

Assessing Traceability – Practical Experiences and Lessons Learned

Gilbert Regan, Miklos Biro, Derek Flood, Fergal Mc Caffery

{gilbert.regan, fergal.mccaffery, derek.flood}@dkit.ie
miklos.biro@scch.at

Abstract. Most existing software systems lack explicit traceability links between artefacts, or if implemented, is often not leveraged to take advantage of the information it can provide to a development or validation team.

Within the medical device domain, as in other safety critical domains, regulation normally requires such systems are certified before entering service. This involves submission of a safety case (a reasoned argument that the system is acceptably safe) to the regulator. A safety case should include evidence that the organisation has established effective software development processes. At the heart of such processes they must incorporate traceability.

However, numerous barriers such as a lack of awareness of traceability and a lack of guidance as to how to implement traceability hamper the its effective implementation. In this paper we address the lack of guidance on traceability implementation by presenting our experience of developing and trialling a traceability assessment model in two medical device organisations. We show that the assessment model was successful in identifying strengths and weaknesses in both organisations implementation of traceability. Through experience of trialling the model we propose an idea to improve it by automation, using the Open Services for Lifecycle Collaboration (OSLC) initiative.

Keywords: Requirements traceability, Traceability assessment, Medical device, Safety critical, Process assessment, Automation

1 Introduction

Traceability has been embraced by many software development standards such as ISO/IEC 15504 [1] and CMMI [2] and its importance is well recognised in the software engineering community. A report on the needs and priorities for software research, commissioned by the United States Department of Defense states that traceability is a practice that facilitates software assurance [3].

Traceability is the ability to follow the life of a requirement in both a forward and backwards direction (i.e., from its origins through its de-

velopment and specification to its subsequent deployment and use, and through all periods of on-going refinement and iteration in any of these phases) [4]. 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 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.

In addition to the many benefits offered by traceability, Industries are often compelled to implement traceability practices by government regulations [5]. For example Food and Drugs Administration (FDA) state that traceability should verify that software requirements are traceable to system requirements and to risk analysis results, a software design implements all of the software requirements, and that all code is linked to established specifications and established test procedures [6]. Similarly the Federal Aviation Authority (FAA) require that a Low Level Requirement (LLR) traces up to a High Level Requirement (HLR) and that each requirement is fulfilled by the source code, that each requirement is tested, that each line of source code is connected to a requirement.

However despite its many benefits and regulatory requirements it seems that *'often the quality of traceability information is poor, or out-of-date due to improper maintenance'* [7]. Reports from the field are few and far between when it comes to traceability. Much research has been conducted on traceability that references studies conducted more than a decade ago. However, we know very little about the traceability practice as it occurs in companies today [8], except for the fact that many software development enterprises do not maintain traceability in an appropriate way [9, 10]. In fact *'most existing software systems lack explicit traceability links between artefacts'* [11].

Numerous reasons have been identified for reluctance in implementing traceability including cost, complexity, building a requirements trace matrix (RTM) is time consuming, arduous and error prone [17], stakeholders having differing perceptions as to the benefits of traceability [12], developers may fear that traces could be used to monitor their work [18], and difficulties with trace tools [19]. Finally almost no guid-

ance is available for practitioners to help them establish effective traceability in their projects and as a result, practitioners are ill-informed as to how best to accomplish this task [8, 20].

To assist medical device organisations in addressing the lack of guidance on how to implement effective traceability, this paper presents the development and validation of a traceability process assessment model (PAM). We present our experience of developing and trialling a traceability assessment model in two medical device organisations and show that the assessment model was successful in identifying strengths and weaknesses in both organisations implementation of traceability. Through experience of trialling the model we propose an idea to improve it by automation, using the Open Services for Lifecycle Collaboration (OSLC) initiative

The remainder of this paper is structured as follows: Section 2 outlines current assessment models' relationship to traceability and the need to automate the assessment and maintenance of traceability. Section 3 outlines current assessment of traceability in medical device standards and guidelines and assessment models such as ISO 15504. Section 4 outlines the methodology used to develop the PAM while section 5 details the structure of the developed PAM. Section 6 presents validation of the PAM by expert review and industry trial while Section 7 discusses how traceability assessment and maintenance could be automated using the Open Services for Lifecycle Collaboration (OSLC) initiative. Finally section 8 concludes the paper.

2 Related Work

A literature review was conducted to determine what other traceability assessment models were available in the general, safety critical or medical device domains. This review returned only one model on traceability compliance/ capability assessment called Med-Trace [12]. Med-trace is a lightweight traceability assessment method, completed in 8 stages, whose goal is to assist medical device organizations to improve their software development traceability process. The authors completed assessments on two medical device companies and were able to identify areas for improvement in each company's traceability process.

There are a number of process assessment models which provide common frameworks for assessing software process capability. These models include ISO 15504 SPICE , Automotive SPICE [13], SPICE 4 SPACE [14], and CMMI. These frameworks assess processes such as software design process, software construction process, software testing process etc. However the frameworks do not include a dedicated traceability assessment process. The frameworks do include traceability assessment but it is spread out across a lot of processes and sometimes difficult to interpret e.g. base practice 4 of the software construction process (Eng. 6) in SPICE states;

“Verify software units. Verify that each software unit satisfies its design requirements by executing the specified unit verification procedures and document the results”.

Explicit traceability is not required in the above statement but it may be implied. It is open to interpretation.

It is important to highlight that traceability has been considered as a key issue by the agile community as well. Scott Ambler, one of the key personalities of the agile movement, states in 1999 that “My experience shows that a mature approach to requirements traceability is often a key distinguisher between organizations that are successful at developing software and those that aren’t. Choosing to succeed is often the most difficult choice you’ll ever make—choosing to trace requirements on your next software project is part of choosing to succeed.” [15]

The same Scott Ambler’s advice in 2013 [16]:
“Think very carefully before investing in a requirements traceability matrix, or in full lifecycle traceability in general, where the traceability information is manually maintained. When does maintaining traceability information make sense?

- Automated tooling support exists
- Complex domains
- Large teams or geographically distributed teams
- Regulatory compliance”

While the above view reflects the reluctance in implementing traceability as discussed in the introduction, it also shows its importance in the case of the medical device domain being both complex and subject to regulatory compliance requirements.

Considering all of the above discussion, the need for the automation of assessing and maintaining traceability is imminent. It is this automation

to which the Open Services for Lifecycle Collaboration (OSLC) initiative opens the way also as discussed in this paper.

3 Traceability Assessment

A Software process provides a framework for the key activities of software development. Good management of the process should provide for a sustained orderly improvement of the process. Software process assessment assist organizations in understanding the current state of their software process by identifying strengths and weaknesses in their process and thus providing focus on areas for improvement. In addition to assessing their own process an organization can use software process assessment to determine the state of a supplier's process.

3.1 Traceability Requirements

To understand the requirements for traceability each of the following standards were analysed and appropriate traceability requirements extracted:

- IEC 62304 - Medical device Software lifecycle processes [17]
- FDA – Guidance documents [6, 18, 19]
- ISO 13485 - Medical devices — Quality management systems [20]
- ISO 14971 - Application of risk management to medical devices [21]

While each of the above standards had different requirements for traceability, all the requirements were easily categorised as a requirement for the Software Development Lifecycle (SDLC), Risk Management or Change Management process. The requirements for the three processes are indicated in Figures 1, 2 and 3. As way of explanation of these figures, the double headed arrow in Figure 1 with the number 13 attached indicates a requirement for bidirectional traceability between code and unit test.

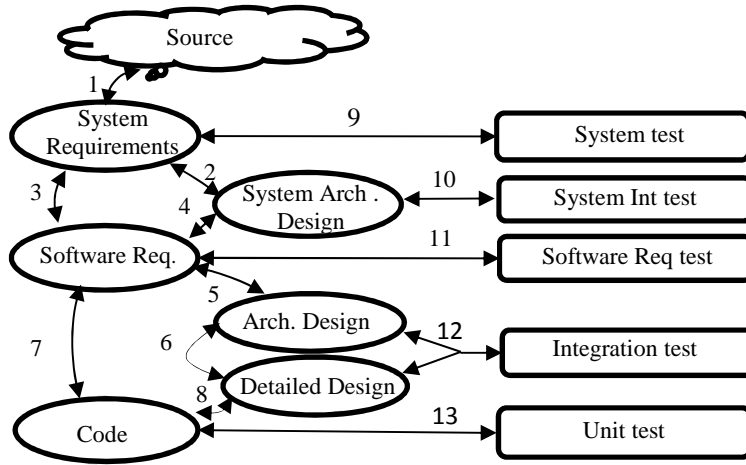


Fig. 1 SDLC and links between phases @Annex E of the Automotive SPICE® PAM

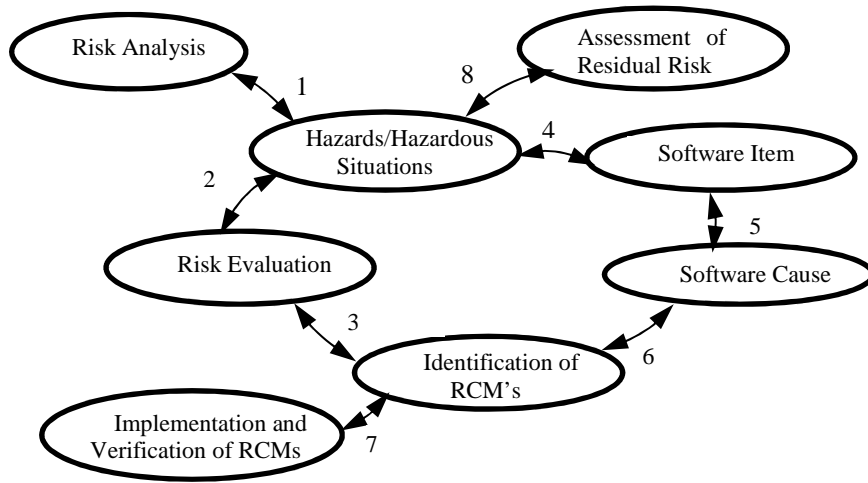


Fig. 2 Risk Management Traceability Overview

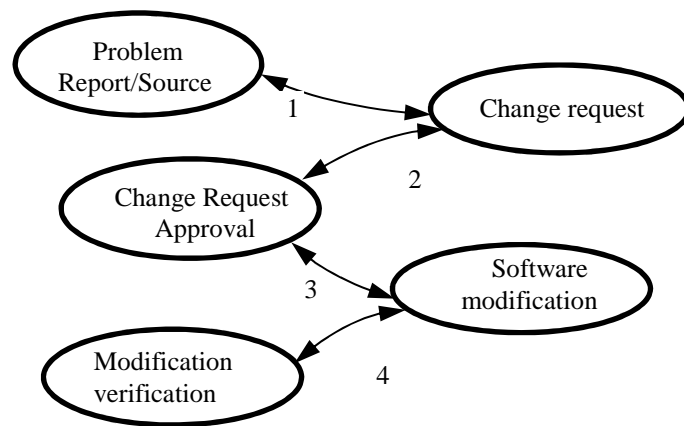


Fig. 3 Change Management Traceability Overview

The above figures thus indicate twenty five separate requirements for traceability as per the medical device standards and guidelines. These requirements along with the best practices for implementing traceability formed the basis for the PAM. The best practices for implementing traceability are the result of an extensive literature review which was completed prior to developing the PAM [22].

4 Research Methodology

The traceability PAM is based on ISO 15504-2 [23] and developed in line with the structure shown in Figure 2.

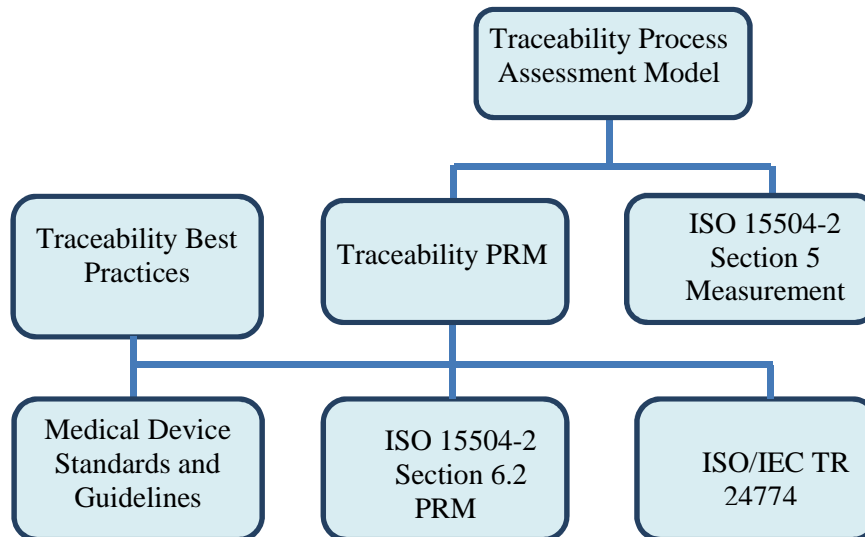


Fig. 4. Traceability Process Assessment Model Approach

It was decided to base the traceability assessment model on ISO/IEC 15504 as this improvement and capability determination model was derived from ISO/IEC 12207 [24] and since ANSI/AAMI/IEC 62304:2006 (Software lifecycle processes for medical device software) is also derived from ISO/IEC 12207 it was determined that there was good synergy between ANSI/AAMI/IEC 62304:2006 and ISO/IEC 15504. Additionally, 15504 is used extensively in other safety critical industries such as the automotive industry (Automotive SPICE), space industry (SPICE 4 SPACE) and the medical device industry (Medi SPICE).

The first stage was to develop a traceability PRM. The PRM was developed using the requirements from traceability (taken from the medical device standards and guidelines), traceability best practices (the result of an extensive literature review), and ISO 15504-2 section 6.2 which sets out the requirements for a Process Reference Model. While ISO 15504-2 details the minimum requirements that a PRM and a PAM should meet, it provides no guidance on how to develop the models i.e. it does not tell you how to transform requirements into a PRM or PAM. To address this issue, this study based the development of the PAM on the Tudor IT Service Management Process Assessment (TIPA) transformation process [25] which is a nine step process that complies with the requirements for PRMs and PAMs as expressed in ISO/IEC 15504-

2. The purpose of technical report ISO/IEC TR 24774:2010 [26] is to encourage uniformity in the description of processes. Uniform description of processes across process reference models allows the combination of processes from different reference models to ease the development of new models and facilitates comparison of models.

Once the PRM was developed the measurement framework and PAM requirements as detailed in ISO 15504-2 were added to the PRM to form the PAM.

5 Structure of Traceability PAM

The traceability assessment framework, illustrated in Figure 2, consists of 4 traceability processes which are Change Management (CM) traceability, Risk Management (RM) traceability, Software Development Lifecycle (SDLC) traceability, and Best Practice traceability.

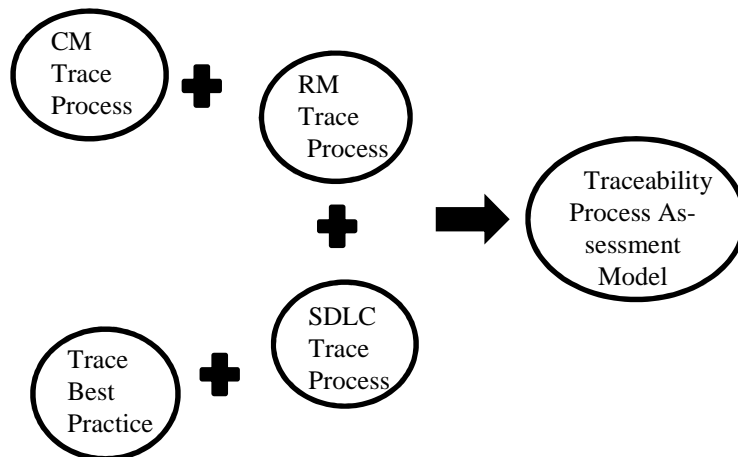


Fig. 5 Traceability Process Assessment Framework

Each of the processes contains: (i) Title; (ii) Purpose, which contains the unique functional objectives of the process when performed in a particular environment; (iii) Outcomes, which are a list of expected positive results of the process performance; (iv) Base practices, whose performance provides an indication of the extent of achievement of the process purpose and process outcomes; and (v) Work Products (WPs) are either used or produced (or both), when performing the process.

The CM traceability process: The purpose of this process is to ensure that traceability is adequately addressed throughout all stages of

the Change management/Problem resolution process by assessing the requirements for traceability as depicted in Figure 3.

The RM traceability process: The purpose of this process is to ensure that traceability is adequately addressed throughout all stages of the risk management process by assessing the requirements for traceability as depicted in Figure 2.

The SDLC traceability process: The purpose of the SDLC Traceability Process is to ensure that traceability is adequately addressed throughout all stages of the SDLC process by assessing the requirements for traceability as depicted in Figure 1

Traceability best practice process: The purpose of the Traceability Best Practices process is to ensure that traceability best practices are established when implementing traceability through the SDLC and the supporting processes of risk management and change management. This is achieved by assessing if a company policy and a standard operating procedure for traceability have been developed, the resources required for successful traceability implementation are made available, and the appropriate techniques for successful implementation are deployed.

6 Validation of PAM

The PAM has been validated initially by expert review and then by trial in two medical device organisations. This section provides an overview of the validation methodology and findings.

6.1 Validation by Expert Review

An initial validation of the traceability PAM has been conducted by four experts who were chosen based on the following criteria; their expertise in: a) ISO/IEC 15504; b) medical device standards; c) requirements traceability; and d) medical device software development. Details of the methodology used for the validation and the findings which resulted from the validation have previously been published [27]. A summary of the changes made to the PAM as a result of 25 comments made during the expert review now follows.

Change management process:

- Amended to include traceability of problem reports to change requests.
- Add note indicating that *bidirectional* traceability is good practice, even though it is not a requirement of IEC 62304

Traceability best practices process:

- Amended so that it will include reference to source documents. As these documents are academic papers the references will be included in an appendix
- Outcome 1 of the Traceability best practice process will be changed from “a company policy on traceability and procedures for its implementation are established” to “Establish a plan for establishing traceability in the organisation/project”.

Risk Management traceability process:

- Note added indicating that *bidirectional* traceability is good practice, even though it is not a requirement of IEC 62304 or ISO 14971
- Outcome 7 to have following note attached: ‘*IEC 62304 requires that risk control measure implemented in software be included in the software requirements specification*’.

SDLC traceability process

- A note is added to Outcome 7 stating; ‘*Where software requirements cannot be addressed in software design (e.g. non-functional requirements, attributes etc.), traceability between software requirements and source code should be directly established*’.
- Amend Outcome 5 of the SDLC traceability to state that this PAM only requires those system requirements that are implemented in software to be traced. Also add note to Outcome 5 stating; if the system is a software only system, then outcomes 1, 2, 3 and 4 can be removed and Outcome 5 can be replaced with “Traceability is provided between software requirements and their source”.

6.2 Validation by Industry Trial

Two organisations were selected for trial of the PAM based on their suitability and accessibility.

Organisation A are a small medical device software company based in Ireland with a total of ten employees which include one programmer,

one software tester and one quality assurance person. The company has been in existence since 2002. Their device has a software safety classification of B, meaning non-serious injury is possible.

Organisation B is a small Product Development & Design Engineering company focused on the Medical Device and Life Science market and is based in Ireland. The company, which was formed in 2007, employ 14 individuals with skills in mechanical, hardware and software engineering. The company are a third party supplier of software to medical device companies and have recently been accredited with IEC 62304 certification.

To perform the assessment an eight stage process [28] was adopted. It is determined that this process will allow for a comprehensive assessment of an organisations traceability compliance without being too onerous in terms of resources that the organisation need to provide.

The assessment results for both Organisations A and B are outlined in Figures 3. Figure 3 is a UML diagram indicating how each organization fulfils the medical device standards requirements for traceability and is as a result of using the PAM to assess traceability through risk management, change management and SDLC process. The dashed links in Figure 3 indicate the links that Organisation A are implementing while the dashed lines ending in a filled circle indicate the links that Organisation A are not implementing. Similarly the solid links indicate the links that Organisation B are implementing while the solid lines ending in a filled circle indicate the links that Organisation B are not implementing.

The results of the assessment indicate that while both organisations are fulfilling many of the medical device requirements for traceability they are both weak in the following areas:

- Tracing from hazardous situation to software item to software cause. This is a particular requirement from IEC 62304.
- Tracing between software requirements and their initial source
- Tracing down to the code level

In following discussion with the organisations, they maintained that the reason for the missing links was that they were simply unaware that they were required.

In addition to these finding the assessment results also report that both organisations are only partially applying the best practices for implementing traceability or else not at all as depicted in Table 1.

The columns F,L,P and N indicate full, largely, partially or No implementation of traceability best practices. So for example neither organisation A or B had a company policy on traceability.

Table 1 Best Practice Implementation Findings

Traceability Best Practices	F	L	P	N
Do you have a documented company policy on traceability? If so, where is it documented?				A B
Have you established a Traceability improvement communication method ? If so, what is it?				A B
Do you have a Traceability Information Model ?				A B
Do you have a documented Standard Operating Procedure for Traceability				A B
Are resources required for traceability made available e.g. Training, Education, Tools, Time?			A B	
Are appropriate techniques deployed for effective implementation of traceability e.g. inventory of current tools and practices, review of traceability information.			A B	

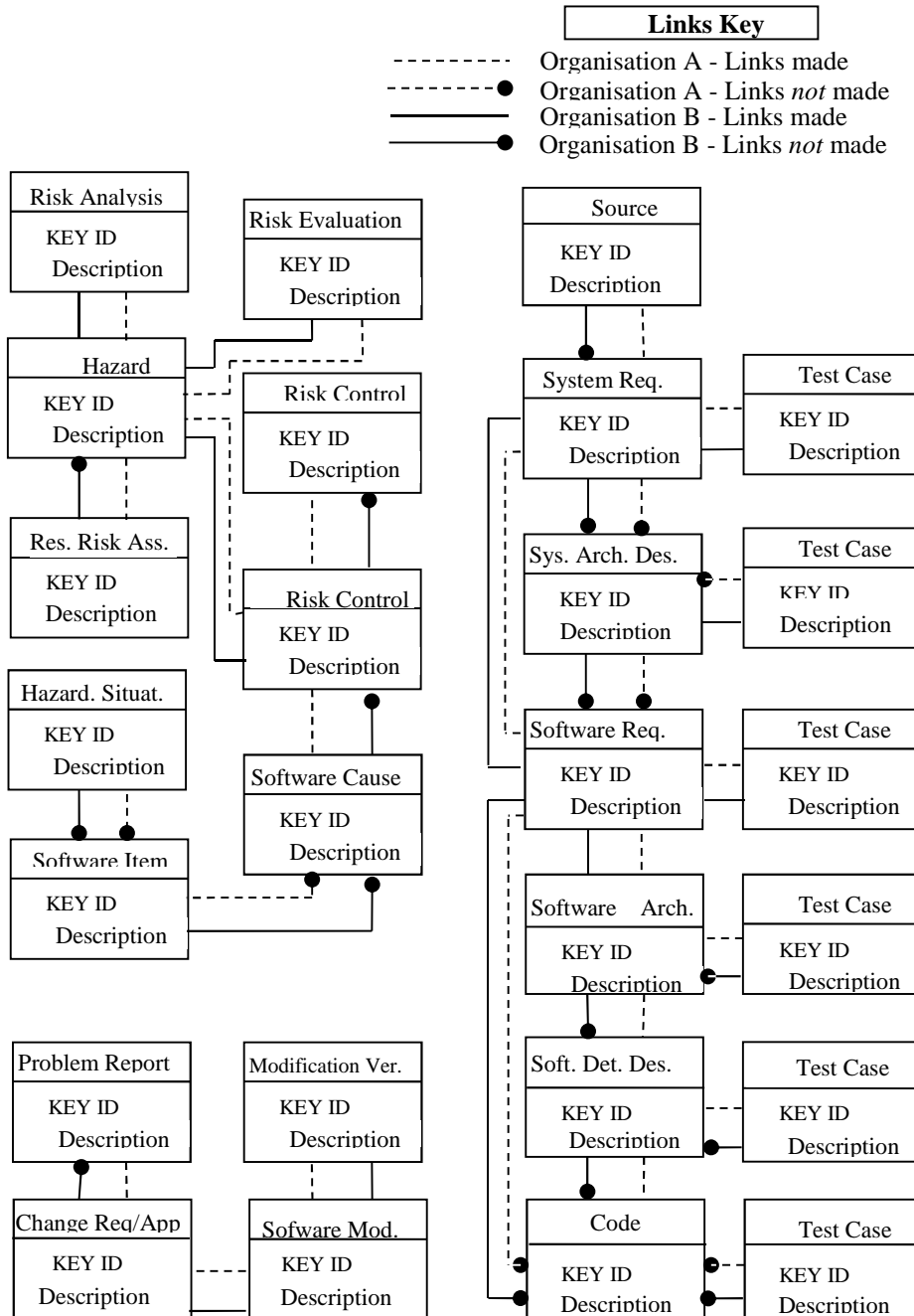


Fig.6 Organisation A and B Implementation of Traceability Requirements

6.3 Discussion

One goal of the assessment was that it would not be too onerous in terms of resources that the organisation needs to provide. For both organisations total contact time between assessor and organisation was approximately 4 hours. This equated to 1.0 hours for steps 1 and 2 of the assessment method (Develop appraisal schedule and conduct overview briefing), 1.5 hours for step 4 (conduct interviews), 1.5 hours for steps 6 and 7 (Deliver findings report and agree improvement plan). Step 8 of the assessment method (Re-assess the SPI Path and Produce a Final Report) is still to be completed as a 3 month implementation period was agreed. It is envisaged that step 8 will require no more than 1.5 hours of contact. Steps 3 and 5 of the assessment method (analyse software documentation and create findings report) were conducted off-site in both assessments and so required no input from either organisation. Due to the limited hours required, both organisations felt the assessment was not onerous and considered the assessment and its findings and recommendations to be very relevant and valuable.

Another goal of the assessment was that it would allow for a comprehensive assessment of an organisations compliance with the requirements for traceability and its implementation of traceability best practices, and provide a pathway for improvement. The assessment findings clearly uncovered a number of weaknesses within both organisations, with Organisation A missing 7 required links and Organisation B missing 15 links. The assessment thus demonstrated the level of non-compliance with the medical device standards requirements for traceability, within both organisations. Additionally the assessment demonstrated that, within both organisations, the level of traceability best practice implementation was extremely low. Although the PAM does not assess the reasons for non-implementation of traceability requirements or best practices, through informal discussion it has emerged that the main reason is that the organisations were simply unaware of the missing requirements and best practices. As a result of the assessment both organisations are now aware of best practices for implementing traceability. Furthermore the recommendations for improvement have not only provided both organisations with a pathway for improvement, but also with goals which they feel are reasonable, achievable and motivational.

While the PAM has been successful in achieving its goals we feel that the assessment of traceability could be further improved. Although the assessment was not overly onerous for the organisations, it did require a couple of visits to the organizations and did put some demand on their resources. Additionally the assessment provides a snapshot of traceability implementation at a single point in time and does not account for artefacts dynamically changing during production and maintenance. Changing artefacts necessitates the need for maintaining the traceability. To address these issues we describe our vision to automate the assessment and maintenance of traceability.

7 Vision to automate the assessment and maintenance of traceability

As discussed earlier, there is imminent need for the automation of traceability assessment and maintenance. Considering the clear definition cited in the introduction, traceability is the ability to establish links (or traces) between source artefacts and target artefacts [2]. According to the state of the art of web technology, we have today the means to identify and to establish links between immense numbers of artifacts which can even be seamlessly traced on the basis of these links.

Our vision is that the processes defined in the Traceability Process Assessment Model of this paper could be executed using a system accessing all of the necessary artifacts which would be accessible on the web (internet or intranet). By consequent, this system would ultimately have full traceability assessment and also resulting traceability maintenance capability.

Application Lifecycle Management (ALM) tool vendors are perfectly aware of this need, and some of the tools [29] contain features supporting a given level of automation. However, current ALM tools have following inherent weaknesses:

- Traceability is basically restricted to the closed ALM system. APIs are available for providing internal data, however, no standardized open form of exchange was made possible before the below discussed OSLC initiative.
- Useful traceability reports can be generated, but they are static while requirements and identified defects are very dynamically

changing artefacts, and may even originate from outside the ALM system.

- Assessors and users may be easily confused by the complexity of the set of widgets, such as buttons, text fields, tabs, and links which are provided to access and edit all properties of resources at any time.
- Assessors and users need to reach destinations such as web pages and views by clicking many links and tabs whose understanding is not essential for the assessment.

Open Services for Lifecycle Collaboration (OSLC) is the recently formed cross-industry initiative aiming to define standards for compatibility of software lifecycle tools. Its aim is to make it easy and practical to integrate software used for development, deployment, and monitoring applications. This aim seems to be too obvious and overly ambitious at the same time. However, despite its relatively short history starting in 2008, OSLC is the only potential approach to achieve these aims at a universal level, and is already widely supported by industry.

The unprecedented potential of the OSLC approach is based on its foundation on the architecture of the World Wide Web, which is unquestionably proven to be powerful and scalable, and on the generally accepted software engineering principle to always focus first on the simplest possible things that will work.

The elementary concepts and rules are defined in the OSLC Core Specification which sets out the common features that every OSLC Service is expected to support using the terminology and generally accepted approaches of the World Wide Web Consortium (W3C). One of the key approaches is Linked Data being the primary technology leading to the Semantic Web which is defined by W3C as providing a common framework that allows data to be shared and reused across application, enterprise, and community boundaries.

The OSLC Core Specification is actually the core on which all lifecycle element (domain) specifications must be built upon. Examples of already defined OSLC Specifications include:

- Architecture Management
- Asset Management
- Automation
- Change Management
- Quality Management

- Requirements Management

We have seen that the Change Management Specification is highly relevant to the traceability process assessment experiences discussed in this paper. As OSLC has become part of the OASIS Open Standards Network, the Change Management Specification is being further developed by the OASIS OSLC Lifecycle Integration for Change and Configuration Management (OSLC CCM) Technical Committee to be approved as a full OASIS standard. OASIS is an international nonprofit consortium with wide impact that brings companies, governments, academia, and individuals together to solve communications challenges.

The Core, briefly mentioned above, defines the resource types, properties and operations to be supported by all OSLC service providers including Change Management (OSLC CM) providers.

Examples of possible OSLC CM Resources include defect, enhancement, task, bug, activity, and any application lifecycle management or product lifecycle management artifacts. Resource types are defined by the properties that are allowed and required in the resource.

The properties defined in the OSLC Change Management Specification describe these resource types and the relationships between them and all other resources. The relationship properties describe in most general terms for example that

- the change request affects a plan item
- the change request is affected by a reported defect
- the change request tracks the associated Requirement
- the change request implements associated Requirement
- the change request affects a Requirement

OSLC is currently at the technology trigger stage along its hype cycle [30]. It is already clear however that it is the approach which has the potential to have a determining impact on the future of the satisfaction of the traceability requirements.

Full traceability of a requirement throughout the development chain and even the entire supply chain was also a major focus point of the European CESAR project (Cost-Efficient Methods and Processes for Safety Relevant Embedded Systems) which adopted interoperability technologies proposed by the OSLC initiative.

Another important European project, completed in 2013 and exploiting OSLC, is iFEST (industrial Framework for Embedded Systems Tools).

CRYSTAL (CRITICAL sYSTEM engineering AccELeration) is a currently running ARTEMIS Innovation Pilot Project (AIPP) whose Interoperability Specification (IOS) is also based on OSLC.

8 Conclusion

In order to improve their implementation of traceability an organisation needs to be able to identify gaps in their existing traceability process. Hence this paper presents the development and validation of a traceability PAM. This PAM is based on the ISO 15504 structure and used the TIPA transformation process for development. Validation of the model, first through expert review and then by trial in two medical device organisations, verifies that the model can identify both strengths and weaknesses in an organisations existing traceability processes. It can also be used to assess the state of a suppliers traceability process.

If our envisioned system, based on the processes defined in the Traceability Process Assessment Model of this paper, could seamlessly access the resources and their relationships using OSLC across all tools applied in the entire software development lifecycle (SDLC), then traceability process assessment and maintenance could be fully automated.

Acknowledgement. This research is supported by the Science Foundation Ireland Principal Investigator Programme, grant number 08/IN.1/I2030 and by Lero - the Irish Software Research Centre (<http://www.lero.ie>) grant 10/CE/I1855.

References

1. ISO/IEC: 15504-5 : An exemplar Process Assessment Model. ISO, Switzerland (2012)
2. SEI: CMMI® for Development, Version 1.3. Improving processes for developing better products and services. SEI, Pittsburgh (2010)
3. Producibility, C.f.A.S.-I.S.: Critical Code: Software Producibility for Defense. (2010)
4. Gotel, O.C.Z., Finkelstein, C.W.: An analysis of the requirements traceability problem. Requirements Engineering, 1994., Proceedings of the First International Conference on Requirements Engineering, pp. 94-101 (1994)
5. Cleland-Huang, J., Gotel, O., Zisman, A.: Preface. In: Cleland-Huang, J., Gotel, O.,

Zisman, A. (eds.) Software and Systems Traceability. Springer, London Dordrecht Heidelberg New York (2012)

6. FDA: General Principles of Software Validation; Final Guidance for Industry and FDA Staff. CDRH, Rockville (2002)
7. Dekhtar, A., Hayes, J.H.: Studying the role of Humans in the Traceability Loop. In: Cleland-Huang, J., Gotel, O., Zisman, A. (eds.) Software and Systems Traceability. Springer, London Dordrecht Heidelberg New York (2012)
8. Mader, P., Gotel, O., Philippow, I.: Motivation Matters in the Traceability Trenches. Proceedings of the 2009 17th IEEE International Requirements Engineering Conference, RE, pp. 143-148. IEEE Computer Society (2009)
9. Klimpke, L., Hildenbrand, T.: Towards End-to- End Traceability: Insights and Implications from Five Case Studies. 2009 Fourth International Conference on Software Engineering Advances, pp. 465-470. IEEE Computer Society. pp. 465-470. (2009)
10. Hayes, J.H., Dekhtyar, A.: Towards Traceable Test-Driven Development. TEFSE'09, Vancouver, Canada (2009)
11. Lucia, A.D., Marcus, A., Oliveto, R., Poshyvanyk, D.: Information Retrieval Methods for Automated Traceability Recovery. In: Cleland-Huang, J., Gotel, O., Zisman, A. (eds.) Software and Systems Traceability, pp. 88 - 111. Springer (2012)
12. McCaffery, F., Casey, V.: Med-Trace: Traceability Assessment Method for Medical Device Software Development. EuroSPI pp. 1.1 - 1.8, Denmark (2011)
13. AutomotiveSIG: Automotive SPICE® Process Assessment Model. (2010)
14. ECCS: Space Product Assurance- Software process assessment and improvement – Part 2: Assessor instrument. ESA Requirements and Standards Division, Netherlands (2010)
15. Ambler, S.: Tracing Your Design. Dr.Dobb's Journal: The World of Software Development (1999)
16. Ambler,
<http://www.agilemodeling.com/essays/agileRequirementsBestPractices.htm#QuestionTraceability>
17. ANSI/AAMI/IEC: 62304:2006 Medical device software—Software life cycle processes. AAMI, Arlington, VA (2006)
18. FDA: Guidance for the Content of Premarket Submissions for Software Contained in Medical Devices. CDRH, Rockville (2005)
19. FDA: Off-The-Shelf Software Use in Medical Devices; Guidance for Industry, FDA Reviewers and Compliance. CDRH, Rockville (1999)
20. ISO: ISO 13485- Medical devices — Quality management systems — Requirements for regulatory purposes. ISO, Switzerland (2003)
21. ISO: ISO 14971:2007 Medical devices — Application of risk management to medical devices. ISO, Switzerland (2007)
22. Regan, G., Mc Caffery, F., Mc Daid, K., Flood, D.: Implementation of traceability

best practices within the medical device domain. EuroSPI 2013. DELTA, Dundalk, Ireland (2013)

23. ISO/IEC: 15504-2: Process assessment — Performing an assessment. ISO, Switzerland (2003)

24. ISO/IEC: 12207: Systems and software engineering — Software life cycle processes. ISO, Geneva, Switzerland (2008)

25. Barafort, B., Renault, A., Picard, M., Cortina, S.: A transformation process for building PRMs and PAMs based on a collection of requirements – Example with ISO/IEC 20000. SPICE, Nuremberg, Germany (2008)

26. ISO/IEC: ~~24774~~ Systems and software engineering — Life cycle management — Guidelines for process description. (2010)

27. Regan, G., Caffery, F.M., Daid, K.M., Flood, D.: The Development and Validation of a Traceability Assessment Model. 14th International Conference, SPICE 2014, vol. 447, pp. 72-83. Springer, Vilnius, Lithuania (2014)

28. McCaffery, F., Coleman, G.: The Development of a Low-Overhead Assessment Method for Irish Software SMEs. Journal of Information Technology Management XVIII, (2007)

29. Gartner, <http://www.techostan.com/docs/quadrant.pdf>

30. Biro, M.: The Software Process Improvement Hype Cycle. Invited contribution to the Monograph: Experiences and Advances in Software Quality (Guest editors: D.Dalcher, L. Fernandez-Sanz) CEPIS UPGRADE Vol. X (5) pp. 14-20 (2009)
<http://www.cepis.org/files/cepisupgrade/issue%20V-2009-fullissue.pdf>