

Conflicts and Synergies among Quality Requirements

Barry Boehm
CS Department
USC
Los Angeles, CA USA
boehm@usc.edu

Xavier Franch
ESSI Department
Universitat Politècnica de Catalunya (UPC)
Barcelona, Spain
franch@essi.upc.edu

Abstract—Analyses of the interactions among quality requirements (QRs) have often found that optimizing on one QR will cause serious problems with other QRs. As just one relevant example, one large project had an Integrated Product Team optimize the system for Security. In doing so, it reduced its vulnerability profile by having a single-agent key distribution system and a single copy of the data base – only to have the Reliability engineers point on that these were system-critical single points of failure. The project’s Security-optimized architecture also created conflicts with the system’s Performance, Usability, and Modifiability. Of course, optimizing the system for Security had synergies with Reliability in having high levels of Confidentiality, Integrity, and Availability. This panel aims at fostering discussion on these relationships among QRs and how the use of data repositories may help discovering them.

Keywords—software qualities, nonfunctional requirements, quality requirements, -ilities, reliability, availability, maintainability, security

I. MOTIVATION

Quality requirements (QRs; also known as Non-Functional Requirements) have often received little attention compared to functional requirements [1], even though empirical studies suggest otherwise (e.g., studies with software architects [2]). It has been argued that neglecting QRs is one of the top ten risks of requirements engineering [3].

One of the key issues on QRs is the impossibility to understand their effects in an isolated manner. Synergies and conflicts among QRs are known from long ago [4][5] both in general and concerning particular types of QRs [6].

The goal of this panel is to foster discussions on the elicitation, representation, analysis and knowledge reuse concerning synergies and conflicts among QRs.

II. KEY QUESTIONS

The key research questions addressed by the panel are:

1) *What type of relationships do exist between QRs?* Synergies (i.e., a QR supporting another) and conflicts (i.e., a QR damaging another) are the two most typical examples of relationships among QRs. Still, other types like plain dependency (a requirement that makes sense just subordinated to another), overlapping or redundancy can be thought of. In addition, the degree to what the relationship is stated (just binary –conflict or not–, a qualification or a number) can be matter of discussion.

2) *How can be these relationships modeled?* Several existing requirement representation notations and models allow representing these relationships. For instance, the NFR language [7] defines different types of qualified relationships like “hurt” or “break” for establishing conflicts and “support” and “help” for synergies. Approaches like [8] establish these types of relationships at the level of software requirement patterns. The level of analysis supported by these representation approaches is a key point in their adoption.

3) *Is it possible to use project data to learn about the relationships among QRs?* There is a clear and increasing trend in exploiting data coming from project repositories like *GitHub*¹, *SonarQube*² or *Jira*³. We may find for instance Behnamghader et al. [9] that investigates the effect of commits in software quality evolution, or the Q-Rapids project [10] which proposes the elicitation of QRs from software and project repositories. The extension of these approaches to investigate relations among these QRs is a potential line of research.

III. PANEL FORMAT

The panel aims at being highly interactive. First, three panelists will provide an initial position with respect to the research questions. Afterwards, the floor will be opened and the panel conducted as a fishbowl panel in which all attendees may take the word and become temporary panelist by replacing one of the current panelists. This highly dynamic format is especially well suited as a brain storming technique.

The two panel organizers will be in charge of moderating the flow of the fishbowl conversation and will take care that opinions are condensed in a few words and written in a post-it stick to the wall. Colours will be assigned to the opinions for different categories: challenges, facts, enablers and evidences.

IV. EXPECTED OUTCOMES

Since time allocated to the panel is short, unrealistic goals need to be avoided. Therefore, we want just to make sure that the opinions expressed in the fishbowl panel are diverse and are well reflected in the testimonial post-its. With this premise, the two panel organizers will apply some classification technique in order to classify the contributions

¹ <https://github.com/>

² <https://www.sonarqube.org/>

³ <https://www.atlassian.com/software/jira/>

by topics. The result will be reported in a short open document and is expected to be the basis of future discussions and events.

As a second goal, we aim at witnessing the emergence of a community of interest in the topic bound to the QRS conference. A next natural step would be to plan ahead for a longer meeting, e.g. a workshop to explore the brainstormed ideas in more depth.

V. CONCLUSIONS

Synergies and conflicts among QRs are still an open topic. The purpose of this panel is to create awareness on the topic and gather qualified opinions and evidences on the current state of the research and the practice.

ACKNOWLEDGMENTS

This work is partially supported by the Q-Rapids project, which has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 732253. The material is also based upon work supported in part by the U.S. Department of Defense through the Systems Engineering Research Center (SERC) under Contract H98230-08-D-0171. SERC is a federally funded University Affiliated Research Center managed by Stevens Institute of Technology. It is also supported by the National Science Foundation grant CMMI-1408909, "Developing a Constructive Logic-Based Theory of Value-Based Systems Engineering". We also acknowledge the support of National

Natural Science Foundation of China No. 91318301, 61432001, 91218302.

REFERENCES

- [1] S. Wagner. *Software Product Quality Control*. Springer, 2013.
- [2] D. Ameller, C. Ayala, J. Cabot, X. Franch. "Non-Functional Requirements in Architectural Decision Making". *IEEE Software* 30(2), 2013.
- [3] B. Lawrence, K. Wiegers, C. Ebert. "The Top Ten Risks of Requirements Engineering". *IEEE Software* 18(6), 2001.
- [4] B. Boehm, J. R. Brown, M. Lipow. "Quantitative Evaluation of Software Quality". ICSE 1976.
- [5] B. Boehm, H. In. "Identifying Quality-Requirements Conflicts". *IEEE Software* 13, 1996.
- [6] D. Mairiza, D. Zowghi. "An Ontological Framework to manage the Relative Conflicts between Security and Usability Requirements". MaRK 2010.
- [7] L. Chung, B. Nixon, E. Yu, J. Mylopoulos. *Non-Functional Requirements in Software Engineering*. Springer 2000.
- [8] S. Renault, O. Méndez-Bonilla, X. Franch, C. Quer. "PABRE: Pattern-based Requirements Elicitation". RCIS 2009.
- [9] P. Behnamghader, R. Alfayez, K. Srisopha, B. Boehm. "Towards Better Understanding of Software Quality Evolution Through Commit-Impact Analysis". QRS 2017.
- [10] L. Guzmán, M. Oriol, P. Rodríguez, X. Franch, A. Jedlitschka, M. Oivo. "How Can Quality Awareness Support Rapid Software Development? - A Research Preview". REFSQ 2017.