

## Software Test: Structured and Efficient Embedded System Tests

### Objectives

This training focuses on the software test process, from test requirements analysis to acceptance testing.

It supplies information about the test phases and established test methods.

You learn how to professionally plan, specify, execute, document and evaluate tests.

Besides covering the test process, this training gives an overview of the overall software development process and the relation and interaction between development and test, enhancing the mutual understanding between the two divisions.

### Participants

Test engineers, test managers, software developers, software architects

### Requirements

Basic knowledge of a higher-level programming language (e.g. C/C++) .

## Software Test: Structured and Efficient Embedded System Tests

### Content

#### Motivation: The Significance of Software Test in Projects

#### Software Quality

- Software standards and quality features

#### Software Requirements Analysis, Test Requirements Analysis

- Test case related description of functional and non-functional requirements (QoS)

#### Test Planning, Test Plan

- Development of a test strategy (based on requirements, risk, quality, ...)
- Creating a test plan according to IEEE 829
- Test effort, test duration, test resources, test staff
- Defining the test target and test exit criteria
- Program analysis metrics (McCabe, Halstead, LoC, ...)
- Failure analysis, risk analysis

#### Test Specification

- Test requirements analysis and test case specification according to IEEE 829
- Determining test cases based on the defined strategy
- Describing test cases (graphical, formal, tabular or as text)
- Creating test scripts according to IEEE 829

#### Static Software Analysis

- Manual: review, inspection, walkthrough
- Automated

#### Whitebox Test at Module Level

- Program flow oriented testing
- Statement, branch, condition, MC/DC, path coverage
- Data flow oriented testing

#### Blackbox Test at Module and System Test Level

- Functionality, function point, requirements coverage

- Equivalence class partitioning, boundary value analysis
- Fault coverage (comparison with similar systems)
- Error guessing, random test, state transition test
- System behavior tests
- Retest, regression test

**Integration Strategies and Integration Test**

- Incremental, top down, bottom up, big bang
- Test target oriented, process oriented, function oriented strategy
- Hardest first
- Input, output, interface coverage
- Development of drivers and stubs

**Test Automation**

- Test automation definitions and concepts
- When is test automation useful?
- Types and selection of test tools
- Hardware-in-the-loop, HIL
- Software-in-the-loop, SIL

**Cost-efficient Testing**

- Test automation
- Design for test
- Model based testing

**Test Analysis and Test Documentation**

- Test analysis metrics
- Test exit criteria
- Test documentation according to IEEE 829
- Automated test analysis and test document creation

**Exercises**

- Fault analysis
- Code review (Fagan inspection)
- Programming a test driver in C
- Determining test data with CTE according to the classification tree method
- Blackbox and whitebox tests with Tessy
- Determining metrics with the tools cccc and CMT++
- Information about further tools

**Trainings**

<b>Price *</b>	<b>Duration</b>
2.050,00 €	4 days
Training code: E-SW-TEST	

\* All prices are exclusive of applicable VAT.

**Coaching**

Unsere Coaching-Angebote bieten den großen Vorteil, dass unsere Experten ihr Wissen und ihre Erfahrungen direkt in Ihren Lösungsprozess einbringen und damit unmittelbar zu Ihrem Projekterfolg beitragen.

Für Ihre Anfrage oder weiterführende Informationen stehen wir Ihnen gern zur Verfügung.