



## SOFTWARE TESTING ESSENTIALS COURSE FOR UAT

The Software Testing Essentials course is a one day introductory course explaining the fundamentals of software testing. 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 COURSE FOR UAT

## 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, tools and how to manage the test process. It also emphasises the importance of User Acceptance Testing (UAT) and how UAT adds value to the project.

## 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 user acceptance testers, project managers, users, business analysts, 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 that are appropriate for UAT
- To understand and apply a powerful monitoring technique known as S-Curves and how to use them during UAT
- To understand the role of UAT within the whole lifecycle and its importance
- To be able to measure the value of UAT testing using a simple yet powerful measurement technique called DDP (Defect Detection Percentage)

## Prerequisites

None.

## Course Content

### Principles of testing

**Importance of Testing** 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. Describes the uses of software testing to assess, achieve and preserve software quality and highlights the importance of prioritising the tests.

**Principles of Testing** explains that looking for errors is the most effective approach to testing and that in order to have justified confidence in the quality of software it is necessary to try to break it.

**Causes of Error** describes a number of common causes of error and, although they cannot be eliminated, shows ways in which their numbers can at least be reduced.

## Testing in the Lifecycle

**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.

## The importance of UAT

**Aims and objectives of UAT** UAT is considered to be the final stage of validation. This is the time to see that the system under test meets the users' needs and provides confidence in its use. This section looks at some of the aims and objectives of UAT – together with some misconceptions.

**Working with Development** In many organisations testing is an us-versus-them mentality. This can be particularly noticeable during UAT. Building relationships is key – and an understanding of key problems from both sides is required.

**Building a strong UAT team** Highlights the key components in building a strong UAT team. Working together to a pre-defined goal is essential. Having trust and commitment is vital. A look at some of the 'rights' and 'responsibilities' of a UAT team.

**Effectiveness of UAT** a simple but powerful measure of the value of UAT.

## Overview of Testing Techniques and Tools applicable for UAT

**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.

**Techniques applicable during UAT** Explains what testing techniques are and highlights the various techniques (systematic and non-systematic) that can be applied during UAT.

**UAT with and without tools.** Describes the different types of tool support that are available for testing throughout the software development life cycle and in-particular what is appropriate for UAT. This session will explain a number of pitfalls in trying to automate the execution process and having the balance between manual and automated testing.

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

