

# What Am I Testing and Where? Comparing Testing Procedures Based on Lightweight Requirements Annotations

Florian Pudlitz<sup>1</sup>, Florian Brokhausen<sup>2</sup>, Andreas Vogelsang<sup>3</sup>

## Abstract:

The article was originally published in the international journal *Empirical Software Engineering* with the title “What am I testing and where? Comparing testing procedures based on lightweight requirements annotations” [PBV20].

**Keywords:** Annotation; Requirements modeling; Test management; Test evaluation; Simulation

## 1 Overview

Software complexity has increased dramatically in many areas in recent years, for example due to increasing automation or stronger interconnectivity between devices. This results in growing challenges in requirements and test management. Nowadays, system requirements are often written in natural language, which makes automated processing more difficult. The goal is to ensure that all requirements are checked despite the increasing complexity of the test cases. Previous test procedures use a transformation of requirements specification into test specification with consideration of the traceability of test results. However, these procedures reach their limits in complex test scenarios in different test levels, because the system runs through several situations automatically. For example, when testing driver assistance systems in real test drives or in traffic simulations with several hundred vehicles, new test approaches are required. In addition, it is not yet possible to make any statements about the similarity of test levels or test scenarios within a test level.

Our approach is based on a Multilevel Markup Language for annotating text passages in natural language requirements [PVB19]. The test engineer has the possibility to mark up text passages and observe them in test runs. After a test run, the log data is evaluated and the results can be displayed in relation to the annotations in the natural language requirements. Manual annotation can be partially automated by using machine learning algorithms [PBV19].

---

<sup>1</sup> Technische Universität Berlin, Fachgebiet Distributed and Operating Systems, Straße des 17. Juni 135, 10623 Berlin, Deutschland, florian.pudlitz@tu-berlin.de

<sup>2</sup> Technische Universität Berlin, Fachgebiet für Fluidsystemdynamik, Straße des 17. Juni 135, 10623 Berlin, Deutschland, florian.brokhausen@tu-berlin.de

<sup>3</sup> Universität zu Köln, Lehrstuhl für Software and Systems Engineering, Weyertal 121, 50931 Köln, Deutschland, vogelsang@cs.uni-koeln.de

Our experimental evaluation, shown schematically in Figure 1, investigates four different evaluation foci based on annotations of the Multilevel Markup Language. First, it investigates large scale usage of the markup language, the annotation scalability. Second, we examine Test Case Alignment, which examines how well the test levels fit the requirements. Third, in Test Stage Compliance, we compare the test levels with respect to the annotations in the requirements. Fourthly, we compare the test scenarios that have been performed, the Test Stage Similarity.

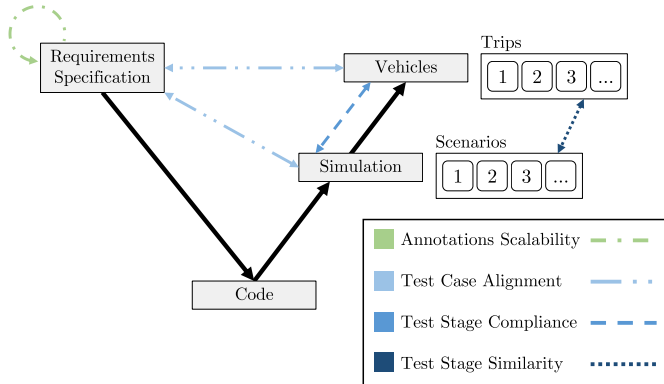


Fig. 1: Schematics of the experimental setup.

We investigate two different test stages. On the one hand, we chose a traffic simulation with 1300 vehicles for all evaluations carried out, since simulations are becoming increasingly important, especially in the automotive sector. On the other hand, we used real driving data from 53 test drives. Investigations have shown that the use of field user data, especially in the automotive area, has been of little importance so far, but will be further expanded in the future [EBV20].

## 2 Results

Our experiment shows how different test levels are linked to natural language requirements. The test engineer receives not only concrete evaluations of the annotations but also holistic statements about test coverage of requirements. With regards to the performed simulation, 75 (25.5%) of the 294 requirements exclusively contain fulfilled annotations. The evaluation of the real trips shows that 80 (27.2%) of the considered requirements contain fulfilled annotations only. In contrast, 75 (25.5%) requirements are entirely unfulfilled within the simulation with regards to the inherent annotations and 42 (14.3%) with respect to real driving data.

## Bibliography

- [EBV20] Ebel, Patrick; Brokhausen, Florian; Vogelsang, Andreas: The Role and Potentials of Field User Interaction Data in the Automotive UX Development Lifecycle: An Industry Perspective. Association for Computing Machinery, New York, NY, USA, p. 141–150, 2020.
- [PBV19] Pudlitz, Florian; Brokhausen, Florian; Vogelsang, Andreas: Extraction of System States from Natural Language Requirements. In: 27th IEEE International Requirements Engineering Conference (RE). 2019.
- [PBV20] Pudlitz, Florian; Brokhausen, Florian; Vogelsang, Andreas: What Am I Testing and Where? Comparing Testing Procedures based on Lightweight Requirements Annotations. Empirical Software Engineering, 2020.
- [PVB19] Pudlitz, Florian; Vogelsang, Andreas; Brokhausen, Florian: A Lightweight Multilevel Markup Language for Connecting Software Requirements and Simulations. In (Knauss, Eric; Goedicke, Michael, eds): Requirements Engineering: Foundation for Software Quality. Springer International Publishing, Cham, pp. 151–166, 2019.