Context-specific Requirements Engineering Quality

Henning Femmer

Technische Universität München, Germany,
femmer@in.tum.de

Extended Abstract

The *Requirements Artefact* is a central entity in the software engineering process: Based on this artefact, designers create an architecture, developers build the system, test managers set up a test-strategy, etc. Because of this central concern of stakeholder needs, RE artefacts must be represented in a form that all stakeholders understand. Therefore, RE artefacts are predominantly written in *natural language (NL)*: Based on a globally distributed questionnaire with 151 industry participants, Mich et al. [9] report that 95% of the participants represent their requirements using natural language. Of these, 16% use structured natural language, e.g. through templates and forms and 79% apply "common natural language"[9].

To minimize risks, projects apply *quality assurance* (QA) for RE artefacts, for example in the form of inspections. In fact, implementing any kind of reviews for RE artefacts is considered one of the key project success factors according to Hoffman and Lehner [3]. The key element in these reviews, as summarized by Katasonov and Sakkinen [6], is a definition of quality, against which an artefact is checked.

We argue that this definition of requirements quality is an inherently *context-specific* concept, because RE is a very diverse activity: As requirements artefacts are only means for software production and not their ultimate goal, their definition of quality must adapt to the way in which they are used in the corresponding project, process or company. This is supported by a recent study [2], in which we showed that checking RE artefacts against a standard definition of quality from ISO/IEC/IEEE 29148 [5] leads to various violations. However, qualitative analysis revealed that practitioners to some extent refuse the ISO definition of requirements quality for this artefact. Reasons were, among others, that the artefact did not need to fulfill various criteria at this point in the process, or that specific criteria cannot be fulfilled for the corresponding domain [2].

Problem Statement: We need a way to define quality of a requirements artefact in a context-specific manner, as well as an efficient method for analyzing the quality of natural language requirements artefacts against such a context-specific quality definition.

Currently there is no understanding of such context-specific RE artefact quality definition. Current research, e.g. Katasonov and Sakkinen [6], as well as cur-

rent requirements engineering standards, such as the IEEE Recommended Practice for Software Requirements Specifications [4] or the ISO/IEC/IEEE 29148 [5], define quality of requirements through a set of generic properties, independent from the context. In contrary, we propose an approach using activity-based quality models, which defines requirements artefact quality based on how they are used in the software engineering process.

As the definition of quality is central in QA, we must furthermore analyze the impact of a context-specific definition of quality onto QA approaches. So far practitioners usually perform QA methods (e.g. reviews) manually due to the requirements representation in natural language. In the current state of practice, context-specific definitions are incorporated into guidelines. However, the produced requirements artefacts very often violate the guidelines in various ways, e.g. requirements artefacts usually contain plenty of passive voice requirements, even though many guidelines discourage this. We argue, that this gap is due to shortcomings in QA: In order to make QA for a context-specific quality definition applicable in practice, a method is required that is able to check a requirements artefact against a quality definition effectively and efficiently. We define Requirements Smells, an approach that searches for violations to common quality criteria based on natural language processing. Requirements Smells are elicited from standards, related work and company-specific guidelines, implemented into an open source quality analysis tool, and evaluated with various requirements specifications from Daimler, Wacker, MAN, and Munich Re.

References

1. H. Femmer, J. Kucera, and A. Vetrò. On The Impact of Passive Voice Requirements on Domain Modelling. In *International Symposium on Empirical Software Engineering and Measurement (ESEM)*, 2014.
2. H. Femmer, D. Méndez Fernández, E. Juergens, M. Klose, I. Zimmer, and J. Zimmer. Rapid Requirements Checks with Requirements Smells: Two Case Studies. In *Proc. of the 36th International Conference on Software Engineering (ICSE14)*, 2014.
3. H. Hofmann and F. Lehner. Requirements engineering as a success factor in software projects. *IEEE software*, (August), 2001.
4. IEEE Computer Society. IEEE Recommended Practice for Software Requirements Specifications. Technical report, IEEE Computer Society, 1998.
5. ISO, IEC, and IEEE. ISO/IEC/IEEE 29148:2011-Systems and software engineering - Life cycle processes - Requirements engineering. Technical report, ISO IEEE IEC, 2011.
6. A. Katasonov and M. Sakkinen. Requirements quality control: a unifying framework. *Requirements Engineering*, 11(1):42–57, Oct. 2005.
7. D. Méndez Fernández. *Requirements Engineering: Artefact-Based Customisation.* PhD thesis, Technische Universität München, 2011.
8. D. Méndez Fernández and S. Wagner. Naming the Pain in Requirements Engineering - Design of a Global Family of Surveys and First Results from Germany - Technical Report. 2013.
9. L. Mich, F. Mariangela, and N. I. Pierluigi. Market research for requirements analysis using linguistic tools. *Requirements Engineering*, 9(1):40–56, Feb. 2004.