Impact of Standards on the Role and Application of Traceability in the Medical Device Domain

Gilbert Regan, Fergal Mc Caffery, Kevin Mc Daid, Derek Flood

{gilbert.regan,fergal.mccaffery, kevin.mcdaid, derek.flood}@dkit.ie

Abstract. Software failure in the medical device domain can lead to injury or death. Controlling this risk is fundamental to producing quality software. To produce quality software, an effective requirements and hazards traceability process is required. Hence traceability is central to medical device software development. It is also an essential requirement for regulatory approval. The necessity for traceability is reinforced by the medical device standards and guidelines. In this paper we outline how traceability is an important part of medical device software development, what standards contain reference to traceability, and which specific clauses within those standards companies should refer to when defining their traceability processes. We also summarise the findings obtained when a lightweight assessment method (Med-Trace), that we created, based upon the traceability practices within these standards, was implemented in two SME organizations.

Keywords: Med-Trace, Medical Device Standard, Software Traceability, Software Process Assessment and Improvement

1 Introduction: Background to Medical device software

Software-based medical devices are playing an increasingly important part in healthcare. Many medical devices must interface with other equipment, connect to hospital and laboratory information systems, and work in high-stress situations. The increased demands on such devices has resulted in increased software complexity and has created formidable development challenges for their manufacturers [1]. This increased complexity has resulted in the need for increased traceability and risk control measures.

In order to market their devices within a country, a medical device development company must comply with the regulatory requirements of that country [2]. Although guidance exists from regulatory bodies on what software activities must be performed, no specific method for performing these activities is outlined or enforced [3]. In 1993 the European Council released Medical Device Directive (MDD) 93/42/EC [4]. The purpose of this directive was to ensure the safety of medical devices placed
on the European market. This directive has been amended by Directives 2000/70/EC [5], 2001/104/EC [6], 2003/32/EC [7] and 2007/47/EC [8]. To this end, in the USA, the Food and Drug Administration (FDA) Center for Devices and Radiological Health (CDRH), independently from the European Council, published guidance papers which include risk based activities to be performed when using off-the-shelf software [9], during software validation [10], and for pre-market submission [11]. These documents however did not enforce any specific activity for performing the software activities, hence manufacturers could fail to comply with expected requirements. Therefore the medical device industry decided to recognise ISO/IEC 12207 [12] (general software engineering process standard) as suitable for general medical device software development. However the Association for the Advancement of Medical Instrumentation (AAMI) identified pitfalls in ISO/IEC 12207 and produced AAMI SW68 [13] (Medical Device- Software Lifecycle Processes) which was based on ISO/IEC 12207. In 2006 a new standard AAMI/IEC 62304 [14] was released which replaced AAMI SW68.

The remainder of this paper is structured as follows; Section 2 discusses the importance of traceability in all domains culminating with the medical device domain. Section 3 identifies traceability requirements within the medical device standards. Section 4 considers the implementation and findings of Med-trace, a traceability assessment model. In Section 5 we draw our conclusions.

2 Traceability

2.1 Introduction

In engineering terms a trace is comprised of a source artifact, a target artifact and the link between them [15]. Traceability is the ability to establish and use these traces. Numerous definitions for traceability exist in the literature but one of the most popular and encompassing is:

"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 all periods of on-going refinement and iteration in any of these phases" [16].

In general, traceability is about understanding a design right through from the origin of the requirement to its implementation, test and maintenance. Traceability allows us to understand aspects such as to whether the customers’ requirements are being met, the specific requirements that an artefact relates to, and the origins and motivation of a requirement. Traceability helps ensure that ‘quality’ software is developed.

2.2 Traceability in all domains

Software systems are becoming increasingly complex. Artefacts such as test cases, requirements documents, source code, design documents, bug reports etc, and the
links between them are created over long periods of time by different people. Creating and maintaining these links is a difficult and expensive task. Therefore most existing software systems lack explicit traceability links between artefacts [17]. Traceability was initially used to trace requirements from their source to implementation and test, but now plays an increasing role in defect management, change management and project management. Increasingly software development is globally distributed across multiple teams and sites which makes traceability even more important [18]. As traceability provides an essential support for developing high quality software systems [19], it is vital to engage an efficient traceability process. Traceability implementation is mandated in many software development standards and many industries, in particular the safety critical industries [20]. For example in the US the Food and Drugs Administration states that code must be linked to requirements and test cases. Safety critical products can be dangerous because failure can result in loss of life, significant environmental damage, or major financial loss [21]. Safety critical systems must satisfy a range of functional and non-functional requirements, including safety, reliability and availability. Regulation normally requires safety critical systems are certified before entering service. This involves submission to the appropriate regulator of a safety case (a reasoned argument that safety requirements have been met and the system is acceptably safe to operate) must be made for a safety critical systems as regulation requires these systems are certified before entering service [22].

2.3 Traceability in the medical device domain

In the medical device domain, software development is a difficult and complex endeavor. Defective medical device software can cause serious injury or death. Therefore safety is a key concern [18]. In the period from 7th Feb 2011 and 7th Feb 2012 the FDA recorded 151 medical device recalls and state software as the cause [23]. The number of devices that have recently been recalled due to software and hardware problems is increasing at an alarming rate [24]. During 2009 the FDA recalled 63 medical devices because of software issues. During 2010 they recalled 107 medical devices for the same reason. It is incumbent on medical device manufacturers to ensure, to the best of their ability, that software-based medical devices are safe and effective. Meeting this responsibility requires expertise in effective risk management practices, familiarity with software safety, and the adoption of a risk management mind-set [1]. Manufacturers must establish effective software development processes that are based on recognized engineering principles appropriate for safety critical systems. At the heart of such processes, they must incorporate traceability. Generally there is a lack of published material regarding traceability in medical device software in addition to a lack of guidance on how to implement traceability effectively in organisations [18]. As traceability is central to the development of medical device software, a traceability assessment and improvement method called MedTrace [20] has been developed (See section 4).
3 Medical device software standards and guidelines

Software traceability is central to medical device software development and essential for regulatory compliance. The need for this compliance is highlighted in many of the medical device software standards and guidelines. In order to understand the generic requirements for traceability and in particular the requirements for traceability within the medical device domain, a literature review of generic, safety critical and medical device domains was conducted. Detailed requirements for traceability, as expressed by the medical device standards and guidelines, are summarised in this section. Table 1, details the number of times (including section numbers for each instance) each standard identifies traceability. Table 2 provides an example of two of these references.

| Table 1: Number of times (and section numbers) each standard impacts traceability |
|---------------------------------|-----------------|-----------------|
| **Standard Title**              | **No.**         | **Section Numbers** |
| ANSI/AAMI/IEC 62304:2006 Medical device software—Software life cycle processes (2006) | 6 | 5.1.1; 5.2.6; 5.7.4; 7.3.3; 8.2.4; B.6.2; |
| Medical Device Directive 2007/47/EC | 1 | ANNEX I 12.1 |
| General Principles of Software Validation; U.S. Food and Drug Administration 2002 | 6 | 3.1.2; 3.2; 5.2.2; 5.2.3; 5.2.4; 5.2.5; |
| Guidance for the Content of Premarket Submissions for Software Contained in Medical Devices; US FDA 2005 | 2 | Page 11; Page 16; |
| Off-The-Shelf Software Use in Medical Devices: US FDA 1999 | 1 | 5.5.1 |
| ISO 14971:2007(E) - Medical devices — Application of risk management to medical devices | 1 | 3.5; |
| IEC/TR 80002-1:2009 - Medical device software Part 1: Guidance on the application of ISO 14971 | 8 | 3.5; 6.3; Table C; Table D |
| ISO 13485 (2003) Medical devices — Quality management systems — Requirements for regulatory purposes | 2 | 7.5.3.2.1; 7.5.3.2.2; |
Table 2: An example of Practice content relating to traceability taken from two standards as referred to in Table 1

<table>
<thead>
<tr>
<th>Standard Title</th>
<th>Process</th>
<th>Practice</th>
<th>Practice Content</th>
</tr>
</thead>
<tbody>
<tr>
<td>ANSI/AAMI/IEC 62304:2006 Medical device software — Software life cycle processes (2006)</td>
<td>Software development Process 5.0</td>
<td>Software development planning 5.1</td>
<td>The manufacturer shall establish a software development plan which should ensure TRACEABILITY between SYSTEM requirements, software requirements, SOFTWARE SYSTEM test, and RISK CONTROL measures implemented in software;</td>
</tr>
<tr>
<td>ANS/AAMI/IEC 62304:2006</td>
<td>Software configuration management process 8.0</td>
<td>Change control 8.2</td>
<td>Traceability change 7.3</td>
</tr>
</tbody>
</table>

Failure in medical device software can have fatal consequences. The gravity of these consequences is highlighted in the medical device standards through reiteration of the necessity to control risks. Traceability can control risk. For example the General Principles of Software Validation (GPSV) [10] states that a software requirements traceability analysis should be conducted to trace software requirements to (and from) system requirements and to risk analysis results. Moreover ISO 14971:2007 [25] requires the manufacturer to establish and maintain a risk control file which shall provide traceability for each identified hazard to a) risk analysis, b) risk evaluation, c) implementation and verification of risk control measures and d) the assessment of the acceptability of any residual risks. The documentation of risk control measures is emphasised by ANSI/AAMI/IEC 62304 [14] which directs the manufacturer to document traceability of the software hazards: from hazard situation to 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. The imperative for risk control is further called for in Off-The-Shelf Software Use in Medical Devices [9], Guidance for the Content of Premarket Submissions for Software Contained in Medical Devices [11] and IEC/TR 80002-1 [26].

There is considerable variance in the level of traceability detail required within the standards. Some of the standards provide very little detail as to between which stages of the Software Development Life Cycle (SDLC) traceability should be provided e.g. ISO 13485 requires an organization to establish documented procedures for traceability and that such procedures shall define the extent of product traceability and the records required. However other standards provide a greater level of required traceability detail such as GPSV which requires traceability from system requirements to software requirements and through each stage of the SDLC including design, code (including modules and functions) and test (traceability from test to detail design, high level design and to software requirements). Moreover the Guidance for the
Content of Premarket Submissions for Software Contained in Medical Devices

state that explicit traceability must exist among requirements, specifications, identified hazards and mitigations and among verification and validation testing.

The ability to trace change is good practice for software development in general and of necessity for medical device software development. The necessity for change management is emphasised in ANSI/AAMI/IEC 62304 when it states that the manufacturer shall create an audit trail whereby each a) Change request b) Problem report and c) Approval of change request, can be traced. It further requires that approved change requests are made traceable to the actual modification and verification of the software.

4 Med-Trace assessments and findings

4.1 Development of the Med-Trace Assessment Method

Due to the safety critical nature of medical device software, a company must meet 'country specific' regulatory requirements in order to market their product in that country. An effective traceability process is a crucial requirement to achieving regulatory compliance. Due to a lack of specific guidance within the medical device standards and documentation, achieving an effective traceability process is problematic, resulting in many medical device companies engaging inefficient traceability processes [20]. Consequently, a method (known as Med-Trace [20]) of assisting medical device software companies to improve their traceability processes and to adhere to the traceability aspects of the medical device software standards (as detailed in section 3) was developed. Med-Trace is a lightweight software traceability process assessment and improvement method for the medical device industry. Med-Trace is based on traceability best practices emanating from software engineering process models (CMMI_R, ISO/IEC 15504-5), software engineering traceability literature and medical device software standards and guidelines i.e. the traceability practices that are expressed in table 1.

4.2 Med-Trace Implementation and observations

This section discusses how the Med-Trace assessment method was implemented in two medical device organizations and the resulting observations. The objectives of the case studies were to demonstrate how Med-Trace could be used to assess the current status of the software traceability processes within similar organisations and to discover the main problems that medical device software development organizations face in terms of traceability.

Med-Trace was implemented in two Small to Medium Sized (SME) medical device organisations. Both organisations developed electronic based medical devices that require compliance with both the FDA and the MDD. One organisation was based in Ireland while the other was based in the UK.
From the Med-Trace assessment the following observations were made across both organisations:

- A member of management was responsible for implementing traceability and its importance in medical device software development was recognised and understood.
- Tracing requirements and managing risk was recognised as difficult and complex.
- There is a lack of detailed guidance on how to implement traceability.
- Their process for software development with regard to traceability needed to be improved and formulised.
- The requirement for relevant training and the ability to record and leverage best practice with regard to traceability also emerged.
- The need for automated tools to manage traceability was recognised as was the serious limitation of using manual tools.
- Financial constraints needed to be considered when adapting automated tools.

Both organisations considered Med-Trace to be worthwhile and very relevant and appreciated the fact the Med-Trace is lightweight. The findings report addressed key areas where improvements were required and both organisations agreed to adapt the resultant traceability process improvement plan and agreed to be reassessed.

6 Conclusions and Future Work

An effective traceability process is essential when developing medical device software due to its safety critical nature. The requirement for effective traceability is mandated by the medical device standards and guidelines and its importance is evident from the number of times traceability is referred to in these standards and guidelines. However, the implementation of an effective traceability process is recognised as difficult and complex.

While effective traceability is mandated by the standards and its necessity was understood by the two organisations who participated in the Med-Trace assessment, there is a lack of detailed guidance in how to best implement an effective traceability process within the medical device software domain. There currently are a challenging number of standards governing medical device software development and to determine the exact traceability requirement from each of these standards can be time consuming. Med-Trace addresses these challenges by providing a lightweight assessment method which may be used to diagnose an organisation’s strengths and weaknesses in relation to traceability in their software development processes.

The limitations (slow, tedious and prone to error) of using manual traceability tools such as Excel is an issue that needs addressing. Automated tools mitigate these limitations to some extent, however there are concerns around these tools such as cost, missing traces, needless traces, training, and the fact that tools require validation in their own right. Automated tools alone don’t provide for accountability and so human intervention in safety critical domains such as medical device software development is necessary. The limitations of existing automated traceability tools imply the need for further development of effective tools for SME organisations operating in the safety critical domain.
To-date, Med-Trace has been applied in two SME organisations and has been well received. It is envisaged that Med-Trace will continue to be refined based on ongoing research and feedback from future assessments. Future plans include a tool to automate Med-Trace with the objective of facilitating its national and international roll out and to encourage its wider use.

7 Acknowledgement

This research is supported by the Science Foundation Ireland (SFI) Stokes Lectureship Programme, grant number 07/SK/11299, 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 10/CE/I1855

8 References


9 Author CVs

Gilbert Regan
Gilbert Regan is a PhD student at Dundalk Institute of Technology. He is a member of the Regulated Software Research Group in Dundalk Institute of Technology and a member of Lero. His main research subject is Software Process Improvement with particular focus on Traceability within Medical Device Software.
**Fergal Mc Caffery**  
Dr Fergal Mc Caffery is a Lecturer with Dundalk Institute of Technology. He is the leader of the Regulated Software Research Group in Dundalk Institute of Technology and a member of Lero. He has been awarded Science Foundation Ireland funding through the Stokes Lectureship, Principal Investigator and CSET funding Programmes to research the area of software process improvement for the medical device domain. Additionally, he has received EU FP7 and Enterprise Ireland Commercialisation research funding to improve the effectiveness of embedded software development environments for the medical device industry.

**Derek Flood**  
Dr. Derek Flood is a post doctoral research fellow with the Regulated Software Research Group at Dundalk Institute of Technology (DkIT). Derek has recently worked as a post doctoral research assistant at Oxford Brookes University on the PACMAD project, examining the usability issues associated with mobile applications. Through this work Derek has published at a number of international conferences such as the IADIS IHCI and INTERACT. His current research interests include Software process improvement for Medical Devices, Spreadsheet risk, Usability of mobile applications, and the psychological impact of mobile applications on end users.

**Kevin McDaid**  
Kevin is a lecturer in Computing at Dundalk Institute of Technology. He leads research in the area of Spreadsheet Engineering in the Software Technology Research Centre. His background is in Mathematics and Statistics and his research focus on the application of statistical methods to decision problems in software development.