

Supporting the Systematic Assessment of Requirements Traceability - A Quality Model

Patrick Rempel¹ and Patrick Mäder²

Abstract: Traceability is an important quality of software requirements and allows to describe and follow their life throughout a development project. The importance of traceable requirements is reflected by the fact that requirements standards, safety regulations, and maturity models explicitly demand for it. In practice, traceability is created and maintained by humans, which make mistakes. In result, existing traces are potentially of dubious quality but serve as the foundation for high impact development decisions. We found in previous studies that practitioners miss clear guidance on how to systematically assess the quality of existing traces. In this paper, we review the elements involved in establishing traceability in a development project and derive a quality model that specifies per element the acceptable state (Traceability Gate) and unacceptable deviations (Traceability Problem) from this state. We describe and formally define how both, the acceptable states and the unacceptable deviations can be detected in order to enable practitioners to systematically assess their project's traceability. We evaluated the proposed model through an expert survey. Participating experts considered the quality model to be complete and attested that its quality criteria are of high relevance. However, experts weight the occurrence of different traceability problems with different criticality. This information is useful for practitioners to quantify the impact of traceability problems and to prioritize the assessment of traceability elements.

Keywords: requirements traceability; traceability quality model; problem classes; assessment

1 Motivation and Challenges

Requirements traceability is a critical element of any rigorous software development process. It provides support for numerous software engineering tasks. However, achieving purposed and trustworthy traceability remains a challenge, which is not yet solved. In industrial practice, traceability is created and maintained by humans who make mistakes [RMK13]. The resulting traceability is often of dubious quality but serves as the foundation for high impact development decisions. Developing safety-critical systems requires the compilation of safety cases arguing that a system is safe for use, which typically involves traceability [Kel99]. In previous studies [RMKC14, MJZC13], we investigated projects that struggled with problems such as a lack of compliance or latent safety risk, which were caused by incomplete or missing traceability data. When reflecting on the results of those previous studies, we realized that no classification of traceability problems was available to systematically assess traceability for structural deficiencies. This lack of a well-defined problem classification makes it difficult for software practitioners to generate an understanding of possible traceability problems and how to recognize them.

¹ Technische Universität Ilmenau, Software Systems Group, Ilmenau, Germany, patrick.rempel@tu-ilmenau.de

² Technische Universität Ilmenau, Software Systems Group, Ilmenau, Germany, patrick.maeder@tu-ilmenau.de

2 A Requirements Traceability Quality Model

In this presentation we propose an enumeration of potential traceability quality problems along with an enumeration of possible conformance assessment results. The entire work has been published at [RM15]. A traceability assessment refers to determining the degree of traceability fulfillment and can lead to four possible results. First, the fulfilling set fully conforms with the set of required data. This kind of result is represented in our quality model by the traceability gate elements. Second, the fulfilling set is *incomplete*, because it misses data in order to fully conform with the set of required data. This state is represented in our quality model by the problem category missing traceability. Third, the fulfilling set is *redundant*, because it contains superfluous data, not necessary to conform with the set of required data. This state is represented in our quality model by the problem category superfluous traceability. Fourth, the fulfilling set is *incomplete* and *redundant*, and thus a composite problem category of missing traceability and superfluous traceability.

3 Results and Conclusions

We conducted a survey with 13 traceability experts to evaluate the completeness and usefulness of the proposed model. The participants considered the proposed assessment model to be *complete* and attested the model high practical relevance. Traceability problems related to *completeness* were consistently rated as more important than problems related to *appropriateness*. We assume this pattern is attributed to the differing problem implications. Problems related to the quality attribute appropriateness imply that unnecessary effort was spent. Problems related to the quality attribute completeness imply that traceability based decisions are made on incomplete data. Especially, within the context of safety critical software, this can be a potential threat to the functional safety of a system.

Acknowledgments We are funded by the German Ministry of Education and Research (BMBF) grant 01IS14026B.

References

- [Kel99] T. Kelly. *Arguing safety—a systematic approach to managing safety cases*. York, 1999.
- [MJZC13] Patrick Mäder, Paul L. Jones, Yi Zhang, and Jane Cleland-Huang. Strategic Traceability for Safety-Critical Projects. *IEEE Software*, 30(3):58–66, May 2013.
- [RM15] Patrick Rempel and Patrick Mäder. A quality model for the systematic assessment of requirements traceability. In *Proc. 23rd IEEE International Requirements Engineering Conference*, pages 176–185, 2015.
- [RMK13] Patrick Rempel, Patrick Mäder, and Tobias Kuschke. An empirical study on project-specific traceability strategies. In *Proceedings of the 21st IEEE International Requirements Engineering Conference*, pages 195–204. IEEE, 2013.
- [RMKC14] Patrick Rempel, Patrick Mäder, Tobias Kuschke, and Jane Cleland-Huang. Mind the gap: assessing the conformance of software traceability to relevant guidelines. In *Proc. 36th International Conference on Software Engineering ICSE*, pages 943–954, 2014.