

Requirements-based testing with defect taxonomies

Michael Felderer¹, Armin Beer²

¹University of Innsbruck, Innsbruck, Austria
michael.felderer@uibk.ac.at

²Beer Test Consulting, Baden, Austria
info@arminbeer.at

Abstract: In this paper we summarize requirements-based testing with defect taxonomies which seamlessly integrates defect taxonomies into the standard test process to improve the effectiveness and the efficiency of testing requirements.

Defect taxonomies which consist of hierarchies of defect categories provide information about the distribution of faults and failures in a project. In practice, most defect taxonomies are only used for the a-posteriori allocation of testing resources to prioritize failures for debugging purposes. *Requirements-based testing with defect taxonomies* (RTDT), which has been defined and evaluated in several previous publications [FB12], [FB13a], [FB13b], [FB14a], [FB14c], [FBP14], exploits the full potential of defect taxonomies to control and improve all phases of the overall test process, i.e., test planning, design, execution and evaluation. Figure 1 summarizes the process steps and artifacts of RTDT.



Figure 1: Artifacts and process steps of requirements-based testing with defect taxonomies

On the basis of the requirements specification the defect taxonomy is created (step 1). Then the requirements are linked to defect categories and validated (step 2). With defect taxonomies especially the requirements quality attributes completeness, ranked for importance, verifiability, traceability, comprehensibility and right level of detail can be reviewed [FB13a]. Afterwards tests are planned (step 3) which results in a test strategy taking defect taxonomies into account. In this test strategy, the test design techniques like equivalence partitioning are assigned to combinations of defect categories and requirements [FB14a]. Test design techniques allow the test strength to be varied on the basis of the priority of requirements and the severity of defect categories. From the test strategy abstract and executable test cases are designed (step 4) and then executed (step 5). Finally, the test results are evaluated and failures are assigned to defect categories (step 6). Due to this traceability between failures and defect categories, the severity values of failures which typically have varying accuracy can be checked and adapted. Thus, more realistic release quality statements and more precise planning of additional hotfixes or releases are possible [FB13b].

Besides the benefit of more realistic release quality statements, the improved effectiveness and efficiency of requirements-based testing with defect taxonomies was shown in industrial case studies performed in a public health insurance institution in Austria [FB13b], [FB12]. Furthermore, in a student experiment we showed that RTDT is independent of a specific type of defect taxonomy [FBP14]. RTDT is independent of a specific domain [FB13b] and we expect similar benefits in other domains besides public health insurance institution where RTDT has already successfully been applied. We therefore plan further empirical case studies on the effectiveness and efficiency of RTDT in different domains as well as on the influence of RTDT on release planning [FB14b] as future work.

Acknowledgement. Parts of the work described in this paper were supported by the project QE LaB – Living Models for Open Systems (FFG 822740).

References

- [FB12] M. Felderer and A. Beer, Estimating the Return on Investment of Defect Taxonomy-Supported System Testing in Industrial Projects, 38th Euromicro Conference on Software Engineering and Advanced Applications, IEEE, 2012.
- [FB13a] M. Felderer and A. Beer, Defect Taxonomies for Requirements Validation in Industrial Projects, 21st International Conference on Requirements Engineering (RE 2013), 2013.
- [FB13b] M. Felderer and A. Beer, Using Defect Taxonomies to Improve the Maturity of the System Test Process: Results from an Industrial Case Study, Software Quality Days 2013, Springer LNBP, 2013.
- [FB14a] M. Felderer and A. Beer, Using Defect Taxonomies for Testing Requirements, IEEE Software, 2014, doi: doi.ieeecomputersociety.org/10.1109/MS.2014.56 (online first).
- [FB14b] M. Felderer and A. Beer, On the Role of System Testing for Release Planning: Industrial Experiences from Comparing Two Products, PROFES 2014, Springer LNCS, 2014.
- [FB14c] M. Felderer and A. Beer, Practical Guidelines for System Testing with Defect Taxonomies, ASQT 2013, OCG, 2014.
- [FBP14] M. Felderer, A. Beer, and B. Peischl, On the role of defect taxonomy types for testing requirements: Results of a controlled experiment, 40th EUROMICRO Conference on Software Engineering and Advanced Applications (SEAA 2014), 2014.