



Embedded Software Testing with C++

Course category	Testing
Training area	Quality
Course code	TC++-401
Duration	4 days
Additional information	Contact Feabhas for delivery dates

Verification and validation are vital parts of the software quality process. This is especially true in a real-time embedded environment, where the system may be subject to strict safety or reliability requirements.

This four-day course introduces the concepts and practices of testing embedded software across the entire development lifecycle. It covers software verification and validation in an embedded environment and is a mixture of lectures and practical exercises.

Attendees will also perform practical exercises in software verification and Agile development practices.

Course objectives:

- To understand the need for testing in an embedded environment
- To appreciate the value of verification and validation as part of the quality process.
- To give an understanding of the test process
- To understand the objectives of Agile Programming methodologies
- To identify the different types of testing and where and when they should be used

Delegates will learn:

- How to create their own test harnesses
- The use of test harnesses for embedded systems
- Integration testing using test doubles
- How to write code in a Test-Driven Development style.

Pre-requisites:

Attendees should have a good working knowledge of the C++ language

Who should attend:

The course is designed for software engineers working in an embedded, real-time environment. The target audience is engineers who have to create and also test their own code.

Duration:

Four days

Course materials:

- Delegate handbook
- Delegate workbook
- Student datakey

Course workshop:

During the practical exercises, delegates will gain hands-on experience with development environments and Open Source testing tools and frameworks.

Why do we test?

- Measuring confidence in a product.
- How and why errors occur
- Testing as quality improvement.

Terminology

- The language of testing

The Swiss Cheese model

- Being effective in your testing.
- A layered approach to testing.

Test Harnesses

- The principles of test harnesses.
- Test harness tools

Test Specification

- Test Conditions
- Test Cases
- Test procedures

Systematic Black Box Testing

- The problems with ad-hoc test creation
- The Classification Tree Analysis method

State-based testing

- Finite state machines
- n-switch testing
- The Chinese Postman algorithm

Software integration

- What is software architecture?
- Integration techniques

Test Doubles

- Testing control-oriented objects
- Test stubs
- Mock objects

Testing legacy code

- Examining object internals
- Code seams
- Mocking legacy objects

Coverage Testing

- The principles of white-box testing
- Coverage metrics
- The problems of white-box testing

Static Analysis

- Types of static analysis tool
- Selecting a tool

Reviews

- Benefits of reviews
- The review process

Design for test

- The SOLID principles
- Dependency injection

Test-driven development

- Test-driven design
- The Russian Doll model

Continuous Integration

- CI servers
- Regression testing

Testing from use cases

- Behaviour-driven development
- Use case modelling
- Generating test cases

Test planning

- The test process

Creating a test plan

Feabhas Ltd - UK Office 15-17 Lotmead Business Park, Wanborough, Swindon, SN4 0UY, UK ·

+44 (0) 1793 792909 · info@feabhas.com · www.feabhas.com