern, Kaiserslautern, Germany. Fraunhofer IRB Verlag, Stuttgart, Germany.

Krasker, G.D. (2004). *Failure Modes And Effects Analysis: Building Safety into Everyday Practice*. HC Pro, Inc.

Wallace, D.R. & Kuhn, R. (2001). Failure modes in medical device: an analysis of 15 years of recall data. *International journal of reliability quality and safety* 8(4), pp. 351-373.

Paper II

Risk Identification by Physicians and Developers – Differences Investigated in a Controlled Experiment

C. Lindholm and M. Höst

Abstract

Risk management is an important process and risk identification is an important part of this process, especially in development of medical software. This paper presents an experiment where physicians, developers and software developers for medical devices are asked to identify risk in a given scenario describing the procurement of a patient monitoring system. It is concluded that multiple roles and thereby different experiences, will affect the risk identification process. Involving multiple roles, for example users and developers in the risk identification process, will result in a more complete set of identified risks than if only one role is included in the process.

Published in proceeding of the workshop of Software Engineering in Health Care (SEHC09), at ICSE, Vancouver, Canada, 2009, pp. 53-61.

1 Introduction

A small fault or mistake made in our daily life might not be so severe but in the health care domain the smallest mistake in development can make the difference between life and death. It is crucial that medical devices do not fail in any way. If they do, they can harm both the patients and the medical staff.

Physicians deal with risks as part of their work in different ways. Physicians working in the intensive care unit, the surgical ward or the emergency care unit comes in contact with a lot of different medical devices and these medical devices can also add different risk to the rest of the risks in the care situation.

All organisations that develop medical devices must have a risk management process according to law (European Council 1993). How strict and detailed this process must be depends on the safety criticality of the product. The same law as the medical device itself regulates all software included in the medical device.

It is crucial for all types of project planning and management to carry out risk management. It is important, as early as possible in the project, to identify all the relevant risks in order to avoid or minimise the effect of the potential problems.

Risk management is often performed in several steps e.g. as described by Hall (1998). A typical process for risk management includes risk identification, risk analysis, risk planning and risk monitoring. Relevant people identify risks during the risk identification and then the risks are prioritised with the respect to the probability of the risk actually occurring and the potential effect it will have if the risk occurs. According to Pfleeger (2000) the prioritising of risks are often decided through discussions where participants see risks in different ways and different values.

The research in this paper focuses on risk identification and is conducted through a controlled experiment, where physicians, developers and medical device developers have identified risks in a given risk scenario and also prioritised risks by giving them risk values based on estimated probability and effect.

The outline of this paper is as follows. After this introduction, related work is presented in Section 2 followed by a description of the experimental design in Section 3. The results are presented in Section 4

and discussed in Section 5. Finally the conclusions are summarised in Section 6.

2 Related work

This paper concerns research about the first step of the risk management process, i.e. the risk identification step. A basic assumption is that if multiple roles, and thereby different experiences, are involved in the identification activity, the resulting list of identified risks will be more complete than if only one role was included. There is not much research about the effects of including many roles in this step, but there is some research on the risk management process and on the risk identification step. For example, in (Li et al. 2008) risk management procedures from 133 projects were analysed in a recent survey, although the focus was not on the roles conducting risk identification.

However, in Kasap & Kaymak (2007) the risk identification step is analysed in some detail also with respect to the participating roles. 11 different techniques for identification are listed, and at least two are directly related to the question of what roles to include. Theses techniques are "brainstorming", where a group of experts are given the task to identify risks, and "cross functional teams", where the teams identifying risk are composed of different functional areas in he organisation. However, in Kasap & Kaymak (2007) no empirical evaluation of the performance of different types of teams are presented.

In Mojtahedi et al. (2008), a related issue is discussed. Instead of discussing what kind of different risks that are identified by different roles it is discussed how to handle that different roles give different priorities to different risks, i.e. that different persons give different assessments of risk probability and risk effect. This is related to how risk value of different risks is combined, although the focus of this paper is not as much on this part.

In Maytorena et al. (2007) it is reported from a study where 51 middle managers were interviewed based on a scenario concerning risk identification. It was found that there was no significant effect, neither of number of years in management role nor of number of years of current job title on risk identification performance. However, there was an effect of "predominant style of information search", e.g. in what way

information is sought and to what extent decisions are based on prior experience. This is obviously not the same as investigating different roles, although it concerns the question of what persons that actually do the risk identification.

Not much research is found about risk management and risk identification in specific in the medical device domain, even though all organisations developing medical device have to have a risk management process according to law. Ratkin (2006) states however that companies are required to have expertise in effective risk management practices, to be familiar with software safety and to be able to adopt a risk management mind-set.

3 Experiment design

The research is conducted through an experiment where the groups of physicians, software developers, and software developers specialised in development of software for medical devices are compared with respect to performance in risk identification. This experiment can a bit more formally be described as a "quasi experiment" (Robson 2002). A quasi-experimental design follows the experimental approach to design but does not involve random allocation of participants to different groups (Robson 2002). Since the subjects involved in this experiment are different categories of professional practitioners, i.e., physicians, developers and medical device developers, this is not possible to randomise over a sample of people.

That is, the independent variable of the experiment is the role, which can have three different values: physicians, software developer, and software developer specialised in development of software for medical devices.

3.1 Research questions

The objective of the research in this paper is to investigate if there is any difference between physicians who are users of systems and developers of systems regarding the view of risks. More specific research questions are:

- RQ1: Which difference can be identified between the numbers of risks identified by users and the number of risks identified by developers?

- RQ2: What are the differences between the kind of risks identified by users and the kind of risks identified by developers?
- RQ3: What are the differences between the groups with respect to which risks they see as important?
- RQ4: What risk overlap can be identified between the professional groups?

For research question RQ1, the dependent variable is the number of people from each role that have found the risk. That is, for each risk it is counted how many physicians that has found it, how many developers that has found it, etc.

For research question RQ2, the dependent variable is the kind of risks identified by the participants from each role related to the defined categories in Table 1 in Section 3.3. The number of risks for each category is counted in total and for each group of participants.

The differences between the groups with respect to which risks they see as important, research question RQ3, the dependent variable is the risk value assigned to the risks by the participants. The risks identified by all three groups have been analysed to find the risks most important for all the participating groups. In order to see the difference between the roles, the risks with highest risk values for each group was analysed and then compared.

In the analyse of research question RQ4 the risks in common for all the three participant groups have been used to identify what risk overlap that exists.

Research question RQ1 have been analysed with statistical test and research question RQ2-RQ4 have been analysed by descriptive statistics.

3.2 The experiment

The experiment was performed during the year of 2008, with 15 physicians, 15 developers and 6 medical device developers as participants. All the participants were presented to the same risk scenario describing the procurement of a patient monitoring system.

The involved subjects are professional developers and physicians working in Sweden. The physicians are all anaesthetists employed at the same clinic in a hospital in Sweden. As an anaesthetist they alter their work between three different units, the intensive care unit, the surgical

ward and the emergency care unit. Working in these three different units involves handling a lot of different medical devices. Before the risk scenario was sent out to the physicians an information meeting were held to explain the experiment and the purpose of the experiment. The risk scenario was not exposed to the physicians at this information meeting. A description of the risk scenario, a reply form and a self-addressed envelope were sent out to the 37 physicians by ordinary mail. The physicians were asked to send in their answers within 2 weeks. After 3 weeks a reminder with the same content was sent out. In total 15 physicians returned their answers, i.e., a reply rate of 15/37 = 40%.

The developers were asked if they would like to participate in the experiment by e-mail. This e-mail was sent out to developers in different Swedish companies developing software and in the e-mail the experiment and the purpose of the experiment was explained. The developers were addressed directly and e-mail was sent out to 26 developers and 15 accepted to take part in the experiment. That is the reply rate of the developers was 15/26 = 58%.

The same risk scenario as the one sent to the physicians was sent to the developers with a minor difference, that some of the medical terms was explained. The risk scenario and reply form was sent by e-mail and the developer returned the reply forms after answering, by e-mail. The most difficult group to get participants from was the medical device developers. Despite 32 different information e-mails directly to individuals and different companies developing medical devices in Sweden and several telephone calls only 6 participants participated in the experiment. This means that the reply rate of this group was 6/32 = 19%. The major reason given for not participating in the experiment was lack of time. The risk scenario and reply form that was sent out to the medical device developers by e-mail were identically as the one sent to the developers. The reply forms received from the medical device developers were also returned by e-mail.

The risk scenario is around 1¼ page long and is written in Swedish describing the procurement of a patient monitoring system. When the risk scenario was written some things considered as risks by the researchers was deliberately incorporated in the scenario but no too obvious risks were chosen. The risk scenario describes the following scenario:

”It is the county council that have decided to buy a new patient monitoring system with the intention to have the same monitoring system at all the units at the hospital, for example at the intensive care unit, the surgical ward and the emergency care unit. The goal for the county council is to rationalise the care and reduce cost for the daily activity. The county council have requested 10 companies for tender and the tender text is presented in the scenario. The tender text is divided in four parts; Dimensioning and functionality, Location, presentation and compatibility, Mobile monitoring and Communication and use. The system shall, for example, have one central server, it shall be possible to monitor 22 patients through wireless communication, and the functionality that the system at least shall include is specified. It shall be possible to use the new system together with other medical devices, the new system must guarantee patient safety, it shall be possible to easy physically transport the patient together with the system, the system should be able to communicate with other wards and external partners, a user should be able to use the system after five hours of training etc. That is, the scenario includes both functional and non-functional requirements. The risk scenario ends with a description of the company the county council have decided to give the contract to. This company is a new, promising and expanding company on the medical device market and they promises to deliver the system in 6 month to the lowest cost. The majority of the staff is developers that recently have taken their degree. That is, the scenario also includes information about the development organisation and a description of the experience and competences of the developers”.

All the participant in the experiment were asked to study the risk scenario, identify risks and write down the risks in their own words (i.e. "free text form") in the reply form. The participants were also asked to estimate the probability of the risk and to estimate the effect of the risk. For the probability, the participants were given a graded scale 1 – 4, where 1 represents that it is very unlikely for the risk to occur, 2 represents that it is unlikely, 3 that is likely and finally 4 represents that it is very likely the risk will occur. A scale was also given for the estimate of the effect if the risk occurs. This scale was graded 1-5 where 1 represents that the effect would be insignificant if the risk would occur 2 that the effect would be acceptable, 3 that the effect would be serious,

4 that the effect would be very serious and 5 the effect would be catastrophic if the risk would occur.

In the analysis the researcher then calculated the risk value, *R*, for each given risk by multiplying the given figure for probability, *P*, by the given figure for effect, *E*, as R = P x E. Thus, the highest possible risk value a risk can get in this experiment is R = 4 x 5 = 20

3.3 Analysis

The data was collected from the reply forms that the participants sent in. All the risks were put together in a digital format. Each risk was given a standardised risk description and a risk identifier by the researcher. The risk with the same content and meaning was counted but registered as the same risk with same risk identifier and risk description.

Each risk was then categorised according to the type of risk in 15 categories shown in Table 1.

These categories have been defined based on both the researchers assumptions and intentions about the present risks in the scenario, and based on the risks that actually were identified by the participants. That is, the identified risks were allowed to affect the categories of risks.

The risk values for each risk and the professional groups were also registered and analysed. The experiment data was analysed with descriptive statistics and statistical tests. Research question RQ1 was analysed with statistical tests, and research questions RQ2-RQ4 were analysed with descriptive statistics.

Table 1. The risk categories

Category	Example of risk
Education	Too short education of the users before using the new system
Delivery time	Too short time to delivery (6 month)
Support	Upgrading of the new system in the future
Cost	The budget is out of control
Alarm	The alarm do not work as intended
Wireless transmission	The wireless transmission do not work as intended
Back up	One central server is not enough
Requirement specification	Vague requirement specification
Security	The security is at risk if there is communication with extern partners
Experience	The developers that recently passed their degree have none or slight experience in developing this kind of system
Company	Because it is a new company there is a risk that the company goes bankrupt
Bidding procedure	Vague tender
Introducing the new system	The new and old system will not run in parallel. The old system is shut down when the new system is installed
Problem with the new system	The new system give the wrong information
Development process risks	The customer does not have time to participate in the process

Research question RQ1 was analysed by comparing how many of the people from every role that identified every specific risk. That is, the following metric was calculated:

$$M_{Role,Risk} = \frac{F_{Role,Risk}}{N_{Role}}$$

where $F_{Role,Risk}$ denotes the number of individuals from a role that found a specific Risk, and N_{Role} denotes how many people there are available from that role, i.e. how many that could have found the risk.

This means that the relative number of people from each group is compared, which is necessary since there are fewer physicians than developers.

This data can be analysed with a repeated measures analysis of variance (ANOVA) design with the following model:

$$M_{Role,Risk} = m + a_{Role} + b_{Risk} + c_{Role,Risk,}$$

where $c_{Role,Risk}$ is an error term which is normally distributed with mean 0. In this model, m represents the overall mean value, a represents the effect of the roles, i.e. that roles do not have to be equally effective in identifying risks, and b represents the effect of the risks, i.e. the fact that all risks are not equally hard to identify. This means that it is possible to test the null hypothesis

$H0$: $a_{Role} = 0$, for all roles,

i.e. that there is no effect of role on the number of identified risk. That is, if it is possible to show that a_{Role} is not 0 (i.e. a_{Role} is not the same for all roles) it is shown that all roles are not equally effective in identifying risks. A repeated measures design was chosen because it takes into account the fact that all risks are not equally hard to identify. For more information about this kind of model and analysis, refer e.g. to Montgomery (2001) or Dalgaard (2002). The data cannot be assumed to be normally distributed so a non-parametric statistical test, i.e. the Friedman test Dalgaard (2002), which is a non-parametric alternative to the above described analysis, was also applied.

3.4 Validity

A major concern in quasi-experiments is the threats to validity. The interpretation of findings is more complex than "true" experimental design according to Robson (2002). It is important to consider the validity threats already during the design of the experiment, so the validity threats can be reduced as much as possible. Validity refers to accuracy of results and validity threats may be dived into four types (Wohlin et al. 2000). These four types are conclusion validity, construct validity, internal validity and external validity.

Conclusion validity is related to the possibilities to draw the correct conclusions regarding the relationship between the independent and dependent variables of the experiment. Threats to conclusion validity can be the use of wrong statistical tests, for example using statistical tests where the statistical power of the tests are to low. To reduce this threat the choice of statistical test have been made very carefully and for example analysis with non-parametric test has also been done.

Construct validity is concerned with the getting the correct measures for the concept and the relationship between the concepts and theories behind the experiment. A threat can be that all the participants do not interpret risk the same way as the researchers intended. The scales for estimating the probability and the effect of the risks can also be interpreted in different ways even if the intention from the researchers has been to make the instructions and the scales as unambiguous as possible. In order to try to mitigate construct validity the instructions and scales were written and defined with the up most intension to be as clear and unambiguous as possible. Two developers and one physician whom not participated in the experiment reviewed and commented the instructions, scales and scenario before use.

Internal validity is affected by factors that affect the measures but that are outside the control of the researchers. A threat to this study and an affecting factor can for example be that the participants read and filled in the replay form under different conditions that was out of the researchers control. We have, however, not seen any signs of this. Another common threat with respect to this is that the participants in different groups have different experience. The physicians have, as it is described in Section 4.1, somewhat longer experience in average than the other two roles. Even if this is a threat to the study, we do not see it as too serious. This distribution could in the future be compared to the typical distribution of the roles, i.e. physicians and software developers, although this has not yet been done.

External validity primarily relates to how general the result of the experiment is for example how representative the problem in the assignment is. Another threat could also be that the participants is not representative of the target population, so to lower this threat in the experiment the subjects asked to participate in the experiment is working as professional developers or physicians. However the physicians have the same specialty and practice in the same clinic at the

same hospital, this can be a threat to how general the results are. In further experiments physicians from different hospitals should be included. All the participants may not have done this kind of risk analysis before or been in contact with this kind of scenario before but however all participants are used to dealing with risks and the scenario is based on a scenario they quite likely could be confronted with. A threat can though be that ca 1/3 of the developers in this experiment have taken a degree at Lund University where courses containing risk analysis have been taught. No checklist for risks was introduced to the participants but this was a deliberate choice of the researches with the intension not to control the participants. However there were no restrictions against using one. The risk scenario itself is a threat because the scenario is fictional but written with the up most intension to be as realistic as possible. Before the use of the scenario, it has been reviewed and commented on by one physician and two developers; none of them have taken part as participants in the experiment.

4. Results

4.1 Results from the controlled experiment

The experiment was preformed by 36 participants, 15 physicians, 15 developers and 6 medical device developers. All the participants were asked to specify how many years they have been working in their profession. They were asked to mark "< 2 years", "2-5 years", or "> 5 years". There are differences in the three groups as can be seen in Table 2, and which is also discussed in Section 3.3. The majority of the physicians and the medical device developers in this experiment have been working in their professions for more than 5 years while the majority of the developers have been working for 2-5 years in their profession.

Table 2. Working years in profession

Year in profession	Physicians	Developers	Medical device developers
< 2	7 %	33 %	33 %
2-5	0	40 %	17 %
>5	93 %	27 %	50 %

The participants have also stated how many minutes they have used for the experiment. The time in average in the different groups varies, and not all the participants have answered this question. 12 of the 15 physicians have spent in average 26 minutes per person on the experiment where 45 minutes is the longest time spent and 5 minutes the shortest time spent. 14 of the 15 developers spend in average 1 hour and 23 minutes per person where the longest time spent is 6 hours and the shortest time spent is 30 minutes. The average time for the third group, medical device developers is 1 hour and 2 minutes per person. 5 of the 6 medical device developers answered and time varied between longest time spent 150 minutes and shortest time spent 20 minutes.

All the risks have been analysed and categorised and the risks that appeared to be the same risk have been counted and registered together. This has resulted in 197 specified risks with a unique identifier and risk description given by the researcher. Out of these 197 risks there are 54 risks that are stated by only one of the groups (risks unique for the group). Developers have identified more risks per person than the other groups as shown in Figure 1. Also here it should be taken in count that the medical device developers are a smaller group.

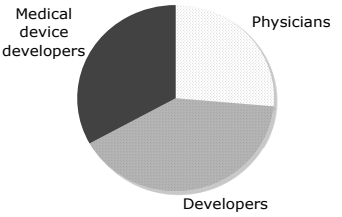

Figure 1. Number of risk/person

The total number of risks identified in this experiment is 390 risks distributed as shown in Table 3.

Table 3. Number of risks

Professional group	Members	Number of risks	Number of risks/person
Physicians	15	130	8.7
Developers	15	200	13.3
Medical developers	6	60	10.0
Total	**36**	**390**	**10.8**

The developers are the group that has identified the largest amount of risks followed by the physicians and medical device developers. However the group with medical device developers is a much smaller group than the other two groups so it could be expected that they should identify less amount of risks. Therefore looking at the number of risks found per person could be more interesting and it can be seen in Table 3 and Figure 1 that the developers found the largest amount of risks per person, followed by the medical device developers and smallest amount of risks per persons was found by the physicians. One possible reason that the developers found more risks per persons can be they spent more time on their risk identification process than the physicians.

Concerning research question RQ1, the resulting values of metric *M,* as described in Section 3.3, for each risk and role is displayed in Figure 2.

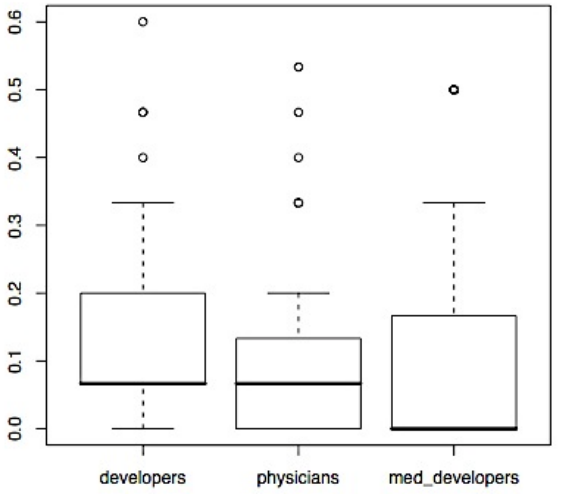

Figure 2. Box-plots of relative number of people identifying each risk (*M*).

Some differences between the groups can be found and the result of an analysis of variance is that there is a significant effect of the role on *M*. The resulting ANOVA-table is depicted in Table 4.

Table 4. Result of ANOVA-test for RQ1.

	Df	F-value	Pr(>F)
Role	2	7.4014	0.0007747 ***
Risk	110	2.9937	
Residuals	220		

A Friedman test for the same hypothesis gives a p-value of 4.785e-06, which means that this too indicates a significant difference.

An analysis of the difference between only physicians and developers also gives a significant difference (p-values are 1.23e-05 for a paired t-test and 1.13e-06 for a Wilcoxon test, which is a non-parametric alternative). That is, there is not only a difference between physicians and developers (from Figure 2 and the ANOVA-test) but there is also difference between physicians and medical device developers.

In order to investigate research question RQ2 and the differences between the kind of risks identified by physician and the kind of risks identified by developers and medical device developers each risk has been categorised according to type of risk in the 15 categories shown earlier in Table 1. The category with the highest number of different risks are “Problem with the new system” it contains 41 different risks with unique risk identifier, these risks relate to problems with the system when it has been delivered and are up and running. “The system gives the wrong information”, “The safety for the patient reduces” and “The hospitals paging system is “knocked out”” is example of risks in this category. All three participant groups have the highest number of different risks with unique risk identifiers in this category. The developers have identified 34 of the 41 risks listed in this category, the physicians 21 and the medical device developers 15 of the listed risks in this category.

The two categories that then follow with 10 different risks in total with unique risk identifiers in each category are “Wireless transmission” and “Development process risks”. In the category “Wireless

transmission" the developers have identified 9 of the 10 risks and the physicians and medical device developers 5 of the 10 risks. When it comes to "Development process risks" the physicians have not identified any development process risks at all but the developers have identified 9 out of the 10 risks and the medical device developers 2 out of the 10 risks.

In order to survey research question RQ3 regarding the differences between the groups with respect to which risks they see as important, all the participants have, and as explained in Section 3.2, given each risk they have identified a risk value, *R*. The highest risk value a risk can receive is $R = 20$. When analysing the 29 risks common for all the three groups a total risk value for each of these risks was calculated by summarising the average risk value given to these particular risks from each group of professionals. Table 5 shows the four risks with the highest total risk value for the risks in common for all the three groups.

Table 5. Total risk value

Risk	Phys.	Dev.	Med dev.	Total risk value
The new and old system does not run in parallel	17.5	16	20	53.5
The delivery of the new system is delayed	20	13.8	13.3	47.1
One central server is not enough	17.5	15.2	13	45.7
The company goes bankrupt	14	10.3	20	44.3

Of the 29 risks in common, 9 belong to the category "Problem with the new system" and 4 belong to the category "Support".

The three different professional groups give different risk value to the different risks. A top ten risk list out of the 197 risks for a physician is not the same top ten lists as for a developer or for a medical device developer. The top four risks for each group are presented in Table 6.

The risks identified and given the highest risk value by the physicians are risks that are not identified at all by the other two groups. Also one of the risks given the highest risk value by the developers is a unique risk, a risk only identified by the developers. None of the risks identified by medical device developers and given high risk value is

unique for the medical device developers as a group however, have the medical device developers given these risks higher risk value than the other two groups. The medical device developers have also a larger amount of risks with the highest risk value, 20, than the other two groups.

Table 6. Top four risks

Physicians Risks	Phys.	Dev.	Med dev.
Vague requirement specification	20	0	0
Old documents is lost	20	0	0
The hospitals paging system is "knocked out"	20	0	0
One central server is not enough	17.5	15.2	13
Developers Risks	Dev.	Phys.	Med dev.
Upgrade of the new system in the future	20	0	0
Problem getting the new system running	20	12	6
One central server is not enough	15.2	17.5	13
No delivery of the system at all	15	0	0
Medical Developers Risks	Med dev.	Phys.	Dev.
The wireless transmission put other units out of order	20	0	4
Vague tender gives low quality	20	0	12
The company goes bankrupt	20	14	10.3
The new and old system does not run in parallel	20	17.5	16

Regarding research question RQ4 and if their is any risk overlap between the groups it can be seen that there are 29 risks out of the 197 risks (15%) that are common to all the three groups, this means that the risk have been identified by at least one participant in each group of professionals.

If the groups are merged and studied, the group of physicians and medical device developers as one group and developers and medical device developers as another group give us two groups equal in size. It was found that developers and medical device developers have 8 risks that are in common for these two groups only, compared to physicians and medical device developers they have only one risk in common exclusive for them. Physicians and developers have the largest amount of risks, 19 risks, in common but this could be explained with that it is the largest group of participants (30).

5 Discussion

Risks and risk management is an important area. It is crucial for all types of project planning and management to perform risk management. All organisations developing medical devices are obliged by law to have a risk management process. Risks are also something that affects physicians whom deals with risks in all care situations. Risks are therefore a major concern for all of the professional groups participating in this experiment.

There is a difference regarding the view of risks between physicians, developers and medical device developers shown in this experiment. Looking at the number of identified risks, developers identified a larger amount of risk per person than physicians. A possible reason for this can be that developers are more used to the process of identifying risks in this way, presented in the experiment than the physicians.

Often most developers have sometimes worked in projects, which make it a familiar working form for them but probably not for physicians. This could explain that all the three groups identified similar types of risks with the exception that none of the physicians identified any development process risks at all.

It was shown that the physicians did not identify any risks that are typical medical risks so the developers could also have identified all risks identified by physicians. The three risks the physicians gave the highest

risk value was not identified by the other two groups but they easily could have. In this experiment no checklist for risks have been used in order not to control the participants. It could be interesting to do this experiment with a checklist and see if it has any effect on the result.

6 Conclusion

The research in this paper presents an experiment where physicians, developers and medical device developers are asked to identify risk in a given scenario in order to investigate if there is any difference regarding the view of risks. Our basic assumption is that multiple roles, and thereby different experiences, will affect the list of identified risks and that it will be more complete than if only one role is included.

It can be concluded that there is a difference between the different professional groups regarding the view of risks in this research study. The different experiences affect the risk identification and also the prioritisation of risks. There is a difference in the number of people from each role that has found a risk and there is a difference in between the groups with respect to which risks they see as important. However there is no distinct difference in the kind of risk identified by the participant groups with one exception that the physicians have not identified any risks that could be categorised in the development process risk category.

The risk overlap between the participant groups are rather small and given that the results can be replicated and generalised we can conclude that it is important to include participants from different professional groups in the risk identification process.

It can be concluded that it is necessary, at least for this kind of system, to include the users in the risk identification process in order to get more complete risk identification. It is not sufficient to only include the developing organisation in identification of risks. Involving different roles in risk identification may probably be advantageous in several types of systems.

The researcher presented in this paper has indicated the necessity of incorporate the user in the risk identification process. Furthers research could involve the development of practical guidelines checklists and workshop processes for medical device industry and the goal in the long

run should be to influence and contribute to standards as for example IEC 62304 and ISO 13485.

References

Dalgaard, *Introductory Statistics with R*. New York: Springer.

European Council (1993). Council Directive 93/42/EEC Concerning medical devices. Luxembourg, Official Journal of the Communities.

Hall, E. M. (1998). *Managing risk: Methods for software systems development*. Reading: Addison Wesley.

Kasap, D. & Kaymak, M. (2007). Risk Identification Step of the Project Risk Management, In *Proceeding of the International conference on management of engineering & technology*, pp. 2116-2120.

Li, J., Conradi, R., Slyngstad, O. P. N., Torchaniano, M., Morisio, M. & Bunse, C. (2008) A State-of-the-Practice Survey of Risk Mangement in Development with Off-the-Shelf Software Components, *IEEE Trans on Software Engineering*, 34, (2), pp. 271-286.

Maytorena, E., Winch, G.M., Freeman, J. & Kiely, T. (2007). The influence of experience and information search styles on project risk identification performance. *IEEE Trans on Engineering Management*, 54 (2), pp. 315-326.

Mojtahedi, S.M.H., Mousavi, S.M. & Makoui, A. (2008). Risk identification and analysis concurrently: Group decision making approach. In *Proceeding of the 4th IEEE International conference on management of innovation and technology*, pp. 499-507.

Montgomery, D. (2001) *Design and Analysis of Experiments*. Wiley,

Pfleeger, S.L. (2000) Risky Business: What We Have Yet to Learn About Risk Management, *Journal of Systems and Software*, 53, pp. 265-273.

Rakitin, S. R. (2006). Coping with defective software in medical devices. *IEEE Computer*, 39(4), pp. 40–45.

Robson, C. (2002). *Real world research* (2nd ed.). Oxford UK: Blackwell Publishers.

Wohlin, C., Runeson, P., Höst, M.; Ohlsson, M.C., Regnell, B., & Wesslén A. (2000). *Experimentation in Software Engineering: An introduction.* Boston: Kluwer Academic.

Paper III

Different Conceptions in Software Project Risk Assessment

M. Höst and C. Lindholm

Abstract

During software project risk management, a number of decisions are taken based on discussions and subjective opinions about the importance of identified risks. In this paper, different people's opinions about the importance of identified risks are investigated in a controlled experiment through the use of utility functions. Engineering students participated as subjects in the experiment. Differences have been found with respect to the perceived importance, although the experiment could not explain the differences based on undertaken role in a development course.

Published in proceeding of the Software engineering track at the 22:nd Annual ACM Symposium on Applied Computing (SAC), Seoul, Korea, 2007, pp. 1422-1426.

1 Introduction

During project planning and management, procedures for risk management are crucial. This is, for example, acknowledged by the presence of risk management issues at level 3 in the Software Engineering Institute's Capability Maturity Model (e.g. CMMI 2002). The objective of risk management is to identify relevant risks as early as possible in a project, in order to avoid or limit the effect of potential problems, such as project delays and cost overruns. More formally, risk management can be defined as "an organized process for identifying and handling risk factors; including initial identification and handling of risk factors as well as continuous risk management (Fairley 2005). Safety critical projects include, as all other projects, a large amount of software. When it comes to risks that are related to the product, e.g., the number of persistent faults in the product, they are very important for two reasons. One reason is that it is important to identify these as early as possible in order to secure the quality of the developed product. The second reason is that it is important to limit the number of problems during the project even if the quality of the product with respect to the number of dormant faults is acceptable when the product is delivered. This is because a large amount of changes during a project deteriorates the structure of the code, which results in new faults later on.

Risk management is often carried out in a number of steps, e.g.: risk identification, risk analysis, risk planning, and risk monitoring (Sommerville 2004). During risk identification, risks are identified by relevant people, e.g. by using checklists and brainstorming techniques. The identified risks are prioritized with respect to their probability of actually occurring in the project and their potential impact. The risks that are expected to have both high probability and large unwanted effects are the most important risks to continue to work with in the process. In the risk-planning step, plans are made in order to either lower the effects of the prioritized risk, lower their probability, or to prepare for what to do if they actually occur. In the monitoring step, the risks are monitored during the course of the project. There are, of course, no clear and objective rules available for how to prioritize the identified risks in the second step. This is instead carried out through discussions and subjective evaluations, where participants have different

values and see the risks in different ways (Pfleeger 2000). This means that it is important to investigate how large differences there are between different participants, and whether it is possible to explain identified differences.

Utility functions (e.g. Wakker & Deneffe 1996) describe how different people value a property. For example, a utility function could describe how people value the expected life-duration after different alternative medical treatments. If the utility function is linear, a life-duration of $2x$ years would be perceived as twice as good as a life-duration of x years. The utility function does, however, not have to be linear, which affects how people make decisions when choosing different treatments. Based on the shape of the utility function it is possible to discuss whether different individuals act as risk-averse, i.e. they tend to avoid risks and choose a lower safe gain, or risk-seeking, i.e. seeking a possible high gain instead of a more certain lower gain.

2 The utility function

2.1 The Trade-off method

The objective of the Trade-off (TO) method is to estimate the utility function for one person. According to the TO method (Wakker & Deneffe 1996), the subject is iteratively asked to compare different "lotteries". A lottery is shown graphically in Figure 1, which should be interpreted as that one of two events (event 1 and event 2) will occur If the probability of event 1 is p, then the probability of event 2 is $1-p$. If event 1 occurs this will result in result 1 and if event 2 occurs this will result in result 2.

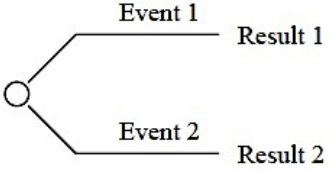

Figure 1: A lottery

An example of possible values in the lottery is shown in Table 1.

Table1: An example of a lottery.

Property	Meaning
Event 1	Design expert NN is unable to follow the project
Event 2	Design expert NN is able to follow the project
Result 1	There will be 10 faults at the acceptance test
Result 2	There will be 3 faults at the acceptance test

In the TO method participants should iteratively compare pairs of lotteries. An example of a pair of lotteries is shown in Figure 2. The upper lottery shows what could happen if one condition is true (an old design is chosen) and the lower shows what could happen if another condition is true (a new design is chosen). The probabilities of the events are assumed to be independents of the conditions, i.e. the probability that design expert NN will be able to participate in the project is the same in the two lotteries. An advantage of the TO-method compared to other methods for eliciting utility functions is that the value of the probability need not be explained to the person using the method.

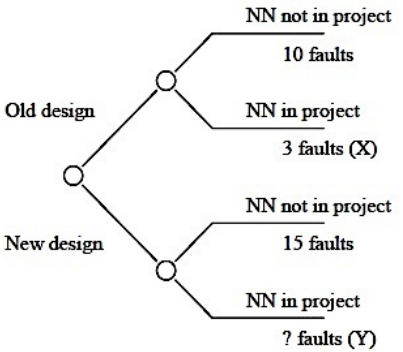

Figure 2: A pair of lotteries.

In the TO method the subject is first asked to select a value of the number of faults in acceptance test in the second lottery (Y in Figure 2) that makes the two lotteries equally attractable. When this has been done the subject is asked to compare two new lotteries. These two lotteries are similar to the first two lotteries, but with value X (see Figure 2) changed to the value that the subject chose in the last question. The subject is now asked to give a new value of Y that makes

these two lotteries equally attractable. This process is iterated in order to give values of the utility function for the result factor.

If the X-value in the first comparison is called x_0, the first Y-value is called x_1, the second Y-value called x_2, etc., then it can be shown that the utility function u, can be estimated as (Wakker & Deneffe 1996),

$$u(x_i) = i \times u(x_1)$$

which can be normalized to $u(x_i)=i \times a$ where $a = 1/n$, and n is the number of Y-values given by the subject. The proof for this is not provided in this paper; instead the reader is referred to (Wakker & Deneffe 1996). In Figure 3 a hypothetical example of a utility function is shown. This example shows a concave curve, i.e. $x_i - x_{i-1} < x_{i+1} - x_i$, $1 < i < n$. Curves can also be linear ($x_i - x_{i-1} = x_{i+1} - x_i$), convex ($x_i - x_{i-1} > x_{i+1} - x_i$), or a combination of these shapes.

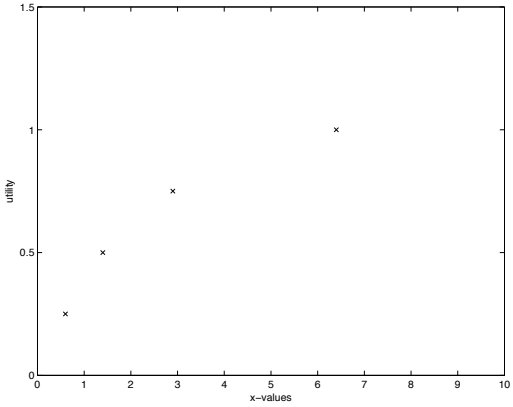

Figure 3: An example of a (concave) utility function.

2.2 Interpretation of utility functions

In most cases utility functions describe properties where a large number is better than a low number (e.g. monetary gain). The factors that are considered in software risk assessment often refer to negative aspects and not to positive aspects. For example, factors such as number of remaining faults, delay, etc. are analysed instead of positive factors such as revenue, life-duration, etc. In Fennema (1999) the typical shape of utility functions for losses is discussed.

If the utility function e.g. for the remaining number of faults is concave (i.e. as in Figure 3) this means that relatively the effect of every

fault is higher if there are few faults than if there are many faults. This means that a person with this interpretation thinks that $2x$ faults is less than twice as serious than if there are x faults. If this person would choose between a fixed value x and a lottery with value 0 with probability 1/2 and value $2x$ with probability 1/2, this person would probably choose the lottery since the expected utility value of the lottery is lower than for the fixed value x. Since this person chooses the lottery instead of the fixed value, we say that a person with a concave utility function is risk seeking. If the function is convex, the value of every fault is higher if there are many faults compared to if there are few faults. This means that a person with a convex utility function is risk averse.

Imagine a situation where a person should compare two different alternative ways of handling a risk in a project. Based on subjective evaluations it might be estimated that one of the alternatives will results in a certain expected number of remaining faults and the other alternative will results in a higher number of expected faults. In this case a person with a concave utility function would probably not see the second alternative as negative as a person with a convex utility function. This will of course affect how different people act during discussion on risk evaluation during risk management. It is therefore interesting to investigate how different individual utility functions for this type of properties are.

3 The experiment

3.1 Objectives

The objective of the research presented in this paper is to investigate the shape of utility functions for factors that are relevant in software project risk management. More specifically, the research questions are as follows:

- RQ1: What is the distribution between convex, concave and linear utility functions for properties that are relevant in software project risk assessment?
- RQ2: Is there any difference between different roles in a project with respect to the shape of the utility functions?

RQ3: Is there any difference between the shapes of the utility functions for normal projects and projects developing safety-critical products.

3.2 Experiment subjects, objects, and context

The research questions are investigated in an experiment where students act as subjects. The experiment was conducted as part of a software engineering project course given at Lund University during the spring of 2005. The students followed programmes in Computer Science, Software Engineering, Electrical Engineering, and Multimedia. The course is attended in the 2:nd year of their university studies.

The course is a project-course where the students work in projects of typically 17 persons in each project. All projects are given the assignment of implementing a number of services for a basic telephone switching system. In the beginning of the course the students are given a basic version of the system where only basic functions such as providing simple telephone calls, managing what happens if the called party is already involved in a telephone call, etc. are provided. Their assignment is to develop more advanced services such as call forwarding, billing, etc. The project group should follow a software development process based on the waterfall model with steps such as project planning, requirements engineering, implementation, and testing. This experiment was conducted during the test-phase of the project, i.e. after the project planning was carried out. In every project groups the students are divided into the following roles: Project leaders (PL), Technical responsibility (TR), Developers (D), and Testers (T).

The experiment was conducted during a seminar where all students participated. At the seminar the seminar-leader first held a lecture on risk management, and then the students carried out the tasks of the experiment.

In the experiment the utility function of every student was elicited with the TO-method. The students were presented with two scenarios (scenario 1 and scenario 2). Scenario 1 is based on the project assignment in the course (translated from Swedish to English):

Assume that there was a design expert (NN) in your project that could decide the design. NN is part of the "technical responsibility"-group of your project and NN has some new ideas about the design that are not exactly as the teachers in the course have thought. The design proposed by NN is called "new design" and the ordinary design, as proposed by the teachers is called

"old design". Based on experience data, the project leaders estimate that there will be a certain number of faults remaining in the product at the acceptance test.

Consider the following four cases:

Case 1A: The old design is used and NN is able to participate in the project. Then there will be 5 faults at the acceptance test.

Case 1B: The old design is used and NN is unable to participate in the project due to illness. Then there will be 6 faults at the acceptance test.

Case 2A: The new design is used and NN is able to participate in the project. Then there will be 2 faults at the acceptance test.

Case 2B: The new design is used and NN is unable to participate in the project due to illness. How many faults can there be at the acceptance meeting if the two designs should be equally attractable?

Scenario 2 is based on another system than they worked with in the course. It describes instead a safety critical system and was presented as follows (translated from Swedish to English):

In an intensive care unit you have surveillance equipment connected to the patient that monitors the patient condition. Different values is continuously registered, such as patient's absorption of oxygen, cardiac activity etc. The values are analysed by software in the surveillance equipment. The surveillance equipment sends an alarm if the analysed values in any way differ form the normal values. If no attention is taken to the abnormal values (i.e. absence of alarm) it can cause severe injury to the patient and in some cases even death. There is a great risk for serious damage if the alarm fails. The personnel need proper training to be able to connect and manage the surveillance equipment correct. Most of the personnel have this type of training, but some times they do not have the training, due to lack of time. If the surveillance equipment is connected the wrong way there is a risk for absence of alarm and the patient are exposed to danger. Now the intention is to try out new software in the surveillance equipment. Consider the following four cases:

Case 1A: Present software is used. The personal are trained on the surveillance equipment. At 7 occasions in a three-month period, there was absence of alarm from the surveillance equipment, despite the fact that there should have been alarms.

Case 1B: Present software is used. In this case personnel who have not received proper training on the equipment use the equipment. At 9 occasions in a three-month period, there was absence of alarm from the surveillance equipment, despite the fact that there should have been alarms.

Case 2A: New software is used. The personal are trained on the surveillance equipment. At 4 occasions in a three-month period, there was absence of alarm from the surveillance equipment, despite the fact that there should have been alarms.

Case 2B: New software is used. In this case personnel who have not received proper training on the equipment use the equipment. How many alarms can be missed if the new software should be equally attractable?

In the TO-method the questions that are asked to the subject should, as it is described in Section 2.1, be based on the previous answer given by the subject. For example, if the subject answered "250" in the last round, then "250" should be one of the results that should compared to in the next round. This means that it is hard to use the TO-method based on completely pre-developed and parameterized instrumentation, e.g. paper forms. For the purpose of this research, a simple tool was developed, see Figure 4. From the screen-shot it can be seen that the appearance of the tool was not identical to the questionnaire that is described in (Wakker & Deneffe 1996), where a decision tree (e.g. Figure 2) was graphically presented to the subjects.

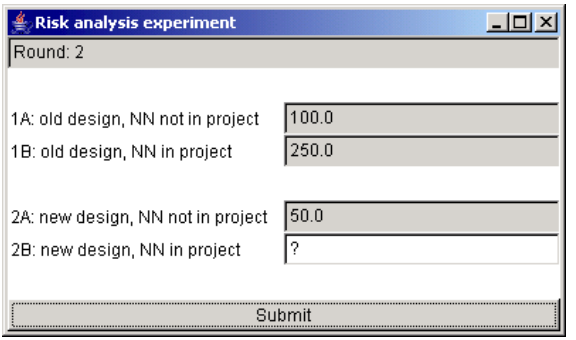

Figure 4. A simple tool, screen for round 2 after answering "250" in round 1.

3.3 Experiment design

All students first worked with scenario 1 and after that with scenario 2. In the analysis the results from each student is characterized as concave, convex, linear or "other". A curve is classified as "other" if it has not the same shape (convex or concave) for all x-values, e.g. the first half of the curve is convex and the second half is concave. In order to investigate research question RQ1 the data from all students are pooled and the number of curves of each shape is analysed.

In order to investigate research question RQ2 the role in project was chosen as independent variable and the number of curves of each shape was chosen as dependent variable. In order to investigate research question RQ3 the scenario was chosen as independent variable and the number of curves of each shape was chosen as dependent variable.

3.4 Validity

In order to evaluate the validity of the study, a checklist from (Wohlin et al. 2000) is used. Validity threats may be classified as conclusion validity, construct validity, internal validity, and external validity.

The *conclusion validity* is related to the possibilities to draw correct conclusions about relations between the independent and dependent variables of the experiment. Typical threats of this type are, for example, to use wrong statistical tests, to use statistical tests with too low power, or to obtain significant differences by measuring too many dependent variables ("fishing and the error rate"). Since there were only moderately many participants in the study, care must be taken when it

is stated that no difference between two groups are found. It can only indicate that there is no difference, which is further discussed in Section 4.

The *internal validity* is affected by confounding factors that affect the measured values outside the control, or knowledge, of the researcher. This may, for example, be that the groups of subjects carried out their assignments under different conditions, or maturation of participants. In order to lower the internal threats in this experiment all students carried out the assignment the same time during a 90 minutes seminar when one of the researchers was present. One threat to this study is that the two scenarios were analysed in the same order by all students. This should be taken into account when the difference between the scenarios is analysed, i.e. when RQ3 is analysed. The reason for letting every participant work with the scenarios in the same order was that it was seen as positive that the students started with a scenario that presents a familiar project and system.

Threats to *construct validity* denote the relation between the concepts and theories behind the experiment, and the measurements and treatments that were analysed. We have not identified any serious threats of this kind.

The *external validity* reflects primarily how general the results are with respect to the subject population and the experiment object. The intention is that the subjects in this experiment should be representative of engineers working with this type of estimation in live projects. As we see it, the largest threat to validity is of this kind. It cannot be concluded with any large validity that the students that participated in this experiment are representative of professional practitioners. Scenario 2 is not in any way related to the students' course work, but scenario 1 was based on the projects that the students participated in the course. However, the scenario was still a hypothetical scenario and it was studied in the testing phase of the project, i.e. after the risk assessment in a real project.

4 Results and analysis

The experiment was conducted with 47 students, but one of them did not hand in any results, which means that there were 46 students that completed the tasks. The number of subjects that completed scenario 1 was 44, since 2 of the subjects were discarded because the scenario was

only iterated three times. The minimum of iterations was set to four times. In scenario 2, 3 subjects were discarded for the same reason so the number of subjects that retained for further analysis was 43.

Table 2. Distribution of utility functions

	Concave	Convex	Linear	Other
Scen.1	20 % (9)	32 % (14)	30 % (13)	18 % (8)
Scen.2	5 % (2)	23 % (10)	58 % (25)	14 % (6)

In order to investigate research question RQ1 the distribution of utility functions were analysed. The distribution between concave, convex, linear and other utility functions for the two scenarios are displayed in Table 2. The result is presented in percent of the total number of subject for each scenario, and in absolute figures in parenthesis. The students had different roles in their project groups. There is a difference in the number of students connected to the various roles. The largest group were developers (18 students) and the smallest group were project leaders (6 students). The values for each role and type of utility function are presented in Table 3.

Table 3. Roles and utility functions

Role	Scen	Concave	Convex	Linear	Other
PL	1	17 % (1)	50 % (3)	33 % (2)	0 % (0)
	2	0 % (0)	40 % (2)	60 % (3)	0 % (0)
TR	1	25 % (2)	50 % (4)	0 % (4)	25 % (2)
	2	25 % (2)	0 % (0)	50 % (4)	25 % (2)
D	1	17 % (3)	28 % (5)	39 % (7)	17 % (3)
	2	0 % (0)	29 % (7)	65 % (11)	6 % (1)
T	1	25 % (3)	17 % (2)	33 % (4)	25 % (3)
	2	0 % (0)	23 % (3)	54 % (7)	23 % (3)

The data has been analysed with a number of chi-2 tests (Sidney & Castellan 1998) as summarized in Table 4. In the analysis, data from people with responses other than convex, concave and linear was discarded.

For RQ1, a chi-2 goodness of fit test was carried out in order to see whether the three shapes were equally probable. Data from both scenarios was pooled. It was clear that the shapes were not equally probable, which shows that the shape that results from the method is not completely random. Concerning RQ1, the most important contribution lies in the fact that different people respond in different ways, and the distribution of the different shapes.

Table 4. chi-2 tests

RQ	Independent variable	p
RQ1	-	0.0006***
RQ2	PL+TR vs D+T	0.66
RQ3	Scenario	0.012*

*significant at the 5% level, **1% level, ***0.1% level

Concerning RQ2 there are too few data points to be able to carry out a Chi-2 test that compares the shapes of each role. Therefore, data from project leaders and "technical responsibility" was pooled and data from developers and testers were pooled, which means that an analysis comparing "management roles" to more developer-oriented roles could be carried out. There is no statistically significant difference.

In the analysis of RQ3 it was found that there is a clear difference between the two scenarios, i.e. the distribution of curves is different for the two scenarios.

5 Discussion and Conclusions

From this study it is possible to conclude that different study participants have different opinions about how serious risks concerning faults remaining after testing are. It is probably possible to generalize this and conclude that different people in the software engineering process are more or less risk seeking. This is important to know in a risk management process. Methods for assessing the level of risk seeking are available (e.g. the TO method), but in most cases it is probably enough to be aware of the differences.

Based on this study, it has not been possible to state that any role is more risk seeking than any other role. This is either because there are too few subjects or that there actually are no large differences. This

means that it is not possible to formulate any simple ways to assess how risk seeking a person is based on the role that he/she has in a project.

It is possible to observe a difference between the two scenarios. In scenario 1 there are more convex (risk averse) curves than in scenario 2. The result from scenario 2 shows dominance of linear utility functions. In scenario 2 a more risk-averse tendency may be expected since the scenario concerns severe injury to patients or even death, but this is not the case. The only explanation that has been found is the fact that the subjects were used to the TO-method and the tool and knew how it works during scenario 2, see Section 3.4. However, this has to be further analysed.

There are, as described in Section 3.4, some threats to the validity of this study and future studies will be adjusted. People with more experience in general and with more experience from their project-roles should be involved in the study. If a similar experiment design is chosen, it should be adapted so that all subjects do not work with both scenarios in the same order. There were reasons for choosing this design in this research, but in further studies it is probably better not to have the same order for all participants.

References

CMMI (2002). CMMI Product Team, Capability Maturity Model Integratio, Version 1.1, Staged representation, Software Engineering Institute, Technical report CMU/SEI-2002-TR-029.

Fairley, R.E. (2005) Software Risk Mangement, *IEEE Software*, pp. 101.

Fennema, H. & van Assen, M. (1999). Measuring the utility of loss by means of the tradeoff method, *Journal of risk and uncertainty*, 17 (3), pp. 277-295.

Pfleeger, S.L. (2000) Risky Business: What We Have Yet to Learn About Risk Management, *Journal of Systems and Software*, 53, pp. 265-273.

Sidney, S. & Castellan, N.J. (1988). *Non-Parametric Statistics for the*

Behavioral Science, McGraw-Hill.

Sommerville, I. (2004). *Software engineering (7th ed.)*. Readings: Addison Wesley.

Wakker, P. & Deneffe, D. (1996). Eliciting von Neumann-Morgenstern utilities when probabilities are distorted or unknown, *Management Science*, 42 (8), pp. 1131-1150.

Wohlin, C., Runeson, P., Höst, M.; Ohlsson, M.C., Regnell, B., & Wesslén A. (2000). *Experimentation in Software Engineering: An introduction.* Boston: Kluwer Academic.

Paper IV

A Case Study on Software Risk Analysis and Planning in Medical Device Development

C. Lindholm, J. Pedersen Notander, and M. Höst

Abstract

Software failures in medical devices can lead to catastrophic situations. Therefore, it is crucial to handle software-related risks when developing medical devices, and there is a need for further analysis of how this type of risk management should be conducted. The objective of this paper is to collect and summarise experiences from conducting risk management with an organisation developing medical devices. Specific focus is put on the first steps of the risk management process, i.e. risk identification, risk analysis, and risk planning. The research is conducted as action research, with the aim of analysing and giving input to the organisation's introduction of a software risk management process. First, the method was defined based on already available methods and then used. The defined method focuses on user risks, based on scenarios describing the expected use of the medical device in its target environment. During the use of the method, different stakeholders, including intended users, were involved. Results from the case study show that there are challenging problems in the risk management process with respect to definition of the system boundary and system context, the use of scenarios as input to the risk identification, estimation of detectability during risk analysis, and action proposals during risk planning. It can be concluded that the risk management method has potential to be used in the development organisation, although future research is needed with respect to, for example, context limitation and how to allow for flexible updates of the product.

Published in Software Quality Journal, 22(3), 2014, pp. 469-497.

1 Introduction

Software has for many years been an important part of large systems in domains such as automotive, telecommunication, and finance. In health care, software is becoming more widespread because of the introduction of new IT-systems, e.g. administration systems and patient journal systems, and the increasing amount of software in medical devices, e.g. monitoring equipment, defibrillators, and pacemakers. In this paper, we consider software intensive medical devices, meaning medical devices where software is essential to the functionality of the device.

Medical devices can be safety–critical devices, which means that they have the potential of causing harm to people or the environment. It is essential to show that safety–critical devices are safe and of high quality. This can be done through the application of a structured development process that is compliant with a safety standard. Examples of standards are IEC 61508, which is a safety standard for electrical, electronic, and programmable electronic safety-related systems, and IEC 61511, which covers integration of components developed according to IEC 61508 (Gall 2008). Even if standards are available, there is still a need to further investigate how development of software can be carried out with these types of requirements.

The focus of this paper is on risk management, which is an important part of a development process for safety critical systems (Leveson 2011; Sommerville 2007). Risk management (Boehm 1991; Hall 1998; Crouhy et al. 2006) includes identification of risks, analysis and prioritisation of risks, and handling and monitoring of risks. In all these steps, it is not enough to only understand a complex product, but the usage of the product must be understood as well. This means that it is necessary to involve several different roles in the work, such as domain experts, technical experts, and process experts. In this study, medical physicians with competence on the monitored medical processes are involved, together with engineers with competence on the software and hardware, and personnel with competence on the required procedures in the organisation. The objective of the presented research is to summarise experiences from conducting risk identification, risk analysis, and risk planning in the development of a medical device. This is achieved by conducting a case study on a software project in the

medical device domain. An earlier preliminary analysis of the data in this paper was presented at the Software Quality Days 2012 (Lindholm et al. 2012). This paper presents an extended analysis of the case study and covers a longer period of time. Compared with the preliminary analysis, this paper also investigates data collected during the planning step (i.e. research question RQ4 in Sect. 3.1) and the interviews with the development organisation. This case study is conducted in the medical device domain, where the risk management process was carried out on a patient monitor system for monitoring intracranial pressure and calculating the cerebral blood flow. It is carried out in an organisation that has experience from product development in general, but not much experience from software development. The organisation had already an existing risk management process for development of hardware, but needed a risk management process adapted to software development. This is a situation that we believe can be of interest for other organisations in the medical device domain, since other organisations face similar challenges.

The risk management method used in the study has a user perspective in the software risk management process. User scenarios were input to the risk identification step, and intended users participated in the risk meetings. A risk meeting in this case is a formal meeting with intended users, representatives from the development organisation, and the researchers. The activities during the risk meetings depended on the part of the risk management process. The activities are further described in Sect. 3.2. The motivation for this study is to get experiences from having a user perspective in risk analysis and risk planning, with the long-term objective to design an improved version of the risk management process. Risk management and usability are separately two well-known research areas. Regarding medical devices, there is an aim from the authorities that human factors shall be addressed in the risk management process. The researchers have not found documentation on how this shall be done in a detailed, practical way and try to address a practical, detailed level in this research. The objective was also to investigate the implications of composing a system from third-party components, used in a safety–critical context, e.g. monitoring devices, pressure sensors, and communication interfaces with regard to risk analysis. In particular, we wanted to understand how

the dependencies between components would affect the risk analysis and the impact of the choice of system boundary.
In Sect. 2, background and related work is presented. In Sect. 3 the case study research method is presented, and in Sect. 4 the risk management process is shown. The results are presented in Sect. 5, and they are discussed in Sect. 6, where the main conclusions also are summarised.

2 Background and related work

2.1 The medical device domain

Several characteristics of the medical device domain contribute to its complexity. One is the work environment where personnel are mobile and often interrupted in their tasks and required to handle unexpected situations when they occur. Garde and Knaup (2006) have identified several other characteristics that contribute to the complexity of health care products. One characteristic is that the treated patient has an unlimited set of characteristics that constantly change and interact. This makes it impossible to categorise patients in the same way as products can be categorised. Two other characteristics mentioned by the authors are that the majority of stakeholders are non-technical professionals, e.g. physicians, nurses, and administrators, and the multitude of medical standards and medical terminology.

The importance of software and embedded systems controlled and managed by software is increasing in the medical device industry, because medical devices are more and more used in the health care sector (Bovee et al. 2001; Linberg 1993; McCaffery et al. 2005). The size of the software in a typical medical device has been growing with time; in some medical devices, the size in lines of code has increased. For example, the software in a typical cardiac rhythm management device is implemented with approximately half a million lines of code (Vishnuvajjala et al. 1996).

Medical software can be divided into stand-alone software, e.g. hospital information systems and active devices for diagnoses, or software that is a component, part, or accessory to a device, e.g. a software algorithm for statistical analysis of pulse oximetry data.

Software-related problems in medical devices can lead to catastrophic failures. The Therac-25 (Leveson and Turner 1993) is a well-known accident where a software fault led to three patients' death

and several patients were injured due to a software-related failure in controlling the therapeutic radiation. Other examples include the incident with software related failures in a pacemaker that caused two patients death, and a multi-patient monitoring system that failed to store the collected data to the right patient (Schneider and Hines 1990).

2.2 Critical factors

Safety and risk management are important in the medical domain in order to avoid hazard situations that can lead to injury and death. Medical device safety (Dhillon 2008) is concerned with failures and malfunctions that introduce hazard situations, and it is expressed with respect to the level of risk. A medical device that frequently fails but without mishaps is considered safe but unreliable, and a medical device that functions normally all the time and regularly puts humans at risk are considered reliable but unsafe. When a medical device, for example an x-ray device or a surgical laser, is classified with unconditional safety, it requires elimination of all risks associated with it. This is carried out in the design process or through appropriate warnings that complements satisfactory design.

When working with risk analysis in the medical device area (Dhillon 2008), there are several critical factors that relate both to the medical device and the usage of the device, such as design, manufacturing including quality control/quality assurance, user training, interaction with other devices, and human factors. The FDA defines the concept ''human factors'' as ''in the broadest sense, a discipline devoted to the effects of user interface design, job aiding, and personnel training in the operation, maintenance, and installation of equipment'' (FDA 1996). When there are users, there are human errors. The concept of human errors include all the occasions when a planned sequence of mental or physical activities do not lead to the intended result and when the failure cannot be related to chance. Cognition and perception are important factors when it comes to human errors (Reason 1990) and should be considered in designing user interfaces as well as in risk management.

Historically, the earliest documented report of human errors in medical device use can be traced back to 1849 when an error in the administration of anaesthetics resulted in death (Dhillon 2008). Today,

human errors in health care are the eighth leading cause of death in US (Dhillon 2008); the costs are phenomenal, and more than 50 % of technical medical equipment-related problems are caused by operator errors (Dhillon 2000). Walsh and Beatty (2002) refer to a wide range of studies that show that 87 % of critical incidents connected to patient monitoring is due to human factor errors. To minimise user errors and understand user-related risks, it is important to have a complete understanding of how a device will be used, and the goal with incorporating users in the risk management process is to minimise usage-related hazards so that the intended user can safely use the medical device. FDA has a specific document (FDA 2000) that gives guidance on how to incorporate human factors into the risk management process. The document describes what tasks to include in the risk management process and what to consider regarding the user environment, the user and the device. None of these tasks are described in detail in the document. Since human factors are critical, the aim of this research is to implement and study the user activities in practice at a detailed level. The users have been incorporated in the risk management process in this case study through the usage of scenarios as input to the risk identification process and through participation of users at the risk meetings during the whole risk process. Usability testing of the user interface has also been done, but the report from the usability testing is beyond the scope of this article.

2.3 Risk management

A risk is ''the probability of incurring a loss or enduring a negative impact'' (Fairley 2005). In the medical device area, it is crucial that the risk of harm is so low as possible. The medical device development organisations have to address different risks regarding patients, users, environment, and third parties (for example, service technicians) (Rakitin 2006). A fault or mistake of a person or a technical failure in the medical device domain can be the difference between life and death. The use of medical software is an inherent risk to patients, medical personnel, and surroundings.

One challenge of an organisation developing medical software is to identify a sufficient set of risks for their products. If more risks are identified, more risks can be eliminated or mitigated. Another challenge is that the software in a medical device needs to comply with the same

laws and regulations as the medical device itself. How strict and detailed the manufacturer's processes have to be depends on the safety classification of the product. Different laws and regulations exist between countries.

Risk management must be included in the development process for a medical device according to European and American law (Commission of the European Communities 1993; FDA 2005). There are also standards that the organisation needs to follow. Concerning risk management for medical devices, ISO 14971 (www.iso.org) needs to be considered. This standard defines the majority of the risk management terms and gives a framework for a risk management process without specifying details about how things should be done.

Risk management is a process for identifying and managing risks (Hall 1998). The risk management process is often divided into the four steps displayed in Fig. 1.

The risk management process for a medical device development organisation must cover all four steps. The research presented in this paper focuses on the three first steps in the process, i.e. risk identification, risk analysis, and risk planning. The reason for this is that these steps are important in the first line of work in the development of a complete risk management process. The research is focusing on a detailed description of each step. The important forth step, risk monitoring, is out of the scope of this study due to the timeframe of the case study.

Various researchers have reported on risk management on software development in general, e.g. Boehm (1991), Hall (1998), Charette (1989), and Jones (1994). In the medical domain, the published research covers often the whole risk management process on a high level, not specifically described step by step. One example is described by McCaffery et al.(2009, 2010) who have developed and tested a software process improvement risk management model (Risk Management Capability Model) that integrates regulatory medical device risk management requirements with the goals and practices of the Capability Maturity Model Integration (CMMI). Schmuland (2005) also investigates the whole risk management process, although he focuses on residual risks, i.e. the remaining risks after the risks have been handled, and how to assess the overall residual risk of a product. It is based on the identification of all the important scenarios. Hegde

(2011) presents a case study of risk management based on ISO 14971 and concludes that the standard as guideline can ensure a safe product with an acceptable level of risk. Then, there are several studies presenting specific methods, for example the use of FMEA in the risk management process (Chiozza and Ponzetti 2009; Xiuxu and Xiaoli 2010; Habraken et al. 2009). There are some researchers that focus on one of the steps in the risk management process.

Figure 1: Risk management process.

In the medical domain, for example, Sayre et al. (2001) in particular studied the risk analysis step. They described an analytical tool for risk analysis of medical device systems, a Markov chain based safety model and argue that this safety model presents significant opportunities for quantitative analysis of several aspects of system safety.

Dey et al. (2007) have identified the need for analysing risk management issues in software development from the developers' perspective with the involvement of the stakeholders. In the medical device area, we have not found any documented research on software risk management processes involving stakeholders or intended users in the process. In our case study, intended users as well as developers and managers from the development organisation were involved in the risk management process. This was achieved by using user scenarios, during the risk identification phase, as a construct for understanding and communicating about risks.

3 Case study methodology

The research in this paper is based on a study of a single case. According to Yin (2003), ''a case study is an empirical inquire that investigates a contemporary phenomenon within its real-life context, specially when the boundaries between the phenomenon and context are not clearly evident''. In software engineering, process improvement

activities are often of a complex nature and cannot be studied in isolation, which means that there is a need for empirical studies in real-world settings like in this study. The research design of a case study is flexible where the research strategy develops during the data collection and analysis (Robson 2002). The flexible design is also reflected in the interviews that have been made during the study. The design allows open-end questions, and they can be specified in advance and developed over time. The flexibility also allows the researcher to clarify questions during the interview session and gives a freedom in the sequencing of questions and in their exact wording.

In action research, there is collaboration between researchers and those who are the focus of the research (Robson 2002). The observations in this study have been done as active observations, meaning that the researchers have been allowed to influence the outcome of the observed activity. The aim was to observe how the activities are performed in their context, not to actually perform the activities. However, during the activities, it was natural for the researchers to give input and support to the development organisation. The aim was also to get information about aspects of the activities by asking questions and giving advice on relevant topics.

3.1 Objectives

The objective of the case study presented in this paper is to give input to the development of a software risk management process in an organisation that develops medical devices. The development organisation has a risk management process for development of hardware, but needs to adapt it to software development. The specific research questions of the study are as follows:

RQ1: What are the experiences from focusing on a sub-system as a part of a larger system?
RQ2: What are the experiences from using the chosen risk identification method?
RQ3: What are the experiences from using the chosen risk analysis method?
RQ4: What are the experiences from using the chosen risk planning method?

That is, RQ1 was defined based on the architecture of the analysed product, while RQ2, RQ3, and RQ4 concern the three main steps of the studied risk management process, i.e. risk identification, risk analysis, and risk planning, with a focus on pros and cons from the used methods. The software risk management process in this case covers only the software development of the new medical device (bedside monitor). In this case, the new device can be regarded as a sub-system since it is a part of a lager system. For example, the new device imports blood pressure values from a patient monitor. The studied steps of the risk management method are presented in Sect. 4.

3.2 Case study process

The research method applied in this study is an exploratory single case study, and the research process is based on the case study process described by Runeson and Höst (2009). The process in Fig. 2 was followed. First of all, it should be said that the research process that carried out was more iterative than what is displayed in Fig. 2. The figure is intended to show the main activities performed and the general order of the activities.

The main objectives of the study were defined based on the general interests of the researchers, and the interests of the development organisation. This was defined in informal meetings between the researchers separately and between the researchers together with the development organisation. The researchers had some knowledge about the development organisation before the case study, based on earlier involvement in the organisation and the developed product.

The preparations for the study were made in the initial phase, which included informal meetings with the development organisation. In the initial phase, the objectives of the study were refined, and the research methodology was decided in more detail. In order to record all relevant information from the activities performed during the study, the first author of this paper was responsible for managing this information in the form of a case study protocol stored as a set of files. A first version of a case study protocol with research questions, early versions of the interview questions for the first interviews, and procedures and protocol for data collection were produced initially. The information in the protocol was maintained and updated over time by logging, for

example, the discussions, risk meetings, participants, and decisions made by the development organisation as well as by the researchers.

Figure 2: Case study process.

The following information was stored as part of the protocol:

- Research questions
- Interview questions and transcripts from interviews
- A log-file where all meetings were listed
- Protocols from the meetings where identified risks are listed and described
- Qualitative observations from meetings in textual form. These observations were formulated after risk meetings and collected by the first author of this paper.

As part of the preparations and as input to the risk meetings, discussions were held with the organisation regarding different risk identification techniques, risk analysis methods, and scales for risk

classification. The development organisation decided, based on these discussions, the design of the software risk process for the first two steps, i.e. risk identification and risk analysis. After the preparations were finished, the first phase of the data collections started.

The data collection was made through two different sources: interviews and observations. All collected data were treated confidentially in order to protect the participants of the study and to ensure that the participants felt free to speak during data collection. In an effort to increase the validity of the study, a technical report with the preliminary analysis results were created so that the participants could review and give feedback on the result. In addition, feedback discussions were held with the participants.

Before the first risk meetings, two interviews were held with participants from the development organisation. The interviews were conducted in order to understand the development organisation's expectations on the new risk management process and to record their experiences from the existing risk management process for hardware.

The data collection through active observations was carried out in three phases:

> Phase 1: Risk identification and risk analysis. Five risk meetings were held where the defined software risk process was used with the researchers as active observers of the process.
> Phase 2: Risk analysis and risk planning. Seven risk meetings were held where an updated version of the software risk process was used, with the researchers as active observers.
> Phase 3: Risk analysis and risk planning. Three risk meetings were held where the risk management process continued with the researchers as active observers.

Before Phase 2, there were meetings with the organisation on how to proceed with the risk management process, i.e. risk planning and updates of the used process. One outcome was that the development organisation decided not to use detectability due to perceived difficulties estimating it during Phase 1. The altered process was then executed during Phase 2 and Phase 3. During the risk analysis, the risks that were considered technical risks, as opposed to user risks, were transferred to a technical risk analysis.

After Phase 3, new interviews were carried out with two representatives from the development organisation. These interviews were made in order to understand the development organisations experiences, lessons learnt, and apprehension of the new risk management process.

The data collection and analysis is further described in more detail in the following subsections.

3.3 Case study context and subjects

The case study was conducted at a department at a large hospital in Sweden, which develops and maintains medical devices. The development organisation has extensive experience in developing and maintaining medical devices, but not with devices including software. The target environment for the new medical device is an intensive care unit (ICU) at the hospital. The case study was conducted from the summer 2010 until spring 2012. A timeline of the study is presented in Fig. 3.

The risk management process was carried out on a patient monitor system for monitoring intracranial pressure and calculating the cerebral blood flow, including both software and hardware. However, the risk management process primarily considered the software component that was developed for the new device (bedside monitor), and focused on identifying user risks. The purpose of the patient monitor system is to monitor a patients' intracranial pressure, calculate the cerebral blood flow, and present it to the medical personnel. The main parts of the system are presented below. Part 1 and 2 of the system have been used before, while part 3 is the one being developed.

Part 1: Pressure sensor placed in the patient's skull.
Part 2: Monitor connected to the sensor. The monitor presents and exports blood pressure values.
Part 3: Bedside monitor, i.e. the new device. It imports blood pressure values from the patient monitor, calculates the cerebral blood flow, and presents it on a screen. It consists of a computer with a screen, an operating system, the Palcom middleware, and the application code. The graphical user interface presents the calculated blood flow and the measured intracranial blood pressure.

Before this project the setup was that the pressure sensor was connected to the commercial patient monitor, which carries out digitalisation of the values and presents the result as a real-time curve on its small screen. The value is sampled every 8 ms (125 values a second), which results in a smooth graph. In the new setup the sampled values are exported in real time from the monitor, using the available serial port to the bedside monitor. The new software, developed in java for the bedside monitor, is structured as four main components:

- Import data from the patient monitor
- Calculate blood flow
- GUI for presentation and interaction
- Storage for measured data, calculated data, and other info such as comments from the nurses and physicians.

Figure 3: Case study timeline

With this arrangement, it is possible to present the blood flow continuously, as a curve, in real time, while other methods only can give single values. One can also view historical data stored on the bedside monitor.

The implementation is well separated where each part is implemented as services in the middleware framework (Svensson Fors et al. 2009). The development of the software has been done iteratively where each part has been enhanced separately. The calculations have been modified and improved iteratively, while the GUI has been developed in cooperation with the physicians and nurses at the clinic, also iteratively.

Participants in the study represent three different groups: the intended users with special domain knowledge (e.g. physicians and nurses), the development organisation (e.g. medical device expert, risk analysis supervisor, and software developers), and the researchers (e.g. process experts and technical experts from academia). At this stage of

the process no representatives from patients' organisations were involved.

3.4 Preparatory discussions and data collection

In this section, the preparatory discussions with the organisation are described as well as the performed risk meetings and interviews, in which data collection was made.

3.4.1 Preparatory discussion

There were two preparatory discussion meetings together with the organisation, Discussion 1 prior to Phase 1 and Discussion 2 prior to Phase 2 (see Fig. 3). The organisation had an existing risk management process for development of hardware, but needed a risk management process adapted to software development. The study began with Discussion 1, which was about the development process, including the risk management process. It was clear from the discussions that the first part of the risk management process should focus on the two first steps in the risk management process, i.e. risk identification and risk analysis. As a result of Discussion 1, a process for risk identification and risk analysis designed for software systems was defined. In the second discussion, Discussion 2, focus was on risk planning, including risk resolution, optimising selection criteria, and how to handle high-severity risks. The whole software risk process is described in Sect. 4.

3.4.2 Data collection

Data was mainly collected from two sources: interviews, the first ones held in the beginning of Phase 1 and the last ones after Phase 3, and active observations, during all three phases.

The first phase, Phase 1, started in September 2010 and ended in December the same year. Five risk meetings were held for approximately 3 h each. At least two representatives from each participant group, the intended users, the development organisation and the researchers were present at the meetings. The scope was risk identification and risk assessment. A risk could both be identified and assessed during the same meeting.

In total, 152 risks were identified and assessed, where 12 were assigned a high-risk value. At the end of Phase 1, approximately 18 man months had been spent on developing the software for the blood pressure monitor. Phase 2 started in March 2011 and ended in June

2011. Seven meetings were held, with the same characteristics as in Phase 1. The scope was risk identification, risk resolution, root causes, action proposals, and effect risks.

In total, 218 risks were identified, 25 identified risks were removed during risk assessment because the team no longer regarded them as user risks or problems that should be a part of the risk management process. 10 of the risks were considered technical risks, as opposite to user risks, and were transferred to the technical risk analysis. Of the remaining 183 risks, 10 were given a high-risk value.

The final phase, Phase 3, started in January 2012 and ended in March 2012. Three meetings were held, with fewer participants than in the earlier phases, although the same participant groups were represented. Furthermore, the meetings were shortened to 2 h because a majority of the participants perceived that 3 h were too long. The scope was risk planning of the remaining risks.

At the end of Phase 3, 225 risks were documented. In addition to risks from earlier phases, risks related to implemented risk actions and planned risk actions were added in Phase 3.

Of the 225 documented risks at the project end, 25 were removed because they were no longer perceived as risks, 11 risks were transferred to the technical risk analysis, and 3 were considered residual risks. The remaining 86 risks were determined to be sufficiently managed.

Data collection during the risk meetings was conducted through active observations by the researchers. The participants had a high awareness of being observed and there was quite a high degree of interaction by the researchers. The purpose of the interaction of the researchers was to capture interesting aspects as well as pros and cons regarding the process. The researchers asked direct questions during the risk meetings, for example, if something was vague regarding the process.

During the risk meetings, the researchers documented their observations individually on paper. These notes contained both direct observations and the researchers' own reflections. The notes, as well as personal reflections, were in most cases discussed by the researchers directly or shortly after the meetings. It is beyond the scope of this paper to present all notes in detail. However, the notes were often written in bullet form with findings like ''unclear for some persons what they mean with normal usage'', ''sometimes hard to know where

we are in the scenario in the discussion'', etc. The notes were understandable and valuable for the researchers in their discussions after the risk meetings.

After the last risk meeting in Phase 1, the notes were compiled into a list of statements, which were recorded in the case study protocol. Each statement was then coded, grouped, and interpreted. The outcome of the analysis was reported back to the development organisation in the form of a technical report, which concluded Phase 1. The report was reused in the feedback discussions after Phase 3.

The second data source was the interviews. The first interview, with two representatives from the development organisation, was carried out in the beginning of Phase 1, before the first risk meeting. The second interview was a follow-up interview after the end of Phase 3.

It was held with the same representatives from the first interviews.

The interviews were conducted as an open dialogue between the researcher and the interviewees. All questions were predefined and open-ended. The following questions were asked in interview 1:

- What risk management process do you have in general?
- What strategies exist at management level?
- Are there different processes for different products?
- What differences do you see for a risk management process for software?
- What types of risks do you usually focus on?
- What challenges do you see in this project?
- What improvements do you want to achieve?
- The following questions were asked in interview 2:
- Describe the new risk management process.
- What are the main differences compared to before?
- What advantages do you see?
- What is difficult with the new process?
- What improvements can you see?
- What was better before?
- What challenges did you see in the introduction of the new process?
- Anything that should have been done in a different way?
- What are the most important experiences from the work?
- What are the differences between risk management in general

and for software?

- What will happen now with the risk management process?

The interviews were made over phone except for one that was carried out face-to-face. They were all done in Swedish and by the same researcher. Since the questions are open and the interviews were conducted in a semi-structured way as a dialogue. The respondents were allowed to talk freely after each question and in some cases follow-up questions were posed, such as ‘‘who do you see as the main authors of scenarios?’’ after the first question in the second interview. All the interviews were recorded and later transcribed.

Observer triangulation (Robson 2002) was achieved by having three researchers participating in the case study, which meant that alternative interpretations and explanations were discussed. Data triangulation was done by collecting data from multiple sources, i.e. interviews and active observation.

3.4.3 Analysis

The analysis, the fourth step in the case study process (Fig. 2), is based on the notes taken by the researchers during the risk meetings and the interviews. After the meetings, the notes were compiled into a list of statements in a protocol, which was distributed among the researchers. In addition, interesting observations and reflections were discussed among the researchers, directly after the risk meetings.

The analysis proceeded with grouping each statement, either as an observation or a reflection. Observations were statements that only described what occurred or was said during the meetings, reflections were statements that contained the researchers’ immediate thoughts about an observation. Next, each statement was labelled with the step of the risk management process, during which it was recorded. After that, the statements were grouped into themes, such as the product, the organisation, the process, methods and experiences. The purpose of the employed coding strategy was to get a better understanding of the material and make it easier to navigate.

At the end of Phase 1, the information about the case study and the preliminary results, as presented by Lindholm et al. (2012), were presented in a technical report. The report was sent to representatives from the development organisation as part of the feedback process. This was done with the purpose of giving the development organisation an opportunity to comment upon the interpretation of the results, and to

resolve potential misinterpretations by the researchers. The outcome was that the development organisation confirmed that their understanding of the process was consistent with the researchers and that only minor details, such as the title of one participants, had to be corrected in the technical report.

After Phase 2 and Phase 3, all the observations and reflections from these two phases, including all the material form the interviews, were analysed by the researchers. The observations and reflections were analysed the same way as the observations and reflections in Phase 1. The transcribed text from the interviews was then labelled with the predefined factors, grouped according to the factors and then discussed by the researchers.

The results from the analysis can be found in Sect. 5. The results and conclusions were coordinated with representatives from the development organisation to get clarification and confirmation of the material.

4 The software risk management process

This section describes the risk management process used by the development organisation, as it was employed during the risk meetings. The first step, risk identification, can be based on different techniques that can be used, such as checklist-based identification, development of prototypes, cost-benefit analysis, and scenario-based analysis (Boehm 1991). In the studied risk management process, a scenario-based identification method was used. A scenario was defined as a chain of events, with a cause-effect relationship that describes a realistic diagnosis sequence during normal use, see Fig. 4. Each scenario can be traced back to at least one requirement for the product. The scenarios cover both normal operation and special circumstances.

Scenarios based on the requirements specification were used as input to the risk identification step. The design of the first part of software risk process is shown in Fig. 5.

The risks were identified through brainstorming, with the medical device expert acting as facilitator during the sessions. For each scenario, all participants suggested possible risks connected to the specific scenario discussed. Thus, all identified risks were considered in the next step, if they obviously were no risks. All identified risks were

documented during the meetings. In the next step, risk assessment, each identified risk was assessed separately according to probability, severity, and detectability. Scales predefined by the Swedish national board of health and welfare was used for probability and severity assessment.

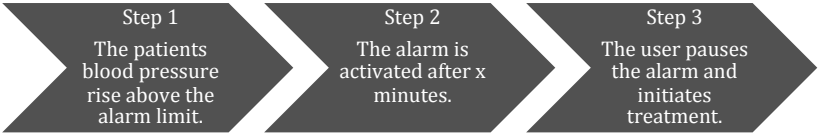

Figure 4: Example scenario

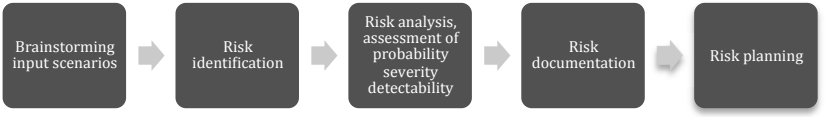

Figure 5: First part of the software risk process

The scales are graded from one to four (low to high). A probability grade of four corresponds to ‘‘fault that will occur each month or more frequently at normal use’’, a grade of one to ‘‘fault will never occur or very unlikely’’. On the severity scale, four correspond to ‘‘death or severe injury’’, and one to ‘‘discomfort or minor injury’’.

The risk value (**R**) was calculated for each risk by multiplying the probability (**P**) with the severity value (**S**), i.e. **R** = **P** x **S**. The highest risk value a risk can have with these scales is **R** = 4 x 4 = 16.

Detectability was estimated according to the three following statements ‘‘if the fault (hazard) always could be detected before a severe situation occurred’’, ‘‘if the fault (hazard) sometimes could be detected’’, or ‘‘if the fault (hazard) never could be detected’’.

Risks were documented in a spreadsheet, which was continuously updated during the risk management process, see Table 1. After risk identification it would contain, id, conditions, risk id, and risk description. After risk assessment, values for the, **S**, **P** and **R**, columns would be added. Traceability was maintained by the risk id, specified on the form Ax.y.z Rn. The first part, Ax.y refers to the scenario that

the risk was identified in, Z to the step in the scenario and Rn is a local unique identifier that allows for more than one risk to be assigned to a particular step in a scenario.

The user scenario in Fig. 6 shows a scenario where the user reads the measured intracranial blood pressure on the bedside monitor, the curve is zoomed as much possible and the user increases the amplitude scale with the help of the menu. The risk identified according to step 3 in the scenario, is the risk in Table 1. The consequence of that risk could be that the patient is given the wrong treatment. This risk was regarded to be a severe risk that can cause the patient severe injury or death so the group therefore gave the risk the severity value 4. The probability that the risk would occur was considered very likely, it could happen each month or more frequently at normal use, so the probability was also given the value 4. Since the risk value became 16, the risk was due to further action.

In the risk planning step risks that required actions were handled. The organisation had decided to proceed with risks with **R** = 8 according to the risk management plan but also with **R** = 6 or risks with **S** = 4 or risks with **S** = 2 plus **P** = '?' (i.e. probability could not be assessed) due to that there had been no prior decision on if a risk should be pursued or not. That strategy implied that some risks with **R** = 4 were handled with the motivation that it increases the quality. The development organisation colour-coded the identified risks. Risks that could be technically prevented were coloured blue, risks that should be investigated to see if they could be technically prevented were coloured purple, and risks that were not to be handled were coloured white. For all the risks that the group decided to proceed with, action proposals were discussed and decided on. In the following iterations, risks where actions had been implemented were reassessed according to the four-graded scales and possible effect risks were identified. The effect risks in its turn were then assessed according to the same scales and the risks with low risk values were left without further action. Remaining risks were assessed and were either accepted, assigned new actions, or left as residual risks. Risks assigned with actions were then investigated by the development organisation, and software parts linked to risks with **R** > 8 were assigned to verification.

Table 1: Risk identified from user scenario A1.1

ID	Conditions	Risk ID	Risk Description	S	P	R
A1.1	The amplitude scale is increased with the menu	A1.1.3 R1	Another user does not see that the amplitude scale is increased	4	4	16

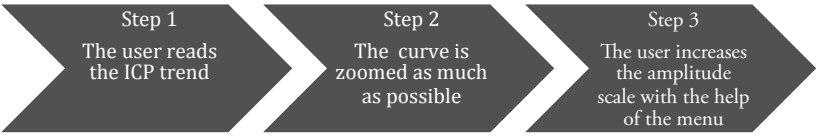

Figure 6: User scenario A1.1.

5 Results

In this section we present our results from observations that were made during the risk meetings described in Sect. 3. The results are grouped, with regard to the research questions, into four categories: system definition (RQ1), risk identification (RQ2) risk analysis (RQ3), and risk planning (RQ4). See Table 2.

Table 5 to get a brief summary of the results. The statements in the tables are traced to the text with ids, on the form **R**x.

In the last section, the development organisation's experiences from the risk management process are presented. The results presented in Sects. 5.4 and 5.5 is new and exclusive material for this article and not covered by Lindholm et al. (2012).

5.1 System definition

This section presents results related to the definition of the system, that is, system boundary as well as system context. In the studied risk management process, the system described in Sect. 3.3 was the object of analysis, although not in its entirety. It was decided that the risk management process should only consider the bedside monitor, in

particular the in-house developed software functions and user interaction with the monitor.

Table 2 Summary of the results concerning the system definition

Area	Summary	ID
System Boundary	The team had to make assumptions about input from external devices and their reliability.	R1
	The team had difficulties deciding whether a risk belonged to the system or the environment.	R2
System Context	The target environment was not defined in detail and information about workload, user experience, and physical layout of the target environment, had to be supplied on the fly.	R3

A consequence of the narrow boundary definition of the analysed system was that the team had to make assumptions about the components that were not included in the analysed system, e.g. the patient monitor and the pressure sensors (R1). These assumptions included details about the input data, as well as the reliability of the excluded components. An example of these assumptions concerned the input from the patient monitor to the bedside monitor and involved the risk that wrong values were shown on the beside monitor. For example, it was assumed that if the manufacturer updates the communication protocol to the patient monitor, this could be the source of such values. The assumption was then made that the manufacturer always informs if this type of update is made. Major interfaces between the analysed system and its environment were also identified at this stage, such as the graphical user interface and some of the technical interfaces between the components in the whole system, e.g. the Ethernet connection between the bedside monitor and the patient monitor. Thus, the technical context of the system was defined.

Other factors of the system context, which had to be defined according to the risk management process, were the target environment and the intended users of the system. The target environment was defined by the team as the ICU and the intended users as nurses and physicians at the ICU. Factors in the environment such as physical and mental working conditions, current practice, and rules, were described when questions about them arose. This was also the case when

questions about differences between categories of intended users arose (R3).

During the risk management process, difficulties with the chosen system boundary were observed. The team had problems deciding whether certain risks were part of the analysed system or if they were outside the system boundary (R2). According to the process, risks outside the system boundary should not be considered. This was most frequently observed for risks that were related to erroneous input data, either from incorrect calibrated or malfunctioning measuring devices, or from problems with the connections to the bedside monitor. For instance, it was not clear to the team if sensor failures should be analysed if the bedside monitor depended on the output of the sensors.

5.2 Risk identification

The studied risk management process puts emphasis on the users and their interaction with the system through the use of scenarios. In the studied process, the scenarios were based on expert knowledge of the target environment and current work practices. In this section, results related to the use of scenarios are presented, i.e. how they were used throughout the risk management process by the risk management team (Table 3).

In the risk identification step, scenarios were used for brainstorming and discussions about potential risks. In practice, the team went through the scenarios at the risk meetings, one by one, step by step, and for each step suggested potential risks. The aim was to record all risks that were suggested, not to reason about any detail. Despite this, a tendency was observed to let the perceived probability of a potential risk or the severity of its consequences influenced which risks were identified (R4). For instance, sometimes the team argued that a risk that was not very probable or had very mild consequences should not be considered a risk at all. The risk that the user on the bedside monitor chooses the wrong comment among the predefined comments is an example of this. The team argued that this would not happen and that the risk should be removed. In the end it was decided to keep the risk in the documentation.

Table 3 Summary of the results concerning risk identification.

Area	Summary	ID
Scenario	There was a tendency to let the perceived probability of a potential risk or the severity of its consequences influence which risks were identified.	R4
	It was unclear if risk identification in a given step should be done independently of the path leading to it or if the identification should be constrained by the scenario history.	R5
	The design of the scenarios impacted the outcome of the risk identification.	R6
	The user representatives dominated the discussions due to the belief that they had better background knowledge than the development representatives.	R7
	Technical risks were not restricted to the scenario they were identified in, as opposed to user risks, which were to a larger degree only valid in a specific scenario context.	R8

When going through the scenarios, it was found that the team had different views on the importance of the ordering of the steps in a scenario (R5). Some team members argued that causality was important, i.e. that the path to the current step under discussion should be taken into account. Other members argued that the current step should instead be analysed independent of the path leading to it. For example, in the scenario in Fig. 4, should risks be identified when pausing the alarm in general or only when a patient has a high blood pressure during 9 minutes?

Furthermore, it became evident that the scenario composition had an impact on the outcome of the risk identification (**R**6). If a scenario were wrongly constructed or unrealistic it would not expose the intended system behaviour and would thus prevent the identification of potential risks. Some of the used scenarios had to be adjusted because they did not describe the system or user behaviour well.

Another aspect that might have some significance on the outcome of the risk identification was the observed difference in activity level between the participants. The user representatives dominated the discussions whereas the developer representatives held a lower profile

(R7). It was the teams' belief that the user representatives had more background information about the scenarios, e.g. medical knowledge, and the target environment, and were thus considered better suited to identify certain risks. Although the developer representatives held a low profile they contributed with valuable insights about the technical nature of the system. In particular, their expert knowledge of the software and the graphical user interface was valuable to the team. In general, they had less influence on the discussion than the user representatives.

Although the scenario-based method focuses on user interaction and user-related risks, some technical risks were found, mostly relating to system interfaces. The technical risks share that they are more general in nature than the user-related risks and they are not bound to a specific scenario (R8). The technical risks were recorded and transferred to the technical risk analysis.

5.3 Risk analysis

The risk analysis was conducted using a method influenced from the development organisation's existing risk management process for hardware products. The method specifies three variables, severity, probability, and detectability that are to be estimated based on normal usage of the system. The process mandates that each risk should be assessed independently of all other risks and that each variable should be estimated in sequence, starting with severity followed by probability, and finally detectability. In this section, observations related to the risk analysis step and estimation of the three risk variables is presented (Table 4).

Prior to conducting the risk analysis, the team had to define what normal usage meant for the actual system. It was defined as the average workload (**R**9), e.g. the average number of patients at the ICU and the average duration a patient is connected to the system. The risk management process does not give any concrete suggestions on how normal use should be defined.

Several challenges were observed regarding the estimation of severity, probability and detectability. For instance, it was not always clear to the team what severity and probability actually meant, and how they were related to each other (**R**10). This issue was solved during the

risk meetings. It was determined from discussions that the severity is the worst case consequence of a risk and the probability is how often the risk occurs, independent of its consequences.

Regarding the estimation of detectability, the team thought it was difficult or even impossible to assign an appropriate value (**R**11). This was often the case when a risk was related to not being aware of an event. Typically, the users were the only participants that could determine if a risk was detectable or not. Due to the difficulties of estimating detectability, the team refrained from estimating a value for most of the risks.

Furthermore, it was observed that, although the assessment of the risk values should be independent of each other, the discussions about severity and probability were sometimes hard to separate. Moreover, in those cases where a detectability value was estimated it sometimes influenced the assessment of probability and severity, i.e. some argued that, if a problem was detected, actions would be taken to prevent it form occurring or result in an accident (**R**12).

As a final observation relating to the estimation of the risk variables, the system definition was seen as impractical when assessing certain risks, because it was too narrow (**R**14). In the analysed system some risks would be perceived as catastrophic, but had the whole system, as described in Sect. 3.3, been analysed they would not. The main reason for this was mainly the built-in safety functions in the patient monitor. For those risks, the team agreed to consider the full system definition when estimating the risk variables. An example of such a risk is that the real-time plot is not displayed on the bedside monitor. The severity was, however, regarded low because of the redundancy of the patient monitor.

The activity levels of the participants were observed, for the same reasons as in the risk identification step. When estimation the severity value, the user representatives had great impact on the results (**R**13). Typically, they would be the only participants that were able to determine the consequence of a particular risk in the target environment. Estimating a risk's probability value required both the users and the developers. Risks associated with user interaction had probabilities assigned based on the current situation at the ICU and on previous experience with similar systems. Technical risks had their

probabilities assigned based on the opinion of the developers. The team did not assign probabilities to pure software-related risks.

Table 4 Summary of the results concerning risk analysis.

Area	Summary	ID
System Context	The risk analysis was made under the assumption of normal use, which was defined as the average workload during a year.	R9
Estimation	It was not clear to the team what severity and probability actually meant, and how they were related to each other	R10
	The team had problems with estimating detectability and refrained from doing it for the majority of the identified risks.	R11
	The estimation of severity, probability, and detectability, was sometimes influenced by the other values, e.g. a low probability would result in a low severity; if a risk is detectable then it is not likely to happen.	R12
	The user representatives, due to their extensive medical domain knowledge, dominated the estimation of severity and, to a lesser extent, probability.	R13
System Boundary	The chosen system boundary was seen as too narrow because it did not include all components of the product. A risk could have catastrophic consequences in the analysed system, but when considering the whole product the risk would be non-existing or less severe.	R14

5.4 Risk planning

In this section we present observations related to how the risk planning was carried out, including discussions and decisions about proposed risk reduction actions during the risk meetings. At this risk meetings the user representatives dominated the discussions since they had more knowledge about the domain as well as more background information about the action proposals, such as medical knowledge and environmental knowledge (**R**15). Further, more time was spent at the meetings discussing implementation of the action proposals rather than

the actual risks they might introduce. It was not uncommon that those discussions continued after a decision was made on what action to implement (**R**16). Sometimes during these discussions possible alternative solutions were also proposed and they were then documented in the documentation together with the other action proposals, which could result in that more than one action proposal was suggested for a risk (Table 5).

According to the provided scales, a risk could not have the risk value zero (**R**18). Thus, it was not clear form the process description how to handle risks that were not seen as risks anymore, e.g. due to actions taken to eliminate it. After some discussions, the team arrived at a solution where such risks were scored out in the risk documentation. Another problem with the scales was the assessment of probability, in particular how to handle risks when the probability could not be assessed. The solution employed by the team was to assign the maximum value, i.e. four to the probability. A consequence of this solution was that risks with severity two or higher automatically became part of the risk planning. Most of these risks were software risks, for example that the software handling the alarm does not receive any input values. They were later assigned for special verification and transferred to the technical risk analysis.

Furthermore, the development organisation decided that a risk should only be reassessed after risk reduction actions had been implemented, but in the later stages of the project, risks were reassessed even if the actions had not been implemented. It could be noticed that this created some confusion between the participants.

Regarding the risk descriptions that were written for each risk when they were identified, it was shown that the explicitness of the description had an impact on the understanding of the risks in later stages of the process (**R**17). This became evident when the risks were going to be reassessed. For some risks, the risk description was perceived as vague and unclear, and the initial meaning of the risks was debated since it was not clear to everyone what the risk really was. Such a vague risk was the risk that an alarm was not observed on time. From the beginning ''on time'' was not defined in exact clock time, and could have different meaning for different users.

Another problem was that for some of the risks there were more than one identified root cause, and sometimes the proposed action for

one of the causes lowered the risk value, but the action proposed for the other cause did not. No decision was made by the development organisation on how to handle the proposed risk reduction actions in these situations.

Table 5 Summary of the results concerning risk planning.

Area	Summary	ID
Risk Mitigation	The user representative had better domain knowledge than the development organisation	R15
	The discussion altered between focusing on the actual risks and the action proposals as such	R16
	The initial risk descriptions have impact on later understanding of the risk context	R17
Process Support	The scales had limitations e.g. a risk could not be assigned the risk value zero.	R18

5.5 The software risk process from the development organisation's point of view

The development organisation had an existing risk management process for development of hardware and wanted to develop a risk management process adapted to software development. In this section, the results from four interview sessions are presented. The interviews were conducted in order to understand the development organisation's expectations on the new risk management process as well as their opinions about the outcome of using it in a real project (Table 6).

The two first interview sessions were held with representatives from the development organisation during the design of the new risk management process, i.e. in the beginning of Phase 1, to get their expectations of the new risk management process and their experiences of the hardware risk management process they had previously used. At the end of Phase 3, the same representatives were interviewed again to get their view on the outcome of using the new process, what they found challenging with the process, and their lessons learnt of using the process in a real project.

Table 6: Summary of the results concerning the organisation's view of their process.

Area	**Summary**	**ID**
Process Adoption	Health personnel working with risk management feels familiar with the new process and it is easy for new personnel to adapt to it.	R19
	A main challenge was to find the time and right competences for the risk analysis team.	R20
Scenario	It is difficult and time consuming to produce relevant user scenarios.	R21
	The scenarios make the software easier to understand, which in turn improves the understanding of potential risks.	R22
Process Support	Provided scales were too limited and their usefulness were not optimal e.g. they did not include the value zero.	R23
Estimation	Probability cannot be estimated for software.	R24

The old risk management process was characterised, according to the development organisation, by extensive checklists and templates. For instance, one interviewee said that they used ''enormous checklists, where you should go through many items, so it became complicated''. The risk analysis also had a tendency to be performed late in the projects, something that the development organisation wanted to avoid in the new process. They wanted to achieve a uniformed way of working for the whole organisation, and a well organised and effective process.

Further, when the representatives from the development organisation were asked in the second interview to reflect on the advantages with the new risk management process, they mentioned that the health personnel working with risk management feel familiar with the new process and that it is easy to learn for new personnel (**R**19). The only challenge that was highlighted was the difficulty of producing relevant user scenarios, which was emphasised by one of the interviewees as ''I don't think it is quite that easy, I think you have to invest time in that'' (**R**21).

Looking at the challenges that the development organisation felt they had to face, before Phase 1, they were related to the new risk

process itself. One identified challenge was that they were afraid that the new process would not catch all risks without the support from checklists and standards. The involvement of many different roles and competences at the same time was also seen as a challenge. After completing the process, the development organisation perceived that the greatest challenge had been that the process was untried and that the process was discussed in parallel with the discussions regarding the product risks. This was especially confusing for the user representatives at the meetings. Another perceived challenge was that the process was seen as time-consuming (**R**20).

The process of risk management of software was new to the development organisation. Comparing hardware and software the development organisation perceived that the difference between a physical object, e.g. a chirurgical instrument, and software is that software is untouchable and harder to understand. The intangible nature of software makes it more difficult to really know what risks can be present, both regarding user risks and technical risks. Furthermore, after experiencing the new risk management process, the development organisation sees a difference with the use of user scenarios. When working with hardware there is a choice between using checklists or user scenarios, but when it comes to software there is no such choice. ''It is easy to find risks when you use user scenarios but it becomes much clearer when we talk about software [than hardware]'' (**R**22). Another difference concerns the probability value according to the interviewees. It is hard to assess when it comes to software and it has to be treated in a different way than for hardware (**R**24). The provided scales were perceived to be too limited, especially since it did not allow the zero value (**R**23). As a result, the probability scale has been redefined. It is now based on the frequency of use of the product rather than calendar months, as well as allowing the value zero.

In the interview after Phase 3 the interviewees were asked to reflect upon lessons learnt from the new risk management process. To summarise, they recommended that the number of participants in the risk analysis team should be restricted and that roles and responsibilities should be clarified in the team. Scenarios should be discussed in depth and be well established in the analysis team. The interviewees were also firm in their belief that detectability should not be included in the analysis, because it triggered discussions that made the severity

estimation difficult. They also stressed that a probability score of zero should be included in the future, as a way to document that a risk had been considered but eliminated. As a suggestion to improve efficiency, strict control of meeting discipline should be enforced, especially when discussing risks and risk mitigation efforts, i.e. risk action proposals. Too much time was spent on discussing implementation details rather than effects of proposed actions. Finally, it was emphasised that technical risks should be separated from user risks at the beginning of the risk management process.

The risk management process will now be adopted to the risk standard ISO 14971 by the development organisation. This will lead to that the terminology will be harmonised with the standard and some more documentation requirements will be added, such as risk management plan and risk analysis report.

6 Discussion and conclusion

In this section, we discuss our results and present the conclusions from our analysis. The discussion is organised, with the aim of addressing the research questions, into five areas: system boundary (RQ1), system context (RQ2 and RQ3), scenario (RQ2), estimation (RQ3), and risk planning (RQ4). We address problems that we found particularly challenging during the studied risk analysis process and that can be considered when a new risk management process is defined. A summary of the main findings is shown in Table 7.

6.1 System boundary

In systems theory, safety is an emergent property on the system level (Leveson 1995). Even so, one of the purposes of the study was to see if it was possible to do risk analysis on a part of the whole system, i.e. the in-house developed software and patient monitoring device.

From our results, we can draw the conclusion that the system boundaries must be set carefully and not without considering dependencies between components.

As observed in the risk identification step, it was necessary to make assumptions about input from external devices, used in the analysed system. Later in the risk analysis step, the existence of these external

devices could be used to argue that certain risks was nonexisting or had low severity.

Before defining the system boundary, it should be clear how components are coupled. Components with low coupling might be analysed independently and components with strong coupling should be analysed together.

Table 7 Summary of the findings of the study.

Area	**Findings**
System Boundary	The system boundaries must be carefully defined considering dependencies between components.
	Couplings between components should be identified so that loosely coupled components can be separated from strongly coupled components in the analysis.
System Context	The system context, i.e. where the system is used, how it is used and by whom, should be described using quantitative measures and example scenarios, providing a risk analysis team with a better foundation for risk identification and risk analysis.
Scenario	Constructing relevant scenarios is challenging. Trade-offs must be made between common case scenarios and special case scenarios. Mixing developer and user scenarios might improve the overall scenario quality as well as attaching contextual information to the scenarios.
	The user representatives dominated the discussions around the scenarios, because of their expert knowledge about medical practices. If not managed correctly this might prevent valuable insights from the development team during risk identification and risk analysis.
Estimation	The order of estimation influenced the outcome of the risk analysis, thus the prescribed order of estimation, e.g. severity, probability, and detectability, should be strictly followed.
	The concept of detectability was not well understood and the provided scale did not give as much help, as was the case with probability and severity. Although, detectability might provide

	valuable information it was considered too difficult to estimate.
Risk Planning	Documenting risk descriptions, which only captures the essence of a risk, is not enough. To be able to understand a risk through the course of the risk management process, additional contextual information is needed.
	There is a tendency to discuss action proposals instead of risks during risk planning. Strictly controlled meetings might keep the discussions on track.
	Mixing action proposals that are implemented with proposals that are not introduces unnecessary confusion. To avoid this, risk analysis should be done prior to implementation.
Risk Management Process	The process is considered effective and easy to adapt to, and it fits well with "the natural flow" of the development process.

6.2 System context

The system context, such as users and physical and psychological work conditions, affects the identification and analysis of risks. It is therefore important that the system context is defined during the analysis.

In the studied risk process, normal use is used as an indicator of how the system will be used in the target environment. Normal use is defined as an average of the workload on the system in the target environment. This is a simple approach, and it gives no detailed understanding about how the system is used and how it affects the risk analysis.

By describing normal use in a more quantitative manner, e.g. using a scenario or use case, a more nuanced picture can be obtained about the usage of the system in its context. The description may not only describe for how long and for how often the system is used, but also where and when. The description could be augmented with special case scenarios where high load and low load could be defined.

6.3 Scenarios

The studied scenario-based risk identification method focuses on user interaction and user related risks. Some technical risks were also identified using the scenarios. The nature of these risks relates primarily to user friendliness and that calculated values are displayed correct. Since technical risks are of a more general nature and not scenario-specific, there is a need for a separate risk identification regarding these risks, preferably performed by the software development team that possesses technical knowledge of the system. There is also a need for risk identification of external factors, for example, process and project risks.

The scenarios have to be designed in order to reflect the system functionality as correctly as possible. It is not possible to determine that a scenario is incorrect based on the assumption that the course of events is unlikely. The balance between plausible scenarios and special cases has to be considered. When the scenarios are designed, a possible way could be to let the users and developers work separately. After the separate design process, the scenarios could then be discussed and decided on in a plenary discussion before the risk identification starts.

The scenarios used in this case have no contextual description attached to them. It could be of value to put a scenario in its context and describe the assumptions made regarding the scenario, for example in terms of describe the working situation, if it is an ‘‘ordinary’’ day with acceptable numbers of patient or a very stressful day with a lot patients with severe traumas. The development organisation has concluded that there is a need for the scenarios to be discussed and firmly established in the risk analysis team and that attached contextual descriptions could be a part of that process.

When the scenarios were discussed step by step, it could be noted that the user representatives, as expected, are the dominant part, since they possess domain knowledge regarding the target environment and medical issues. The developers had a more peripheral role and were consulted regarding technical aspects of the system.

A possible solution to the dominance factor could be to have very strict control of the meetings, with the ambition to get the opinion from all the participants, for example give specific time slots to each participant.

6.4 Estimation

The qualitative nature of estimating the value of the risk quantities, in particular that it is based on the participants' subjective opinions makes the result quite uncertain. It is important to define and separate the estimation of different values, e.g. severity, probability and detectability, and to strictly apply the predefined scales.

Detectability was not estimated for the majority of the risks due to several reasons. The scale was considered imprecise and did not assist the participants in the estimation effort, as the scales for severity and probability did. Another problem was that the concept of detectability was not well understood. The used scale defines three levels of detectability: a risk is, never, sometimes or always detected. It was found that these words lack precision and are subject to personal interpretation. For instance, does always mean that a risk is always detected, most of the time or only when it can be observed? The scale gives a false impression that detectability can be measured quantitatively although it is a qualitative property. Instead of detectability, we would suggest that it is better to use observability. If a risk is observable, then it can be detected. Using the scale, a risk is either directly observable, indirectly observable, or unobservable.

In addition to the observed problems, it could be argued that detectability should be considered as a mitigating factor and be estimated during the risk treatment step. There exists at least two counter-arguments for this: first, the expert knowledge that is required to determine the detectability might not be available when risk treatment is performed; secondly, the detectability value would give additional information when prioritising risks for further analysis and treatment.

After completing the risk analysis the development organisation decided, based on the encountered problems, to remove detectability from the process. Although this simplifies the process, it removes potentially important information about risks. There is a need for further research on how to define and estimate detectability of identified risks.

6.5 Risk planning

The documented risk descriptions have an impact on the risk planning process, because the descriptions influence the understanding of the risk context. In the studied process, risk descriptions typically only contained a very short summary of the nature of the risk. To lower the risk of misunderstanding and misinterpretation later in the process, a solution might be to extend the risk descriptions with contextual information about the risk. For instance, a risk context document could be added and linked to the risk descriptions. It could describe, for example, where, when, and how the risk emerges, as well as who is operating the device. The additional information should make the risk easier to understand and remember in later parts of the risk process.

During the risk planning process, the discussions had a tendency to focus on the action proposals instead of the actual risks. A possible solution to that is to have very strict control of the meetings, e.g. disallow in-depth discussions about proposed actions. Instead, separate ''proposal meetings'' should be arranged if there is a need for in-depth discussions about action proposals. This solution was adopted by the development organisation.

Risk reduction actions increase the complexity of the system, which have implications for risk assessment. One particular challenge arises when there is a mix of actions that is already implemented and actions proposed to be implemented. A solution to this problem would be to wait with the implementation of actions until after all risks have been discussed and assessed. Furthermore, the process should specify how to handle situations were there is more than one root cause of a risk and how the proposed actions shall be managed for the different root causes.

6.6 The risk management process

The studied risk management process focuses on the user interface when software is involved. Sometimes, the focus might be too high on the user interface, according to one of the representatives from the development organisation. However, since it is well known that many risks are related to the usage of a system and the user interface, e.g. Dhillon (2008) reports that 50 % of technical medical equipment-

related problems are caused by operator errors, it is important that the user interface stays in focus.

The goal with the new risk management process for the development organisation was to get an effective process that is easy to adopt. In addition, they wanted a process that makes it possible to begin the risk management process earlier in a project. After Phase 3, the representatives from the development organisation stated that the new risk management process is now used in another project and that it was easy to adapt to that project.

6.7 Validity threats

Validity of this kind of study can for example be analysed with respect to construct validity, internal validity, external validity, and reliability (Yin 2003).

Construct validity reflects to what extent the factors that are studied really represent what the researcher have in mind and what is investigated according to the research questions. In this study, there were several different roles with different types of expertise involved. This could be a potential threat since there is always a risk of misunderstandings. One aspect that lowered this threat is that both the technical experts and the process experts had a long tradition of working together with the medical experts, which means that they had good knowledge of the investigated product and the usage of it. However, even if the technical expert and the process expert have this knowledge, the risk cannot be ruled out totally.

It can also be noted that if, for example, medical terms were misunderstood by the researchers or the process experts, this would probably be a larger problem for the result of the conducted risk analysis than for the research results presented in this paper. The research was conducted as part of the risk analysis attempt and not seen as something completely different by the participants. There was a wish to do as good a risk analysis as possible, which we also think is good for the research results.

Internal validity is important in studies of causal relationships. We have not identified any significant relations of this kind, which means that this risk is not seen as serious.

External validity is concerned with to what extent it is possible to generalise the findings, and to what extent the findings are of interest to

people outside the investigated case. The study was conducted with a limited set of participants from one single project. This means, of course, that the results cannot automatically be generalised to other organisations and projects. Instead, it must be up to the reader to judge if it is reasonable to believe that the results are relevant also for another organisation or project. Especially, an organisation that is used to risk management in general, but not for software systems, can be in a similar situation as was the case here. However, it should be noted that the focus on a specific case is the typical situation in a case study. The case is studied in detail in order to learn as much as possible from it.

Reliability is concerned with to what extent the data and the analysis are dependent on the specific researchers. The reliability was addressed by conducting both the data collection and the data analysis as a group of researchers instead of one single researcher. The preliminary results were also sent to the other participants in the form of a technical report. This made it possible for the other participants to find possible error by the researchers.

6.8 Key contributions

It can be concluded that the risk management method used in this case study has the potential to be used in a medical device development organisation or similar organisations.

Regarding the risk management method, it was found that the system boundaries must be carefully defined and the nature of the couplings between components identified. The system context can be described using quantitative measures, such as usage frequencies and example scenarios. By attaching contextual information to the scenarios, the risks are easier understood and remembered over time and the overall scenario quality may also be improved. Mixing development and user scenarios may also be considered to improve the overall scenario quality. During the analysis of the risks, the prescribed order of estimations should be strictly followed since it influences the outcome of the risk analysis. In the risk planning process, the risk analysis should be carried out prior to implementation to avoid unnecessary misunderstandings.

Future research regarding the risk management method is needed with respect to, for example, detectability, context limitation, and how to allow for flexible update of the product.

Acknowledgments

The authors would like to gratefully acknowledge the persons involved in this case study. The authors would also like to acknowledge Gyllenstiernska Krapperup-stiftelsen for funding the research studies of Christin Lindholm. This work was also partly funded by The Swedish Foundation for Strategic Research under a grant to Lund University for ENGROSS-ENabling GROwing Software Systems. Prof. Boris Magnusson is acknowledged for the support in the study and the writing of this paper.

References

Boehm, B. (1991). Software risk management: Principles and practices. *IEEE Software,* 8(1), pp. 32–41.

Bovee, M. W., Paul, D. L., & Nelson, K. M. (2001). A framework for assessing the use of third-party software quality assurance standards to meet FDA medical device software process control guidelines. *IEEE Transactions on Engineering Management*, 48(4), pp. 465–478.

Charette, R. N. (1989). *Software engineering risk analysis and management.* McGraw-Hill Software Engineering Series, New York: McGraw-Hill.

Chiozza, M. L., & Ponzetti, C. (2009). FMEA: A model for reducing medical errors. *Clinica Chimia Acta*, 404(1), pp. 75–78.

Commission of the European Communities (1993). Council Directive 93/42/EEC EEC.

Crouhy, M., Galai, D., & Mark, R. (2006). *The essentials of risk management.* Maidenherd: McGraw-Hill.

Dey, P. K., Kinch, J., & Ogunlana, S. O. (2007). Managing risk in software development projects a case study. *Industrial Management and Data Systems,* 107, pp. 284–303.

Dhillon, B. S. (2000). *Medical device reliability and associated areas.* Boca Raton: CRC press Taylor & Francis Group.

Dhillon, B. S. (2008). *Reliability technology, human error and quality in health care.* Boca Raton: CRC press, Taylor & Francis Group.

Fairley, R. E. (2005). Software risk management. *IEEE Software*, May/June, pp. 101.

FDA (1996). Do it by design: An introduction to human factors in medical devices.

FDA (2000). Medical Devise Use-Safety: Incorporating Human factors Engineering into Risk Management.

FDA (2005). Food, Drug and Cosmetic Act section 201(h).

Gall, H. (2008). Functional Safety IEC 61508/IEC 61511. The Impact to Certification and the User, In *Proceedings of IEEE International conference on computer systems and application,* pp. 1027-1031.

Garde, S., & Knaup, P. (2006). Requirements engineering in health care: the example of chemotherapy planning in paediatric oncology. *Requirements Engineering*, 11(4), pp. 265–278.

Habraken, M. M. P., Van der Schaal, T. W., Leistikow, I. P., & Reijnders-Thijssen, P. M. J. (2009). Prospective risk analysis of health care processes: A systematic evaluation of the use of HFMEA in Dutch health care. *Ergonomics,* 52, pp. 809–819.

Hall, E. M. (1998). *Managing risk: Methods for software systems development*. Reading: Addison Wesley.

Hegde, V. (2011). Case study: Risk management for medical devices. In *Proceedings of reliability and maintainability symposium (RAMS),* Lake Buena Vista, Florida, USA. pp. 1-6.

Jones, C. (1994). *Assessment and control of software risks.* Englewood: Prentice-Hall.

Leveson, N. G. (1995). *Safeware: System safety and computers.* Reading: Addison-Wesley.

Leveson, N. G. (2011). *Engineering a safer world: Systems thinking applied to safety, engineering systems.* Cambridge: MIT Press.

Leveson, N. G., & Turner, C. (1993). An investigation of the Therac-25 accidents. *IEEE Computer*, 26, pp.18–41.

Linberg, K. R. (1993). Defining the role of software quality assurance in a medical device company. In *Proceeding of 6th annual IEEE symposium on compute-based medical systems*, pp. 278–283.

Lindholm, C., Pedersen Notander, J., & Höst M. (2012). A case study on software risk analysis in medical device development, *In Proceeding of 4th software quality days (SWQD 2012)*, Vienna, Austria. Pp. 143-158.

McCaffery, F., McFall, D., Donnelly, P., Wilkie F. G., & Steritt, R. (2005). A software process improvement lifecycle framework for the medical device industry. In *Proceeding of 12th IEEE international conference and workshops of the engineering of computer-based systems (ECBS**0**05)*, pp. 273–280.

McCaffery F., Burton J., & Richardson I. (2009). Improving software risk management in a medical device company. *In Proceedings of international conference on software engineering (ICSE),* Vancouver, Canada. pp. 152-162.

McCaffery, F., Burton, J., & Richardson, I. (2010). Risk management capability model for the development of medical device software. *Software Quality Journal*, 18, pp. 81–107.

Rakitin, S. R. (2006). Coping with defective software in medical devices. *IEEE Computer*, 39(4), pp. 40–45.

Reason, J. (1990). *Human error.* Cambridge: Cambridge University Press.

Robson, C. (2002). *Real world research (2nd ed.).* Oxford, UK: Blackwell Publishers.

Runeson, P., & Ho¨ st, M. (2009). Guidelines for conducting and reporting case study research in software engineering. *Empirical Software Engineering*, 14(2), pp. 131–164.

Sayre K., Kenner J., & Jones P. (2001). Safety models: an analytical tool for risk analysis of medical device systems. *In Proceedings of 14th IEEE symposium on computer-based medical systems (CMBS'01)*, Maryland, USA, pp. 445-451.

Schmuland, C. (2005). Value-added medical-device risk management. *IEEE Transactions on Device and Materials Reliability,* 5(3), pp. 488–493.

Schneider, P., & Hines, M.L.A. (1990). Classification of Medical Software. In *Proceedings of the IEEE symposium on applied computing*, pp. 20–27.

Sommerville, I. (2007). *Software engineering (8th ed.).* Readings: Addison Wesley.

Svensson Fors D., Magnusson B., Gestega°rd Robertz S., Hedin G., & Nilsson-Nyman E. (2009). Ad-hoc composition of pervasive services in the PalCom architecture. *In Proceedings of the ACM international conference on pervasive services (ICPS'09),* pp. 83–92.

Vishnuvajjala, R.V., Subramaniam, S., Tsai, W.T., Elliot, L., & Mojedehbaksh, R. (1996). Run-time assertion schemes for safety-critical systems. In Proceedings of the 9th IEEE symposium on computerbased medical systems, pp. 18–23.

Walsh, T., & Beatty, P. C. W. (2002). Human factors error and patient monitoring. *Physiological Measurement*, 23(3), pp. 111–132.

Xiuxu, Z., & Xiaoli, B. (2010). The application of FMEA method in the risk management of medical devices during the lifecycle. In *Proceedings of 2nd international conference on e-business and information system security (EBISS),* China, pp.1-4.

Yin, R. K. (2003). *Case study research: Design and methods (3rd ed.).* Beverly Hills: Sage. Author Biographies

Paper V

Introducing Usability Testing in the Risk Management Process in Software Development

C. Lindholm and M. Höst

Abstract

Human beings make errors and that is nothing that we can avoid completely. We can however lower the risk of people doing wrong in situations where, for example, medical devices are used. The overall objective of the research presented in this paper is to investigate how usability testing can contribute to software risk management process in the medical device domain. Experience has been collected from both the risk management process and usability testing in a development project of a medical device. It can be concluded that usability tests can give valuable input to the risk management process. Usability tests can indicate risks that are not identified in the risk management process and render the possibility to verify if risks with high risk value actually cause the presumed problems.

Published in proceeding of the workshop of Software Engineering in Health Care (SEHC13), at ICSE 2013, San Francisco, USA, pp. 5-11.

1 Introduction

Medical devices and systems have an important role in today's health care and they are frequently used in different situations by different user categories. The software part in medical devices has increased over the years and plays a more and more dominant role.

A study by Walsh & Beatty (2002) shows that approximately 87% of all incidents in medical environments, where patient monitoring takes place, are due to human factors. To lower the incident rate it is thereby important to include human factors in different ways in the development process of medical devices. The purpose of this case study on a patient monitoring system is to investigate the possibilities of utilising usability testing as a contribution to the risk management process. Since risk management as well as usability are important areas in the development process of medical devices and other safety critical systems here is a need for research to investigate how these two areas can interact in a beneficial way and to implement to role of the user in different ways in the development process.

2 Background and related work

The user is a key player in the usability field and defined as "any human that might handle, operate and otherwise interact with a medical device though the device user interface" according to the standard IEC EN 62366, Medical Devices – Application of Usability Engineering To Medical Devices (IEC 2007).

Human factors engineering (HFE) is defined (ANSI/AAMI 2009). as the application of knowledge about human capabilities and limitations to design and development of devices, systems, tools, organisations and environments. Where as the process of human factor engineering (HFE) extends to all medical devices and has emphasis on risk management and lifecycle. There are several standards involving usability and ANSI/AAMI HE 75-2009, Human Factors Engineering – Design of Medical Devices (ANSI/AAMI 2009) and the third edition of the medical electrical equipment standard EN 60601-1 (EN 2006) are example of standards where usability is an integrated part of the standard.

Usability testing is regarded as a major technique for developers to use in the development process (Daniels et al 2007) in order to comply with Human Factors Engineering – Design of Medical Devices (ANSI/AAMI 2009) and EN 60601-1 (EN 2006). According to Dumas and Redish (1999) usability testing means focusing on the users and on how the users use the products to be productive. Usability testing is thereby a powerful method in system development based on prototyping (Kushniruk 2002). There is a difference between the usability engineering process and the risk management process, for example, in decision making. The risk management process defines unacceptable risks, while in the usability engineering process risk are associated with usability and the design and development process for the user interface (IEC 2007). A usability engineering process focuses on all known or foreseeable hazards related to the medical user interface and not only those with unacceptable risk, like risk management process mostly do.

3 Research method

The qualitative research in this paper is based on an empirical study in a real world setting, since process improvement activities in software engineering because of their complexity are very hard to study in isolation. The aim of qualitative research is to investigate and understand phenomena within its real life context (Robson 2002; Yin 2003).

3.1 Objective

The overall objective of the research in this case study is to investigate how usability testing can contribute to the software risk management process in the medical device domain. More specifically the objectives are as follows:

- To investigate what type of problems and potential risks can be identified through usability testing.
- To investigate if the problems and potential risks identified through usability testing are the same problems and risks identified during a risk management process.

- To examine how the results from the usability testing can be used in the risk management process.

The objectives are investigated in a single case study at a department at a large Swedish hospital that has extensive experience in developing and maintaining medical devices, but not devices including software. The development process of a patient monitor system with an intensive care unit (ICU) as the target environment was studied during the case study. The first objective is illuminated by the results from the usability tests, the second objective by the comparison of the results from the software risk management process and the usability tests and the last objective is based on the prior findings.

3.2 The case study context

The case study was conducted at a department at a large hospital developing and maintaining medical devices and was performed from the summer 2010 to spring 2012. The case study contains two main parts, the software risk management process and the usability testing. The focus in this paper lies on the usability testing and the conjunction between the usability testing and the software risk management process. The software risk management process is described in detail by Lindholm et al. (2012).

The risk management and usability testing was carried out on a patient monitor system for monitoring a patient´s intra-cranial pressure, calculate the cerebral blood flow and present it to the medical personal on a beside monitor. The patient monitor system consists of three main parts; a) Pressure sensor placed in the patient's skull, b) Patient monitor (connected to the pressure sensor) that presents and exports blood pressure values, c) Bedside monitor, the new device which import the blood pressure values from the monitor and calculate the cerebral blood flow. The patient monitor system includes both software and hardware, although the risk management process focuses only on the software, and the usability test only on the user interface for the medical staff.

3.3 Case study process

The overall case study contains two discussion phases and three data collection phases where the first usability test is part of Phase 2 and the second usability test is part of Phase 3, see Figure 1. The discussion

phases and Phase 1 focus only on the software risk management process (Lindholm et al. 2012).

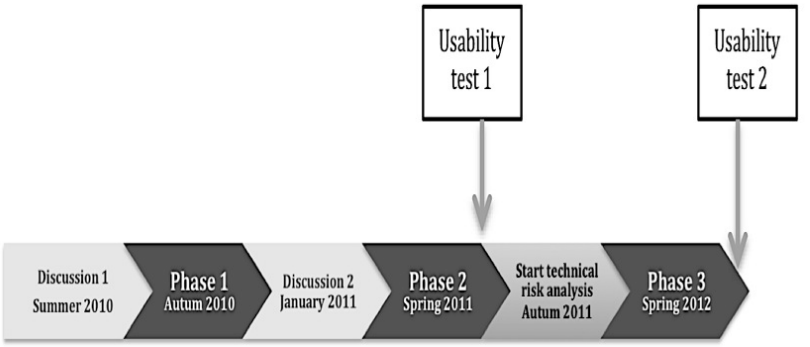

Figure 1. Case study timeline

The detailed case study process for the usability testing part is based on the case study process described by Runeson and Höst (2009). The objectives were defined and three research questions were identified, before the preparation of the two usability tests were done and the tests were carried out with participants. The data was collected and documented during the usability tests followed by the analysis of the documentation. Results reported from the usability test were sent to the development organisation and the results from the first usability test had impact on the changes of the user interface. Feedback from the development organisation was then received.

3.4 The usability testing

According to Nielsen (1992) is it enough to run a usability test with a small number of users (4 ± 1), and Virzi (1992) suggests that a usability test involving 5 participants can yield 80% of the possible findings. In this case study 4 test users participated in the first usability test, and 5 test users in the second. In the second usability test there were 5 test users available at the usability test occasion so it was decided to engage all five in the test. The test users in the first usability test were 2 nurses and 2 enrolled nurses in the age of 26-33. The selection criteria were that they had not worked with the tested system before, but were experienced in using monitor equipment. Gamer et al. (2009) describe that novices are important test persons since they encounters most of

the serious problems and also make the most errors. The test users in the second usability test were selected from the same premises and consisted of 3 nurses and 2 enrolled nurses in the age of 31-51. The test users for both test occasions were selected by the development organisation and the test facilitator and the observer prepared the test scenarios for the two usability tests. The aim of the performed usability tests was to find as many as possible of the most problematic problems. Different test scenarios were designed, for example that the test person should identify different curves on the screen, make notes and react to alarms. The usability tests were held at the intensive care unit at the hospital and preformed on the bedside monitor connected to a patient monitor. However the patient monitor was not connected to any patient. Instead the values were simulated with a simulator.

The test method that was used was "Active intervention" (Dumas and Redish 1999). However, the test person was also encouraged to think out laud (Nielsen 1992; Sharp et al. 2007) while using the system and verbalise her thoughts.

The test facilitator gave the test persons simple instructions about what to do, and encouraged them to express their thoughts. The test facilitator asked for example the test person to explain what she would do next and why. Each usability test session lasted for about 30 minutes and after each session the facilitator and observer took a few minutes to summarise and write down the things that struck them as complement to the log written during the test session. The first usability test identified 12 usability problems and the second usability test identified 16 usability problems. After each usability test the problems were presented in a test rapport supplemented with change suggestions. These reports were sent to the development organisation. The usability problems and the change suggestions were discussed by the development organisation and resulted in a major change of the user interface after the first usability test.

3.5 The software risk management process

The software risk management process, applied in this case study focuses on user risks and the first three first steps of the risk management process, i.e. risk identification, risk analysis and risk planning. There were in total 15 risk meetings held from the summer 2010 until the spring 2012. Three different groups of participants were

represented at the meetings; a) The intended users with special domain knowledge, e.g. physicians and nurses, b) the development organisation, e.g. medical device expert, risk analysis supervisor, and developers, and c) the researchers, e.g. process experts and technical experts from academia. At least two representatives from each group of participants were present at the meetings. For the first step, the risk identification, scenarios were chosen by the development organisation to be the main risk identification source. A scenario was defined as a chain of events, with a cause-effect relationship that describes a realistic diagnosis sequence during normal use. The risks were identified through brainstorming on each scenario and where all participants suggested possible risks connected to the specific scenario. Then each risk was assessed separately according to probability, severity and detectability. Scales predefined by the Swedish national board of health and welfare was used for probability and severity assessment and all identified risks were documented during the meetings. Both scales are four-graded (1-4). The risk value, R, was calculated for each risk by multiplying the probability, P, by the given figure for severity, S, i.e., $R = P \times S$. The highest risk value a risk in this study can have is $R = 4 \times 4 = 16$. Detectability was assessed according to the three following statements "if the fault (hazard) **always** could be detected before a severe situation occurred", "if the fault (hazard) **sometimes** could be detected" or "if the fault (hazard) **never** could be detected".

3.6 Data collection and analysis

The data collection from the software risk management process was carried out through active observations by the researchers at fifteen risk meetings. All the risks were documented during the meeting in Excel by the development organisation. In total 225 risks were identified out of which 25 risks were removed since they were not regarded as actual risks after more careful consideration.

The data collection from the usability tests was made at the usability test sessions at the intensive care unit, the first test in May 2011 and the second in May 2012. Each usability test took approximately 30 minutes and the observer logged all the actions. All observations were written down during the sessions and then transcribed on computer, resulting in reports on test results. The transcribed results were used by the facilitator and observer to identify the usability problems. First the

facilitator and the observer identified the problems separately, then they compared the results, discussed the identified problems and then discussions resulted in one list of usability problems for each test. The lists were complemented with change suggestions and resulted in written test reports that were sent to the development organisation.

The data in this study have been collected from the risk documentation from the risk meetings and the documented test results from the usability test sessions.

Each of the 26 identified usability problems were sorted into three different categories based on what functionality or feature each user problem was connected to. The three categories are:

A: Alarm, problems connected to the alarm function

C: Comments, problems connected to the commenting function

D: Different usability problems, problems connected to different functions.

The usability problems in each category was given a unique identifier, for example "A2-1,2", where A stands for the category A, 2 is a serial number and 1 means registered in usability test 1, and 2 registered in usability test 2 . After that, the usability problems were classified by using the failure qualifiers defined in the classification of usability problems (CUP) scheme by Vilbergsdottir et al. (2006) shown in Table 1. The usability problems can be classified differently, another way would for example as described by Keenan et al. (1999) be with primary categories and subcategories. It was decided to use the failure qualifiers (Vilbergsdottir et al. 2006) since they are straightforward, easy to understand, easy to categories after the usability test, and suitable for the user problems identified during the usability test. Each usability problem was documented with its unique identifier, a description of the usability problem, the failure qualifier, and the number of test persons that had that particular usability problem during the usability test. Each documented usability problem was then compared to each documented risk from the risk management process. For the usability problems where a corresponding problem was covered by a risk in the risk documentation, the usability problem was compiled together with the risk. To the documentation of the usability problems, the risk's unique identifier, the risk description, and the initial risk values was added. The usability problems were then sorted in two categories, those connected to an identified risk and those that were not connected to an

identified risk. This procedure was repeated again after usability test 2. Recurring usability problems were especially marked and sorted to a special category. Some risks were reassessed due to actions taken to lower the risk, the new risk values was also added to the corresponding usability problem.

Observer triangulation (Robson 2002) was implemented by having three researchers in the risk management part and two researchers during the usability test part of the case study. All collected data was treated confidential in order to protect the participants of the study and to ensure freedom during data collection. The participants have been very cooperative and were also given the right to review the findings and give feedback.

Table 1. Failure qualifier based on (Vilbergsdottir et al. 2006).

Abbre-viation	Explanation
M	**M**issing, when the test participant fails to find something in the user interface that she expected to be present.
IMM	**I**ncongruent **M**ental **M**odel, when the user interface is unclear, because it does not match the test participant´s mental model or her previous experience.
I	**I**rrelevant, when the user interface contains information/object that, while perhaps true, does not contribute to system services and is not needed
W	**W**rong, when the test participant can notice that something has gone wrong e.g. apparent programming bug.
B	**B**etter way, when the test participant suggests that something in the user interface could have been done differently.
O	**O**verlook. Sometimes the test participant is given a task but she overlooks an entity in the user interface i.e. the user does not see the existing entity or fails to realize that she is supposed to interact with it.

3.7 Validity

The construct validity concerns to what extent all people involved understand and use terms correctly in a consistent way. There is of course a risk that participants in the risk management or the usability study misunderstand each other and that the researchers misinterpret people in the study. We have been aware of this risk and tried to make sure that we understand the participants. The internal validity concerns to what extent causal relationships are misinterpreted or based on unknown factors. Since this type of relationship is not the focus of the study, this is not seen as a problem in the study. Concerning the reliability, the analysis is carried out by comparing identified risks and problems seen in the usability analysis. No major problems are seen with respect to this. The external validity is harder to judge since this is the first study conducted in one case setting. The results can probably be of interest for other projects where risk management is carried out for a medium sized software system. Especially, the results can be valid if the organisation is new to software development.

4 Results

4.1 Usability problems

During the two usability tests, 26 different usability problems were identified in total. Two of the usability problems were the same problems identified in both tests (i.e. A3-1,2 and C1-1,2 in Table 2).

The majority of the usability problems concern the commenting functionality. The user interface for the commenting functionality was changed between the two usability test, although there were still usability problems connected to the commenting functionality registered after the second usability test. Finding the function in the user interface, how to add a comment, the use of medical staff identification, and how to save a comment are some examples of usability problems registered with respect to the commenting functionality. Two of the users actually pressed the wrong button when trying to save a comment and then believed that they had saved it. Here it can be noticed that the physicians at the intensive care unit find it highly desirable that the all categories of medical staff adds comments in the system.

Usability problems were also found for the alarm functionality, such as how to interpret the alarm and how to reset the alarm when it started to signal. The alarm function is vital and since two of the users, one in each test round did not notice the alarm at all, the functionality was highlighted in the development organisation. There was an in-depth discussion about adding acoustic alarm as a complement to the visual alarm, but the final decision was to avoid acoustic alarm due to the risk of alarm fatigue. Compared to an ECG-machine for heart surveillance, an alarm on the bedside monitor is not equally unguent to attend to, which also favoured having only a visual alarm. The visual alarm functionality was however redesigned after the first usability test.

The usability problems are classified according to what type of problem as presented in Table 1. The classification shows that IMM - **I**ncongruent **M**ental **M**odel and O - **O**verlook are the dominant types of usability problems in this case see Figure 2. An IMM problem is when the user interface is unclear, because it does not match the test participant´s mental model or her previous experience, and an O problem is when the user does not see the existing entity or fails to realize that she is supposed to interact with it.

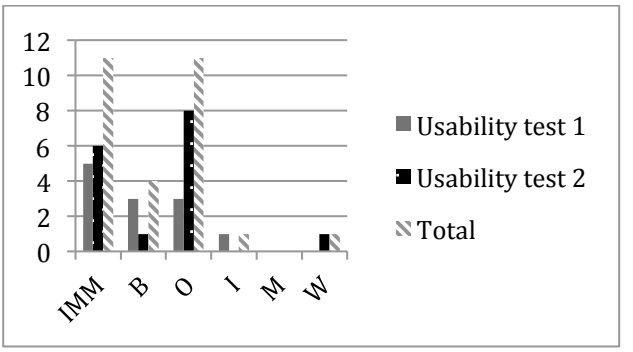

Figure 2. Number of usability problems per failure qualifier

It was quite obvious that the design of the alarm function and the commenting function was not in compliance with the users' previous experiences of these types of functions. Concerning the commenting function, the users had trouble seeing the existing entity (failure qualifier O), more precisely where and how to add a comment in the system and also the use of standard comments. The users came with

some suggestions about how things could be done better (failure qualifier B), and when it concerned the user manual they were unanimous regarding append more images in the manual.

4.2 Usability problems versus risks

If we compare the usability problems identified through the usability testing and the risks identified during the risk management process it is found that 11 out of the 26 usability problems had been identified as risks during the risk meetings. Out of these 11 usability problems, 3 usability problems were uniquely identified during the first usability test, 6 were uniquely identified during the second usability test, and two usability problems were identified during both usability tests, see Table 2. For the risks connected to the usability problems, severity, *S*, and probability, *P*, were estimated at the risk meetings, and the risk values, *R*, were calculated for each risk, $R = P \times S$. The highest risk value a risk in this study can have is $R = 4 \times 4 = 16$. Initially 2 of these risks had low risk values (R = 2, R = 4) and 4 had high risk values (R = 8, R = 9, R = 12, R =16) and 4 risks was given risk value zero, since the severity and probability for these risks was regarded very low. The second last usability problem in Table 2, i.e. D2 was identified as a risk but was regarded as a strict technical risk, so the estimation and handling of this risk was postponed to a later technical risk meetings. Risk value 8 was set by the development organisation as the limit for high risk-value All identified risks with risk value 8 or above were handled and dealt with. The two usability problems, A2-1 and A3-1,2 scored high risk values concerning the users' perception of the alarm function. The alarm functionality and its related risks rendered most discussions during risk meetings. There were different opinions among the participants, but the discussions resulted in a major redesign of the alarm functionality and the development organisation together with the users finally decided not to implement an acoustic alarm. The risk with only using visual alarm was put as residual risk. However the usability problem, C1-1,2, adding a comment in the system, was a problematic function for all four users in the first usability test and all five users in the second usability test. It was given a relativity low risk value (R = 4) during the first part of the risk management process with no redesign as consequence. The probability was set to 4, which corresponds well with the result of the usability test and the severity was set to 1, "discomfort

or minor injury to the patient". The physicians' great desire that the commenting function should be widely used by all the medical staff is not caught in the risk management process in the beginning. As a result of the first usability test, that pinpointed the problems with this functionality, a redesign was decided. The risk was reassessed and the probability value was lowered to 2, which resulted in a new lower risk value, R = 2. This was too optimistic since the second usability test performed on the new design and after the reassessment, showed that it still was a problematic function for all the test users. There were two more risks connected to usability problems that were reassessed (i.e. C9-2 and D6-2) due to actions taken to lower these risks. The reassessment resulted in low risk values (R=3, R=4) presented in Table 2 with italic, underlined figures. Interesting to notice here, is that the usability problems C9-2 and D6-2 were found by all of the participants during the second usability test. The action taken had not lowered the risks to extent as expected by the risk management group.

There were 15 usability problems that were **not** caught in the risk management process, 6 of them were identified in the first usability test, and 9 in the second usability test. These problems are presented in Table 3. All of these fifteen usability problems, found in the second usability test were all new problems, not found in the first usability test. It was mainly usability problems concerning the commenting function that was not documented as risks during the risk management process. There were also several problems in the D category, with problems for example regarding the touch screen, the user interface, and user manual, that was caught in the usability tests and that were not documented as risks during the risk management process. However, the problem that the users have with finding the commenting function was identified as a risk but with low risk value. On the other hand, the risk that the users would not find their way through the commenting function when for example entering text and saving the added comment was not identified as a risk.

For example two users thought that they had saved the comment they had entered in the system but they had not, since they pressed the wrong button for saving the comment. For four of the usability problems found in the second usability test, all five test users noticed them. Two of these faults concern the commenting function, i.e. C10-2 and C12-2, and two concern different interface functions, i.e. D7-2

and D9-2. After the first usability test the development organisation took all the found problems under consideration and the discussions led to actions regarding all the problems except replacing the touch-screen to a more sensitive one. The system was updated and a new version was released before the second usability test.

Table 2. Usability problems connected identified risks

Id. usability problem	Description usability problem	Failure qualifier	Risk S	P	R
A2-1	The user does not know the cause of the alarm, does not know how to interpret the alarm.	IMM	4 *3*	2 *2*	8 *6*
A3-1,2	A visual alarm was simulated; the user did not notice the alarm. The user does not see the entity.	O	4 -	3 -	12 *0*
C1-1,2	The user is given the task to add a comment in the system, but the user have trouble to find the way to do it. The user failed to find the way even if the entity existed.	O	1 *1*	4 *2*	4 *2*
C9-2	The users did not perceive the button to press for changing time for the comment. The user does not see the entity.	O	3 *3*	3 *1*	9 *3*
D2-1	The change of the graphs due to user action is unclear to the user.	IMM	-	-	-
D5-2	The users did not perceive that the graphical scales where changed. The user does not see the entity.	O	4 *4*	4 *1*	16 *4*

If we look at the different types of usability problem versus identified risks, we find that the dominant class is IMM-Incongruent Mental Model, when the user interface is unclear to the user, see Figure 3. There is a slight dominance of problems that were not at all highlighted in the risk management process and those that was highlighted in the risk management process but had got a low risk value, which meant that no action was taken according to them, although they proved to be a problem for the users. For the category O – Overlook, there is a slight dominance of problems highlighted in the risk management process which indicates that it is easier to identify items the users may overlook For the usability problems classified as B – **B**etter way, I – **I**rrelevant, and W – **W**rong, there were more usability problems identified as risks than not identified as risks. The users did

not find anything missing that they had expected to be there (i.e. M – Missing) when they took part in the usability test.

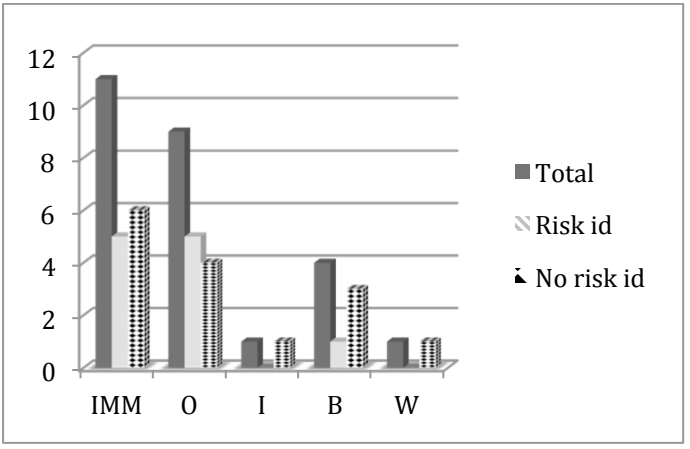

Figure 3. Usability problems per failure versus identified as risks or not

5 Discussion and conclusion

In the usability test there were two functionalities that generated most of the usability problems, it was the commenting function and the alarm function. The medical device used before (i.e. a sensor connected to the patient and a printer) and was replaced by the new bedside monitor had none of these two functionalities, so they were new to the users for this kind of monitoring. The users are used to alarms, but then mostly a combination of acoustic and visual alarms, and not only visual alarms as in this case. Commenting functions are available in some medical devices, for example in continuous EEG (Electroencephalography) monitoring of patients in intensive care, but it is not a common functionality in medical devices for monitoring overall.

One of the dominating types of usability problems found during the usability test were IMM, **I**ncongruent **M**ental **M**odel, when the user interface is unclear, because it does not match the test participant's mental model or her previous experience. For some of the functionality causing IMM problems, the users lacked experience, and for some of the functionality the users' mental models were not the same as the

developers'. The users expect the user interface to follow their logic and not the software's or the developers' logic, so when there is a mismatch it will show as a problem. Since active intervention was used during the test, it gave the test facilitator and observer a good understanding of the users' problems and also their mental model of the product. The other dominating type of usability problems were O, **O**verlook, the users do not see the existing entity or fails to realise that they are supposed to interact with it. The users and developers perceive things differently. Things that are obvious for the developers are not even noticed by the users, and the users see and interact with the medical device in their context and on the basis of their domain knowledge. In this case, user representatives have been part of the development process and the risk management process but there have not been representatives from the whole user spectra. Since the users are novices to the tested system and lack the experience from it, it may have affected their self-confidence and made them more critical and inclined to suggest improvements. If we look at the type of usability problems according to risk, for those problems both found in the risk analysis and during usability tests there was a dominance of the Overlook category. The participants in the risk management process identified more risks with users interacting and finding the functionality than risks concerning the users' mental model of the functionality and the workflow. The Overlook problems are probably more concrete and easier to imagine for the developers when looking at the user scenarios.

There were 15 usability problems found during the usability tests that were not identified as risks. Several of these usability problems imply risk and should be handled in the risk management process, especially the four usability problems that were found by all the users. If we then consider usability problems identified during both risk analysis and usability test there were four problems with high risk value, so there seemed to be a good match between high risk value and problems for the users. However not a total match because one of the risks (connected to usability problem A3-1, 2) had a high risk value but only one user in each usability test had that problem. The probability value for that specific risk was set quite high during the risk analysis and according the results from the usability tests it should maybe not have been give such a high value after all. There were also two identified risks with high risk values in the risk analysis but they were not identified as

usability problems in the first usability test. The design was changed without regarding the usability test results and these changes generated usability problems (C9-2 and D5-2) for all the users in usability test two. This indicates the need to verify if an identified risk really is a problem to the users before any changes are made.

Table 3. Usability problems not connected to identified risk

Id. usability problem	Description usability problem	Failure qualifier	Users that found the problem	Comment
C2-1	A text label on a button is not understood by the user, so the user does not press the button to perform the given task.	IMM	3	The fourth marked spontaneously that the button should have a better text label
C10-2	The standard comments in the system is not noticed by the users and therefore not used. The user does not see the entity.	O	5	
D1-1	The user did not notice the text information.	I	3 (4)	1 user did see the text information after a while and 3 did not see it at al.
D7-2	Users had trouble pressing the button "Back" due to its position on the screen.	W	5	

It can be concluded that usability tests can give valuable input to the risk management process. Usability tests can indicate risks that are not identified in the risk management process and give a possibility to verify if risks with high risk value actually cause the presumed problems. It is also possible to capture "problem functionality" e.g. for functionality that is new or unknown to the user. Usability testing also catches problems that are good risk candidates, where the functionality is unclear to the users and where the developers and the users have

different mental models. Timing is important when it comes to usability testing connected to the risk management process. The time must be right, so no changes are made only based on the risks, before the usability test is performed. The usability tests can for example verify that a risk with a high risk-value actually is a problem for the users before any changes are made. Risk values are assumptions so if they can be identified in additional ways before any action is taken, effort and time can be saved due to the avoidance of unnecessary changes.

References

ANSI/AAMI (2009). ANSI/AAMI HE75:2009 *Human factors engineering – design of medical devices.* Arlington VA: Association for the advancement of medical instrumentation.

Daniels J., Fels S., Kushniruk A., Lim J. & Ansermino J.M. (2007). A framework for evaluating usability of clinical monitoring technology. *Journal of Clinical Monitoring and Computing,* 21, pp. 323-330.

Dumas, J.S. & Redish, J.C. (1999). *A practical guide to usability testing.* Exeter: Intellect Books.

EN (2006). EN 60601-1, *Medical electrical equipment - Part 1: General requirements for basic safety and essential performance.* http://www.sis.se/en/health-care-technology/medical-equipment/general/ss-en-60601-1. 2013.

Gamer K., Liljegren E., Osvalder A-L. & Dahlman S. (2002). Application of usability testing to the development of medical equipment. Usability testing of a frequently used infusion pump and a new user interface for an infusion pump developed with Human Factors approach. *International Journal of Industrial Ergonomics*, 29, pp. 145-159.

IEC (2007). IEC 62366:2007, *Medical devices – application of usability engineering to medical devices.* http://www.iso.org/iso/catalogue_detail.htm?csnumber=38594. 2013.

Keenan, S.L., Hartson, H.R., Kafura, D.G, & Schulman, R.S. (1999). The usability problem taxonomy: A framework for classification and analysis. *Empirical Software Engineering,* 4, pp. 71-104.

Kushniruk, A. (2002). Evaluation in the design of health information systems: application of approaches emerging from usability engineering. *Computers in biology and medicine,* 32, pp. 141-149.

Lindholm, C., Pedersen Notander, J. & Höst, M. (2012). A Case Study on Software Risk Analysis in Medical Device Development. In *Proceedings of Software Quality: 4th International conference (SWQD 2012)*, pp. 143-158.

Nielsen, J. (1992) The usability engineering life cycle, *Computer,* 25(3), pp. 12-22.

Robson, C. (2002). *Real world research* (2nd ed.). Oxford UK: Blackwell Publishers.

Runeson, P. & Höst, M. (2009). Guidelines for conducting and reporting case study research in software engineering. *Empirical Software Engineering,* 14(2), pp. 131-164.

Sharp, H., Rogers, Y. & Preece, J. (2007). *Interaction design: beyond human-computer interaction (2nd ed.)* West Sussex: John Whiley & Sons, Ltd.

Walsh, T. & Beatty, P. C. W. (2002). Human factors error and patient monitoring. *Physiological Measurement*, 23(3), pp. 111–132.

Vilbergsdottir, S.G., Hvannberg, E.T. & Law, E. L-C (2006). Classification of usability problems (CUP) scheme augmentation and exploitation. In *Proceedings of NordiCHI 2006*, pp. 281-290.

Virzi R.A. (1992). Refining the test phase of usability evaluation: How many subjects is enough? *Human Factors,* 34(4), pp. 457-471.

Yin, R. K. (2003). *Case study research: Design and methods* (3rd ed.). Beverly Hills: Sage.

Paper VI

Validation of a Software Risk Management Process, Involving User Perspective

C. Lindholm

Abstract

The medical device domain constantly evolves and changes and more and more user groups use medical devices. In addition to medical professionals, patients, relatives and the general public are also using medical devices, for example heart starters that are available at public places and different mobile medical applications. Users handling medical devices make errors, but by involving users in the risk management process it is possible to lower the risk of these errors.

This paper presents an evaluation of the value of complementing a traditional risk management process with emphasised user perspective and the evaluation of parts of the new risk management process is made in an organisation developing medical devices. The main goal of the new risk management process is to integrate users and user perspective and to introduce usability testing, as an included part in the process. The research was conducted as action research with the aim to evaluate the user perspective parts of the new risk management process.

In conclusion the proposed risk management process is found to support the practitioners in their work with user risks and risk management.

.

Submitted to a journal

1 Introduction

A medical device that fails can bring harm to both patients and medical professionals. In the risk management process a major challenge is to assure safety and prevent harm. Research indicates an increasing importance and use of software in medical devices (Allen 2014; Bovee et al. 2001; Chunxiao et al. 2013; Méry & Kumar Sigh 2010). Software, medical devices connected to each other and users are contributing to the complexity of the medical device domain. Users handling medical devices make errors, but involving users in the risk management process can lower the risk of user errors. More and more user groups, in addition to medical professionals, patients, relatives and the general public, use medical devices. The medical device domain constantly evolves and changes. Mobile applications that can help people manage their own health and promote a healthy living are now being used and added to the medical device spectra. Recently the Food and Drug Administration (FDA) has issued guidance for mobile medical applications (FDA 2013) to meet the changing domain.

To be able to handle user risks in a comprehensive way, different user groups with different perspectives, need to be involved in the risk management process. This paper introduces the first version of RiskUse, a medical device software risk management process with an emphasised user perspective. The risk management process is developed in close contact with the development organisation and the purpose of the risk management process is to provide practitioners, mainly risk managers with a software risk management process that has a defined user perspective, is easy to apply and includes hands-on recommendations on how to use the process. The aim is also to present a risk management process that allows the development organisation to perform adequate risk management activities that can ensure that the developed software is safe from a user perspective. The main goal is to integrate users with different perspective in the software risk management process and to introduce usability testing, as an integrated part in the risk management process contributing to the goal of involving end-users in the process.

RiskUse is developed based on contributions from prior research, more specifically based on empirical knowledge about the state of practice, human factors and case study results of the risk management

process (Lindholm et al. 2014), and usability testing (Lindholm & Höst 2013) complemented with in-depth studies of risk management regulatory requirement, standards and guidelines within the medical device domain. The terminology used in RiskUse is adapted to the terminology used by regulatory bodies and standards within the medical device domain.

The evaluation of the first version of RiskUse was carried out in a case study using an action research approach, according to observations of risk meetings, supplemented with interviews and observations of usability testing. The goal of the case study presented in this paper, is not only to evaluate the risk management process, but also to make improvement proposals to the development organisation, based on the results from applying RiskUse in a medical device development project. The results can also be used to further improve RiskUse.

The outline of this paper is as follows; related work is presented in Section 2 followed by the presentation of objectives and research design in Section 3. The developed risk management process, i.e. RiskUse, is presented in Section 4 and the results are presented in Section 5. Finally, Section 6 presents discussions and conclusions.

2 Related work

One of the challenges an organisation developing medical software have, is to identify a sufficient set of risks for their products. Given potential for harm, inadequate medical device software can cause, the risks have to be successfully addressed in the work with safety and risk management. The organisation addresses different risks regarding patients, users, the environment, and third parties (for example service technicians) (Ratkin 2006). Another challenge for the development organisations is to comply with the regulatory requirements of the country in which they wish to market their medical devices. How strict and detailed the manufacturer's processes have to be depends on the safety classification of the product. Specific standards regarding risk management are ISO 14971 (ISO 2012) for application of risk management to medical devices, and IEC 80001-1 (IEC 2010) for application of risk management for IT-networks incorporating medical devices. Users and usability are important factors to consider in the medical devices domain, and IEC 62366 (IEC 2007) is a standard for

application of usability to medical devices. To provide some high-level guidance in achieving regulatory compliance there is guidance document published. IEC/TR 80002-1 (IEC/TR 2009) guidance on the application of ISO 14971 to medical device software, FDA's Do it by design, an introduction to human factors in medical devices (FDA 1996) and Medical device use-safety, incorporating human factors engineering into risk management (FDA 2000), are examples of such guidance documents. However, no real detailed guidance or special methods are provided by the authorities, demonstrating how regulatory compliance shall be achieved. The risk management process, RiskUse, presented and evaluated in this paper is meant to offer a process with detailed guidance how to perform adequate risk management activities incorporating users, usability and usability testing in the process.

Several approaches and strategies are used in order to address risk management within the medical device domain. Fault Tree Analysis (FTA) (Hyman 2002; IEC 2006a; Krasich 2000) and Failure Modes and Effects Analysis (FMEA) (Chiozza & Ponzetti 2009; IEC 2006b; Jain et al. 2010; Xiuxu & Xiaoli 2010) are often used for tracing possible risks in medical devices software or systems. FTA is a top–down analysis method where undesirable end events are identified and then all contributing factors, determine which failures are most critical. Whereas FMEA is a bottom-up analysis method used to identify each potential failure mode for all the parts in the system and trace negative effects though up the system. Failure Modes and Effects Criticality Analysis (FMECA) is an extension to FMEA where the severity ranking of the failure modes is made and allows prioritisation of countermeasures (Becker & Flick 1997). Another failure mode method is Healthcare Failure Mode and Effect Analysis (HFMEA™) developed by the United States Department of Veterans Affairs' National Center for Patient Safety, it is based on multidisciplinary teams, identifying possible failure modes using graphical described health care processes (Habraken et al. 2009). Hazard and Operability Studies (HAZOP), not so widely used, is a qualitative method for identifying hazards and operational problems with the use of guide words (more, less etc.) (McDermid 1995). The medical device regulatory requirements require production and postproduction monitoring of the medical device for discovering additional or unexpected severe risks. Corrective Preventive Action (CAPA) system is used in some cases, to collect, organises and

trace failures. The failures are evaluated for risk, severity and necessary action (Bills & Tartal 2008; Lozier 2010).

When working with risk management in the medical device area (Dhillon 2008) there are several critical factors that relate both to the medical device and the usage of the device, such as design, manufacturing including quality control/quality assurance, user training, interaction with other devices, and human factors. Walsh & Beatty (2002) refer to a wide range of studies that show that 87% of critical incidents connected to patient monitoring is due to human factor errors. The concept human factors are described by the FDA as "a discipline that seeks to improve human performance in the use of equipment by means of hardware and software design that is compatible with the abilities of the user population" (FDA 2000). In order to get safe and effective medical devices, human factor considerations regarding the user environment, the users and the medical device itself, have to be a part of the risk management process (FDA 2000). In the European standard IEC 62366 (IEC 2007) the term usability engineering is used. The terms human factors engineering and usability engineering are often used interchangeably for the process of achieving highly usable devices. The user is a key player in the usability field and defined as "any human that might handle, operate and otherwise interact with a medical device though the device user interface" (IEC 2007). According to Dhillon (2008) human errors in health care are the eighth leading cause of death in the US and the costs are high and more than 50% of technical medical equipment-related problems are caused by operator errors (Dhillon 2000). To minimise user errors and understand user-related risks, it is important to have a complete understanding of how a device will be used and the goal with incorporating users in the risk management process is to minimise usage-related hazards so the intended users can safely use the medical device. The users and user participation is therefore an important part of the risk management process presented in this paper.

Various researchers have presented risk management in software development in general, over the years (Boehm 1991; Bubenski 2014; Charette 1998; Hall 1998; Jones 1994). In the medical domain the published research in most cases covers the whole risk management process at a high level, or focus on one of the steps in the risk management process. McCaffery et al. (2019, 2010) focus on the whole

risk management process and have developed and evaluated a software process improvement risk management model (Risk Management Capability Model) that integrates regulatory medical device risk management requirements with the goals and practices of the Capability Maturity Model Integration (CMMI). Schmuland (2005) investigates residual risks, i.e. the remaining risks after the risks have been handled, and how to assess the overall residual risk of a product. It is based on the identification of all the important scenarios. Hegde (2011) presents a case study of risk management based on ISO 14971 and concludes that the standard as guideline can ensure a safe product with an acceptable level of risk. McHugh et al. (2014) are studying the use of agile practices when developing software in the medical device domain and concludes that insufficient coverage of risk management activities are considered as one of the barriers to agile adoption, by the development organisation. However, McHugh et al. (2014) have concluded that the FDA General principle of software validation, accept iterative software development models and that they thereby enables for the use of agile practices. When agile processes are tailored to meet the need of regulated environments and supported by the appropriate tools, the agile approach is highly suitable in a regulated environment according to Fitzgerald et al. (2013). Gary at al. (2011) are also arguing that agile practices can contribute to safety critical software development and that they allow including activities related to risk reduction such as FTA and FMEA.

3 Research methodology

The aim of flexible research design also called qualitative research design is to investigate and understand phenomena within its life context (Robson 2002; Yin 2003). It relies on changes to the research design based on new information during the study process, e.g. change of research questions and data sources, and the design is intended to evolve over time as the researchers gain more knowledge (Robson 2002). The qualitative research presented in this paper is based on a case study in a real world setting at a large Swedish hospital, thus have extensive experience in developing and maintaining medical devices. The flexible research design (Robson 2002) is reflected in the design of the case study, where the research strategy has been allowed to develop

during the data collection and analysis and the interviews were made according to a semi-structured approach.

The aim of the performed case study (Robson 2002; Runeson et al. 2012) was to perform a formative evaluation of parts of the new risk management process, RiskUse presented in Section 4. The aim was also to get experience and feedback for further improvement and development of the process and to ensure that RiskUse is both usable and useful (Rogers et al 2011). The intention of a formative evaluation is to support the development process and to improve (Robson 2002). The study was based on participatory research (Robson 2002) where the researcher collaborated with the participants in the risk management process, the usability testing and in the interviews.

The evaluation was carried out using an action research approach, according to the observations. Avison et al. (1999) recommend action research when the research is done in a real life context, since it is a way to associate research and practice, through an iterative process involving both researchers and practitioners working together in different activities. The researcher works iteratively to try out theories in real situations with practitioners. Gain experience and feedback and then modify the theories and try the theories again. The goal of the case study was to evaluate parts of RiskUse and to make improvement proposals to the development organisation, based on the results from applying RiskUse in a project. Action research according to Shull et al. (2008) aims to intervene in the studied activities and to improve the situation in contradiction to other empirical research methods that only attempt to observe the situation as it currently are. Therefore action research was considered applicable and used. The research process is described in Section 3.2.

3.1 Objective

The main objective of this study was to evaluate the user perspective parts of RiskUse, i.e. to evaluate the value of complementing a traditional risk management process with an emphasised user perspective. With the aim to evaluate parts of RiskUse in a real life context, the following research questions were addressed:

RQ1: what value can a risk management process, involving a user perspective bring to an organisation developing medical devices? More specific:

a) Use cases – how can predefined use cases improve a risk identification process?
b) Risk control - how can risk control with user involvement, be implemented in a risk management process?
c) Usability testing - is it beneficial to in cooperate usability testing in the risk management process?

RQ2: how can the proposed risk management process be further improved?

User perspective in the research questions refers to the use of use cases as input in the risk identification, users as participants at the risk meetings and usability testing as input to the risk control part of the risk management process.

Traceability and documentation are important parts of a risk management process. Since these two parts were introduced in a new way to the development organisation, traceability and documentation were also evaluated during the case study presented in this paper.

3.2 Research design

RiskUse presented in Section 4 are the result of a research conducted in several steps outlined in Figure 1. The boxes with a folded corner in Figure 1 present the source of the input data, the box with rounded corners, represents documents and the boxes with frame represent RiskUse. A combination of empirical research methods has been applied.

To successfully transfer knowledge and technology from research to practice, close cooperation and collaboration between researchers and practitioners is needed (Seaman 1999). However, it is generally challenging to transfer research results into industrial practice (Zhang & Xie 2013) and the process of transfer is complex, involving many roles and phases (Buxton & Malcolm 1991).

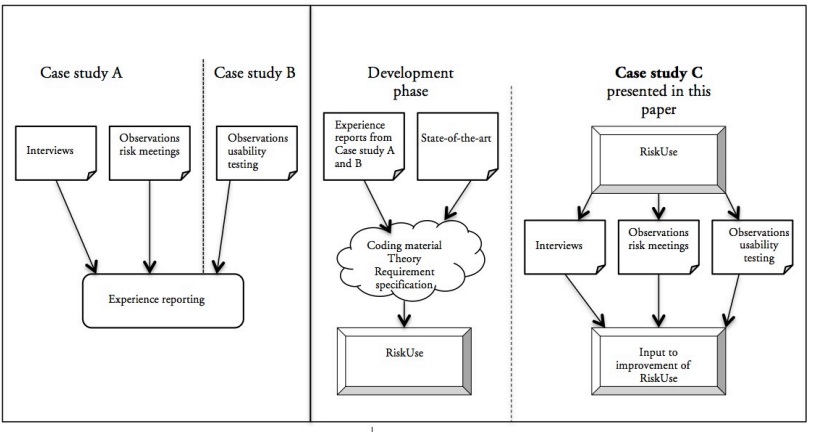

Figure 1. The research process

The risk management process, RiskUse was developed in close contact with the development organisation that has extensive experience in developing and maintaining medical devices, but not with devices including software. A limited candidate risk process was developed in cooperation between the researcher and the organisation, covering the three first steps of the risk management process, risk identification, risk analysis and risk planning (e.g. Case study A). The data collection, analysis and results from Case study A is presented by Lindholm et al. (2014) presenting observations from the risk meetings and the interviews. To further involve users in the risk management process usability testing was used as an integrated part of the process and validated in Case study B presented by Lindholm and Höst (2013). Usability testing was chosen because usability testing is considered as one of the most powerful ways (Daniels et al. 2007) and perhaps the most powerful one (Kushniruk 2002) to evaluate usability. With the aim to further develop the limited candidate risk process, state-of-the-art were further studied and accounted for, with special focus on laws, regulations, standards and guidelines within the field. Based on experiences from Case study A and Case study B and from the state-of-the-art studies theory was generated, a requirement specification was written and RiskUse was developed based on the requirement specification (Development phase in Figure 1). Before RiskUse was evaluated through the case study presented in this paper, the differences between RiskUse and the risk management process used before by the

development organisation were identified and documented by the researcher. The differences are presented in Section 3.3.1. The data collection was made through observations of risk meetings, interviews and observations of usability testing and the results and finding are presented in this paper. The findings are also used to further improve RiskUse.

3.3 The context

All three case studies presented in Figure 1 were conducted at the same large Swedish hospital together with the development organisation that has extensive experience in developing and maintaining medical devices but minor experience with devices including software. The development project was the same case project in both case study A and B but at different stages of the project. In case study C, the project was another development project involving another medical device system and its context is presented below. For further details on the context of case study A and B see Lindholm et al. (2014) and Lindholm and Höst (2013).

3.3.1 Case study C context

This section presents the study context of case study C. RiskUse was evaluated in case study C and the evaluated parts of the new risk management process is described in detail in Section 4. The case study was conducted during spring 2014.

The researcher identified the main difference between RiskUse and the risk management process used before in the development organisation. The main differences were:

a. The development organisation did not use, use cases in the risk management process.
b. The development organisation had no defined process to create traceability between requirements, hazards and other documents.
c. The development organisation had no defined process to reuse of parts of the documentation.
d. The development organisation had no defined process for risk control.

e. The development organisation did not use usability testing as part of the risk management process.
f. The development organisation had a different scale for assessment of probability.
g. The development organisation had no specified time limits for the risk meetings,.
h. The development organisation had no uniformed content design of the risk management report and risk management plan.

The target system in case study C is a system for advanced care of patients in their home environment (the itACiH project), where IT/MT Service department, Region Skåne are responsible for the regulatory parts. The system is under development. Before this type of patients had to be hospitalised and treated, but with this technology the idea is that they can stay at home instead and be monitored and treated. The intended use of the system is to be a connection between multiple devices (home units) for medical purpose at the patient's home, mobile units and a ward unit. The system is not a medical record system; the medical information is documented separately in the patient's medical record. It is trained medical professionals that use the system and for the use of the home units, the users are trained persons surrounding the patient. The main system components are presented in Figure 2.

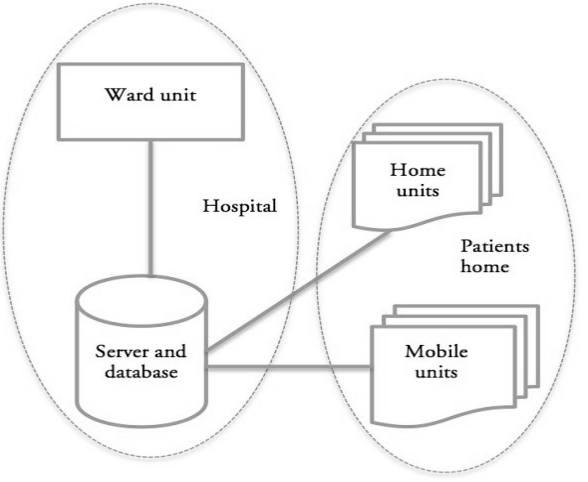

Figure 2. System components

Ward unit: presents real-time incoming data and historical data from the mobile units and the home units. The unit is also a secondary warning device for the devices connected to the system. The ward unit is placed in the ward at the hospital responsible for the treatment of the patient.
Server and database at the hospital: sorts and storages the incoming data from the other units, for example images, measured values and different kinds of documentation.
Home units placed at the patient's home: Sends data to the server. The development organisation's software developers have specially developed the home units and the type of unit depends on the need of the patient. The home unit can, for example, be a weighing machine, a device for measuring blood pressure or a video camera.
Mobile units: communicates with the server. The mobile units contain for example checklists for different medical activities and tools for valuation of the patient's condition. The medical professionals bring the mobile units with them when they are going home to the patients.

The risk management process was carried out on packages of functionality on the ward unit and mobile units, on adding home units to the system and on editing and communicating drug prescription lists between the different units (e.g. ward unit, mobile units and the server). The development organisation's overall risk management process is designed to comply with ISO 14971 (ISO 2012) but each step in the risk management process was not tailored and tried out for the case project. During the case study there were some changes according to risk management personnel. This was reflected in that different procedures had been used during the project. The change of personnel during the case study did not affect the use of the new risk management process, but it allowed the involved researcher to be of assistants in writing the risk management report and also update the risk management plan according to the new risk management process. The development organisation found new risk management process beneficial and decided to continue to use the new process.

The participants at the risk meetings represent three different groups: *intended users* with special domain knowledge (e.g., physician and nurse), the development organisation (e.g. risk manager and software developers) and researcher (e.g., process expert from

academia). At this stage, no patients were involved in the risk management process.

The usability testing was carried out on one of the tools on the ward unit. The tool is based on a guide on how to treat a patient in palliative care. 5 test users participated in the usability test, 4 nurses and 1 enrolled nurse According to Nielsen (1992) is it enough to run a small number of test users (4 ± 1) and Virzi (1992) suggests that a usability test involving 5 participants can yield 80 % possible findings. The selection criteria were that the test users have none or very little experience of the tested tool. The test users were selected by the development organisation. The test facilitator and the observer prepared the test scenarios for the usability tests with the aim that the performed usability test should find as many as possible of the substantial problems.

"Active intervention" (Dumas 1999) was the test method that was used, when the test persons, for example, is asked to explain what they would do next and why, as they perform the tasks. However the test person was also encouraged to think out loud (Nielsen 1992; Roger et al. 2011) while using the tool and verbalise her thoughts. The test facilitator gave the test persons simple instructions about what to do, and encouraged them to think out laud. The test facilitator for example asked the test person to explain what she would do next and why. Each usability test session lasted for about 30 minutes and after each session the facilitator and observer took a few minutes to summarise and write down the things of interest as complement to the log written during the test session. After the usability test were the problems presented in a test rapport supplemented with change suggestions and sent to the development organisation. The usability problems and the change suggestions were discussed by the development organisation and resulted in a change of the user interface.

3.4 Data collection and analysis

The data collection and analysis from the Development phase and Case study C displayed in Figure 1 is presented in this section. All collected data were treated confidentially in order to protect the participants and to ensure that they felt free to speak during data collection.

In the Development phase the data was gathered from two sources: from the experience reports from Case study A and Case study B and state-of-the art document studies (e.g. laws, regulations, standards and guidelines).

The data in Case study C was gathered from three sources: from interviews with the development organisation, observations from risk meetings and observations from the usability testing.

3.4.1 Experience reports and state-of-the-art

The experience reports in Figure 1 come from the three main sources: observations during risk meetings, interviews with the development organisation presented in Lindholm et al. (2014) and observation from usability testing presented in Lindholm and Höst (2013). The results and experiences in the experience reports were coded with codes presented in Table 1.

The state-of-the-art in Figure 1 include studies of different laws, regulations, standards and guidelines related to risk management and usability in the medical device domain. Each document was read and all data related to risk management and usability was extracted and coded with the codes in Table 1. Six of the standards, for example, ISO 14971 (ISO 2012) and IEC 62366 (IEC 2007) stand for the majority of the collected data.

The data from the experience reporting and the state-of-the-art documents was grouped and interpreted by the researchers. The outcome of the analysis was then used to build theory, then write a requirement specification for the risk management method and then develop the method RiskUse.

Table 1. Codes for collected data

Abbr.	Code	Explanation
P	Preparations	Preparations of and input to the risk meetings
R	Risk team	Structure of the risk team
RM	Risk meeting	The form and documentation from the meetings
U	Use cases	Factors effecting use cases
SO	Scales overall	Information about different scales
S	Severity	Different scales and measures of severity
P	Probability	Different scales and measures of probability
D	Documentation	Document in the whole risk management process
I	Identification	Factors effecting risk identifications
A	Analysis	Factors effecting risk analysis
C	Control	Factors effecting risk control
M	Monitoring	Factors effecting risk monitoring
T	Traceability	Information about how to achieve traceability

The data from the experience reporting and the state-of-the-art documents was grouped and interpreted by the researchers. The outcome of the analysis was then used to build theory, then write a requirement specification for the risk management method and then develop the method RiskUse.

3.4.2 Interviews

The interviews in case study C were divided into two sessions: four interviews held before the risk meetings and four interviews held after the risk meetings. The interview sessions took place at the development organisation during the spring 2014 except one that was a telephone interview. The interviews made before the risk meetings were conducted in order to record the development organisation's experiences from the risk management method former used in projects, (e.g. before the new risk management process was introduced). The interviewees were participants from the development organisation and

the specific project: one risk manager, one user representative (e.g. nurse) and two software developers. In the second session of interviews were the interviews made in order to understand the organisations experiences and apprehension of the introduced risk management method. The interviewees had the same professional roles as the interviewees in the first interview session.

A semi-structured interview approach (Robson 2002) was used for all the performed interviews. The questions were predefined and open-ended and the interviews were conducted as an open dialog between the researcher and the interviewees. The respondents were allowed to talk freely after each question and in some cases follow-up questions were posed. The questions and example of follow-up questions are presented in Figure 3. The interview guide was discussion among the researchers before the interviews were held.

All the interviews were conducted in Swedish, face-to-face and recorded by the same researcher. The recordings were later transcribed and analysed according to Runeson et al. (2012). The transcribed material was coded with the codes presented in Table 1 and the coded statements was then grouped and discussed by the researchers.

3.4.3 Observations from risk meetings

The data collection during the risk meeting in case study C was conducted through active observations by the researcher.

The active observations took place during spring 2014 at three risk meetings. The purpose of the interaction was to capture interesting aspects regarding parts of the new risk management process. It is important to ensure that those being observed are not constantly thinking about that they are observed (Seam 1999). Since the case study has an active research approach the practitioners and researcher worked together and the researcher, aimed to be a natural part of the team as much as possible. The researcher, directed the risk meetings in collaboration with the development organisations risk manager and the researcher also looked for and logged signs indicating that the participants' way of acting was affected of the researcher being present.

Interview guide

Questions before the risk meetings (QB)

QB1. What professional role do you have?
QB2. What advantages do you see in the risk management process?
QB3. Any parts of the risk management process that could be improved?
QB4. What challenges do you see in this project regarding the risk management process?
QB5. What improvements would you like to achieve with a new risk management process?

Questions after the risk meetings (QA)

QA1. What are the main differences compered to before?
QA2. What is difficult with the new risk management process?
QA3. What improvements can you see?
QA4. What challenges do you see in the new risk management process? And in risk management in general?
QA5. Is it difficult to engage participants to attend and be active at risk meetings?

Figure 3. Interview guide

During the risk meetings, the researcher documented the observations on paper and it was both direct observations and personal reflections. The notes were short statements with findings like "discussions about if the risk that should be registered should only be risk within the scope (6 month) or for the system in the future", "new preconditions resided to the use case", etc. After the last risk meeting the notes were compiled into a list of statements and each statement was then coded with the codes in Table 1. The coded statements were then grouped, interpreted and discussed among the researchers.

Relevant data from the different sources e.g., interviews and active observations were compered and triangulated. Interpretation of the collected data involved identifying the parts of the data relevant to a

specific research question. The result from the analysis can be found in Section 5.

The results and conclusions were coordinated with representatives from the development organisation to get clarification and confirmation of the material.

3.4.4 Observations from usability testing

The data collected from the usability tests in case study C was made at the usability test sessions during May 2014. The tests were performed on the target system used in the hospital ward, where the tests also took place. The observer logged all action during the usability test session and each session took approximately 30 minutes. The observer wrote down all the observations during a test session and after the session discussed the facilitator and the observer the session. All the notes from the sessions were transcribed and compiled into a test results report. The facilitator and the observer then used the test report to identify usability problems. The facilitator and the observer first identified the problems separately, after that was the identified problems discussed and the result from the discussion was a list of usability problem for each test case complemented with change suggestions. Each of the 16 identified usability problems were sorted into different categories based on what functionality each user problem was connected to. The usability problem in each category was given a unique identifier (e.g. serial number) and then the classification of usability problems (CUP) scheme by Vilbergdottir et al (2006) shown in Table 2, was used to classify the usability problems by using failure qualifiers.

The unique identifier, a description of the usability problem, the failure qualifier and the number of test persons that had that particular usability problem during the usability test, was recorded for each usability problem. Each of these usability problems was then compared to each identified risk in the risk management process regarding the functionality tested for usability. When there was a usability problem corresponding to an identified risk, the risk and the usability problem were compiled together by adding the unique risk identifier, the description of the risk and the initial risk value to the information about the usability problem. The usability problems were then sorted into two categories, those connected to an identified risk and those that were not.

Table 2. Failure qualifier based on Vilbergsdottir et al. (2006)

Abbre-viation	Explanation
M	**M**issing, when the test participant fails to find something in the user interface that she expected to be present.
IMM	**I**ncongruent **M**ental **M**odel, when the user interface is unclear, because it does not match the test participant´s mental model or her previous experience.
I	**I**rrelevant, when the user interface contains information/object that, while perhaps true, does not contribute to system services and is not needed
W	**W**rong, when the test participant can notice that something has gone wrong e.g. apparent programming bug.
B	**B**etter way, when the test participant suggests that something in the user interface could have been done differently.
O	**O**verlook. Sometimes the test participant is given a task but she overlooks an entity in the user interface i.e. the user does not see the existing entity or fails to realize that she is supposed to interact with it.

Observer triangulation (Robson 2002) was implemented by having two researchers in the usability test part of the case study and the interpretation of the collected data involved identifying the parts of the data relevant to the specific research questions (e.g., RQ1c and RQ2). In Section 5 is the result from the analysis presented.

3.5 Validity

In this section, threats to validity in relation to the research design and data collection in Case study C are discussed and also the steps to mitigate the threats. For details on the validity discussion concerning Case study A and B see Lindholm et al. (2014) and Lindholm and Höst (2013).

The threats to validity are discussed according to four perspectives on validity proposed by Yin (2002) construct validity, internal validity, external validity and reliability. Validity must be addressed in all research steps prior to the analysis phase and not only in the analysis phase (Runeson et al. 2012). Construct validity, external validity and reliability have in this case study been addressed in the research steps

before the analysis as suggested by Runeson et al. (2012). According to Robson (2002) is the researcher an important factor for the quality of a flexible design study. In this study have the researchers involved previous experiences in conducting empirical research both case studies and interview studies.

Construct validity reflects how well the chosen research method has captured the concepts under study and what is investigated according to the research questions. In the interview situation and during the risk meetings there is a risk that the researchers and practitioners may use different terms and have different frames of reference that may lead to misunderstandings. To reduce this risk, the concepts were explained during the interviews and the definitions of terms and concepts in the risk management process were adapted to the standards in the medical device domain. At the beginning of the risk management meetings the exact proceedings for the meeting was explained by the risk manager and researcher.

Reactivity (Robson 2002), the presences of a researcher might affect the participants and thereby influence and limit the outcome. The participants might act or respond after assuming expectations or hide facts. To reduce this treats the participants were guaranteed anonymity regarding all collected data, i.e. the interviews and observations not to be shown or used by researchers outside the case study or by other organisations. At the risk meetings the researcher also looked for and logged signs indicating that the participants' way of acting was affected of the researcher being present.

The selection of interviewees may give an unbalanced and limited view of the concept, so to obtain a good representation of different aspects, where interviewees holding different roles selected (e.g., risk manager, user representative, software developer).

Internal validity is affected by factors that are outside the control researcher and where there are causal relationships among factors. In this study no significant causal relations have been identified, so the risk is not seen as serious.

External validity is concerned with to what extent the findings are applicable and of interest outside the investigated case. External validity is also concerned with the ability to generalise the results i.e. in this case the applicability of the new risk management process. This study is based on the new risk management process, which in its turn is based

on empirical data from previous studies (e.g. Case study A and B) but with a limited set of participants from two single projects. This means that the results cannot automatically be generalised to other organisations. To support generalisations and allow external comparison the context and characteristics of the projects has been presented as extensive as possible under given confidentiality constraints. Some of the key characteristics of the projects may be general for other projects with similar context, which may make the results relevant also for other organisations. To show generalised results the new risk management process has to be independently used in a larger amount of projects.

Reliability concerns the extent to which the research is dependent to specific researchers. In this case there is a risk that researcher bias has influenced the risk management process and the evaluations of the new risk management process. In order to mitigate this risk the perspective of several researchers was included in the study. The reliability was also addressed by applying triangulation to the data collection, collecting data from both observations and interviews. The researchers discussed the interview questions, the codes as well as the coded material from the interviews and observations.

To further increase the validity of this study the final version of this paper was reviewed and approved by participants from the development organisation and feedback were given from the development organisation during the case study process.

4 The risk management process, RiskUse

The purpose of the risk management process, RiskUse is to provide practitioners, mainly risk managers with a software risk management process that has a well defined user perspective, is easy to apply and includes hands-on recommendations how to use the process. The aim is to present a risk management process that allows the development organisation to perform adequate risk management activities that can ensure that the developed software is safe from a user perspective. This section gives a general description of RiskUse with focus on the parts evaluated

in the case study presented in this paper. For a detailed description of RiskUse, see (Lindholm 2014).

4.1 RiskUse - phases

RiskUse consists of five phases displayed in Figure 4. The different phases are defined in more detail in this section.

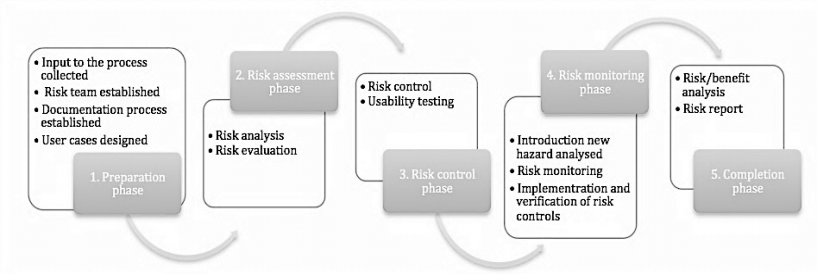

Figure 4. The risk management process, phases and steps.

1. The preparation phase is the first phase of the risk management process and the phase where the risk manager makes the preparations for the entire process. First is input to the risk management process collected, consisting of the requirement specification, the descriptions of intended use, intended users and system context for the system. The documentation is used to get a general understanding of the system, to establish a risk team with the right combination of competencies and used as a foundation in the use case development process. Then the risk manager establishes the risk team. The team shall consist of: a) Developers - with knowledge on how the system or device is designed, are produced, functions and how it shall be used. b) Users - who represents different user groups. The description of intended users can be used as support in the selection process. c) Risk manager - who is responsible for the entire risk management process and chair of the risk meetings. When possible an assistant risk manager shall be selected who will be responsible for the documentation at the risk meetings. The documentation process is established and the documentation shall at least consist of: a) Risk management plan - including description of traceability and scales of assessment and risk acceptance, b) Risk meeting documentation each risk meeting shall be documented, c) Risk

management report including risk/benefit analysis of the residual risks and post product information. The use cases are designed and based on the system's functionality and then used as use case-based identification method of hazards at the risk meetings. More details about use cases are found in section 4.1.1.

2. The risk assessment phase: the phase where the risk meetings are held and the hazards and hazardous situations are identified from the use cases by the risk team. Risk(s) are estimated from each hazardous situation and then analysed and evaluated. During the risk analysis, the known and foreseeable hazards and hazardous situations are identified through the use cases and documented in the risk meeting documentation. For each identified hazardous situation the associated risks are estimated according to defined scales. The risk value is calculated for each risk by multiplying the severity with the probability value. The severity, probability and risk values are documented in the risk meeting documentation. During the risk evaluation each identified hazardous situation with associated risk(s) shall be evaluated and documented in the risk meeting documentation. More specifically the risk team decides if risk reduction is necessary. For each risk that meets one or several criteria for risk reduction or mitigation according to defined criteria, one or several risk control measures shall be established.

3. The risk control phase: the phase contains the risk control process and the assignment of risk control measures to usability testing. The control measures are discussed, decided and documented in the risk meeting documentation. This shall be done for each identified hazard that needs to be reduced or mitigated. A decision shall also be made, if the risk control measure is suitable for usability testing and if so, it is be documented in the risk meeting documentation. The risk control measures are during the risk control phase assigned to developers, i.e., which developer becomes responsible for implementing and verifying the assigned measure. Follow-up dates are decided and documented. The assignment can be done during the risk meeting or after the risk meetings. All risk control measures shall result in one or several requirements in the product requirements specification. The risk control measures are traceable by the use of the unique hazard id, specified on the form UC.x.y Hx. The risk values have to be re-assessed

and possible residual risks identified. Risk control measures suitable for usability testing is incorporated in the usability process and assigned to usability testing. After the usability testing is performed a new iteration of the risk evaluation phase is done. The use case descriptions could be used as a foundation for the test cases in the usability test.

4. The risk monitoring phase: the phase where the introduction of new hazards is discussed and identification of residual risks is performed. For each identified risk control measure, the introduction of new hazard and hazardous situations is discussed. If a new hazard and hazardous situation are identified, a new unique hazard id on the form UC.x.y Hx shall be generated and incorporated in the in the risk meeting documentation. The hazard id shall be coloured blue, and a new iteration of the risk assessment phase and risk control phase is done. During the risk monitoring, appropriate risk control measures are discussed and decided and the risks shall be analysed again according to the defined scales. The remaining risks, the residual risks that do not meet the acceptance criteria, one of two options shall be chosen, either further risk control or risk benefit/analysis. Then the assigned developers shall perform verification and validation of the implementation of the risk control activities. The results of verification and validation shall be balanced against the documented values for severity, probability and risk documented in the risk meeting documentation, a new analysis of the risk values are made. If the risk values, not are lowered enough new risk control measures has to be decided, implemented and verified

5. The completion phase: the last phase where the residual risks are handled and the risk management report is written. For the residual risks not meeting the acceptance criteria and were further risk control is not applicable a risk/benefit analysis shall be made.
Review data shall be gathered to support the conclusion that the medical benefits of the medical device (entire medical device or particular features of the medical device) outweigh the residual risk. The review data shall be documented. A risk management report shall be written prior to release for commercial distribution. A review of the risk management process shall be done and the result shall be documented in the risk management report. Information important for

the production and post-production phase that are gathered and documented during the risk meetings, for example, warnings in the graphical user interface, labelling and special training shall be documented in the risk management report.

4.1.1 Use cases

The use cases shall be written before the risk meetings and the risk manager or other members of the development organisation do it. The use cases are then used as input during the risk meetings. At the meeting the risk manager first makes a walk-though of the use case(s) and alteration are made if needed. The risk manager then guides the discussion throughout the meeting and each step in a use case is discussed according to hazard.

Each use case description shall contain:

Unique id: on the form UC.x.y were UC.x refers to the use case and y to the step in the scenario.

Requirement specification: Specification of the relevant requirement specification, for example the document number.

Requirements: Specification of the user requirements (UR) and the product requirements (PR) relevant to the use case, for example: UR1 [PR6]

Preconditions: Preconditions for the use case, for example: The patient is already registered in the system.

Use case: Each step in the use case is defined, for example: UC 5.1 Chose new evaluation scale.

Comments: Important issues from the risk meeting concerning the use case, for example: the value shall not be displayed for the user.

4.1.2 Usability testing

Usability testing can indicate hazards that are not identified in the risk management process and render the possibility to verify if risks with high risk value actually cause the presumed problems. The usability test shall be made according to well-recognised methods. The results are then analysed and mapped against the risk values documented in the risk meeting documentation. Reassessments of the concerned risks are made at a risk meeting, risk control actions are decided and the risk meeting documentation is updated.

4.1.3 Traceability

Traceability of the hazards is maintained in the risk management process by the hazard id, specified on the form UC.x.y Hx. The first part, UCx refers to the use case that the hazard was identified in, y to the step in the use case and Hx is a local unique identifier that allows for more than one hazard to be assigned to a particular step in a use case. By colour blue the hazard id generated for a new hazard identified after risk control measures, new hazards are traceable and easy to track in the risk meeting documentation.

To maintain traceability to the requirements, both to the product and user requirements, each use case description contains the unique identifiers for the requirements relevant for the specific use case. There is a chain of traceability through the process and also backwards see Figure 5.

Figure 5. Traceability chain.

In the usability test cases are documented, which hazards they test and the results of the usability testing are mapped back to the hazards. The same approach can be used according to other verification and validation activities.

4.1.4 Documentation

All the planned risk management activities shall be documented in the risk management plan. The description of intended use and intended users could be reused from the product project plan and also reused in the risk management report. Endeavour as hands-on descriptions as possible.

The risk management report shall show that the risk management activities have been performed according to the risk management plan and that the overall residual risks are acceptable. Information important for the production and the post-production phase that are gathered and documented during the risk meetings, for example, warnings in the graphical user interface, labelling and special training shall be documented in the report. Material from the risk management plan can

be reused such as a description of the product, intended use, intended users and the risk management process.

4.1.5 Risk meetings and risk control

At the risk meetings there should be at least one participant from each group of participants; the intended users, the developers and risk managers. It is the risk manager, who organises and documents the risk meetings. A risk manager is also the chair of the risk meetings. Predefined use cases and description of intended use, scales for estimation and criteria for risk acceptance are used as input to the risk meetings. At the risk meetings the hazards are identified through brainstorming, with the risk manager as facilitator. For each step of the use case, all participants suggest possible hazards and hazardous situations connected to the specific use case step discussed. Characteristics that could affect safety of the device, stressful situations, environmental factors and transportation of the medical device should also be taken into consideration. All the identified hazards and hazardous situations are documented in the risk meeting documentation. In the next step, the risk analysis is the associated risk(s) for each identified hazardous situation estimated according to defined scales for estimating severity and probability. The risk value (R) is calculated for each hazard by multiplying the severity (S) with the probability (P) value. In the risk evaluation step is a decision made for each risk if risk reduction is necessary. The decision is based on defined criteria for risk reduction. For all the decided to proceed on, risk control measures shall be discussed and decided on and assigned to usability testing if suitable. The effects of the decided risk control measures are discussed and analysed according to the same scales. New hazards generated from the risk control measures shall be documented and analysed. The remaining risks are accepted, assigned new further measures, or left as residual risks. The implementation of risk control measures is, if possible assigned to a named developer.

It is recommended that the facilitator of the risk meeting have a strict control of the meeting with the ambition to get opinions from all the participants and thereby avoid dominance factors. Explicitly addressing each participant or giving each participant a specific timeslot can for example accomplish it. Another important factor is to define and separate the estimation of severity and probability and to strictly

apply the predefine scales, so that the estimation of the different values do not affect each other during the discussions.

Technical risks identified during the meeting are often of a more general nature and not use case-specific, there is a need for handling them separately, recorded them at the meeting and then transfer them to technical risk analysis.

During the discussions about risk control measures it is common that possible alternative solutions and improvement measures that are not true risk control measures are focused on. This solutions and measures should be documented separately and be discussed on another kind of meeting.

4.1.6 Iterative development

The risk management process can be used in linear development processes but also in iterative development processes. The overall risk management process is the same, but with a smaller scope and new use cases for each iteration. The scope of each iteration is defined in the risk management plan and new risk teams may be added over time due to member changes. The activity in the risk assessment phase and risk control phase stay the same and the usability testing is made for suitable functionality within the on going iteration.

Either small risk management reports are written at the end of each iteration where repeated unchanged parts of the report creates a framework to which the small reports are added or a new complete report for each iteration. However, parts such as description of traceability, risk analysis method and risk estimation scales can be reused.

The different phases in the risk management process can run in parallel for the different iterations. For example, when risk monitoring in on going in one iteration, the preparation and risk assessment can start for another iteration.

5 Results

In this section, the results from the interviews with the development organisation and the observation made during the risk meetings and usability test are presented. The results are presented with regard to the research questions presented in Section 3.1 and complemented by a

section presenting additional findings. In this case study, parts of the new risk management process, (e.g., preparation phase, risk assessment phase and risk control phase) were evaluated in a real life context. The monitoring phase and completion phase need to be studied over time and that is beyond the scope of the research presented in this paper.

5.1 Use cases

During the preparation phase use cases were written, covering the functionality that should be assessed according to risk. The risk manager, one of the intended users and the researcher, took part in the writing of all the use cases. The use cases were designed as described in Section 4.1.1. The use cases were then used as input during the risk meetings and the researcher acted as risk manager during the meetings, guiding the discussion, though the meetings and each step in a use case was discussed according to hazard. Brainstorming was used explicitly giving everyone the opportunity to speak. The users added domain knowledge and also knowledge about working conditions, current practice and laws and rules, the developers contributed with technical knowledge. No role was dominating the discussions since everyone explicit was addressed.

Since no type of input had been used at prior risk meetings within the project, the use of predefined use cases was a new experience for the participants. According to the interviewees they had at prior meetings decided during the meetings what would to be discussed and assessed. This approach rendered a lot of discussion that took a lot of the meeting time. It was perceived as negative by the participants, “It is exhausting to participate in a risk analysis, you must stay focused all the time and if you have to spend time coming up with things, a lot of valuable time will be lost”.

As recommended in the RiskUse the meeting time was decided to be maximum two hours, “just right, if you have a break in the middle” according to one of the interviewees. Most of the use cases were covered on estimated time, except one use case that had to be transferred to the next meeting. Each use case starts with the preconditions that apply for the use case, new preconditions were identified during some of the risk meetings and these were added to respective use case. The comment field in the use cases was frequently used since during the meetings information arose that was interesting and important for future

implementation, for example, regarding decisions about what information should be reviled to different users.

There were technical risks, meaning risks arising from activities like design, technical processes, implementation and test procedures, for example risk regarding configuration management and transmission failures identified at the risk meetings. These technical risks were sorted out and left to technical risk meetings, because they did not concern the users. During the meetings it was also concluded that some of the functionality should be handled and implemented the same way, for example, how to save and display information. This type of functionality was marked in the use cases and gathered for risk analysis at a separate risk meeting. This risk meeting was held outside the scope of this case study. The design of some of the functionality covered by the use cases had not reached far enough when the use cases were discussed during the risk meetings; this was identified as a problem. Since it was not decided how to implement the functionality, risk analysis might be a waste of time if another solution was chosen and the risk analysis had to be done over again. The risk analysis of this functionality was postponed until design decisions were made.

The developers were invited to the risk meetings covering the functionality they are responsible for. The developers appreciated this since they perceive it as negative to participate if they have nothing to add, "If I do not have anything to contribute with it is better that I work with something else".

All the participants were positive towards using use cases as input to the risk management process. According to one of the interviewees "It was easy to work with this explicit use case" and according to another "It gave a greater security and safety in what is needed to be discussed and at what level".

Instead of being handed the use cases at the risk meetings some of the participants had preferred to get them in advance and thereby been given the opportunity to gain insight into the use cases before the meetings.

5.2 Risk control

For each identified hazard, to be reduced or mitigated, the appropriate risk control measures shall be discussed, decided and documented in

the risk meeting documentation. The risk values shall also be reassessed based on the planned implementation of risk control measures. After the implementation of the measures, shall a follow-up on the risk values be made. The interviewees described a lack of follow-up procedures after the risk meetings, a procedure where the implemented measures are described and followed-up.

The risk control measures were discussed during the risk meetings and sometimes several measures according to one hazard were decided for implementation. The different measures were separated in the documentation to make traceability possible. Each hazard's risk values (e.g. severity and probability) were re-assessed based on the implementation of the measures. If the risk still remained, further risk control measures had to be considered or the hazard transferred to risk/benefit analysis. In this case study no hazards were transferred to risk/benefit analysis, but one new hazard was identified due to planned risk control measure. The risk control measures suitable for usability testing were assigned to usability testing sessions.

During the risk control phase, risk control measures shall be assigned to developers, i.e. who becomes responsible for implementation and verification the assigned measures. The assignment and the decision about follow-up date can be done during the risk meeting, but in this case it was not possible due to the work planes were not ready. It was decided that the risk manager should be responsible for the assignment procedure and responsible for it was carried out. As one of the interviewees reflected in the interviews at the end of the case study, "The challenge with risk management is time and recourses, that someone owns the process and keep the contacts. As I see it, the risk meetings are no longer not the most difficult thing, it is to follow-up the risks as sad and document when they are implemented".

5.3 Usability testing

Within the scope of this case study a parts of the functionality were decided for usability testing. The part contains 25 different functionalities and was tested with the help of 27 test cases. In the design of the test cases, the use cases have been used as input.

In total 16 usability problems with different severity was identified during the usability testing. The distribution is presented in Table 3.

The identified usability problems have been divided into three groups according to the severity of the usability problems. The first group, 'Serious', contains problems that cause the test users serious problems and the problems need to be adjusted. Six of the usability problems are found in this group and four of these problems are IMM problems (see Table 2) when the user interface is unclear. None of these serious usability problems were identified as possible hazards during the risk assessment phase. The next group of usability problems is 'Less serious'; problems not so serious, but is it desirable that they are adjusted. In this group is seven of the usability problems found, three of them are IMM problems and three are B problems where the test participant has suggested better to solve things. In this group were four of the usability problems identified as possible hazards where the severity was estimated at four (e.g. catastrophic) and the probability of occurrence was not estimated because of the dependents of software.

Table 3. Results from usability testing

Severity	Number of usability problems	Category	Identified hazards
Serious	6	IMM – 4 B – 1 M – 1	0
Less serious	7	IMM – 3 B – 3 M – 1	4
In addition	3	M – 2 O – 1	2

Risk control measures had been assigned to each of these hazards. Since this functionality caused problems for the test users during the usability testing, it may be reasonable to assume that it may cause problems in the real use too. If a problem would occur, the result would be catastrophic and based on that it seems adequate to implement the risk control measures. In the last group, 'In addition', problems that should be adjusted if time so allow are sorted. There are three usability problems found in this group and there are M problems (Missing, see Table 5), when the test users have failed to find some functionality expected by the user to be there and O problems (Overlook, see Table 2), when user does not see an existing functionality. Two of the

usability problems were identified as possible hazards and one of the hazards were given the severity value 2 and the probability also value 2. In the risk team it was decided to reduce this risk even if the risk value was below the criteria of risk acceptance. Three different risk control measures were assigned to reduce this risk. Since only a few of the test users had minor problems with this functionality it would be desirable to discuss this hazard and risk control measures again. The other possible hazard was given severity value 4 and probability of occurrence was not estimated because of the dependents of software. Even this hazard would preferable be discussed again, even if created minor problems for the users.

The results show that usability tests can give valuable input to the risk management process. Usability tests can indicate risks that are not identified in the risk management process and, as in this case at least six or preferable more hazards should have been identified during the identification of hazards. It also gives the possibility to verify if the assessed risk values correspond to the identified usability problems, for example, if risks with high values actually cause the presumed problems. The dominating type of usability problems found during the usability test were IMM (Incongruent Mental Model, see Table 2), and for the functionality, causing IMM problems, the users' mental models were not the same as the developers'. The users expect the user interface to follow their logic and not the software's or the developers' logic, so when there is a mismatch, it will show as a problem. Since the active intervention was used during the usability test, it gave the test facilitator and observer a good understanding of the users' problems and also their mental model of the tested functionality. The differences between mental models should be considered in the further work with the design of the functionality.

5.4 Traceability

To comply with the regulatory requirements of the medical device domain it is essential to have traceability from requirements, including risks, throughout the entire development and maintenance process (Casey & McCaffery 2003). Traceability of the hazards is maintained by the hazard id, specified on the form UC.x.y Hx and thereby linked to the right use case which in its turn links to the right requirements. The identified hazards were easily given their id during the risk

meetings and if a new hazard was identified after risk control measures had been decided, generated hazard id was coloured blue in the documentation. This procedure made it easy to identify that new hazards had been discussed.

Decided risk control measures generated in some cases new requirements and after the risk meetings the risk manager checked and updated both the product and user requirements specifications according to what had been decided, after that the use cases was updated with the new requirements.

Some of the risk control measures were identified and documented as suitable for usability testing and when the test cases for the usability test were designed; the use cases were used as input and linked to the test case. This way of working makes it possible to trace the different user problems back to the hazards.

5.5 Documentation

Documentation concerning the risk management process could be divided in the actual documents produced during the entire risk management process and the documentation made during the risk meetings. The wish from the organisation was to get risk documents, easy to produce and maintain, has a uniform format and complies with standards. The interviews indicated that the prior risk reports were difficult to understand and varied according to contents and layout. The documentation during the prior risk meeting had been made in a predefined spreadsheet, but there were problems understanding what to register in the different columns. Another problem identified by risk management was that hazard descriptions written during the risk meetings were difficult to interpret and understand afterwards, they need to be more explicit and unambiguous.

The risk meeting documentation in this case study was made in a redesign spreadsheet regarding the content and the spreadsheet was also complemented with explanation on how to interpret the different headings in the sheet. During the risk meetings the on going documentation was displayed to all the participants and emphasis was put on the formulation of hazard descriptions, so that all participants experienced that the description was explicit and unambiguous. Colour coding of the documentation was also made, sometimes the colouring was made during the meeting and sometimes after the meeting.

During the risk meetings it was observed that that there were quite a lot of discussion regarding a hazard and the hazard description was first filled in before the description of harm, so it might beneficial to switch the order between these two columns. The description of the cause could be the same or very similar to the description of the hazard and sometimes the description was even left out. It was also observed that more than one risk control measure could be decided for the same hazard, so to maintain traceability the different measures were denoted a), b) and so on. Each hazard was discussed; one at the time and the documentation regarding that hazard was completed before moving to the next hazard. According to the risk manager "It was much easier to work with the spreadsheets now, knowing what to write in the different columns and make things complete and not filling in something here and something there, it felt much more comprehensible and structured".

The risk management plan and the risk management report are two important documents within the risk management process. The researcher updated the risk management plan according to RiskUse. The plan contains the content specified according to guidelines in RiskUse [0] and the researcher's goal was to write a hand-on, practical plan, where the work procedures are easy to follow and it is easy to update the plan, if, for example, participants or work procedures are changed. The risk management plan was reviewed and approved by the organisation and is now in use. In this case the risk management process is not a linear development processes, but an iterative development processes where the product is divided into packages and risk analysed one package at the time. The scope of each package is defined in the risk management plan and a description of the risk team connected to the package was described since there were changes of personnel.

The risk management report shall show that the risk management activities have been performed according to the risk management plan and that the overall residual risks are acceptable. Due to the iterative way of working, other working procedures and the scope of the entire project, it is not beneficial to wait and write the risk management report at the end of the project. It was decided that a risk management report for each package should be written and then compiled to a final risk management report at the end of the project. The researcher

designed and wrote the risk management report covering the package within the scope of this case study. With the aim of making the report as understandable as possible, easy to use as a framework for the next risk management report and to be a part of a final risk management report for the project. Material from the risk management plan was reused such as the description of the product, intended use, intended users and the risk management process. The report was then complemented with the results from the risk evaluation, risk control and important information for the production and post-production phases that was gathered and documented during the risk meetings, warnings in the graphical user interface as an example. The completed report was reviewed and approved by the organisation.

5.6 Additional findings

Some additional findings were made based on the collected data and not covered by the defined research questions. These findings will be presented in this section.

All the participants experience the participation in the risk management process, RiskUse as a positive and important. However, to work in an iterative way is perceived to hamper the risk management work, since it varies how far the design work have reached (e.g. if a solution is decided or not) when the risk analysis is made.

Two of the interviewees clearly pointed out the need of a specific person with competence in the risk area and that is given the ownership over the entire risk management process and they also reflected that the presence of the researcher at the risk meetings have brought security into the discussions. The discussion at the risk meetings had always felt open-minded and the introduction of RiskUse had not changed that. According to the use of another scale for estimating probability, the new scale was perceived as "more natural" and easier to use relative to the prior one.

The developers desiderate special technical risk analysis meetings on a regular basis where the technical risks, for example, risk regarding database storage and communication protocols, could be discussed and the developers often consider risk when they develop. "Maybe it had made the risk analysis easier if you had collected the risk in some way, along the way". It would also be desirable for the developers if their role in the risk meetings were clarified and more specified.

5.7 Value and further improvements

The predefined use case used as input to the risk meetings were perceived as easy to work with and made the participants feel safe and secure in the discussions. Since the use cases are already predesigned, the agenda on what to discuss is already decided before the meeting, avoiding discussion on the scope of the meeting and the time and effort are spent the right issues. The limitation of the meeting time is also perceived as beneficial since it makes it possible for the participants to stay focused during the entire meeting.

The use of unique hazard identity linked to the use cases creates traceability between use cases and the hazard documentation and also the requirements. It also creates links to the test cases used for usability testing. This gives the organisation the traceability demanded by regulatory requirements.

According to the documentation, a better understanding of what to document during the risk meetings was requested from the risk management and the use of the spreadsheet designed according to the new risk management process fulfilled that request. The way of documenting also contributes to traceability. The work approach with the risk management plan and risk management report can give the organisation documents that are easier to follow in the line of work, are easier kept updated and are tending the organisations iterative way of working. This must, however, be evaluated over time.

The risk management process gives the organisation the possibility to have a follow-up procedure on the hazards and risk control measures by assigning developers to each risk control measure. To assign developer at risk meetings was not possible due to the work planning, but this can be done after the risk meetings instead and recommended to the organisation.

Risk values are assumptions so if they can be identified in an additional way before action is taken, effort and time can be saved, due to avoidance of unnecessary changes. The use of usability testing can give that valuable input to the risk management process by indicating risks that are not identified during the risk management process also to verify if the assessed risk values correspond to the identified usability problems.

During the case study future improvements were identified. The use cases should preferably be sent out to the risk meeting participants before the meetings, giving them a chance to read the use cases in advance. It should a maybe also be possible for the participants to submit comments regarding the contents of the use cases or if something has to be updated, added or removed.

Regarding the risk management process, it would be desirable to insert clear checkpoints between the design phase and risk assessment phase so that the solutions are ready and decided. Then there will be no guessing during the risk meetings or postponed risk meetings due to uncertainty regarding design and solutions.

It would also be beneficial to have a formal risk management process regarding the technical risks that are intertwined with RiskUse. Technical risks are more general in its nature than user related risks and the technical risk are not bound to a specific use case. Preferable should the risk manager be in charge of the management of the technical risk and the user risks and the process should be regarded as a uniformed process with different parts. External factors as for example process and project risks are not included in RiskUse but they ought to be considered in an overall risk management process.

Changing the order in which harm and hazard description are filled in might improve the documentation at the risk meetings. Since hazards in most cases are discussed before harm it could be a more appropriate order to have the hazard description first. Regarding the description of cause it was identified that sometimes this deception was left out or had the same text as the hazard description. The cause description might be removed as an own heading and the cause should be written in the hazard description. For all hazards were risk control measures are assigned, it should be mandatory to describe cause in the hazard description.

Since it has not been possible within the timeframe of the case study, to evaluate the process of risk control regarding assigning risk control measures to developers and to set follow-up dates for follow-up after implementation, this has to be further evaluated.

6 Discussion and conclusion

This paper introduces the first version of RiskUse and an evaluation of the process in a case organisation. RiskUse is developed in close collaboration with the organisation developing medical devices. The main goal is to integrate users and user perspective in the software risk management process in the medical device domain and to introduce usability testing, as an integrated part in the risk management process contributing to the goal of integrating users. In conclusion the risk management process is found to support the practitioners in their work with risks and risk management. It can also be concluded that the process has the potential to be used in a medical device organisation and bring value to the organisation. The risk management process is also found to be easy to understand and apply, according to the practitioners participating in the case study.

The use of use cases makes the risk meeting participants feel safe and secure in the discussions during the meetings and make the discussions focus on the right things, saving effort and time. The use cases are perceived easy to work with, however, it takes time to write them. The time is depending on the level of the use cases, more detailed use cases takes more time to write. It is more favourable to strive for writing use cases on the same level during a project to get a uniformed input at the risk meetings. The case study indicates that the benefits outweigh the disadvantage.

Users attending the risk meeting also bring the user perspective into the process. A prior case study (Lindholm et al. 2014) showed that the user representatives dominated the discussions whereas the developer representatives held a lower profile. During the risk meetings in this case study everyone participating were explicitly addressed and no role was seen dominating the discussions. However the effect of participants' personalities cannot be ruled out.

To further address the user perspective in the process, usability testing is used. The results in both this case study and the case study made by Lindholm and Höst (2013) show that usability tests can give valuable input to the risk management process. The product can be more safe and reliable since usability tests can indicate risks that are not identified in the risk management process and effort and time can be saved, due to avoidance of unnecessary changes. The focus on the user

and user interface might see high but since it is well known that many risks are related to the usage of a system and the user interface, e.g. Dhillon (2000) reports that 50 % of technical, medical equipment-related problems are caused by operator errors, it is important that the users and user interface stay in focus

Moreover the results show that traceability of hazards can be achieved by implementing it as described in RiskUse and if the recommendations regarding the documentation are followed, it might bring a more "comprehensible and structured way of working", according to the risk manager in the case study. Regarding the hazard descriptions, it was identified by Lindholm et al. (2014) that the explicitness of the description had an impact on the understanding of the risks in later stages of the process. This has also been identified as a problem of the organisation, since they have a problem understanding the prior hazard descriptions. Therefore, additional emphasis is put on obtaining as explicit hazard descriptions as possible.

The risk control phase could not be fully evaluated within the timeframe of the case study. To assign developers to risk control measures at risk meetings was not possible due to the work planning, so this need to be further evaluated. Concerning further work, it includes considering the identified improvements presented in Section 5.7 as well as further evaluation of the risk management process. The two last phases, the monitoring phase and the completion phase and need to be evaluated and require an evaluation over time and in addition the whole risk management method. Broad generalisations of the results can therefore not be made according to that this is the first evaluation of the risk management process, only involving one organisation. However the case study shows that the risk management process is applicable and the positive results provide a strong argument to continue the evaluation and to promote the risk management process. RiskUse is tailored for the medical device domain, adapted to the regulatory requirements within the domain, concerning for example, traceability, documentation, terminology, residual risks and post-production information. The process is above all emphasised on bringing user perspective into the process as medical standards impose. It is out of the scope of study to investigate the usefulness of the proposed risk management process for other types of organisations. However, it can

be assumed that other types of organisations, handling user risks can use the risk management process.

Acknowledgments

The author would like to gratefully acknowledge all the persons involved in this case study and their organisation that have made the data collection possible for this research. Thanks also for sharing your thoughts and time and your willingness to answerer my questions.

References

Allen, S. (2014). Medical device software under the microscope. *Network Security,* 2, pp. 11-12.

Avison, D., Lau, F., Myers, M. & Nielsen, P.A. (1999). Action Research, *Communication of the ACM*, 42(1), pp. 94-97.

Becker, J.C. & Flick, G. (1997). A practical approach to failure mode, effects and criticality analysis (FMECA) for computing systems. In *Proceeding of the IEEE High-Assurance Systems Engineering Workshop,* pp. 228-236.

Bills, E. & Tartal, J. (2008). Integrating Risk Management into the CAPA Process. *Biomedical instrumentation and technology,* 42(6), pp. 466-468.

Boehm, B. (1991). Software risk management: Principles and practices. *IEEE Software*, 8(1), pp. 32–41.

Bovee, M.W., Paul, D.L. & Nelson K.M. (2001). A framework for assessing the use of third-party software quality assurance standards to meet FDA medical device software process control guidelines. *IEEE Transactions on engineering management,* 48(4), pp. 465-478.

Bubenski, V. (2014). A novel approach to software quality risk management. *Software testing, verification and reliability,* 24, pp.124-154.

Buxton, J.N. & Malcolm, R. (1991). Software technology transfer. *Software Engineering journal,* 6(1), pp. 17-23.

Casey, V. & McCaffery, F. (2013). A lightweight traceability assessment method for medical device software. *Journal of Software: Evolution and Process,* 25(4), pp. 363-372.

Charette, R. N. (1989). *Software engineering risk analysis and management.* McGraw-Hill Software Engineering Series, New York: McGraw-Hill.

Chiozza, M. L. & Ponzetti, C. (2009). FMEA: A model for reducing medical errors. *Clinica Chimia Acta*, 404(1), pp. 75–78.

Chunxiao, L., Raghunathan, A. & Jha, N.K. (2013). Improving the trustworthiness of medical device software with formal verification methods. *IEEE Embedded systems letters,* 5(3), pp. 50-53.

Daniels J., Fels S., Kushniruk A., Lim J. & Ansermino J.M. (2007). A framework for evaluating usability of clinical monitoring technology. *Journal of Clinical Monitoring and Computing,* 21, pp. 323-330.

Dhillon, B. S. (2000). *Medical device reliability and associated areas*. Boca Raton: CRC press Taylor & Francis Group.

Dhillon, B.S. (2008). *Reliability technology, human error and quality in health care.* Boca Raton: CRC press, Taylor & Francis Group.

Dumas, J.S. & Redish, J.C. (1999). *A practical guide to usability testing.* Exeter: Intellect Books.

FDA (1996). Do it by design: An introduction to human factors in medical devices.

FDA (2000). Medical Devise Use-Safety: Incorporating human factors engineering into risk management.

FDA (2013). Mobile medical applications. Guidance for industry and Food and drug administration staff.

Fitzgerald, B., Stol, K-J., O'Sullivan, R. & O'Brian, D. (2013). Scaling agile methods to regulated environments: An industry case study. In *Proceedings of IEEE International conference on software engineering (ICSE 2013),* pp. 863-872.

Gary, K., Enquobahrie, A., Ibanez, L., Cheng, P., Yaniv, Z., Cleary, K., Kokoori, S., Muffih, B. & Heidenreich, J. (2011). Agile methods for open source safety-critical software. *Software: Practice and Experience,* 41(9), pp. 945-962.

Habraken, M. M. P., Van der Schaal, T. W., Leistikow, I. P., & Reijnders-Thijssen, P. M. J. (2009). Prospective risk analysis of health care processes: A systematic evaluation of the use of HFMEA in Dutch health care. *Ergonomics*, 52, pp. 809–819.

Hall, E. M. (1998). *Managing risk: Methods for software systems development*. Reading: Addison Wesley.

Hegde, V. (2011). Case study: Risk management for medical devices. In *Proceedings of reliability and maintainability symposium (RAMS*), pp. 1-6.

Hyman, W.A (2002). A generic fault tree for medical device error. *Journal of Clinical engineering,* 27(2), pp. 134-140.

IEC (2006a). IEC 61025, *Fault tree analysis (FTA),* http://www.iec.ch. August 2014.

IEC (2006b). IEC 60812, Analysis techniques for system reliability - Procedure for failure mode and effects analysis (FMEA), http://www.iec.ch. August 2014.

IEC (2007). IEC 62366:2007, *Medical devices – application of usability engineering to medical devices.* http://www.iso.org. August 2014.

IEC (2010). IEC 80001-1 *Application of risk management for IT-networks incorporating medical devices -- Part 1: Roles, responsibilities and activities,* http://www.iso.org. August 2014.

IEC/TR (2009). IEC/TR 80002-1:2009, *Medical device software -- Part 1: Guidance on the application of ISO 14971 to medical device software.* http://www.iso.org. August 2014.

ISO (2012). ISO 14971:2012 *Medical devices -- Application of risk management to medical devices,* Geneva, Switzerland. ISO.

Jain, R.K., Ananthakrishnan, T.S., Mandalik, S.A. & Jindal, G.D. (2010). Risk analysis of medical instruments – case study of cardiac output monitor. In *Proceeding of 2nd International conference on reliability, safety and hazard (ICRESH),* pp. 637-641.

Jones, C. (1994). *Assessment and control of software risks*. Englewood: Prentice-Hall.

Krasich, M. (2000). Use of fault tree analysis for evaluation of system-reliability improvements in design phase. In *Proceeding of annual reliability and maintainability symposium,* pp. 1-7.

Kushniruk, A. (2002). Evaluation in the design of health information systems: application of approaches emerging from usability engineering. *Computers in biology and medicine,* 32, pp. 141-149.

Lindholm, C., & Höst, M. (2013). Introducing usability testing in the risk management process in software development. In *Proceeding of the Workshop on Software Engineering in Health Care (SEHC13), at ICSE* 2013; pp.5-11.

Lindholm, C., Pedersen Notander, J. & Höst, M. (2014). A Case Study on Software Risk Analysis and Planning in Medical Device Development. *Software Quality Journal*, 22(3), pp. 469-497.

Lindholm C (2014). Guidelines RiskUse 1.0, http://lup.lub.lu.se/record/4882539

Lozier, T. (2010). Streamline Your CAPA Process: Use Risk Assessment to Improve Quality and Compliance. *The Quality Assurance Journal,* 13(1-2), pp. 37-40.

McCaffery, F., Burton J. & Richardson I. (2009). Improving software risk management in a medical device company. In *Proceedings of the International conference on software engineering (ICSE),* pp. 152-162.

McCaffery, F., Burton, R. & Richardson, I. (2010). Risk management capability for the development of medical device software. *Software quality journal,* 18, pp. 81-107.

McDermid, J.A., Nicholson, M., Pumfrey, D.J. & Fenelon, P. (1995). Experiences with the application of HAZOP to computer-based systems. In *Proceedings of the conference on 10th annual Computer Assurance Systems Integrity, Software Safety and Process Security* (COMPASS'95), pp. 37-48.

McHugh, M., McCaffery, F. & Casey V. (2014). Adopting agile practices when developing software for use in the medical domain. *Journal of software: evolution and process,* 26, pp. 504-512.

Méry, D. & Kumar Singh, N. (2010). Trustable formal specification for software certification. In *Proceedings of 4th International Conference on Leveraging Applications of Formal Methods, Verification, and Validation. (*ISoLA'10), pp. 312-326.

Nielsen, J. (1992) The usability engineering life cycle, *Computer,* 25(3), pp. 12-22.

Rakitin, S. R. (2006). Coping with defective software in medical devices. *IEEE Computer*, 39(4), pp. 40–45.

Robson, C. (2002). *Real world research* (2nd ed.). Oxford UK: Blackwell Publishers.

Rogers Y., Sharp, H. & Preece J. (2011). *Interaction design: Beyond human – computer interaction*, (3rd ed.), West Sussex, UK: Wiley.

Runeson, P., Höst, M., Rainer, A. & Regnell, B. (2012). *Case study research in software engineering: Guidelines and example*s. Hoboken, New Jersey: Wiley.

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

Schmuland, C. (2005). Value-added medical-device risk management. *IEEE Transactions on Device and Materials Reliability,* 5(3), pp. 488–493.

Shull, F., Singer J. & Sjøberg, D. I. K. (2008). Guide to advanced empirical software engineering. Chap Introduction, pp. 1-5, London: Springer-Verlag.

Walsh, T. & Beatty, P. C. W. (2002). Human factors error and patient monitoring. *Physiological Measurement*, 23(3), pp. 111–132.

Vilbergsdottir, S.G., Hvannberg, E.T. & Law, E. L-C (2006). Classification of usability problems (CUP) scheme augmentation and exploitation. In *Proceedings of NordiCHI 2006*, pp. 281-290.

Virzi R.A. (1992). Refining the test phase of usability evaluation: How many subjects is enough? *Human Factors,* 34(4), pp. 457-471.

Xiuxu, Z. & Xiaoli, B. (2010). The application of FMEA method in the risk management of medical devices during the lifecycle. *In Proceedings of 2nd International conference on e-business and information system security (EBISS)*, China, pp. 1-4.

Yin, R. K. (2003). *Case study research: Design and methods* (3rd ed.). Beverly Hills: Sage.

Zhang, D. & Xie, T. (2013). Pathways to Technology Transfer and Adoption: Achievements and Challenges (Mini-Tutorial), In *Proceeding of the 35th International conference on software engineering (ICSE),* pp. 951-952.