Supporting Requirements Update during Software Evolution

Eya Ben Charrada[1] Anne Koziolek[2] Martin Glinz[3]

Abstract: Keeping the requirements specification up-to-date is crucial for several maintenance and evolution tasks. Nevertheless, due to time and budget constrains, software maintainers usually apply changes to the code only and leave the requirements unchanged, so they rapidly become obsolete and useless. In this work, we propose an approach for automatically identifying what requirements are likely to be impacted when the code is changed. We use a two-step approach. First, we identify the code changes that are likely to have an impact on requirements based on some heuristics. Second, we trace the relevant changes back to the requirements in order to identify those that are likely to be impacted [**?**]. The output of the tracing is a list of requirements that are sorted according to their likelihood of being impacted. We applied our approach to three case studies and could identify for each case between 70% and 100% of the impacted requirements within a list that includes less than 20% of the total number of requirements in the specification [**?**]. We are currently exploring ways to extend the approach to other types of software artifacts such as tests.

Keywords: Requirements evolution, requirements update, impact analysis, traceability, artifact synchronization

[1] Department of Informatics, University of Zurich, Switzerland, charrada@ifi.uzh.ch

[2] Department of Informatics, Karlsruhe Institute of Technology, Germany, koziolek@kit.edu

[3] Department of Informatics, University of Zurich, Switzerland, glinz@ifi.uzh.ch