Medical Device Software Traceability

Fergal Mc Caffery, Valentine Casey, M.S. Sivakumar, Gerry Coleman, Peter Donnelly, and John Burton

1 Introduction

Software is becoming an increasingly important component of medical devices, as it enables often complex functional changes to be implemented without having to change the hardware (Lee et al., 2006). With increasing demands for greater functionally within medical devices, the complexity of medical device software development also increases (Rakitin, 2006). This therefore places increased demands for appropriate traceability and risk management processes and tools.

Due to the safety-critical nature of medical device software it is important that highly effective software development practices are in place within medical device companies. Medical device companies must comply with the regulatory requirements of the countries in which they wish to sell their devices (Burton et al., 2006). To tackle these issues, governments have put in place regulatory bodies whose role is to define regulatory systems for medical devices and to ensure that only safe medical devices are placed on the market (Mc Caffery et al., 2010a). Although guidance exists from regulatory bodies on what software activities must be performed, no specific method for performing these activities is outlined or enforced (Mc Caffery et al., 2010b).

To this end, in the USA, the Food and Drug Administration (FDA) Center for Devices and Radiological Health (CDRH) has published guidance papers which include risk-based activities to be performed during software validation (US FDA Center for Devices and Radiological Health, 2002), pre-market submission (US FDA Center for Devices and Radiological Health, 2005) and when using off-the-shelf software in a medical device (US FDA Center for Devices and Radiological Health, 1999). Although the CDRH guidance documents provide information on which software activities should be performed, they do not enforce any specific method for performing these activities. The obvious implication of this is that medical device manufacturers could fail to comply with the expected requirements.

F. Mc Caffery (✉)
Regulated Software Research Group, Lero, Dundalk Institute of Technology, Dundalk, Ireland
e-mail: Fergal.McCaffery@dkit.ie

J. Cleland-Huang et al. (eds.), *Software and Systems Traceability*,
DOI 10.1007/978-1-4471-2239-5_15, © Springer-Verlag London Limited 2012

Therefore, within the medical device industry a decision was made to recognize ISO/IEC 12207 (1995) (a general software engineering life cycle process standard) as being suitable for general medical device software development. However, the Association for the Advancement of Medical Instrumentation (AAMI) software committee carefully reviewed ISO/IEC 12207 and decided that, due to a number of shortfalls, it was necessary to create a new standard specifically for medical device software development. The AAMI used ISO/IEC 12207 as the foundation for their new standard "AAMI SW68, Medical device software – Software life cycle processes" SW68 (2001). In 2006, a new standard AAMI/IEC 62304 (2006) was released that was based on the AAMI SW68 standard.

In 1993, the Council of the European Communities published the Council Directive 93/42/EEC (1993), the "Medical Device Directive" (MDD), on medical devices. The MDD is intended to ensure the safety of medical devices placed on the market in the European Union, and has the backing of national legislation in member states. Amendments to this directive occurred via Directives 2000/70/EC (2000), 2001/104/EC (2001), 2003/32/EC (2003), and 2007/47/EC (2007).

Whenever we mention medical device guidelines within this chapter we refer to the following medical device standards and guidelines: IEC 62304, FDA, the MDD, ISO 14971 (2007), EN 60601-1-4 (2000), TIR 32 (2005), IEC 80002-1 (2009), IEC 62366 (2007), GAMP 5 (2008), IEC/TR 61508 (2005), ISO 13485 (2003) and IEC 60812 (2006).

In this context, we embarked on a study of Software Traceability, which is critical to the requirements and safety aspects of software for medical devices. Within this chapter we include the following sections:

2. Requirements for traceability in the context of software development for medical devices;
3. The development of a software traceability process assessment method (Med-Trace) for determining the capability of a medical device software development organization to perform regulatory compliant and effective traceability;
4. Implementation of Med-Trace within two medical device software development organizations;
5. How each of the two assessed organizations plan to improve traceability;
6. Challenges the medical device software industry is facing in terms of implementing traceability;
7. Foundation for further research in this area and how Med-Trace may be rolled out to assist organizations.

2 Requirements for Medical Device Software Traceability

In order to understand the requirements for traceability in the context of medical device software development we conducted a literature review of generic practices for software traceability and in particular a review of the medical device standards requirements for traceability.

2.1 Traceability Literature Review

The literature review was undertaken in three stages and focused on:

- Generic software development and traceability;
- Safety-critical software development and traceability;
- Medical device software traceability requirements.

2.2 Traceability for Generic Software Development

"Requirements traceability refers to the ability to describe and follow the life of a requirement in both a forwards and backwards direction – i.e. from its origins, through its development and specification, to its subsequent deployment and use, and through periods of on-going refinement and iteration in any of these phases" (Gotel and Finkelstein, 1997). An important focus of requirements traceability is identifying how high level requirements are transformed into low level requirements and how these are subsequently implemented in the software product.

Initially requirements traceability was utilized as an aid in tracing requirements from customer/stakeholder needs to implementation and final verification before delivering the product to the customer. The role traceability plays has expanded and it has become an important tool in the software development activities of project management, change management, and defect management (Nuseibeh and Easterbrook, 2000). This is particularly relevant as software development is increasingly globally distributed across multiple teams and sites (Casey, 2010; Damian and Moitra, 2006). It is therefore essential to have an effective traceability process in place as it provides an essential support for developing high quality software systems (Espinoza and Garbajosa, 2008).

When considering generic software development, two of the most popular process assessment and improvement frameworks are the Capability Maturity Model® Integration (CMMI®) (CMMI Product Team, 2006) and ISO/IEC 15504-5 (2006) and Liao et al. (2005). Both recognize the importance traceability plays and incorporate it in their respective models. Each model was reviewed in detail with regard to the requirement for effective traceability and how this was addressed.

2.3 Traceability for Safety-Critical Development

Software products are increasingly being deployed in complex, potentially dangerous products such as military systems, cars, aircrafts and medical devices. Software products for these areas can be critical because failure can result in loss of life, significant environmental damage, or major financial loss (Kannenberg and Saiedian, 2009).

Traceability is especially vital for critical systems which must satisfy a range of functional and non-functional requirements, including safety, reliability and availability (Mason, 2005).

Within the safety-critical software arena, different standards/certifications are available for different industries. These include DO-178B (1992) for the Aerospace industry, with Automotive SPICE (2005) and ISO 26262 (2009) being required in the Automotive industry. IEC 60880 (2006) describes the European standards for certification of nuclear power generating software and IEC/TR 61508 (2005) describes a general-purpose hierarchy of safety-critical development methodologies that have been applied to a variety of domains ranging from medical instrumentation to electronic switching of passenger railways. Requirements traceability is an important clause in all the above mentioned standards/certifications.

In addition to the software development life cycle, a software safety life cycle has also to be implemented for safety-critical systems. It is crucial to maintain traceability between the software safety requirements, the decisions taken during design, and their actual implementation in the code. This is a complex task and needs to be performed whilst the system is being developed and not after the development has finished (Panesar-Walawege et al., 2010).

2.4 Medical Device Software Traceability Requirements

A detailed review was undertaken of the medical device guidelines with regard to traceability. A key point to emerge from this study is that while requirements traceability is essentially part of risk management, hazard traceability is of equal importance in medical device software development. The most relevant findings regarding traceability are presented here in summary.

2.4.1 ANSI/AAMI/IEC 62304:2006

In 2006, ANSI/AAMI/IEC 62304:2006 (*Medical Device Software – Software Life Cycle Processes*) was released. Traceability plays a key role in this standard and is defined as the “Degree to which a relationship can be established between two or more products of the development process” (ANSI/AAMI/IEC 62304, 2006). It is specifically addressed in the following sections of the standard: Section 5.1 states that “the manufacturer shall establish a software development plan for the development activity”. This plan shall address “Traceability between system requirements, software requirements, software system test, and risk control measures implemented in the software”. Section 5.2 specifies that “the manufacturer shall verify and document that the software requirements are traceable to the system requirements or other source.” Section 5.7 states that “the manufacturers shall verify that the software system test procedures trace to the software requirements”. In section 7.3 Verification of Risk Control Measures the standard specifies that “the Manufacturer shall document traceability of software hazards as appropriate: From the hazardous situation to the software item. From the software item to the specific software cause.

From the software cause to the risk control measure and from the risk control measure to the verification of the risk control measure".

As part of the Configuration Management Process in section 8 the standard specifies that "the manufacturer shall create an audit trail whereby each change request, problem reports and approval of change request can be traced".

Traceability is also addressed in B.6 Software Maintenance Process which states "It is especially important to verify through trace or regression analysis that the risk control measures built into the device are not adversely changed or modified by the software change that is being implemented as part of the software maintenance activity".

2.4.2 Medical Device Directive and Amendments

The European Medical Device Directive (MDD) (European Council, 2003) mentions traceability twice, but only in relation to the calibration of test equipment: In 2007, Directive 2007/47/EC added the following amendment to section 8 of the MDD: "For devices which incorporate software or which are medical software in themselves, the software must be validated according to the state of the art taking into account the principles of the development life cycle, risk management, validation and verification" (European Council, 2007). It is in this context that effective software requirements and risk management traceability are essential to achieve state of the art validation.

2.4.3 General Principles of Software Validation

The US FDA CDRH *General Principles of Software Validation; Final Guidance for Industry and FDA Staff* document (US FDA Center for Devices and Radiological Health, 2002) provides guidance on validation and traceability in medical device software development. The scope of the document outlines that traceability is an important activity that provides support to achieve a final conclusion that software is validated. Under section 3.1.2 it states: "the validation of software typically includes evidence that all software requirements have been implemented correctly and completely and are traceable to system requirements". In section 3.2 it specifies that "software validation includes confirmation of conformance to all software specifications and confirmation that all software requirements are traceable to the system specifications". The document goes on to outline in section 5 that traceability is key across almost all of the software development processes and especially in relation to the requirements, design, construction and test processes.

2.4.4 Premarket Submissions for Software Contained in Medical Devices

The FDA CDRH document *Guidance for the Content of Premarket Submissions for Software Contained in Medical Devices* (US FDA Center for Devices and Radiological Health, 2005) provides information to industry regarding the documentation to include in premarket submissions for software devices, including standalone software applications and hardware-based devices that incorporate software. In this document traceability analysis is defined as linking together the product design requirements, design specifications, and testing requirements. It also provides a means of tying together identified hazards with the implementation and testing of the mitigations. It also states that traceability analysis should be included as part of the premarket submission for Moderate and Major level of concern medical devices.

2.4.5 Off-The-Shelf Software Use in Medical Devices

The FDA CDRH *Guidance for Industry, FDA Reviewers and Compliance on Off-The-Shelf Software Use in Medical Devices* (US FDA Center for Devices and Radiological Health, 1999) document was developed to address the many questions asked by medical device manufacturers regarding what they need to provide in a pre-market submission to the FDA when they adopt Off-The-Shelf (OTS) software. With regard to traceability it states: "The introduction of new or modified OTS components to a product baseline may impact the safety of the product. Therefore a safety impact assessment of the medical device must be performed and the associated hazards documented in a Failure Modes and Effects Analysis (FMEA) table. Each hazard's consequence should be provided and expressed qualitatively; e.g., major, moderate, or minor. Traceability between these identified hazards, their design requirements, and test reports must be provided".

2.4.6 ISO 14971:2007

ISO 14971:2007 (*Medical devices – Application of risk management to medical devices*) is the de-facto standard on risk management for medical devices. The FDA recognize the standard (US FDA Center for Devices and Radiological Health, 2002) and agree compliance with it as acceptable for pre-market submissions in the US (US FDA Center for Devices and Radiological Health, 2005). In the EU, conformance with the standard is also acceptable for meeting the requirements of the medical device directives. In section A.2.3.5 the standard defines the risk management file as: "Where the manufacturer can locate or find the locations of all the records and other documents applicable to risk management. This facilitates the risk management process and enables more efficient auditing to the standard. Traceability is necessary to demonstrate that the risk management process has been applied to each identified hazard."

2.4.7 IEC/TR 80002-1:2009

IEC/TR 80002-1:2009 (*Medical Device Software – Part 1: Guidance on the application of ISO 14971 to medical device software*). Though this technical report does not add to, or otherwise change, the requirements of ISO 14971:2007, it does provide direction on how the standard can be implemented specifically for medical device software. The technical report states: "The software process should set up a system that makes traceability possible, starting from the software-related hazards and the software risk control measures and tracing their implementation to the corresponding safety-related software requirements and the software items that satisfy those requirements. All of these should be traceable to their verification".

2.4.8 ISO 13485:2003

ISO 13485:2003 (*Medical devices – Quality management systems – Requirement for regulatory purposes*). The standard specifies requirements for a quality management system that can be used by an organization for the design and development, production, installation and servicing of medical devices, and the design, development, and provision of related services (ISO 13485, 2003). With reference to traceability, the standard states in section 7.5.3.2.1: "The organization shall establish documented procedures for traceability. Such procedures shall define the extent of product traceability and the records required". It goes on in section 7.5.3.2.2 with reference to "Particular requirements for active implantable medical devices and implantable medical devices" to state: "In defining the records required for traceability, the organization shall include records of all components, materials and work environment conditions, if these could cause the medical device not to satisfy its specified requirements. The organization shall require that its agents or distributors maintain records of the distribution of medical devices to allow traceability and that such records are available for inspection. Records of the name and address of the shipping package consignee shall be maintained."

2.4.9 Traceability for Medical Device Software Development

Software development for medical devices can be a difficult and complex endeavour compared to other domains. Safety is a key area which must be successfully addressed given the potential for harm that defective medical device software can cause. An analysis of medical device recalls by the FDA in 1996 (Wallace and Kuhn, 2001) found that software was increasingly responsible for product recalls: In 1996, 10% of product recalls were caused by software-related issues. The standards and guidelines created to overcome this have already been discussed, but problems still persist. In the period the 1st November 2009 to 1st November 2010 the FDA recorded 78 medical device recalls and state software as the cause (Medical & Radiation Emitting Device Recalls, 2010).

Our literature review highlighted there was a limited amount of published material regarding implementation challenges and advances in the field of traceability in medical device software. This was in contrast to other sectors in the same context e.g., automotive and aerospace software development. Another important aspect to emerge from our literature review was that while there is a requirement to address traceability, and undertake traceability analysis, there is limited guidance available to help implement traceability effectively in organizations. This finding is in line with a review of guidance for all aspects of medical device software development which took place in 2009 (Mc Caffery and Dorling, 2009).

3 Development of the Med-Trace Assessment Method

One of the main aims of the Regulated Software Research Group in Dundalk Institute of Technology is to support the growth of a medical device software development industry within Ireland. Therefore, as traceability is central to the development of regulatory compliant software development we decided to develop an assessment method specifically to assist companies to adhere to the traceability aspects of the medical device software standards.

The Adept method (Mc Caffery et al., 2007) was previously developed to provide a lightweight assessment of software processes from CMMI® and ISO/IEC 15504-5 and was not domain specific. The Adept method provides an organization with a choice of 12 process areas that may be assessed using Adept. However, based upon previous research four of these process areas are considered to be important to the success of any software development company and these processes are therefore mandatory – Requirements Management, Configuration Management, Project Planning, Project Monitoring & Control. Therefore, the organization only can select 2 of the process areas to be assessed from the remaining 8 process areas. Adept consists of eight stages, the main stage involves an assessment team conducting process area interviews for each of the 6 selected process areas with appropriate members of the assessed organization. Based upon these interviews a findings report consisting of a set of strengths, issues and recommendations as to how to address the highlighted issues is produced.

Med-Trace is a new lightweight assessment method that provides a means of assessing the capability of an organization in relation to medical device software traceability. Med-Trace is based upon Adept but whereas Adept relates to generic software development processes Med-Trace is specific to the traceability process with medical device software development organizations. Med-Trace enables these software development organizations to gain an appreciation of the fundamental traceability best practices based on the software engineering traceability literature, software engineering process models (CMMI®, ISO/IEC 15504-5), and the medical device software guidelines and standards. Med-Trace may be used to diagnose an organization's strengths and weaknesses in relation to their medical device software development traceability practices.

3.1 Med-Trace Stages

Med-Trace is composed of eight stages (see Fig. 1). The assessment team typically consists of two assessors who conduct the assessment between them. It is essential that the assessors are trained in how to conduct a Med-Trace assessment and have the requisite knowledge of the requirements for medical device software traceability.

The purpose of stage 1 of a Med-Trace assessment is to "Receive Site Briefing and Develop Assessment Schedule". This involves a preliminary meeting between the assessment team and the organization wishing to undergo a Med-Trace

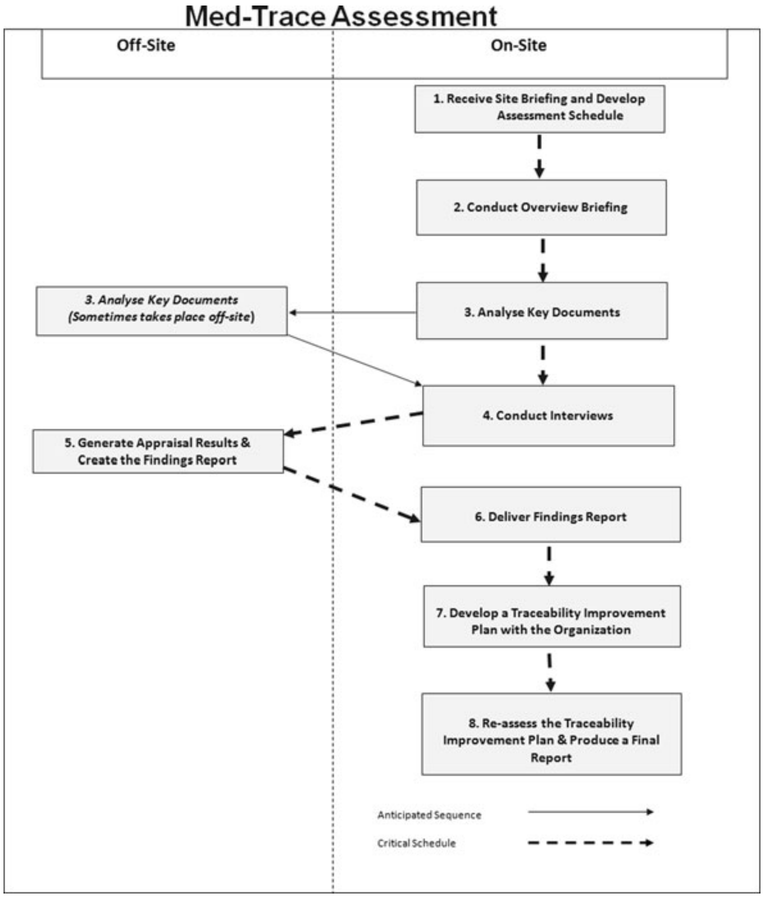
Med-Trace Assessment
Off-Site
On-Site
1. Receive Site Briefing and Develop Assessment Schedule
2. Conduct Overview Briefing
3. Analyse Key Documents (Sometimes takes place off-site)
3. Analyse Key Documents
4. Conduct Interviews
5. Generate Appraisal Results & Create the Findings Report
6. Deliver Findings Report
7. Develop a Traceability Improvement Plan with the Organization
8. Re-assess the Traceability Improvement Plan & Produce a Final Report
Anticipated Sequence
Critical Schedule

Fig. 1 Stages in a Med-Trace assessment

assessment. The assessment team discuss the main drivers for the organization embarking upon a Med-Trace assessment and what can be achieved. Based on the outcome of that discussion an assessment schedule is prepared and agreed.

The purpose of stage 2 is to “Conduct Overview Briefing” During this stage the lead assessor provides an overview of the Med-Trace assessment to members of the organization who will be involved in the subsequent stages of the assessment. This includes what the assessment will involve and cover. What will be required and expected of the participants will also be outlined.

The purpose of stage 3 is to “Analyse Key Documents”. The objective of this stage is to provide insight into relevant process documentation and artifacts which refer or relate to traceability. These are collected, analysed and discussed by the assessors and they record their findings. The first 3 stages are normally performed on the organization's premises, but the documentation collected in stage 3 is sometimes taken off-site as it can then be used to assist with the generation of additional questions for stage 4.

The primary source of data for a Med-Trace assessment is gathered through a series of interviews conducted in stage 4. Therefore the purpose of stage 4 is to "Conduct Interviews". At this stage a set of scripted questions (Appendix: Sample Scripted Med-Trace Questions) are used as the foundation for asking questions that are based upon the software traceability literature search, traceability practices within the CMMI® and ISO/IEC 15504-5 models, and traceability practices that are required by the medical device industry. References are provided in Appendix: Sample Scripted Med-Trace Questions to show the sources of these questions. The assessment team return onsite and key staff members from the organization are interviewed. Each interview is scheduled to last approximately 1.5 hours. At each interview two assessors and one or more representatives from the organization are present. The lead assessor conducts the interview based on the scripted questions and evaluates the responses. The second assessor prepares interview notes based on the responses and may ask additional questions if clarification is required on specific points.

The purpose of stage 5 is to "Generate Assessment Results and Create the Findings Report". This is a collaborative exercise between the assessors to develop the findings report and takes place off-site. The evaluation and interview notes are analysed and discussed in detail from each interview. The findings from all the interviews and from the results from document analysis (undertaken at stage 3) are then considered and the assessment results generated. Based on these results the findings report is prepared and finalised. The resultant findings report consists of a list of strengths, issues and suggested actions for improving traceability.

The purpose of stage 6 is to "Deliver the Findings Report". This stage takes place on-site and involves the lead assessor presenting the findings report to management and participating staff in the organization. Stage 7's purpose is to "Develop a Traceability Improvement Plan with the Organization". This involves the assessors collaborating with management and staff from the organization to collectively develop a pathway towards achieving highly effective and regulatory compliant traceability practices. The findings report provides guidance to the assessed organization and will focus upon practices that will provide the greatest benefit in terms of the organizations business goals with regard to traceability, in addition to quality and compliance. The collaborative aspect of this step is essential as the relevant management and staff take a key part in developing the improvement plan and they ultimately have ownership of it. In these circumstances they are motivated to ensure its successful implementation.

The purpose of stage 8 is to "Re-assess the Traceability Improvement Plan and Produce a Final Report". As part of this stage the assessed organization is revisited approximately 3 months after the completion of stage 7. Progress is reviewed against the recommended improvement path. The outcome of this stage is an updated improvement path and a final report detailing the progress that has been accomplished along with additional recommendations.

4 Implementation of Med-Trace

In this section we discuss how we implemented the Med-Trace assessment method in two medical device organizations. The objective of performing both case studies was to demonstrate how Med-Trace could be used within similar sized and types of organizations (albeit in different countries) to assess the current status of their software traceability processes. We felt that it was important to illustrate the findings from implementing Med-Trace in more than one organization so observations could be made in relation to both the findings and the performance of Med-Trace. Additionally, we wanted to discover what the main issues are that medical device software development organizations face in terms of traceability. We present the process improvement objectives that were collaboratively agreed by both organizations to improve their respective traceability process. We also outline our observations from the findings of undertaking both assessments.

4.1 Implementation in MedSoft

We implemented a Med-Trace assessment in a Small to Medium Size (SME) Irish medical device organization, MedSoft (a pseudonym). MedSoft develop electronic based medical devices that require compliance with both the FDA and the MDD. MedSoft sought a resource-light method to obtain guidance as to how they could improve their software development traceability process, which Med-Trace provided.

4.1.1 Med-Trace Assessment Recommendations Provided to MedSoft

Based on the analysis of the results from the Med-Trace assessment, and in collaboration with MedSoft staff, an improvement plan was developed with the following recommendations:

1. The organization will initiate steps to measure the time spent on traceability and evaluate its effectiveness.
2. The task of performing traceability, in future, will be identified as part of the project plan and adequate time will be allocated to undertake this important task.
3. Good practices which are employed while performing the traceability process will be documented in an efficient format and disseminated to relevant parties as and when required.
4. Project managers will mandate the use of traceability while conducting impact analysis, promoting its usage as a management tool and thus enabling the capture of information for management use.
5. The software development life cycle will contain milestones which will not permit further advancement to other phases/stages of the life cycle until the requirements for traceability are satisfied.

6. A mechanism for tracing the open bugs/known issues to the safety/hazard/risk management system and linking them to the requirements will be made available and utilised.
7. The organization will evaluate tools for the process of automating traceability and requirements management. A tool will then be selected and implemented.

4.2 Implementation in MedNorth

We also undertook a Med-Trace assessment in a UK based medical device organization, MedNorth (a pseudonym). Like MedSoft, MedNorth is an SME and develop electronic-based medical devices that require compliance with both the FDA and the MDD. MedNorth also sought a resource-light method to obtain guidance as to how they could improve their software development traceability process.

4.2.1 Med-Trace Assessment Recommendations Provided to MedNorth

Based on the analysis of the results from the Med-Trace assessment (the MedNorth response to one of the Med-Trace scripted questions is illustrated in Table 1), and in collaboration with MedNorth staff, a pathway was developed as follows:

1. The process for software development traceability and for meetings between the various parties involved will be formalised and documented.
2. A formal training program will be introduced to ensure the adoption of best traceability practices for requirements and risk management.
3. The current Excel-based traceability application will be replaced with an appropriate automated traceability tool.

Table 1 MedNorth response to a Med-Trace scripted question

Question	Response
What kind of resources are provided for the activity of traceability management?	MedNorth developed a dedicated process specifically for traceability that provides coverage of hardware and software. Part of this process involves meetings between parties that are involved in the development of various components that must work together in order to produce the final medical device product. MedNorth feel that the inclusion of these meetings as part of their traceability procedure is a good way of bringing everyone together from the different areas (i.e. software, hardware, mechanical) to ensure that everyone is fully aware of what is required from them and to help ensure that nothing slips within the overall project. The project manager in MedNorth has overall ownership of traceability.

4. Terminology usage with regard to traceability will be standardised and a formal definition of both risk and hazard agreed. A formal method for quantifying probability of harm will also be introduced and deployed.
5. A defined traceability and validation procedure will be developed, implemented and monitored to verify the activities of the staff that perform the traceability and validation function.
6. A formal procedure will be developed and implemented to facilitate mapping from the design documentation to the software code.
7. Resources will be allocated to enable the full implementation of the Ideagen tool. This tool has already been purchased to allow digital signatures to be recorded at each development stage, but it had not been properly implemented in the organization.

4.3 Observations from the 2 Med-Trace Implementations

In both organizations the importance traceability plays in medical device software development was understood and a member of the management team was responsible for its implementation. The dual role of tracing requirements and managing risk and hazards were appreciated, but were recognized as complex and difficult to achieve. The lack of detailed guidance on how best to implement traceability was highlighted as a problem by both organizations. While they both employed a process for software development with regard to traceability this needed to be improved and formalized. The requirement for relevant training and the ability to record and leverage best practice with regard to traceability also emerged.

The serious limitations of utilising manual tools such as Excel, to manage traceability and the need for automated tools was recognized, and required addressing. It was also appreciated that this had to be undertaken with due care and within the financial and temporal constraints of both organizations.

Both organizations welcomed the opportunity to participate in a Med-Trace assessment. The fact that it was lightweight and specifically addressed traceability was considered worthwhile and very relevant. The findings reports addressed key areas where improvements were required and this was confirmed in consultation with the management and staff of both organizations. The adoption of the development pathway provided realistic goals and the collaborative process provided motivation for their achievement. Both organizations are implementing their respective development pathways and have agreed to be reassessed (part of stage 8 of the Med-Trace assessment method).

5 Medical Device Software Industry Traceability Challenges

Due to the critical nature of medical device software and the potential harm failure can cause, the implementation of an effective traceability process is essential. Therefore, to ensure validity, software requirements traceability analysis needs to be conducted to trace software requirements to (and from) system requirements, and to risk analysis results. While this is mandated by the medical device guidelines it is recognized by the industry as a difficult and complex endeavour. This is not helped by the fact that organizations have highlighted the lack of detailed guidance and direction as to how this can be successfully achieved.

A key factor which has been highlighted by the Med-Trace assessments and the literature is the importance of incorporating automated traceability tools into the development process. Especially, considering that many medical device software development organizations employ manual systems like Excel for traceability (Denger et al., 2007). This is a real challenge, which needs to be addressed. There is also a requirement to define and formalise processes which specifically facilitate effective traceability. These need to be supported by resources to provide relevant training and infrastructure.

While the need to provide requirements traceability cannot be underestimated, the necessity to provide traceability for each identified hazard is of equal importance. Risk management is a key activity for medical device software development and hazards have to be traced to risk analysis, risk evaluation and the implementation and verification of the risk control measures.

The number of standards and guidelines which govern medical device software development is also a challenge. To determine the exact requirements of each document with regard to traceability can be time consuming. The information provided can also lack the level of detail required to successfully implement these requirements.

When comparing generic and medical device software development the key difference lies in the mission critical nature and potential for harm which can be inherent in medical device software. Therefore, as risk is a key factor, requirements and hazard traceability both need to be addressed. It is somewhat surprising in these circumstances that tools are used less in medical device software development than in other software development domains (Denger et al., 2007). However, upon closer inspection of the medical device standards there is perhaps a reason in that such tools will also have to be validated in order to achieve regulatory compliance. The use of new automated tools require validation (including Risk and Hazard Analysis/Management) in their own right prior to their use as part of the Quality Management System. This is a very time consuming and costly exercise, especially for a SME. The more complex the tool, the more time, effort and cost associated with the validation and roll-out of the tool.

6 Foundation for Further Research in This Area

The work presented here will be used as the basis for further research in the area of medical device software traceability. It will also be utilized in Medi SPICE (Mc Caffery et al., 2010; Mc Caffery and Dorling, 2009) a software process assessment and improvement model specifically for the medical device industry. The Regulated

Software Research Group is currently developing Medi SPICE in collaboration with international standards bodies and the medical device industry.

Med-Trace will continue to be refined based on the results of ongoing research and feedback from future assessments and practitioners. The goal is to roll out Med-Trace nationally and internationally to assist with traceability. Given the positive response it has received, it is envisaged that research will be undertaken into the development of a tool to automate Med-Trace. The objective of the tool will be to facilitate the international roll out of Med-Trace and encourage its wider use. It is planned that the tool will also collect metrics which will be automatically passed back to the Regulated Software Research Group for analysis. This will assist with the future development of Med-Trace and Medi SPICE.

Acknowledgments This research is supported by the Science Foundation Ireland (SFI) Stokes Lectureship Programme, grant number 07/SK/I1299, the SFI Principal Investigator Programme, grant number 08/IN.1/I2030 (the funding of this project was awarded by Science Foundation Ireland under a co-funding initiative by the Irish Government and European Regional Development Fund), and supported in part by Lero – the Irish Software Engineering Research Centre (http://www.lero.ie) grant 03/CE2/I303-1. The research presented in this chapter was partially funded by the ARTEMIS Joint Undertaking of the European Commission, under grant agreement n° 100022 (CHARTER).

Appendix: Sample Scripted Med-Trace Questions

Question	**Source –** Software Traceability Literature	**Source –** Medical Device Standards
What kind of resources are provided for the activity of traceability management?	Ramesh (1998)[a]	
Is there a documented procedure in place for traceability? Is training provided on traceability and to what extent is explicit knowledge made available on software traceability	Ramesh (1998)[a]	
Implementation of traceability – Forward, Backward Traceability and the Relationship between Requirements (Dependent Requirements), Traceability tracking from the safety perspective and traceability to hazards/risk management	de Leon and Alves-Foss (2006)[a]	

(continued)

Question	**Source –** Software Traceability Literature	**Source –** Medical Device Standards
Where does traceability start – market requirements, product roadmap, system specifications? Where does proper requirement tagging start and how is it documented? Does any tool support this? How is safety classification in traceability achieved?		
How is traceability established between System Requirements, Software Requirements, and Software System testing?		Section 5.1.1 (ANSI/AAMI/IEC 62304:2006, 2006)[a]
How are software requirements traceable to system requirements and how is this verified?		Section 5.2.6 (ANSI/AAMI/IEC 62304:2006, 2006)[a]
How is traceability demonstrated between the software requirements and software system testing?		Section 5.7.4 (ANSI/AAMI/IEC 62304:2006, 2006)[a]
What traceability activities are undertaken during the design phase?		Section 3.2 (US FDA Center for Devices and Radiological Health, 2002)[a]
What traceability activities are undertaken during the coding and construction phase?		Section 5.2.4 (US FDA Center for Devices and Radiological Health, 2002)[a]
How are software systems test procedures traced to software and verified? What elements of system test procedures need to be traced? What are the difficulties in tracing? How does updating of results happen and how are they traced?		Section 5.7.4 (ANSI/AAMI/IEC 62304:2006, 2006)[a]
How are risk control measures traced to the software requirements?		Section 7.3.3 (ANSI/AAMI/IEC 62304:2006, 2006)[a]
How is traceability established between the risk control measures implemented in software?		Section 6.3 (ISO 14971:2007, 2007)[a]
The standard IEC 62304 specifies that the manufacturer shall document traceability of software hazards as appropriate: How is such complex traceability achieved? What are the tools available for achieving this?		Section 7.3.3 (ANSI/AAMI/IEC 62304:2006, 2006)[a]
How is traceability undertaken from the software related hazards and the software risk control measures to the corresponding safety-related software requirements and the software items that satisfy those requirements?		Section 3.5 (ISO 14971:2007, 2007)[a]

(continued)

Question	**Source –** Software Traceability Literature	**Source –** Medical Device Standards
How is software requirements traceability analysis conducted to trace software requirements to (and from) system requirements to risk analysis results?		Section 5.2.2 (US FDA Center for Devices and Radiological Health, 2002)[a]
What documentation do you use to provide traceability to link together design, implementation, testing, and risk management?		US FDA Center for Devices and Radiological Health (2005)[a]
In a software release, there is usually a process of noting down the known errors/known bugs. Is there a concept of traceability from these known bugs to the requirements or any other technical documentation?		Section 5.1.1 (ANSI/AAMI/IEC 62304:2006, 2006)[a]
How is the process of traceability measured and managed for effectiveness? Is there a way of consolidating feedback periodically on how well this process is performed?	Ramesh (1998)[a]	
To what extent has the organization automated traceability? What kind of tools are available which you think are useful for your organization? Have you evaluated them?	Higgins et al. (2003)[a], Feldmann et al. (2007)[a]	

[a] Denotes the relevant reference from the Software Traceability Literature or Medical Device Standards & Guidelines on which the question is based

References

AAMI TIR32:2004: Medical Device Software Risk Management. AAMI, Arlington (2005)

ANSI/AAMI SW68:2001: Medical Device Software – Software Life Cycle Process. AAMI, Arlington (2001)

ANSI/AAMI/IEC 62304:2006: Medical Device Software—Software Life Cycle Processes. AAMI, Arlington (2006)

Automotive SIG Automotive SPICE Process Assessment Ver. 2.2. August 2005

BS EN 60601-1-4:2000 Medical Electrical Equipment, Part 1 – General Requirements for Safety. BSI, London (2000)

Burton, J., Mc Caffery, F., Richardson, I.: A risk management capability model for use in medical device companies. In: International Workshop on Software Quality (WoSQ '06), Shanghai, China, May 2006. ACM, New York, NY, pp. 3–8

Casey, V.: Virtual software team project management. J. Brazil. Comp. Soc. **16**(2), 83–96 (2010)

CMMI Product Team: Capability Maturity Model® Integration for Development Version 1.2. Software Engineering Institute. Pittsburgh, PA (2006)

Damian, D., Moitra, D.: Global software development: How far have we come? IEEE Softw. **23**(5), 17–19 (2006)

de Leon, D., Alves-Foss, J.: Hidden implementation dependencies in high assurance and critical computing systems. IEEE Trans. Softw. Eng. **32**(10), 790–811 (2006)

Denger, C., Feldmann, R., Host, M., Lindholm, C., Shull, F.: A snapshot of the state of practice in software development for medical devices. In: First International Symposium on Empirical Software Engineering and Measurement, Madrid, Spain, 2007, pp. 485–487

DO-178B: Software Considerations in Airborne Systems and Equipment Certification. RTCA, USA, 1st Dec 1992

Espinoza, A., Garbajosa, J.: A proposal for defining a set of basic items for project-specific traceability methodologies. In: Proceedings of the 32nd Annual IEEE Software Engineering Workshop, Kassandra, Greece, pp. 175–184 (2008)

European Council: Council Directive 93/42/EEC Concerning Medical Devices. Official Journal of the European Communities, Luxembourg (1993)

European Council: Council Directive 2000/70/EC (Amendment). Official Journal of the European Union, Luxembourg (2000)

European Council: Council Directive 2001/104/EC (Amendment). Official Journal of the European Union, Luxembourg (2001)

European Council: Council Directive 2003/32/EC (Amendment). Official Journal of the European Union, Luxembourg (2003)

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

Feldmann, R.L., Shull, F., Denger, C., Host, M., Lindholm, C.: A survey of software engineering techniques in medical device development. In: Joint Workshop on High Confidence Medical Devices, Software, and Systems and Medical Device Plug-and-Play Interoperability, Cambridge, MA, USA, 25th–27th June 2007, pp. 46–54

GAMP 5:2008: A Risk-Based Approach to Compliant GxP Computerized System. ISPE, Florida (2008)

Gotel, O., Finkelstein, A.: Extended Requirements Traceability: Results of an Industrial Case Study. In: Proceedings of the 3rd International Symposium on Requirements Engineering, Annapolis, MD, USA, 6th–10th Jan 1997, pp. 169–178

Higgins, S.A., de Laat, M., Gieles, P.M.C., Geurts, E.M.: Managing requirements for medical IT products. IEEE Softw. **20**(1), 26–33 (2003)

IEC 60812:2006: Analysis Technique for System Reliability – Procedure for Failure Modes and Effects Analysis (FMEA), 2nd edn. IEC, Geneva, Switzerland (2006)

IEC 60880:2006: Nuclear Power Plants – Instrumentation and Control Systems Important to Safety – Software Aspects for Computer-Based Systems Performing Category A Functions. IEC, Geneva, Switzerland (2006)

IEC 62366:2007: Medical Devices – Application of Usability Engineering to Medical Devices. IEC, Geneva, Switzerland (2007)

IEC/TR 61508:2005: Functional Safety of Electrical/Electronic/Programmable Electronic Safety Related Systems. BSI, London (2005)

IEC/TR 80002-1:2009: Medical Device Software Part 1: Guidance on the Application of ISO 14971 to Medical Device Software. BSI, London (2009)

ISO 13485:2003: Medical Devices — Quality Management Systems — Requirements for Regulatory Purposes, 2nd edn. ISO, Geneva, Switzerland (2003)

ISO 14971:2007: Medical Devices — Application of Risk Management to Medical Devices, 2nd edn. ISO, Geneva (2007)

ISO/DIS 26262: Road Vehicles – Functional Safety. ISO, Geneva, Switzerland (2009)

ISO/IEC 12207:1995: Information Technology — Software Life Cycle Processes. ISO, Geneva, Switzerland (1995)

ISO/IEC 15504-5:2006: Information Technology — Process Assessment — Part 5: An Exemplar Process Assessment Model. ISO, Geneva, Switzerland (2006)

Kannenberg, A., Saiedian, H.: Why software requirements traceability remains a challenge. Cross Talk: The Journal of Defense Software Engineering **22**(5), 14–17 (2009)

Lee, I., Pappas, G., Cleaveland, R., Hatcliff, J., Krogh, B., Lee, P., Rubin, H., Sha, L.: High-confidence medical device software and systems. Computer **39**(4), 33–38 (2006)

Liao, L., Qu, Y., Leung, H.: A software process ontology and its application. In: Workshop on Semantic Web Enabled Software Engineering, Galway, Ireland, Nov 2005

Mason, P.: On traceability for safety critical systems engineering. In: Proceedings of the 12th Asia-Pacific Software Engineering Conference, 2005, Taipei, Taiwan, 15th–17th Dec 2005

Mc Caffery, F., Burton, J., Casey, V., Dorling, A.: Software process improvement in the medical device industry. In: Laplante, P. (ed.) Encyclopedia of Software Engineering, vol. 1. CRC Press Francis Taylor Group, New York, NY (2010a)

Mc Caffery, F., Dorling, A.: Medi SPICE: An overview. In: International Conference on Software Process Improvement and Capability Determinations (SPICE), Turku, Finland, 2nd–4th June 2009, pp. 34–41

Mc Caffery, F., Dorling, A., Casey, V.: Medi SPICE: An update. In: International Conference on Software Process Improvement and Capability Determinations (SPICE), Pisa, Italy, 18–20 May 2010. Edizioni ETS, pp. 195–198 (2010b)

Mc Caffery, F., Taylor, P.S., Coleman, G.: Adept: A unified assessment method for small software companies. IEEE Software – Special Issue SE Challenges in Small Software Organization **24**(1), 24–31 (2007)

Medical & Radiation Emitting Device Recalls: FDA. http://www.accessdata.fda.gov/scripts/cdrh/cfdocs/cfres/res.cfm. Accessed 25 Nov 2010 (2010)

Nuseibeh, B., Easterbrook, S.: Requirements engineering: A roadmap. In: International Conference on Software Engineering, Limerick, Ireland (2000), pp. 35–46

Panesar-Walawege, R., Sabetzadeh, M., Briand, L., Coq, T.: Characterizing the chain of evidence for software safety cases: A conceptual model based on the IEC 61508 Standard. In: Third International Conference on Software Testing, Verification and Validation, Paris, 6th–10th Apr 2010, pp. 335–344

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

Ramesh, B.: Factors influencing requirements traceability practice. Communications ACM **41**(12), 37–44 (1998)

US FDA Center for Devices and Radiological Health: General Principles of Software Validation; Final Guidance for Industry and FDA Staff. CDRH, Rockville (2002)

US FDA Center for Devices and Radiological Health: Guidance for the Content of Premarket Submissions for Software Contained in Medical Devices. CDRH, Rockville (2005)

US FDA Center for Devices and Radiological Health: Off-The-Shelf Software Use in Medical Devices; Guidance for Industry, Medical Device Reviewers and Compliance. CDRH, Rockville (1999)

Wallace, D.R., Kuhn, D.R.: Failure modes in medical device software: An analysis of 15 years of recall data. Int. J. Reliability, Quality, Safety Eng. **8**(4) (2001)