

Course ID

SWT

Course Duration

2 days

Course Title

Effective Software Testing

Related Courses

- Principles of Software Engineering (SWENG1, 2 days)
- Software Engineering: An Advanced Tutorial (SWENG2, 3 days)
- Software Project Management (SWPM, 2 days)

Aimed At

Information Technology (IT) practitioners, including testers, analysts, developers, and managers with current or anticipated near-future responsibilities for software testing.

Group Size

5-25

Prerequisites

None

Course in a Nutshell

The goal of software testing is to increase to an acceptable level the user's confidence that the software will behave correctly under all circumstances of interest. This course focuses on well-defined practical techniques for meeting this goal, ranging from unit testing of individual software components and integration testing of sets of interrelated components, to system and acceptance testing by the developer, acceptance testing by the user, and ultimately maintenance and regression testing after the software has been placed in production.

Customize It!

- *Are you a member of a user organization, interested in acceptance testing of software at your site? We can create a version of the course that addresses acceptance testing in detail.*
- *Similarly, if you are a member of a system test organization, we can orient the course more toward system testing methods.*
- *If you are a developer intent on learning how testing fits in an agile environment, we can structure the material to emphasize agile methods.*

Learn How To

- Decide when and where to use black box and white box testing techniques
- Design test cases
- Prepare test specifications and test plans
- Test effectively in the face of tight deadlines without significant increase in project risk
- Exploit configuration management
- Incorporate testing into agile projects

Course Outline

- Introduction
 - Testing goals
 - Definitions
 - White box and black box
 - Testing types
 - Reporting results
- Test design techniques
 - Boundary value analysis
 - Equivalence partitioning
 - Decision table design
- White box testing
 - Unit and integration testing
 - Cyclomatic complexity
 - Test coverage criteria
- Black box testing
 - System and acceptance testing
 - Risk-based testing
 - Exploiting use cases
 - Performance testing
 - Test coverage criteria
- Testing in the production phase
 - Regression testing
 - Role of configuration management
 - Adaptive, corrective, perfective maintenance
- Test management
 - Preparing the test plan
 - Use of automation
 - Testing facilities: Hardware and software
- Course summary
 - Lessons learned
 - Where to go from here
 - Discussion and evaluations

**How You Will
Learn**

- An instructor who is highly experienced with software engineering tools and techniques will present this course in interactive lecture format.
- Along with the lecture, we will employ quick exercises, extended workshops, and interesting group activities to reinforce and enrich the class.
- You will receive a printed Participant Handbook to help you remember and retain what you learned in class and apply it after you're back on your job.

Revised

April 13, 2008 f