



TESSY®

TEST SYSTEM

automated  
**unit/integration tests**

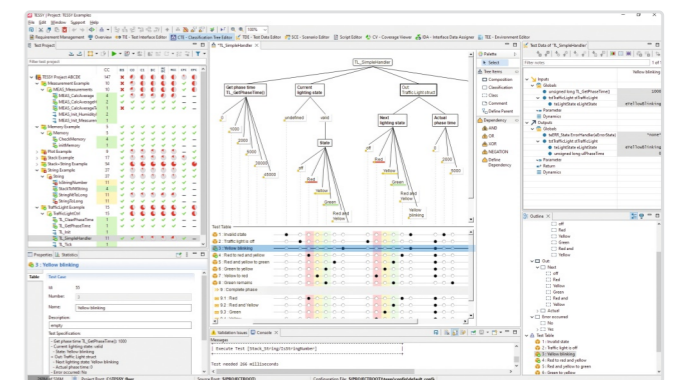
**Razorcat is focused on products and services for testing of embedded software and systems.**

Since 1997 we develop software test tools which are continuously improved to meet the steadily growing requirements of today's development processes for safety-critical software and high quality standards.

*Our team consists of experts with in-depth experience in software development, tool adaptations, interfaces and customized solutions as well as development and testing of safety-critical software. Our experts share their knowledge within seminars or consulting services. We are always providing the best quality, in shortest time and with highest efficiency!*



*TESSY is qualified for safety-related software development according to IEC 61508/ISO 26262 as well as IEC 62304 and EN 50128.*



Razorcat Development GmbH

info@razorcat.com · www.razorcat.com

PRODUCTS + SUPPORT · Head office · Witzlebenplatz 4 · 14057 Berlin · Phone: +49 (0) 30 - 53 63 57 - 0

SERVICES + TRAINING CENTER · Im Ermisgrund 3 · 76337 Waldbronn · Phone: +49 (0) 7243 - 561 58 - 0





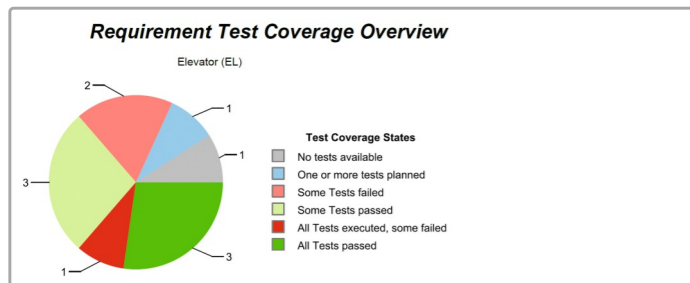
**TESSY**® as one of the first unit test tools is today's leading solution for testing of embedded software. Designed to support development and testing according to standards, TESSY is well established to be used for high-quality products and safety-critical applications.

### Key features for unit and integration testing

- Intuitive test design and automated workflows
- Manage, link and trace requirements
- Edit test data within spread sheets and user code
- Support for testing of software variants
- Fault injection testing
- Define time-based component testing scenarios
- Test execution on hosts, simulators and hardware
- Plot test results graphically
- Automatic Code Access (CA) checking
- Analyze code coverage in flow chart graphics
- Accumulated line-based Hyper Coverage (HC)
- Generate customizable reports in several formats
- Command line scripting for continuous integration
- Integrated Classification Tree Editor (CTE)
- Calculation of software metrics (McCabe)
- Testing effort estimation and tracking
- Auditing of test changes via textual test scripts
- Change based regression testing
- Automatic checking of test case quality

### Efficient testing and traceability

TESSY reduces manual tasks to a minimum making the verification process scalable, transparent and less susceptible to errors. Generation of test specifications, test execution and reporting are automated by TESSY to provide consistent traceability which is also required for certification.

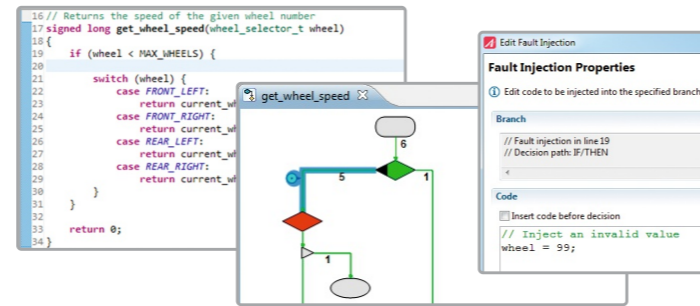


### Performing requirement reviews

TESSY provides a comprehensive requirement management with interfaces to third party tools like DOORS or Polarion. Full requirements coverage can be achieved by tests or ASCII/PDF based check lists and review protocols.

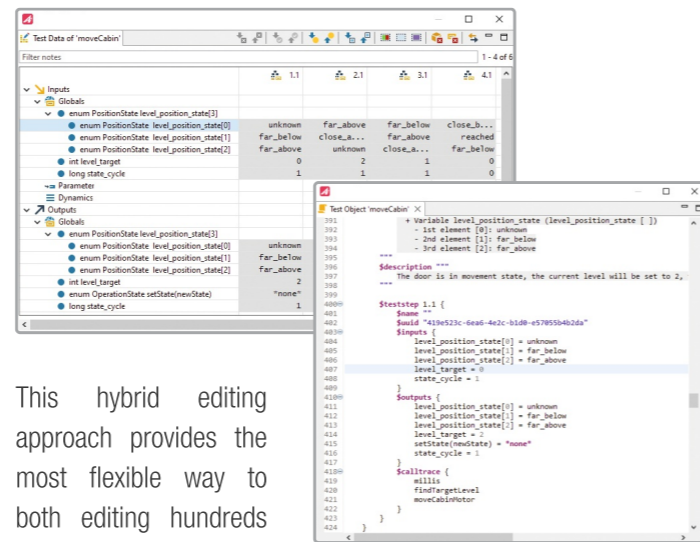
### Fault injection for robustness testing

Unreachable branches and abnormal operating conditions can thoroughly be tested using TESSY's automated fault injections that are based on the control flow of the source code.



### Two-way editing of tests: GUI and textual form

The script perspective allows textual editing of tests as well as review of changes in a dedicated test scripting language. All test data can be converted from the script format to the TESSY internal format and vice versa.



This hybrid editing approach provides the most flexible way to both editing hundreds of test data values using a dedicated table-based GUI as well as comfortable editing of large and distributed testing code fragments within a full featured ASCII based editor.

### Change based regression testing

TESSY detects changes in the source code automatically. Based on these changes TESSY pre-assigns the interface elements and automatically adapts the user code. For fast and straightforward regression testing only tests for changed code will be executed.

### Continuous testing with Jenkins

The command line interface of TESSY and a dedicated Jenkins plugin for TESSY offer powerful testing setups on continuous integration platforms like Jenkins.

### Easy data handling

Comfortable spread sheet editors with issue highlighting and access within the Classification Tree Editor make data handling easy. Use numerous import/export formats for convenient exchange of test data and requirements.

### Continuous adaptations

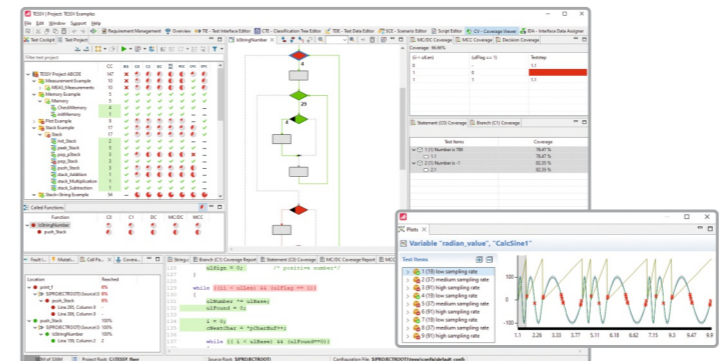
The broad range of supported compilers, target debuggers and interfaces to third party tools is continuously enhanced. Adaptation of TESSY for specific systems, interfaces or optional features can be offered on demand.

### Test driver generation

TESSY automatically generates the complete test framework including stub functions and all external references.

### Code coverage analysis

TESSY includes eight different coverage measurements which can be selected individually or according to norms and standards. Coverage results are visualized in a graphical flow chart linked with colored source code views as well as in textual form.



Powerful navigation through the flow chart easily reveals uncovered branches and conditions being spotlighted within the code view.

### Detecting hidden or untested source code

As soon as tests have been set up, the Code Access (CA) feature automatically detects which lines of code have not been tested. Additionally, all coverage results will be accumulated into the Hyper Coverage (HC) to reveal any unreached source code lines.

### Test the quality of your tests

Mutation testing of all passed tests is one of the additional options for the test execution in batch mode. Also the initialization of output variables with different test data patterns and the check for existence of at least one verification within a test can be applied automatically.

*Automate your test process*



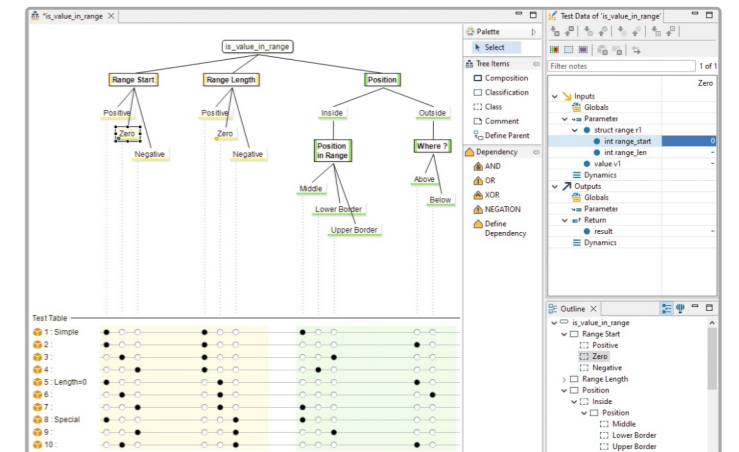
### Systematic test design with the integrated Classification Tree Editor (CTE)

CTE® is a comfortable graphical editor for the Classification Tree Method (CTM). This method is an intuitive and systematic way to transform a (functional) specification into a set of error sensitive and low-redundant test case specifications.

Test relevant aspects and their recursive partitioning in equivalence classes build the classification tree.

Test cases are defined in a second step by combining classes of the tree to specify inputs and expected results within a combination table.

The resulting test case specifications are generated automatically and the test cases can immediately be executed.



This method covers the aspect of model-based testing and the requirements of standards. It is well applicable for black box testing based on the module design specification. The visualization of the test specification leads the tester through the test design and reduces complexity.

### Key features for systematic test specification

- Create error sensitive and low-redundant test cases
- Create, import and export classification trees
- Automated tree generation based on function interface
- Provides a data dictionary for interface variables
- Automatic tree updates on source code changes
- Define dependency rules between test aspects
- Automated test case generation
- Shows inconsistencies with automatic validation checks
- Provides statistics data for tree and test table
- Provides customizable auto-layout
- Convenient assignment of test data