
Reference: QSE-SWQL-TG

Topic: Prototype of a Generator for Parameter Creation of Unit Tests

Course-Type: Seminar, Project, Bakk-/Masterthesis

Start: As soon as possible

End: To be defined

Industrypartner: Software Quality Lab

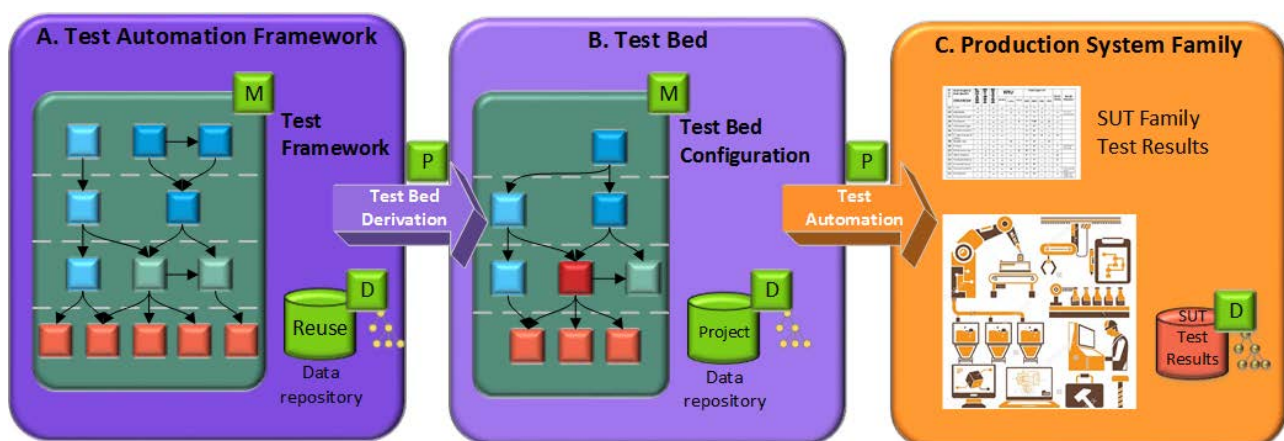
Contact: Kristof Meixner (kristof.meixner@tuwien.ac.at)
Dietmar Winkler (dietmar.winkler@tuwien.ac.at)
Stefan Biffel (stefan.biffel@tuwien.ac.at)

Background:

Testing software systems, especially of existing software artifacts, is often challenging. Although software developers are supported at different stages by various testing frameworks, the decision how and which tests are implemented is left to the developers themselves. This way, due to pressure of time and cost, all too often tests are implemented, that poorly represent the logic of the program and are thus insufficient for critical systems.

Approaches, that aim at a better selection of test input parameters and hence an improvement of testing, are, for example, symbolic execution [1], equivalence class partitioning [2], or boundary value analysis [3]. Framework for such methods partly exist, but are then usually applicable for a particular class of programming language or a specific testing framework (like Junit). Furthermore, such tools are, in many cases, not integrated in open source IDEs like Eclipse.

In the context of a conceptualization and development of a flexible 'Test Automation Framework' (shown in the figure), this topic aims at the development of a concept and an implementation of a prototype for a test generator.



This prototype should allow the selection of reasonable test input parameters for unit testing a particular method. Besides this, the concept should be open to back different programming languages of a family and should provide cross-border test framework support.

The complexity of this topic requires an iterative and incremental approach, which, depending on the course type and effort, offers several sub-topics to be investigated and worked on.

This topic is provided and supervised in cooperation with our industry partner Software Quality Lab (<https://www.software-quality-lab.com/>).

Tasks:

- Analysis of source code and test data
- Identification and evaluation of means and methods for test case generation
- Identification and evaluation of means and methods for parameter analysis
- Conceptualization of a model for parameter class representation
- Conceptualization of a model for the representation of source code
- Prototypical development of a generator or editor which is able to find test parameters based on input parameters
- Evaluation of the prototype

Expertise:

For this topic a set of skills is recommended (at least two are mandatory).

- SQS course (software quality assurance)
- Java programming skills
- Data modeling
- Empirical evaluation, e.g. case study, pre/post comparison
- Eclipse RCP (is an advantage)
- C/C++ (is an advantage)

[1] King, J. C. (1976). Symbolic execution and program testing. *Communications of the ACM*, 19(7), 385-394

[2] Ostrand, T. J., & Balcer, M. J. (1988). The category-partition method for specifying and generating functional tests. *Communications of the ACM*, 31(6), 676-686.

[3] Ramachandran, M. (2003, September). Testing software components using boundary value analysis. In *Euromicro Conference, 2003. Proceedings. 29th* (pp. 94-98). IEEE.