



SOFTWARE TESTING ESSENTIALS COURSE FOR DEVELOPERS

The Software Testing Essentials course is a one day introductory course explaining the fundamentals of software testing aimed at Software Development.

The course explains the importance of testing and reasons why we test software. It looks at how testing fits into the overall Development Lifecycle and what can be done to improve effectiveness and efficiency of testing within our organisation.

SOFTWARE TESTING ESSENTIALS FOR DEVELOPERS

Course Objectives

To provide delegates with an understanding of the principles of testing and explain different testing terminology. The course also provides an overview of the different testing techniques and tools applicable for the Development Department.

Who Will Benefit?

This 1-day course is appropriate for anyone new to testing or who wants a basic understanding of the concepts of testing. It will benefit developers, testers and members of the QA team.

Skills Gained

- Learn about the principles and importance of testing and how it can make a difference within your organisation
- To recognise 6 common types of bug that exist within our software and the strategies to use to attack and destroy them
- To gain an understanding of the various types of testing tools applicable for the development team and where they fit into the overall lifecycle
- To understand and apply some Black and White box testing techniques to assist the programmer in the testing of the system

Prerequisites

None.

Course Content

Testing Concepts

Why is testing necessary discusses how much project effort is typically spent testing and shows that although the purpose of testing is to assess the quality of the software, there can only be confidence in this assessment if the testing itself is of good quality. This section explains some of the principles of testing and why faults get into our software and some practical ways to reduce the fault levels.

The test process breaks down the testing process into a number of distinct activities to show what can be done to improve effectiveness and efficiency. It also describes the attributes of a good test case and explains a test process that focuses on identifying and designing good test cases. Also describes some options on how to document test cases.

The V life cycle model uses the V-Model software development life cycle to explain the relationship between testing and development

activities and describes the different types of testing activities, their purposes and benefits.

Levels of testing explains the main levels of testing including unit testing, integration testing, system testing, acceptance testing, system integration testing and maintenance testing. Also compares these with regression testing.

Behavioural Based Techniques

Types of bug explains that there are many different types of bug that we can find in our software. Each of these will require a different technique to find and destroy.

What is a testing technique? Explains what testing techniques are and the different types available. Also outlines a number of techniques and the differences between them. This section explains the advantages of using testing techniques.

Black Box, White Box and Coverage concepts The difference between Black and White Box testing, what coverage is and how it can be used to give an objective measure, in some part, of test 'thoroughness'. Also highlights the dangers of considering structural coverage alone.

Equivalence Partitioning and Boundary Value Analysis These two techniques are applicable to all levels and many types of testing. They are simple in concept and yet powerful in use and frequently form the foundation of effective testing.

State Transition Testing although particularly suited to state transition systems, this technique can be used effectively wherever a state graph or state chart can be used to model the behaviour of a system or subsystem.

Exploratory Testing this section explains what Exploratory Testing is and how it can be used during Development. It looks at how Exploratory Testing can be a powerful technique to complement the other systematic techniques.

Structural Based Techniques

Code Coverage Concepts These techniques can be used to give an indication of the thoroughness of testing in terms of structural coverage of the source code and to help identify new tests that complement existing ones. Techniques covered include statement, decision, branch, condition combination, LCSAJ and Path Testing.

Testing Tools that will assist Developers Explains the different types of testing tool that are best suited to developers and how the tools can help developers test more effectively and efficiently.

Booking & Information +44 (0)8702 406172
 courses@grove.co.uk
 www.grove.co.uk

