The Development and Validation of a Traceability Assessment Model

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

Dundalk Institute of Technology, Dundalk, Ireland

{gilbert.regan, fergal,mccaffery, kevin.mcdaid, [derek.flood}@dkit.ie](mailto:derek.flood%7d@dkit.ie)

**Abstract.** Regulation normally requires critical systems to be certified before entering service. This involves submission of a safety case - a reasoned argument and supporting evidence that stringent requirements have been met and that the system is acceptably safe. A good safety case encompasses an effective risk mitigation process which is highly dependent on requirements traceability. However despite its many benefits and regulatory requirements, most existing software systems lack explicit traceability links between artefacts. Reasons for the lack of traceability include cost, complexity and lack of guidance on how to implement traceability.

To assist medical device organisations in addressing the lack of guidance on how to implement effective traceability, this paper aims to present the development and validation of a traceability process assessment model and the actions to be taken as a result of the validation. The process assessment model will allow organisations to identify strengths and weaknesses in their existing traceability process and pinpoint areas for improvement.

**Keywords : traceability, requirements traceability, process assessment, safety critical software**

1. Introduction

Manufacturers of safety critical software must ensure their software meets stringent guidelines and is safe to use as intended. Guidelines such as DO-178B (aerospace) [[1](#_ENREF_1)], EN50128 (railway) [[2](#_ENREF_2)] and IEC 62304 (medical devices) [[3](#_ENREF_3)] represents industry consensus opinion on the best way to ensure safe software, e.g. IEC 62304 provides a framework of life cycle processes with activities and tasks necessary for the safe design and maintenance of medical device software. Traceability is an important tool in ensuring that a rigorous software development process has been established and that software is safe, hence these guidelines provide specific guidance for the creation and maintenance of traceability e.g. IEC 62304 states that the manufacturer shall create an audit trail whereby each: a) Change request, b) relevant Problem report, and c) approval of the Change request can be traced.

However despite its many benefits and regulatory requirements, most existing software systems lack explicit traceability links between artefacts [[4](#_ENREF_4)]. Numerous reasons have been identified for reluctance in implementing traceability including cost and complexity. Other reasons include the task of building a requirements trace matrix (RTM) is time consuming, arduous and error prone [[5](#_ENREF_5)], 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 [[6](#_ENREF_6)], the need for documentation can cause resentment among developers who may fear that traces could be used to monitor their work [[7](#_ENREF_7)], difficulties with trace tools including selecting between available tools, and difficulties configuring a general purpose tool or developing a custom tool [[8](#_ENREF_8)]. Finally almost no guidance 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 [[9](#_ENREF_9), [10](#_ENREF_10)].

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). To be effective, organisations need to know how well their current traceability process helps them achieve their goals. Additionally an assessment of a process will lead to an increased understanding of the actual performance and management of activities, and the potential for improvement.

1. 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 [[9](#_ENREF_9)]. Med-trace is a lightweight traceability assessment method, completed in 8 stages, whose goal is to assist medical device organisations 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/IEC 15504 SPICE [[11](#_ENREF_11)], Automotive SPICE [[12](#_ENREF_12)], SPICE 4 SPACE [[13](#_ENREF_13)], and the Capability Maturity Model CMMI [[14](#_ENREF_14)] among others. 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.

1. Methodology

It was decided to base the traceability assessment model on the ISO/IEC 15504 (SPICE) assessment model for two reasons;

1. ISO/IEC 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).
2. ISO/IEC 15504 is derived from ISO/IEC 12207 [[15](#_ENREF_15)] and since 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 IEC 62304:2006 and ISO/IEC 15504.

The PRM was developed using the requirements from traceability (taken from the medical device standards and guidelines), and ISO/IEC 15504-2 section 6.2 which sets out the requirements for a Process Reference Model. Additionally it was felt necessary to assess the best practices for implementing traceability. These best practices (23 in total) were the result of an extensive literature review [[16](#_ENREF_16)].

While ISO/IEC 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. The TIPA transformation process complies with the requirements for PRMs and PAMs as expressed in ISO/IEC 15504-2. The TIPA transformation process contains the following steps [[17](#_ENREF_17)];

1. Identify elementary requirements in a collection of requirements
2. Organise and structure the requirements
3. Identify common purposes upon those requirements and organize them towards domain goals
4. Identify and factorize outcomes from the common purposes and attach them to the related goals
5. Group activities together under a practice and attach it to related outcomes
6. Allocate each practice to a specific capability level
7. Phrase outcomes and process purpose
8. Phrase the Base Practices attached to Outcomes
9. Determine Work Products among the inputs and outputs of the practices
10. Traceability PAM

The traceability PAM, illustrated in Figure 1, consists of 4 traceability processes which are Change Management (CM) traceability, Risk Management (RM) traceability, Software Development Lifecycle traceability, and Best Practice traceability. 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.

**Fig.1.** Traceability PAM

**The CM traceability process**: Five requirements for traceability, extracted from IEC 62304 and depicted in Figure 2, have been grouped to form a change management traceability process. The double head arrows represent bi-directional traceability, even though ‘bidirectional’ is not a requirement of IEC 62304 but is included because it is considered good engineering practice.

Chang Request

Change Request Approval

Regression Analysis

Software Modification

Modification verification

Problem Report/Source

Link 1

Link 2

Link 3

Link 4

Link 5

**Fig. 2.** Change Management Traceability Requirements

From these above requirements a common purpose (to ensure that traceability is adequately addressed throughout all stages of the Change management/Problem resolution process) was developed. A list of expected results (Outcomes) generated from performing the process was established, and five base practices which provide a definition of the tasks and activities needed to accomplish the process purpose and fulfil the process outcomes were identified as follows;

Base Practice 1: Establish bidirectional traceability between each change request and relevant problem report (Link1).

Base Practice 2: Establish bidirectional traceability from each change request to approval of the change request (Link 2).

Base Practice 3: Establish bidirectional traceability from each approval of change request to software modifications (Link 3).

Base Practice 4: Establish bidirectional traceability from each software modification to verification of the modification (Link 4).

Base Practice 5: Establish bidirectional traceability from each software modification to regression analysis (Link 5).

Additionally the PAM contains a set of assessment questions whose objective is to determine if the base practices are being implemented and how successful the organisation is at achieving the process purpose.

# 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 following application of bi-directional traceability: between analysis of risk to the identification of hazards; between hazardous situation and software item; between software item and specific software cause; between each hazard to estimation of risk of each hazard; between each risk estimation to evaluation of acceptability of the risk; between hazards and identification and implementation of risk control measures; between implementation and verification of risk control measures; and between residual risk to assessment of acceptability of those risks.

**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 following application of bi-directional traceability: between software requirements and system requirements; between software requirement and software architectural and software detailed design; between software detailed design and source code; between software requirements and source code; and between each phase of the SDLC and test for that phase.

**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.

1. Validation of the PAM

An initial validation of the traceability PAM has been conducted by expert review. Experts were chosen based on the following criteria; their expertise in a) ISO/IEC 15504, b) medical device standards and c) requirements traceability and d) medical device software development.

**Expert 1** is a provisional ISO/IEC 15504 assessor and is a member of the ISO/IEC JTC 1 SC7 WG10 [standardisation](http://en.wikipedia.org/wiki/Standardization) subcommittee working on the ISO/IEC 15504 standard.

**Expert 2** has thirteen years industrial experience in software development and is a member of the ISO/IEC JTC 1 SC7 WG10 [standardisation](http://en.wikipedia.org/wiki/Standardization) subcommittee working on the ISO/IEC 15504 standard.

**Expert 3** has worked in the field of systems engineering for forty five years and worked as senior staff engineer on medical device software for a large multinational. He is an INCOSE certified ESEP and an ACM distinguished engineer.

**Expert 4** has forty five years in software engineering. He has eighteen years in medical device software including developing processes that included traceability and has seventeen years in international standards activities for software engineering, medical device software and medical IT networks.

Each reviewer was asked to fill in a questionnaire which focused on the PAMs ‘fit for purpose’. A number of minor comments (mainly about terminology) and major comments were returned. The major comments are listed below.

a) Why would anyone want to assess traceability in isolation to all the other pro cess activities?

b) Change management and Problem resolution might be considered by some to be separate processes.

c) Outcome 2 (Traceability is provided from each change request or problem report to analysis/evaluation of the change request/ problem report) and Outcome 4 (Traceability is provided from each denial of change request/problem report to reason for denial) of Change Management traceability process are not outcomes for traceability

d) Outcome 2 of Risk Management traceability process (Traceability is provided from each hazard to estimation of risk of each hazard) does not look like an outcome for traceability but for risk estimation.

e) Outcome 3 of Risk Management (Traceability is provided from each risk estimation to evaluation of the acceptability of the risk) does not look like a traceability related outcome. This should belong to risk acceptance criteria instead.

f) Outcome 6 of Risk Management traceability process (Traceability is provided from any residual risk(s) to the assessment of the acceptability of those risks). I don’t see this as a traceability related requirement. Each residual risk has to have the justification attached to it, which comes from the risk evaluation.

g) In the SDLC traceability process, some of the outcomes are duplicating the contents of 80002-3 PRM.

h) Outcome 2 (Bi-directional traceability is established between system requirements and software requirements, including system architectural design) of the SDLC process is a requirement of architectural design and of software requirements analysis process. Not sure it should be duplicated from there.

i) Outcomes 4, 5 and 6 of the SDLC process (Bi-directional traceability is established between software requirements, software architectural design, software detailed design and source code, and between software requirements and source code, and between each phase of the SDLC and Test Specification for that phase) could be deleted and replace by 1 outcome such as “Ensure consistency of software requirements throughout software development lifecycle”.

j) Traceability best practices process: Not sure how you can call a process a best practice? Is it not more like Planning? I would integrate both planning the traceability and implementing the plan within the other three processes.

k) Outcome 1 of the Traceability best practice process should be “Establish a plan for establishing traceability in the organisation/project” and not “a company policy on traceability and procedures for its implementation are established”.

l) Some of these process outcomes in the Traceability best practices process are already required in 62304;2006 section 5.1.1

m) In Traceability best practices process: Why does this process table not have a link back to source documents (as in the other processes)

n) In Traceability best practices process, Improvement would appear to be outside the scope of Outcome 1(a company policy on traceability and procedures for its implementation are established). Therefore BP2 (Establish traceability improvement communication method) is not clearly defined.

o) Most products are upgrades or enhancements to previous products. As such, the issue of traceability, releases and upgrades become very important. It seems to me that your best practices consider only the greenfield case, which is the exception rather than the rule. How does one trace when building an upgrade to an existing product? Start from scratch? Just do the delta? Etc.

p) Often, a product is to be released in multiple countries, with product variations for different regions. I don’t see any discussion of best practices under those circumstances. In a broader context, your assessments seem to ignore the existence of product lines

q) My personal opinion is that the entire hazard mitigation process is flawed (the standard approach, not just yours). If you look closely at your risk mitigation process you will see that it is a “stovepipe”, e.g. a look at a single requirement and risk pair, or related requirements associated with a single risk. The world is moving away from this view. Very often risks are associated with combinations of hazards, or a chain of flaws rather than a single (requirement, hazard, mitigation) tuple. Furthermore, in systems engineering, we look at the entire environment, rather than a narrow stovepipe view of the tuple (e.g. fukishima).

r) Who does something is almost as important as what is being done e.g. if the QA checks on tracing are done by someone reporting to the project manager, I would invariably expect deterioration in the quality of the work done.

s) Your process best practices suffer from the same inadequacy as just about all such processes, a lack of detail and implementation guidelines. It is one thing to say “Ensure consistency between system architectural design and software requirements” and quite another to do it. What if there are impedance mismatches between the architecture and software tooling?

t) The Change management process does not identify traceability of problem reports to change requests. The change management traceability process seems to consider problem reports and change requests as the same thing. This is not correct. I don’t believe that 62304 requires that the origin of a change request be specified unless the change request is the result of a problem report.

u) IEC 62304 also requires that risk control measures implemented in software be included in the software requirements. So traceability from risk control measures in software to software requirements should be included.

v) It is not clear why the 14971 traceability needs to be bidirectional.

w) For the SDLC traceability process: In outcome 3, all system requirements may not be implemented in software, so there can only be traceability between the software requirements and the system requirement, not traceability between the system requirements (or at least not all of them) and the software requirements. 62304 does not require bidirectional traceability, only traceability from software to system.

x) For the SDLC traceability process: Outcomes 4 and 5 are not required by 62304. I think outcome 4 should be stated more clearly as traceability between requirements and architecture, traceability architecture and detail design, and traceability between detailed design and code. The way it is currently stated makes it sound like everything must trace to everything.

y) For the SDLC traceability process: Outcome 6 indicates that there should be a test specification for each phase of the SDLC. Phase is not a term used in 62304, but assuming that phase = 62304 software development activity then there would not be testing at each phase since tests require code to have been developed. 62304 does require that verification tasks for each development activity be planned and that acceptance criteria be identified, but it does not require a verification specification for each development activity.

To assist with their analysis, the comments were then categorised as shown in Table 2. The ‘C:\Users\regang\AppData\Local\Microsoft\Windows\Temporary Internet Files\Content.IE5\ZA9O9D22\MC900434713[1].wmf’ indicates comments on which the authors agree to amend the model. The ‘x’ indicates comments that will not cause the model to be amended. Categories are:

* Structure - Comments related to the structure of the PAM
* Terminology – Comments related to terminology used
* Outcomes - Comments related to outcomes in the PAM
* Duplication - Comments relating to duplication of traceability requirements from standards
* Scope - Comments relating to scope of PAM
* General – Comments that are general in nature

**Table 1.** Categorisation of reviewer comments

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Categ-ory** | **Review Comments** | | | | | | | | | | | | | | | | | | | | | | | | |
|  | a | b | c | d | e | f | g | h | i | j | k | l | m | n | o | p | q | r | s | t | u | v | w | x | y |
| Structure |  | C:\Users\regang\AppData\Local\Microsoft\Windows\Temporary Internet Files\Content.IE5\ZA9O9D22\MC900434713[1].wmf |  |  |  |  |  |  |  | C:\Users\regang\AppData\Local\Microsoft\Windows\Temporary Internet Files\Content.IE5\ZA9O9D22\MC900434713[1].wmf |  |  | C:\Users\regang\AppData\Local\Microsoft\Windows\Temporary Internet Files\Content.IE5\ZA9O9D22\MC900434713[1].wmf |  |  |  |  |  |  | C:\Users\regang\AppData\Local\Microsoft\Windows\Temporary Internet Files\Content.IE5\ZA9O9D22\MC900434713[1].wmf |  |  |  |  |  |
| Termin-ology |  |  |  |  |  |  |  |  |  |  | C:\Users\regang\AppData\Local\Microsoft\Windows\Temporary Internet Files\Content.IE5\ZA9O9D22\MC900434713[1].wmf |  |  | x |  |  |  |  |  |  |  | x |  |  |  |
| Outcome |  |  | x | x | x | x |  |  | x |  |  |  |  |  |  |  |  |  |  |  | C:\Users\regang\AppData\Local\Microsoft\Windows\Temporary Internet Files\Content.IE5\ZA9O9D22\MC900434713[1].wmf |  | C:\Users\regang\AppData\Local\Microsoft\Windows\Temporary Internet Files\Content.IE5\ZA9O9D22\MC900434713[1].wmf | C:\Users\regang\AppData\Local\Microsoft\Windows\Temporary Internet Files\Content.IE5\ZA9O9D22\MC900434713[1].wmf | x |
| Duplic-ation |  |  |  |  |  |  | x | x |  |  |  | x |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Scope |  |  |  |  |  |  |  |  |  |  |  |  |  |  | C:\Users\regang\AppData\Local\Microsoft\Windows\Temporary Internet Files\Content.IE5\ZA9O9D22\MC900434713[1].wmf | C:\Users\regang\AppData\Local\Microsoft\Windows\Temporary Internet Files\Content.IE5\ZA9O9D22\MC900434713[1].wmf | x |  |  |  |  |  |  |  |  |
| General | x |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | x | x |  |  |  |  |  |  |

1. Discussion

The review comments are discussed under their categorisations.

**Structure:** Two expertsmade similar comments (b and t) that change management and problem resolution are different processes and that the change management process does not identify traceability of problem reports to change requests. On reflection, this is considered to be correct, therefore the change management process is to be amended to include traceability of problem reports to change requests.

Comment ‘j’ considers the title of the Traceability best practice process to be incorrect and that it would be better to integrate planning the traceability and implementing the plan within the other three processes. As the purpose of this process is to assess if known best practices are established and applied when implementing traceability, the process name will be amended to Traceability best practice management. However the integration of the best practices with the other three processes is not considered, as the best practices are generic across the other three processes and would mean a lot of repetition.

The traceability best practices process does not have a link back to source documents, as with the other processes (comment ‘m’). The source documents are academic publications and, on reflection, it is agreed that amending the process to include references to these documents in an appendix would improve the process.

**Terminology:** Comment k isconsidered to be an improvement and soOutcome 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”.

Comment ‘n’ asserts that ‘improvement’ (in base practice 2) would appear to be outside the scope of Outcome 1(a company policy on traceability and procedures for its implementation are established) in Traceability best practice process. However, it is thought that communication of improvement should be part of company policy and so will remain a base practice for outcome 1.

Expert 4 remarks that ISO 14971 does not require bidirectional traceability through risk management (comment ‘v’). While this is correct, ‘bidirectional’ traceability is considered good practice and so a note will be added to the risk management traceability process indicating that it is good practice, even though it is not a requirement of IEC 62304 or ISO 14971.

**Outcome:** Comment ‘c‘ remarks that Outcomes 2 and 4 (Traceability is provided from each change request or problem report to analysis/evaluation of the change request/ problem report, and Traceability is provided from each denial of change request/problem report to reason for denial) of change management traceability process are not outcomes for traceability. As IEC 62304 states thatthe manufacturer shall create an audit trail whereby each Change request, relevant Problem report and approval of the Change request can be traced [Class A, B, C], it is the interpretation of this study that outcomes 2 and 4 are outcomes for traceability and considered good practice.

Comments ‘d‘ ‘e‘ and ‘f‘ consider outcomes 2,3 and 6 (Traceability is provided from each hazard to estimation of risk of each hazard, Traceability is provided from each risk estimation to an evaluation of the acceptability of the risk, and Traceability is provided from any residual risk(s) to the assessment of the acceptability of those risks) of the risk management process to be outcomes for risk estimation, risk acceptance criteria and risk evaluation, and not outcomes for traceability. This study considers this not to be the case as ISO 14971 states that the risk management file shall provide traceability for each identified hazard to the risk analysis(which includes estimation of risk), risk evaluation (evaluation of the acceptability of the risk), the assessment of the acceptability of any residual risk(s).

Comment ‘i’ states that outcomes 4,5 and 6 (Bi-directional traceability is established between software requirements, software architectural design, software detailed design and source code, and between software requirements and source code, and between each phase of the SDLC and Test Specification for that phase) of the SDLC traceability process could be replaced by one outcome i.e. “Ensure consistency of software requirements throughout software development lifecycle”. This point was considered during development of the SDLC traceability process, but for reasons of clarity it was felt better to have three separate outcomes. Comment ‘x’ asserts that outcome 4 and 5 are not required by IEC 62304, which is true, however they are required by the FDA’s GPSV. As outcome 5 is only required where requirements cannot be addressed in the software design (e.g. nonfunctional requirements), outcome 5 will be removed and a note added to the process to address this issue.

In addition to the above, for outcome 3 of the SDLC traceability process, all system requirements may not be implemented in the software (comment ‘w‘). This point is accepted, therefore the process will be amended to include only system requirements that are included in the software.

Expert 4 maintains that IEC 62304 requires that risk control measures implemented in software be included in the software requirements (comment ‘u‘). This study considers this comment to be correct and as a result traceability from risk control measures in software to software requirements should be included. The risk management traceability process will be amended to reflect this.

Finally, comment ‘y’ states that IEC 62304 does not require a verification specification for each development activity but does require that verification tasks for each development activity be planned and that acceptance criteria be identified. However GPSV requires that traceability is established from unit tests to detailed design and to source code, and from integration tests to high level design.

**Duplication:** This category consists of comments on outcomes that relate to the outcomes being duplicated from other models and processes. For example comment ‘g‘ states that some of the outcomes in the SDLC traceability process are duplicating the contents of 80002-3 PRM (yet to be officially published) or is a requirement of architectural design and of software requirements analysis process. Part of the rationale behind the traceability PAM is to include all requirements for traceability in one model. IEC 80002-3 PRM, in addition to any individual medical device standard or guideline, does not include all the requirements for traceability through the SDLC and supporting processes of risk management and change management.

Comment ‘l’ states that some of the outcomes in the traceability best practice process are already required in IEC 62304:2006 Section 5.5.1. However it is determined that this does not seem to be the case as section 5.1.1 contains detail about what the software development plan shall address and does not contain any of the outcomes in the traceability best practice process.

**Scope:** This category contains comments which relate to the scope of the PAM. Comments ‘o‘ and ‘p‘ make the point that the traceability best practice model does not consider best practice for traceability across product lines, that the model just considers the greenfield case. On reflection it is agreed that product lines do need to be considered in the traceability best practice process and the process needs to be amended to reflect this.

Expert 4 feels that the whole hazard mitigation process is flawed (comment q). While there may be some truth in this, ISO 14971 is the current standard for the application of risk management to medical devices and therefore is the standard that will be adhered to.

**General:** This category contains general comments on the traceability PAM. Comments ‘r‘ and ‘s‘ relate to the fact that the PAM does not contain any details as to ‘who does what ‘ and ‘how is it done‘. While these comments are true, it is not the place of a PAM to detail the ‘who’ and ‘how‘, but to assess ‘what’ traceability is implemented. The development of a traceability implementation roadmap will answer the ‘how’ question and is considered future work for this study.

The final comment (comment ‘a‘) questions the need for a traceability assessment model. It is the view of this study that traceability of hazards to implementation and verification of mitigations is important in producing safety critical software, and that tracing user requirements to software requirements and to test of those software requirements is important in product validation. Additionally numerous publications have highlighted the fact that most software systems don’t employ explicit traceability between artefacts (despite the fact the existing assessment models such as ISO/IEC 15504 and CMMI assess for traceability) and that one reason for this is a lack of guidance on how to implement traceability. The traceability PAM will assert focus on traceability to identify strengths and weaknesses in existing traceability processes and set the foundation for improvement. Additionally the traceability assessment model, developed in this study will assess for traceability implementation best practices.

1. Conclusion

To assist medical device organisations improve their traceability, a traceability assessment model has been developed. This model, which consists of four processes, is based on the ISO/IEC 15504 structure and used the TIPA transformation process for development. By assessing for all traceability requirements from the medical device standards and guidelines and by assessing for traceability implementation best practices, this traceability assessment model will assist medical device organisations understand their actual traceability performance and management of activities, and the potential for improvement.

Four experts (two ISO/IEC 15504 experts, one medical device standards expert and one medical device industry expert) reviewed the model and returned twenty five review comments. Nine of the twenty five review comments were accepted for change as it was felt they would improve the model. Most of the changes are to the structure and outcome categories, with two coming from the terminology category. Based on the review feedback and resulting amendments, the model is now ready for pilot assessment within two medical device organisations.

1. References

[1] RTCA, "DO-178B Software Considerations in Airborne Systems and Equipment Certification," ed. Washington, DC, 1992.

[2] CENELEC, "Railway applications – Communications, signalling and processing systems – Software for railway control and protection systems”,," ed, 2001.

[3] ANSI/AAMI/IEC, "62304:2006 Medical device software—Software life cycle processes," ed. Arlington, VA: AAMI, 2006.

[4] A. D. Lucia*, et al.*, "Information Retrieval Methods for Automated Traceability Recovery," in *Software and Systems Traceability*, J. Cleland-Huang*, et al.*, Eds., ed: Springer, 2012, pp. 88 - 111.

[5] J. Cleland-Huang, "Just Enough Requirements Traceability," presented at the Proceedings of the 30th Annual International Computer Software and Applications Conference - Volume 01, 2006.

[6] A. Kannenberg and D. H. Saiedian, "Why Software Requirements Traceability Remains a Challenge," *CrossTalk The Journal of Defense Software Engineering,* p. 5, 2009.

[7] M. Jarke, "Requirements tracing," *Commun. ACM,* vol. 41, pp. 32-36, 1998.

[8] G. Regan*, et al.*, "The Barriers to Traceability and their Potential Solutions: Towards a Reference Framework," presented at the 38th Euromicro Conference on Software Engineering and Advanced Applications, Cesme, Turkey, 2012.

[9] F. McCaffery and V. Casey, "Med-Trace: Traceability Assessment Method for Medical Device Software Development," presented at the EuroSPI Denmark, 2011.

[10] P. Mader*, et al.*, "Motivation Matters in the Traceability Trenches," presented at the Proceedings of the 2009 17th IEEE International Requirements Engineering Conference, RE, 2009.

[11] ISO/IEC, "15504-5 : An exemplar Process Assessment Model," ed. Switzerland: ISO, 2006.

[12] A. SIG, "Automotive SPICE® Process Assessment Model," ed, 2010.

[13] ECCS, "Space Product Assurance- Software process assessment and improvement – Part 2: Assessor instrument," ed. Netherlands: ESA Requirements and Standards Division, 2010.

[14] S. E. Institute, "CMMI® for Development, Version 1.3," in *Improving processes for developing better products and services*, ed, 2010.

[15] ISO/IEC, "12207: Systems and software engineering — Software life cycle processes," ed. Geneva, Switzerland: ISO, 2008.

[16] G. Regan*, et al.*, "Implementation of traceability best practices within the medical device domain," presented at the EuroSPI 2013, Dundalk, Ireland, 2013.

[17] B. Barafort*, et al.*, "A transformation process for building PRMs and PAMs based on a collection of requirements – Example with ISO/IEC 20000," presented at the SPICE, Nuremberg, Germany, 2008.