Software Qual J
DOI 10.1007/s11219-013-9222-2

A case study on software risk analysis and planning in medical device development

Christin Lindholm · Jesper Pedersen Notander · Martin 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.

Keywords Risk management · Risk analysis · Risk planning Software development · Medical device development

C. Lindholm (✉) · J. P. Notander · M. Höst
Software Engineering Research Group, Department of Computer Science, Faculty of Engineering, Lund University, Box 118, 221 00 Lund, Sweden
e-mail: christin.lindholm@cs.lth.se

J. P. Notander
e-mail: jesper.notander@cs.lth.se

M. Höst
e-mail: martin.host@cs.lth.se

Published online: 02 October 2013

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 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

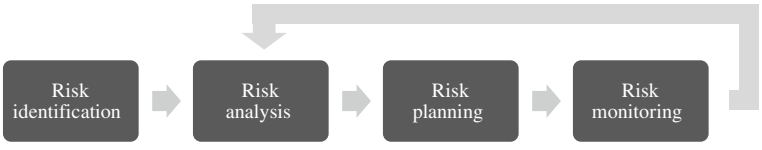

Fig. 1 Risk management process

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. 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.

Fig. 2 Case study process

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.

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

Fig. 3 Case study timeline

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

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 discussions

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.

3.5 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.

Fig. 4 Example scenario

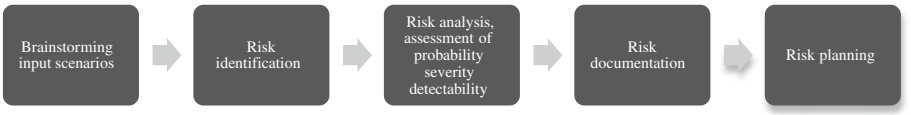

Fig. 5 First part of the software risk process

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 were used for probability and severity assessment. 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. $R = P \times S$. The highest risk value a risk can have with these scales is $R = 4 \times 4 = 16$.

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 $R \geq 8$ according to the risk management plan but also with $R \geq 6$ or risks with $S = 4$ or risks with $S \geq 2$ 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

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

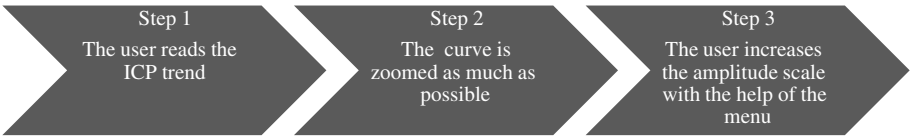

Fig. 6 User scenario A1.1

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.

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

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

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.

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

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

would not happen and that the risk should be removed. In the end it was decided to keep the risk in the documentation.

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 $\times$ minutes?

Furthermore, it became evident that the scenario composition had an impact on the outcome of the risk identification (*R*6). If a scenario was 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 are 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

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

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.

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

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

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.

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

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

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 non-existing 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.

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), 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), 465–478.

Charette, R. N. (1989). *Software engineering risk analysis and management*. New York: Intertext.
Chiozza, M. L., & Ponzetti, C. (2009). FMEA: A model for reducing medical errors. *Clinica Chimia Acta, 404*(1), 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*, 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, p. 101, 2005.
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, IEEE International Conference on Computer Systems and Applications.
Garde, S., & Knaup, P. (2006). Requirements engineering in health care: the example of chemotherapy planning in paediatric oncology. *Requirements Engineering, 11*(4), 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*, 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.
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*, 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 2012*, Vienna, Austria.
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'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.
McCaffery, F., Burton, J., & Richardson, I. (2010). Risk management capability model for the development of medical device software. *Software Quality Journal, 18*, 81–107.
Rakitin, S. R. (2006). Coping with defective software in medical devices. *IEEE Computer, 39*(4), 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., & Höst, M. (2009). Guidelines for conducting and reporting case study research in software engineering. *Empirical Software Engineering, 14*(2), 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.
Schmuland, C. (2005). Value-added medical-device risk management. *IEEE Transactions on Device and Materials Reliability, 5*(3), 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., Gestegård 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 computer-based medical systems*, pp 18–23.

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

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

Author Biographies

Christin Lindholm She is the Faculty Program Director for the Computer and Electrical Engineering programmes at LTH School of Engineering at Campus Helsingborg, Lund University, Lecturer and PhD. student in Software Engineering at Lund University. Her main research area is software in medical devices, especially safety critical medical devices and systems with a special interest in software risk management software quality and development processes.

Jesper Pedersen Notander He is a Ph.D. student in Software Engineering at Lund University. He received an M.Sc. degree from Lund University in 2008 and has prior to his Ph.D. studies been working for two years in the Swedish aerospace and defence industry. His research is focused on flexibility in safety–critical software and software risk management. The research is mainly conducted through empirical methods such as case studies, controlled experiments and surveys.

Martin Höst He is a Professor in Software Engineering at Lund University, Sweden. He received an M.Sc. degree from Lund University in 1992 and a Ph.D. degree in Software Engineering from the same university in 1999. His main research interests include software process improvement, software quality, and empirical software engineering. The research is mainly conducted through empirical methods such as case studies, controlled experiments, and surveys. He has published more than 40 papers in international journals and conference proceedings.