Quantitative Metrics for Requirements Quality a posteriori

Tatiana Chuprina and Vincent Aravantinos
Fortiss GmbH
Munich, Germany
emails: {chuprina, aravantinos}@fortiss.org

***Abstract*—The question of ”how to measure the quality of requirements?” remains problematic due to its subjectivity.**

In this paper we present various quantitative metrics for assessing the quality of requirements assuming a relation between requirements quality and maturity of requirements, i.e., number of changes to be done in requirements document. We consider the number of corrections done in a requirements document within the requirements engineering (RE) and implementation stages; as well as their influence on the time for development process.

Importantly, the proposed metrics for assessing the quality of requirements is only usable after project completion. Therefore, this method is a good tool for empirical studies, even though not usable to assess requirements quality during a project.

I. Introduction

A. Problem

Measuring the quality of requirements remains problematic [1], [2] due to subjectivity of quality definition. There are only few quantitative metrics to measure the quality of requirements. All of them are looking at intrinsic characteristic of requirements, e.g., whether the requirement's statement includes any syntactic defects or conforms to qualitative attributes such as, unambiguity, completeness or consistency. Such approach depends on the requirements' statement and, therefore, decreases an accuracy of the measurement. In other words, the clear baseline for the comparison of the requirements' quality is vague. Thereby, proposing the quantitative metrics for measuring requirements' quality, we attempt to solve the problem with the subjective evaluation of quality and provide a uniform and more precise measure.

B. Contribution

We present various quantitative metrics for assessing the quality of requirements assuming a relation between the requirements quality and the number of requirements' corrections to be done within RE and system implementation workflow. Comparing with existing approaches, discussed in section II, our method considers the quality of requirements with respect to the process, measuring the number of changes and time-consumption during RE and implementation phases. We consider the changes in requirements document done within requirements engineering and implementation stages [3], and their influence on the development time, instead of analyzing the internal characteristics of the requirements' statements. The suggested metrics take into account the maturity of the requirements, resulting in a number from 0 (bad) - 1 (good) for a quality assessment. A developed system, which has passed an acceptance test by a customer, is considered as a baseline for the resulting product. Importantly, the proposed metrics are usable to assess the quality of requirements only after project completion, since this quality is based on the final requirements.

The presented approach can be considered for empirical studies; we plan to employ it in our study for doctoral thesis to analyze an influence of requirements' categorization on quality aspect.

II. Related Work

A. What does "quality" mean?

Despite multiple publications about requirements quality and its assessment, the term "quality" is still subjective [2], [4]. Industry standards [5] specify characteristics and criteria, which presumed effective for improving requirements quality, e.g, completeness, unambiguity. Additionally, the research community provided several types of quality definition and methods for its assessment. For example, Lamsweerde provides a defect-based checklist to inspect requirements for possible flaws and errors in [6]; Pohl proposes a framework defining dimensions of quality: the specification dimension, the representation dimension, the agreement dimension [7]. This approach considers such imprecise and subjective attributes as requirements adequacy or pertinence, thereby purporting uncertain assessment. Instead of analyzing the meaning of the requirements, our method takes into account a maturity of the whole requirements artifact (RA).

Another approach implies syntactic check of the requirements text for improving its comprehension, correctness, ambiguity and other akin characteristics e.g. [8], [9]. All these metrics apply intrinsic inspection of requirements and assess qualitatively the requirements' statement. In contrary, our metrics provide a *quantitative* analysis of a full document with the requirements.

In comparison with methods described above, activity-based quality models shift their approach from inherent properties of the requirements to the context of process, and propose a meta quality model [10], [11]. Furthermore, the ”quality question” also turns to a consideration of how the requirements quality impacts the project success and the relation between them in scientific community [12], [13], so as among practitioners [14].

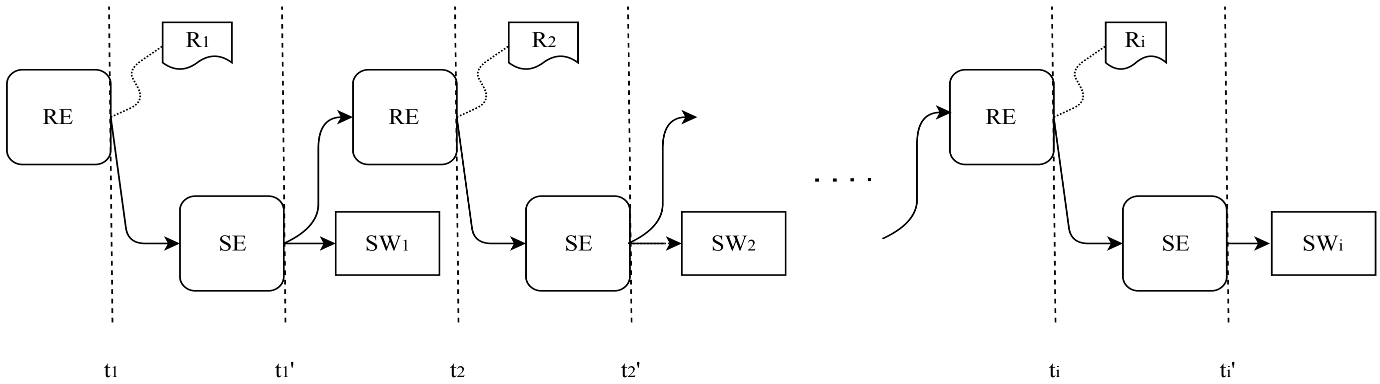

Fig. 1. Requirements engineering and development processes

In contrast to the approaches, our metrics grant a quick and lightweight method for the quality assessment considering the number of the requirements' corrections.

A series of studies about the relation between quality and project outcomes, [15], [13], [16] triggers our research to the proposed metrics. Scientific papers investigating maturity of requirements [17], [3] and their complexity [18] give a base for the idea of transforming a relation between requirements quality and number of changes in requirements document into quantitative measurement, but they described the quality aspects within the ongoing process. The metrics presented in this paper consider a requirements artifact as a "black box" for its maturity and a process of adjusting the requirements at RE and implementation phases, in a quantitative way, once a project is over.

III. Metrics

Figure 1 provides a graphical explanation of the metrics. The workflow is divided into requirements engineering (RE) and system implementation (SI) processes' iterations along the time line: $t_1, t_2, ..., t_i$ - time spent for requirements elaboration during RE phases (RE iteration); $t_1', t_2', ..., t_i'$ - present the time spent for development process based on the provided requirements (SI iteration). The output of every RE iteration is RE artifact (depicted as $R_1, R_2, ..., R_i$); and every SI phase results into SI artifact, e.g., software architecture (shown as SW), till the end of the SI phase, which leads to the release of a product and end of the project.

Consequently, to calculate the *maturity index* of the requirements, the following metrics should be determined:

$$\mu_1(R_j) = i - j \tag{1}$$

end of the number of iterations for the requirements R_j;

$$\mu_2(R_j) = t_i - t_j \tag{2}$$

amount of time (in hours) between initial and the last phases of RE process applying the requirements R_j;

Every RE artifact (a document with requirements) has its index of maturity. The range of this index varies from 0 to 1: 0 means "bad" and 1 indicates "good" quality. The maturity index is inferred from a number of iterations (a certain amount of changes applied to requirements' document) and the time spent for RE and implementation process. The more mature a requirement is, the less changes (iterations) the artifact requires, and the shorter the time for the development process. *Thus, the better quality of the requirements: the higher index of the requirements maturity* (Equation 3).

In Equation 3, the maturity index indicates, how far the considered requirements with the certain maturity parameters (presented as a sum of the metrics for the RE artifact - $\sum \mu_n(R_j)$) from their "good" state (defined as 1)

$$\textit{maturity index} = \frac{1}{\sum \mu_n(R_j)} \tag{3}$$

where (R_j) is a considered RE artifact; j is an initial iteration for calculation.

The input for the first iteration of RE process is a "raw" requirements. During the RE phase, the requirements become elaborated (changed with respect to a project's demands); the result of this process is a corrected RE artifact, that is an input to the next stage (SI). Therefore, we consider the initial RE artifact as a document $\mu(R_j) = 0$.

IV. Conclusion

The presented metrics has its pros and cons. The advantages of this approach described further.

Firstly, the proposed metrics provide a uniformity in measurements of requirements quality. They set a distinct baseline for comparison of requirements quality between different projects. The quantitative assessment provides an unambiguous characteristic of quality, taking into account maturity of the requirements, time spent for RE and development process.

The next advantage is, it is a simplicity of the provided metrics in comparison with some of the considered in section II methods.

Additionally, the described metrics can be used in empirical measurements to assess the quality of requirements only after project completion.

On the other hand, we have to mention also problems of this approach. The provided metrics take in to account time for RE and development processes and amount of requirements changes, but doesn't consider the quality of project. Another point is, that the approach has not been yet applied at practice. However, we consider it to apply in our study regarding requirements categorization approach.

REFERENCES

[1] D. M. Fernández, S. Wagner, M. Kalinowski, M. Felderer, P. Mafra, A. Vetro, T. Conte, M. Christiansson, D. Greer, C. Lassenius, T. Männistö, M. Nayabi, M. Oivo, B. Penzenstadler, D. Pfahl, R. Prikladnicki, G. Ruhe, A. Schekelmann, S. Sen, R. O. Spínola, A. Tuzcu, J. L. de la Vara, and R. Wieringa, "Naming the pain in requirements engineering: Contemporary problems, causes, and effects in practice," *CoRR*, vol. abs/1611.10288, 2016. [Online]. Available: http://arxiv.org/abs/1611.10288

[2] J. Mund, "Measurement-based quality assessment of requirements specifications for software-intensive systems," Ph.D. dissertation, Technical University Munich, 2017.

[3] B. Farbey, "Software quality metrics: considerations about requirements and requirement specifications," *Information and Software Technology*, vol. 32, no. 1, pp. 60 – 64, 1990, special Issue on Software Quality Assurance. [Online]. Available: http://www.sciencedirect.com/science/article/pii/095058499090047U

[4] H. Femmer, "Requirements engineering artifact quality: Definition and control," Ph.D. dissertation, Technical University Munich, 2017.

[5] "Iso/iec/ieee international standard - systems and software engineering – life cycle processes –requirements engineering," *ISO/IEC/IEEE 29148:2011(E)*, pp. 1–94, Dec 2011.

[6] A. van Lamsweerde, *Requirements Engineering: From System Goals to UML Models to Software Specifications*, 1st ed. Wiley Publishing, 2009.

[7] K. Pohl, "The three dimensions of requirements engineering: A framework and its applications," *Information Systems*, vol. 19, no. 3, pp. 243 – 258, 1994, fifth International Conference on Advanced Information Systems Engineering (CAISE '93). [Online]. Available: http://www.sciencedirect.com/science/article/pii/0306437994900442

[8] A. Ferrari, G. Lipari, S. Gnesi, and G. O Spagnolo, "Pragmatic ambiguity detection in natural language requirements," in *2014 IEEE 1st International Workshop on Artificial Intelligence for Requirements Engineering, AIRE 2014 - Proceedings*, 08 2014.

[9] *A New Quality Model for Natural Language Requirements Specifications*, 2006.

[10] S. Wagner, K. Lochmann, L. Heinemann, M. Kläs, A. Trendowicz, R. Plösch, A. Seidi, A. Goeb, and J. Streit, "The quamoco product quality modelling and assessment approach," in *2012 34th International Conference on Software Engineering (ICSE)*, June 2012, pp. 1133–1142.

[11] H. Femmer, J. Mund, and D. M. Fernández, "It's the activities, stupid!: A new perspective on re quality," in *Proceedings of the Second International Workshop on Requirements Engineering and Testing*, ser. RET '15. Piscataway, NJ, USA: IEEE Press, 2015, pp. 13–19. [Online]. Available: http://dl.acm.org/citation.cfm?id=2820704.2820710

[12] K. E. Emam and N. H. Madhavji, "Measuring the success of requirements engineering processes," in *Second IEEE International Symposium on Requirements Engineering, March 27 - 29, 1995, York, England*. IEEE Computer Society, 1995, pp. 204–213. [Online]. Available: https://doi.org/10.1109/ISRE.1995.512562

[13] M. I. Kamata and T. Tamai, "How does requirements quality relate to project success or failure?" in *15th IEEE International Requirements Engineering Conference (RE 2007)*, Oct 2007, pp. 69–78.

[14] J. Beatty and C. Hokanson, "Project value delivered. the true measure of requirements quality," The Magazine for RE Professionals from IREB, Tech. Rep., 2014.

[15] J. M. Verner, K. Cox, S. J. Bleistein, and N. Cerpa, "Requirements engineering and software project success: an industrial survey in australia and the U.S," *Australasian J. of Inf. Systems*, vol. 13, no. 1, 2005. [Online]. Available: http://journal.acs.org.au/index.php/ajis/article/view/73

[16] R. Ferrari, I. Noorwali, and N. H. Madhavji, "Towards a theory on the interaction between requirements engineering and systems architecting," in *2015 IEEE Fifth International Workshop on Empirical Requirements Engineering (EmpiRE)*, Aug 2015, pp. 17–20.

[17] V. R. Basili and D. M. Weiss, "Evaluation of a software requirements document by analysis of change data," in *Proceedings of the 5th International Conference on Software Engineering*, ser. ICSE '81. Piscataway, NJ, USA: IEEE Press, 1981, pp. 314–323. [Online]. Available: http://dl.acm.org/citation.cfm?id=800078.802544

[18] V. Antinyan, M. Staron, A. Sandberg, and J. Hansson, "A complexity measure for textual requirements," in *2016 Joint Conference of the International Workshop on Software Measurement and the International Conference on Software Process and Product Measurement (IWSM-MENSURA)*, Oct 2016, pp. 148–158.