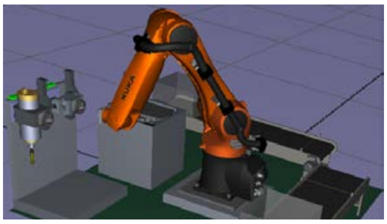
# Test Automation for Modular Embedded Systems Test Beds



**The flexible test automation framework for a modular embedded systems test bed enables flexible configurations of test automation tool chains. Test processes can be adapted and configured according to customer needs efficiently.**

## Goal

Systematic testing of automation systems (*Hardware-in-the-Loop*) requires the collaboration of different testing tools in context of a defined test process. Heterogeneous tool landscapes typically require the manual integration of selected tools, implemented by software experts and supported by test engineers. Thus, automated testing is limited to subdomains of the automation system. In addition, limited flexibility and extendibility of established tool chains increases maintenance effort of software experts.

Main goal is to develop a modular test automation framework to enable (a) creating test scenarios and test cases by domain experts, (b) support these activities with automatable concepts, and (c) execute generated test cases automatically.

## Implementation

The test automation prototype has been developed by the Software Quality Lab GmbH and TU Wien and enables the flexible configuration of tool chains for supporting of test automation processes. The seven-layer architecture supports domain experts in using established software tools at defined layers of the test automation framework and provides related test automation methods for the flexible integration into existing tool chains. Based on the modular concepts of the tool chain can be addressed efficiently.

Defined interfaces between the layers and tools facilitate a seamless information exchange, e.g., between test management, test case abstraction and specification, implementation, execution (i.e., by using simulation or real-world devices), and reporting. Domain experts can focus on designing test scenarios and test cases on test management level without taking software code into consideration.

## Illustrative Use Case

Embedded within the test automation framework, the test process includes four basic steps (see Figure 1):

1. *Experts define test scenarios*, e.g., by using a *Jira Plug-In.*
2. Definition of *abstract test cases & test data* with Keyword-Driven Test Approaches (e.g., *Specflow*).
3. Generation/Mapping of abstract test cases and concrete test code with the *Jira Plug-In*.
4. *Automated execution* of test cases and *test report generation* with *Jenkins.*

Figure 2 illustrates the basic use case including tools and data flows in context of the test process steps.

1. *Definition of abstract test cases based on Keyword-Driven Tests.*
2. *Generation of concrete test cases*, execution, and reporting.
3. *Software Test Code Base* as foundation for test case definition.

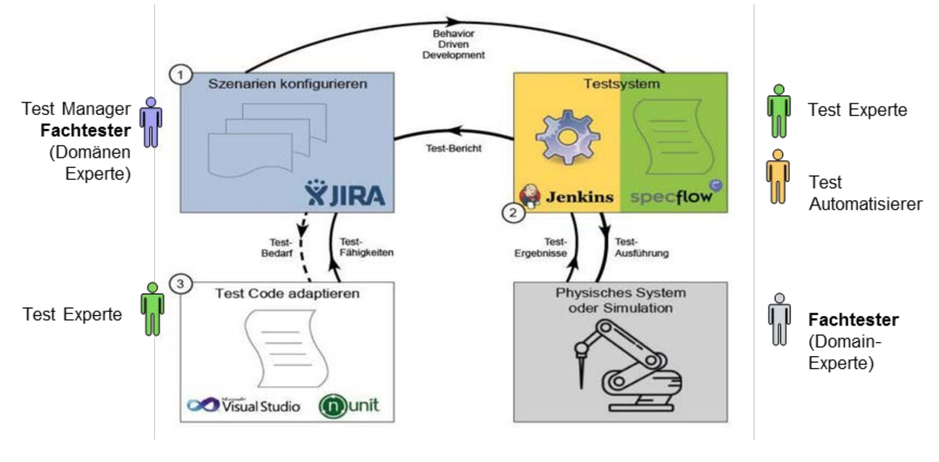


Figure 2: Illustrative Use Case for Test Process Application.

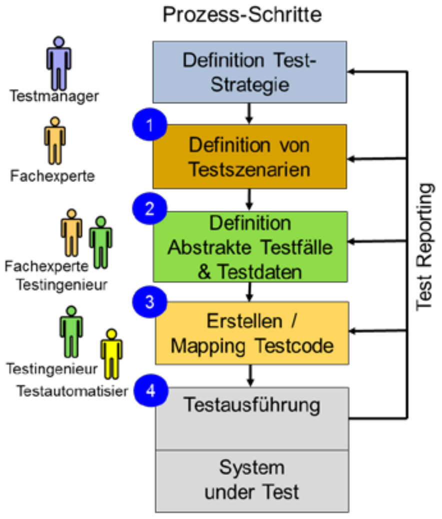


Figure 1: Test Process Steps.

## Technical Specification

- Modular tool kit for test automation.
- Flexible configuration of tool chains based on defined interfaces for tools at different test automation layers.
- Support of „Best-of-Breed" tools.

### Benefits for Customers

- Test scenario/case description designed by experts.
- Efficient mapping of abstract test scenarios and executable test code.
- Automated test case execution and reporting.

**Contact:**

Prof. Dr. Stefan Biffl
TU Wien
stefan.biffl@tuwien.ac.at
qse.ifs.tuwien.ac.at

Johannes Bergsmann
Software Quality Lab GmbH
Johannes.Bergsmann@software-quality-lab.com
www. software-quality-lab.com