

Embedded Software Testing with C

Course category	Testing
Training area	Quality
Course code	TC-401
Duration	4 days
Price exc VAT	£2100.00
Additional information	Available on request

Testing is a vital aspect of verifying the design of a software system. This is especially true in a real-time embedded environment, where the system may be subject to strict safety or reliability requirements.

Engineers require a good knowledge of how to test their code but in addition they must understand other techniques to find and eliminate errors earlier in the design lifecycle.

Supporting testing are the concepts of configuration and change management, vital skills required for managing industrial software.

A four day course covering testing software in a real-time embedded environment. The course is a mixture of lectures and practical exercises.

Course objectives:

- To understand the need for testing in an embedded environment
- To develop practical testing skills
- To give an understanding of the test process and how it fits with the development lifecycle
- To identify the different types of testing and where and when they should be used.
- To introduce good testing practices

Delegates will learn:

- How to perform static analysis of code
- The creation of black box test harnesses
- The creation of white box tests
- An understanding of software metrics
- How to create tests from Use Cases

Pre-requisites:

Attendees should have a practical working knowledge of the C language.

Who should attend:

This 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. This course is also highly beneficial for test engineers new to software testing in a real-time environment.

Duration:

- Four days

Course materials:

- Delegate handbook

Course workshop:

At least a third of the course is devoted to practical exercises. Delegates will gain hands-on experience developing test harnesses, static analysis and software metric analysis.

Introduction to Testing:

- Why test software?

Concepts of Testing:

- Why exhaustive testing is not practicable
- The concept of good test/bad tests
- Categories and levels of test
- Test cases and test procedures

Test Harnesses:

- Principles of test harnesses
- Creating a test harness
- Commercial test tools

Dynamic Testing:

- Structured basis testing
- Equivalence Partitions

- Boundary Value Analysis
- Problems with polymorphic code

Model-Based Testing:

- Synthesis- versus Analysis- models
- Generating tests from state diagrams

Coverage Testing:

- Why perform White-box tests?
- Coverage measures – Statement, Branch, Condition, Path and others
- Coverage testing tools

Code Reviews:

- Benefits of reviews
- Review process
- Checklists

Static Analysis:

- Static analysis concepts
- The use of the compiler for static analysis
- The implications of ISO C standard Annex F
- Static analysis tools
- Coding standards

Metrics:

- Why gather metrics?
- Using metrics to manage and control testing
- Metrics for test

Software Integration:

- Importance of planning your integration.
- Top-down vs Bottom-up Integration
- Practical integration models

Testing from Use Cases:

- Introduction to use cases
- Calculating test cases
- Structured Basis testing for use cases
- Generating test cases from use cases

Regression Testing:

- Purpose of regression tests
- The build process

Configuration Management:

- Why is configuration management vital?
- Configuration items
- Version control
- Change Management
- CM tools

Test Management:

- The test process
- How the test process relates to the software V-model
- "Design by contract"
- Test-driven development
- Agile development processes

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

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