Implementation of Traceability Best Practices within 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

Requirements validation, compliance verification and impact analysis are important activities that are performed during the software development lifecycle. Traceability of requirements through the software development lifecycle (SDLC) is essential in the development of safety critical software. Organisations such as the Food and Drug Administration and the Federal Aviation Authority in the United States require traceability as part of their approval process. However, despite its criticality there is extensive digression in the practices and usefulness of traceability across development projects. Many projects' traceability efforts are simply focused on satisfying regulations and do not leverage the many benefits of traceability. Traceability, if fully implemented is an important tool for managing system development and there are a number of published best practices to help companies with this implementation. By means of a literature review we record a list of the commonly accepted best practices for traceability implementation. Furthermore, through interviews with two medical device companies we report that a number of these practices are unfamiliar to these companies and why an even greater number of these practices are not applied.

Keywords

Requirements traceability, Traceability best practices, Medical device, Safety critical

1 Introduction

Traceability is the ability to establish links (or traces) between source artefacts and target artefacts [1]. Requirements tracing is concerned with recovering the source of requirements and predicting the effects of requirements [2]. 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 aids impact analysis, component re-use, change management and generally supports an improvement in competitive advantage. Traceability helps ensure that 'quality' software is developed [3].

Safety critical industry standards, such as the medical device standards mandate that traceability be implemented e.g. the Food and Drugs Administration (FDA) state that the validation of software typically includes evidence that all software requirements have been implemented correctly and completely and are traceable to system requirements and to risk analysis results, that the software design implements all of the software requirements and that all code is linked to established specifications and

established test procedures[4]. IEC 62304:2006 [5] is a harmonised standard which defines the life cycle requirements for medical device software and requires traceability between system require-ments, software requirements, software system test, and risk control measures implemented in soft-ware, and that the manufacturer shall create an audit trail whereby each change request, relevant problem report and approval of the change request can be traced. In addition to these requirements regulation normally requires critical systems are certified before entering service. This involves sub-mission of a safety case - a reasoned argument and supporting evidence that requirements (non-functional requirements such as safety, availability and reliability) have been met and that the system is acceptably safe [6].

However, despite its many benefits and regulatory requirements, traceability is a tool that often is grudgingly implemented, and, if implemented is often not leveraged to take advantage of the infor-mation it can provide to a development or validation team. Numerous reasons have been identified for reluctance in implementing traceability including cost and complexity. Reasons include the task of building a requirements trace matrix (RTM) is time consuming, arduous and error prone [7], there are few metrics for measuring the return on investment for traceability, stakeholders within an company have differing perceptions as to the benefits of traceability [8], the need for documentation can cause resentment among developers who may fear that traces could be used to monitor their work [9], diffi-culties with trace tools including selecting between available tools, and difficulties configuring a gen-eral purpose tool or developing a custom tool [10]. Finally almost no guidance is available for practi-tioners to help them establish traceability in their projects and as a result, practitioners are ill-informed as to how best to accomplish this task [11, 12].

Notwithstanding the lack of guidance, there are a number of commonly accepted best practices which can direct practitioners with the implementation and maintenance of traceability. In section 2 of this paper we present twenty three best practices which we have identified from literature [8, 13-23]. In section 3, we   discuss our findings from interviewing two small to medium sized (SME) medical device companies about their application of these best practices. We then analyse our findings in section 4 before providing our conclusions in section 5.

2   Traceability best practices

A review of the literature has revealed the following commonly accepted best practices (BP) for the efficient implementation and maintenance of requirements traceability through the SDLC.

BP 1: Adopt a company policy for traceability [8, 14]. A company must adopt consistent practices for requirements management, including traceability. Different stakeholders can have different viewpoints on the need/value of traceability.If all stakeholders buy into the traceability policy, it greatly increases the chances of success. As Kannenberg and Saiedian note, "Perhaps the best way to deal with the problem of different stakeholder viewpoints on traceability is to create an organizational policy on traceability to apply uniformly to all projects."

BP 2: Implement a standard operating procedure (SOP) for traceability. In clinical research, the *International Conference on Harmonization* (ICH) defines SOPs as "detailed, written instructions to achieve uniformity of the performance of a specific function". SOPs help address ' the lack of detailed guidance on how to implement traceability [11]' as they are an integral part of a successful quality system as it provides individuals with the information to perform a job properly, and facilitates con-sistency in the quality and integrity of a product or end-result and help ensure compliance with gov-ernmental regulations.

BP 3: Develop a traceability information model (TIM) [15]. A TIM models the traceable artifact types (i.e. requirements, design, code etc.) and their permitted trace links as a unified modeling language (UML) class diagram. The benefits of a TIM are that it [16]:
- Ensures consistent results in projects with multiple stakeholders;
- As traceability is also used by people who did not create it, these people need to know how it has been defined and what to expect from it;

- As tracing is a complex task, a TIM provides a guideline to ease its set up and allows for the validation of changes;
- Coverage analysis is only possible after having defined what the expected coverage is;
- A TIM is a necessary precondition for automated traceability handling, validation and analyses

BP 4: Provide tool support for traceability [15]. Creating and maintaining traceability can be a time consuming and complex task. Requirements management/ traceability tools provide features for establishing, maintaining and navigating trace links and can display information in matrix or trace slice format. It is worth considering tool support when [1]:
- A project has many requirements,
- When more than one person, site or company is doing the engineering and requirements management work, and where there is a need to share and align artefacts and traces.
- When requirements and other engineering artefacts, including their traces, are being used and reused in multiple ways, such as within other projects and within product families.
- When the engineering personnel are performing repetitive and administrative tasks to enable requirements management.
- When a long project or product life is expected, or when there are many customers with likely change requests to manage.

BP 5: Keep it simple [18] Capture "just enough information". Limit number of data points. Capturing too much information makes the system burdensome, complex and error prone, however one must ensure that enough information is captured so as to make it useful.

BP 6: Create traces incrementally [15]. Stakeholders are empowered to create trace links incrementally within the context of their daily work. This avoids the situation of traceability being deferred until the end of a project where its perceived purpose is for regulatory reasons. Creating traceability as the project progresses allows stakeholders to benefit from traceability knowledge throughout the project.

BP 7: Unique identifiers must be adopted for requirements and business rules [14, 18-20]. To permit traceability, each requirement must be uniquely and persistently labelled so that you can refer to it unambiguously through the project. Unique identifiers follow the requirement throughout the workflow and are never reused or reassigned.

BP 8: A responsible party must take ownership of traceability [8, 14]. Gathering and managing requirements traceability data must be made the explicit responsibility of certain individuals or it won't happen. Further, errors will occur if someone who is not familiar with the system or the requirements attempts to make updates, errors will abound. Updates must be practised consistently or traceability will degrade and become untrustworthy.

BP 9: Practice value based requirement tracing instead of full tracing [8, 14, 17]. Value-based requirement tracing prioritises all of the requirements in the system, with the amount of time and effort expended on tracing each requirement depending on the priority of that requirement. This can save a significant amount of effort by focusing traceability activities on the most important requirements. However, value-based tracing requires a clear understanding of the importance of each requirement in the system; it may not be an option if full tracing is a requirement of the customer or the development process standards used for the project.

BP 10:Clearly identify stakeholder or source [18]. Associate each requirement with a named person or customer or other source of the requirement such as a regulation or requirement from a standard.

BP 11: **Educate the team about the concepts and importance of requirements tracing** [8, 19]. Many companies do not train their employees regarding the importance of traceability and traceability is not emphasized in undergraduate education.

BP 12: Centralise - Requirements traceability should be documented centrally using some type of log e.g. a traceability matrix [21].

BP 13: Audit/Review the traceability information periodically to make sure it is being kept current [19, 21].

BP 14: Inventory Current Processes and Tools [22]. Review methods and practices to ensure they are not outdated? There are always new tactics and development methodologies that can give you a competitive advantage for each product or project, but are they compliant?

BP 15: Start tracing at the beginning of analysis [18]. It is considered much harder to implement traceability if you wait until the end of the project. Less value to the team and company

BP 16:Periodically communicate improvements from the practice of traceability [18]. Quantify savings from early defect detection or elimination of missing requirements

BP 17: Never use traceability as a measurement in performance reviews [8, 18, 23]. Individuals may be concerned that traceability data will be used against them in performance reviews or as a threat to their job security.

BP 18: Bidirectional Traceability [13]. Good traceability practices allow for bidirectional traceability, meaning that the traceability chains can be traced in both the forwards and backwards directions.

BP 19: Identify the key individuals [19] who will supply each type of link information and the personnel who will co-ordinate the traceability activities and manage the data.

BP 20: Explore Terminology and Meaning [22]. Get all participants to agree on one naming convention per artefact (if possible), especially if there are legal or compliance reasons for consistent artefact names. There may be situations where an agreement is not possible, in which case these artefact names should be documented or mapped to each other, even if they are essentially the same. For example, customers may call something a bug, QA may call it a defect, Development may call it an issue, and Engineering may call it an anomaly, but ultimately each group may be referring the same thing. Map these terms to each other and move on.

BP 21: Model traceability queries [15]. Traceability queries cover basic life-cycle activities such as finding all requirements associated with currently failed test cases or listing all mitigating requirements associated with a given hazard.

BP 22: Visualize trace slices [15]. In safety critical systems trace links established between hazards, faults, mitigating requirements, design, implementations, and test cases are of particular importance. Therefore, instead of presenting traceability material in the form of trace matrices, generate visualizations of trace slices in which the hazard is the root node, and all direct and indirectly traced artefacts that contribute towards mitigating the hazard are shown as a tree.

BP 23: Evaluate traces continually using a dashboard [15]. Tracing benefits are often not realized directly by the people performing the tracing tasks. Furthermore, the current status of the traceability effort is often not visible to individual stakeholders or to the project manager. A dashboard that displays the tracing progress for a project can be effective for tracking and managing the tracing goals of the project and also for motivating team members to create appropriate trace links.

3 *Best practices in practice*

3.1 Introduction

To examine the application of traceability best practices we interviewed two SME companies operating within the medical device domain. These two companies were selected because they were both SME's developing medical devices which contained software or developed devices which are standalone software. While both companies were considered SME's they were a different size (20 employees as against 70) and had different levels of experience in developing medical device software. The contrasting size and level of experience of these companies was chosen to provide a broader overview of the experience of traceability in the medical device domain.

Prior to conducting the interview with the companies, a questionnaire was developed to examine the company's overall approach to traceability. The questionnaire includes a set of direct questions were used to determine the companys' application of traceability best practices e.g. have you adopted a company policy on traceability or have you considered adopting one? One direct question was used for each of the twenty three best practices.

In addition to examining the best practices for traceability adopted by the companies, the interview also examined their compliance with the medical device standards, the difficulties that are encountered while implementing traceability and opportunities that may exist for improving their traceability process. A full discussion of these topics is beyond the scope of this paper, however the interested reader is referred to [24]for more details.

3.2 Company Profile

Company A
This company is solely located in Ireland and has a total workforce of less than 20 people. The software systems are provided on-line and have been available for up to 11 years now. The company have 2 full time developers mainly working on upgrades or specific customer requests. Their products are rated as software safety classification I, meaning they may not present much risk of illness or injury. The company uses the Waterfall model for their software development.

Two people were interviewed together and they answered questions both together and individually. The first individual's position was as Quality Control Manager and they had been with the company for eighteen months and had many years' experience in this position. The second individual was a Project Manager and had been with the company for 6 months.

Company B
This company's headquarters are based in the UK, but it has a research and development and manufacturing facility based in Ireland. This study was carried out in the Ireland facility. The company employs 60 to 70 people and sometimes employs contractors on a part time basis so the numbers can fluctuate. The products are marketed globally into the primary care market, secondary care, occupational health, sports medicine and clinical trials. Their products are rated as software safety classification II, meaning non-serious injury to the patient or operator of the device is possible due to a defect in the device. The company uses the V model for their software development.

Two people were interviewed separately from this company. The first person was the Chief Technical Officer and had been with the company for approximately ten years. The second person was the Software Development Manager and he had been with the company for approximately ten years.

3.3 Findings

Table 1 below summarises Company A & Company B's responses to questions about their application of traceability best practices. The first column refers to the best practice numbers as detailed in section 2 of this paper.

Table 1: Implementation of Traceability Best Practices

Best Practice	Company A	Company B
BP 1	No company policy on traceability in place and no plans to implement one. Have never considered it.	No company policy on traceability in place and have not considered implementing one.
BP 2	No traceability SOP in place. No plans to implement one.	A traceability SOP is in place. It basically details how to complete the trace matrix.
BP 3	No TIM in place. Have never considered its implementation.	No TIM in place. Would consider its implementation.
BP 4	No tool support for traceability except general purpose spreadsheet. Not considering trace tool support at present.	No tool support for traceability except general purpose spreadsheet but are actively considering the purchase of a tool.
BP 5	Had a consultant to review this. Find it difficult to know exactly what links to make.	Knowing how much information to capture while at the same time being efficient is difficult.
BP 6	Not something the company has thought about. Would probably need to review	Trace matrix mostly deferred towards the end of the project so stakeholders don't benefit from traceability knowledge throughout the project
BP 7	Each requirement is uniquely identified.	Each requirement is uniquely identified
BP 8	One of the developers is fully responsible although this is not documented.	It is the responsibility of the software development manager to ensure that traceability gets implemented.
BP 9	Don't practice VBRT. Trace every requirement because operating in safety critical domain.	Don't practice VBRT. Trace every requirement because operating in safety critical domain.
BP 10	Source of requirements is identified but not in trace matrix. A developer keeps a record of this himself (not in a company document).	Source of requirements is identified.
BP 11	No education of employees as to the benefits of traceability..	No education of employees as to the benefits of traceability
BP 12	A traceability matrix is in place.	A traceability matrix is in place
BP 13	No formal audit/review of traceability information. It is left to the developer to ensure it is correct so any audit is on an ad-hoc basis.	No formal audit/review of trace information except at end of project where it is signed off by the software develoment manager.
BP 14	Had a consultant who reviewed traceability technique but made no significant change. No formal plan in place for review/audit of traceability approaches.	Company are currently reviewing their approach by considering purchase of trace tool. No formal plan in place for review/audit of traceability approaches.
BP 15	In most cases the matrix gets completed towards the end of the project.	The matrix does not get maintained as and when it should.
BP 16	Improvements made due to traceability are not communicated within the company.	Improvements made due to traceability are not communicated within the company.
BP 17	Traceability data is never used in staff performance reviews.	Traceability data is never used in staff performance reviews.
BP 18	Bi-directional tracing available through trace matrix but difficulty in tracing back from Technical Spec. to Functional Spec.	Bi-directional tracing available through trace matrix and in-document tracing.
BP 19	One developer is responsible for all links	Key individuals were identified.
BP 20	The company had yet to engage in naming conventions and this issue was causing some confusion.	The company has engaged in naming conventions.
BP 21	Had not considered or were aware of this practice	Had not considered this practice
BP 22	Had not considered or were aware of this	Had not considered this practice

	practice	
BP 23	Had not considered or were aware of this practice	Had not considered this practice

4 Discussion of Traceability Findings across both Companies

The first point to consider is the number of practices that each company implement with Company A implementing eight practices and Company B implementing 10 practices. The companies agreed that this was quite a low number considering the authors stated twenty three best practices and offered two main reasons for this. The first reason is that the companies were unaware of some of the practices and the defense for this was that they did not have the in-house traceability expertise. The second reason is that the companies generally viewed traceability as "a pain in the backside" and that "they would probably not bother with it if it wasn't for regulatory purposes."

Another point to note is that Company B implemented every practice that Company A implemented (i.e. BP 7, 8, 10, 12, 14, 17 , 18, 19) plus BP 2 and BP 20. The reason for this commonality may be that it is difficult to have any form of traceability in place without implementing these common practices i.e. traceability will not happen unless someone is made responsible and the easiest way to centrally document traceability is through a trace matrix which also somewhat facilitates bi-directional tracing. With regard to BP 14 Company A have had a consultant review their processes (including their traceability process) because as a result of an amendment to the Medical Device Directive [25] (where stand alone software can now be considered a medical device) their product is now considered a medical device. Company B have reviewed their process and have concluded that the addition of a trace tool would improve their process and are currently considering different tool options.

Company B have implemented two practices more than Company A, one of which is a SOP for traceability and we believe the reason for this may be due to the following two factors. Company B have been developing medical device software for more than thirty years and so has a lot of experience in developing medical device software relative to Company A. In addition to this Company B, being a bigger company, has more resources than Company A.

The lack of interest in traceability (outside of regulatory compliance) which the companies openly admit to, is perhaps better illustrated by the best practices that have not been applied (e.g. having no company policy or SOP in place). Being unaware of or not considering certain best practices only emphasises this indifference. The companies were not interested in the many benefits that requirements traceability offers with one interviewee stating 'I know I should say things like impact analysis but the truth of the matter is we only use traceability for ensuring all requirements have been tested completely and the matrix is very useful when an auditor comes in'.

5 Conclusion

The implementation and maintenance of traceability varies greatly between development projects. Many reasons have been put forward for this including cost, complexity and lack of guidance on how to implement traceability. However from the literature we have identified twenty three best practices, all or some of which a company can use to improve its traceability.

To assess the application of traceability best practices within the medical device domain we interviewed two medical device companies and found that Company A and B applied eight and ten practices respectively, with the eight practices been applied by Company A also been applied by Company B. This commonality of best practice application is mostly due to these common practices been basic practices necessary for any form of traceability implementation. Company B is a bigger and more experienced company in medical device software development and this is a contributor to it applying more practices than Company A. Another finding was that companies did not implement some of the practices because they simply were unaware of them, due mainly to the fact that they did not have the traceability expertise in-house. Finally companies found traceability to be 'a pain in the backside' and 'would not bother with it if it wasn't for regulatory reasons' and so were only interested in the minimum implementation necessary. This viewpoint did not stimulate the application of traceability best practices and meant that these companies were not availing of the full benefits that traceability has to offer.

6 Acknowledgement

7 References

1. Gotel, O. and P. Mader, Acquiring Tool Support for Traceability, in Software and Systems Traceability, J. Cleland-Huang, O. Gotel, and A. Zisman, 2012, Springer.

2. IEEE, Guide to the Software Engineering Body of Knowledge, SWEBOK, Editor. 2004.

3. Regan, G., et al., Impact of Standards on the Role and Application of Traceability in the Medical Device Domain, in EuroSPI. 2012, DELTA: Vienna, Austria. p. 2.11.

4. FDA, General Principles of Software Validation; Final Guidance for Industry and FDA Staff. 2002, CDRH: Rockville.

5. ANSI/AAMI/IEC, 62304:2006 Medical device software—Software life cycle processes. 2006, AAMI: Arlington, VA.

6. Mason, P., On Traceability for Safety Critical Systems Engineering, in Proceedings of the 12th Software Engineering Conference. 2005, IEEE Computer Society. p. 272-282.

7. Cleland-Huang, J., Just Enough Requirements Traceability, in Proceedings of the 30th Annual International Computer Software and Applications Conference - Volume 01. 2006, IEEE Computer Society. p. 41-42.

8. Kannenberg, A. and D.H. Saiedian, Why Software Requirements Traceability Remains a Challenge. CrossTalk The Journal of Defense Software Engineering, July/Aug 2009.

9. Jarke, M., Requirements tracing. Commun. ACM, 1998. 41(12): p. 32-36.

10. Regan, G., et al., The Barriers to Traceability and their Potential Solutions: Towards a Reference Framework, in 38th Euromicro Conference on Software Engineering and Advanced Applications. 2012, IEEE: Cesme, Turkey. p. 319- 322.

11. Caffery, F.M. and V. Casey, Med-Trace: Traceability Assessment Method for Medical Device Software Development, in EuroSPI 2011: Denmark. p. 1.1 - 1.8.

12. Mader, P., O. Gotel, and I. Philippow, Motivation Matters in the Traceability Trenches, in Proceedings of the 2009 17th IEEE International Requirements Engineering Conference, RE. 2009, IEEE Computer Society. p. 143-148.

13. Westfall, L. Bidirectional Requirements Traceability. 2006 [cited 2013 06/02/2013]; Available from: http://www.compaid.com/caiinternet/ezine/westfall-bidirectional.pdf.

14. Marchis, A. An Introduction to Requirements Traceability. Requirements.com 2011 [cited 2013 06/02/2013]; Available from: http://www.requirements.com/Topics/RequirementsTraceability/tabid/87/Default.aspx.

15. Mader, P., et al., Strategic Traceability for Safety Critical Projects. IEEE Software, 2013(In Press).

16. Mader, P., O. Gotel, and I. Philippow, Getting back to basics: Promoting the use of a traceability information model in practice, in Proceedings of the 2009 ICSE Workshop on

Traceability in Emerging Forms of Software Engineering. 2009, IEEE Computer Society. p. 21-25.

17. Rochimah, S., W.M.N. Kadir, and A.H. Abdullah. An Evaluation of Traceability Approaches to Support Software Evolution. in Software Engineering Advances, 2007. ICSEA 2007. International Conference on. 2007.

18. Thomas, B., Ensuring Project Success with Requirements Traceability, in Software Systems Best Practice Conference. 2012: Anaheim, California.

19. Morckos, M., Requirements Traceability. 2011, University of Waterloo: London.

20. Tvete, B. Introducing efficient requirements management. in Database and Expert Systems Applications, 1999. Proceedings. Tenth International Workshop on. 1999.

21. CDC. Unified Process Practice Guide - Requirements Traceability. 2006 [cited 2013 Feb 20]; Available from: http://www2.cdc.gov/cdcup/library/practices_guides/default.htm.

22. Seapine, Six Exercises to Strengthen Traceability. 2011, Seapine Software.

23. Ramesh, B., et al. Implementing requirements traceability: a case study. in Requirements Engineering, 1995., Proceedings of the Second IEEE International Symposium on. 1995.

24. Regan, G., et al. Investigation of Traceability within a Medical Device Organization. in International SPICE Conference on Process Improvement and Capability dEtermination in Software, Systems Engineering and Service Management. 2013. Bremen, Germany.

25. Council, E., Medical Device Directive 2007/47/EC. Official Journal of the European Union, 2007.

8 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 DKIT 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. 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 was formally a post doctoral research assistant at Oxford Brookes University on the PACMAD project. His current research interests include Software process improvement for Medical Devices and Usability

Kevin McDaid
Kevin is a lecturer in Computing at Dundalk Institute of Technology. He leads research in the area of Spread sheet 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.