As of 20.05.2019

Embedded Software Test Object-Oriented for C++: Best Practices for Class and Component Tests

Objectives

This training focuses on tests that, due to their close relation to the programming language, are usually performed by the developers themselves and are thus also referred to as developer tests.

You get to know the development and test process and the related dependencies, extensions and interdependencies.

By making use of the synergies, this enables you to perform tests efficiently.

Static and dynamic test processes and methods for creating test cases are introduced for the optimized test of an embedded system consisting of software and hardware.

This training highlights the particularities related to the test of object-oriented systems.

Through numerous exercises, you are able to apply the respective methods when testing a system that comprises software and hardware.

Participants

Software developers, hardware developers, test engineers

Requirements

A good knowledge of the programming language C++ is a prerequisite for testing object-oriented programs.

Embedded Software Test Object-Oriented for C++: Best Practices for Class and Component Tests

Content

Fundamentals of Testing

- Why is software testing necessary?
- Causes of software defects
- Constructive and analytical quality assurance
- Seven principles of testing
- The fundamental test process
- The psychology of testing

Testing Throughout the Software Lifecycle

- Software development models
- Standards for testing
- Test levels: component test, integration test, system test, acceptance test
- Developer tests (unit test, class test)
- Test types: functional test, non-functional test, structural test
- Re-testing, regression test, maintenance test
- Developing testable software
- Definition: unit, module, component
- Deriving test cases and test data from specifications
- Difference between debugging and testing

Static Test Techniques

- Review process: document Review, code Review, inspection, walkthrough
- Static analysis with tools

Dynamic Test Techniques

- Blackbox test: equivalence partitioning, boundary value analysis, decision table test, state transition test, use case test
- Whitebox test: statement coverage, decision coverage, condition coverage
- Experience-based techniques: error guessing, exploratory testing
- Systematic approach for test case development
- Choosing a suitable test technique
- Assessment of test techniques

Metrics for Complexity Measurement

- Lines of code, McCabe cyclomatic complexity, Halstead metric
- Applying metrics in the test process

Design for Test

- The S.O.L.I.D principles
- Single responsibility, open closed
- Liskov substitution
- Interface segregation, dependency inversion

Testing Object-oriented Software

- Testing modal and non-modal classes
- Testing member functions of a class
- Testing inherited functions and state machines by means of flattening
- Testing class hierarchies
- Testing associations
- Testing strict and non-strict inheritance
- Testing polymorphic class hierarchies
- Test patterns
- Using Google Test and Google Mock

Test Driven Development, TDD

- Advantage of TDD
- Embedded TDD strategies
- TDD example

Integration of Hardware and Software

- Test execution on hardware

Further Test Activities during the Test Process

- Risk management
- Configuration management
- Software quality attributes according to ISO 9126
- Test documentation according to IEEE 829
- Test tool overview

Practical Exercises

- Execution of a code review
- Execution of blackbox tests using unit tools and mock tools
- Deriving test cases from requirements specifications
- Determining code metrics using the tool cccc
- Exercise for testing classes and class hierarchies using unit test tools and mock tools
- Replacing unavailable units by means of stubs, mocks or fakes
- The Arm/ Keil µVision and a Cortex®-M evaluation board are used for test execution on hardware.

Trainings

Date	Price *	Duration
01.07.2019 – 04.07.2019	2.050,00 €	4 days
24.02.2020 – 27.02.2020	2.050,00 €	4 days

* Price per attendee, in Euro plus applicable VAT.

As of 20.05.2019

Training code: E-TEST-OO

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.