Unit Testing (2 Days)

This course is designed for testers and software developers who want to learn how to test software at a detailed level. The process taught in this course can be applied to many different technologies and development environments.

The course covers both functional and structural testing, with numerous examples and templates. You will learn the terminology, process, and challenges of testing in the real world.

As a result of attending this seminar, you should have a good working knowledge of unit testing and what it takes to design and conduct an effective unit test of software, regardless of the technology.

Return on Investment

- Learn how to find software defects early in the development lifecycle before they become more costly and risky to fix.
- Understand how to design a unit test.
- Learn which tools can help you perform unit testing more effectively.
- Get developers involved in testing.
- Advance your career by reinforcing your testing expertise.

Who Will Benefit

- Software developers
- QA and Test Managers
- Test analysts
- Testers

The program requires no testing knowledge or experience. Extensive technical knowledge is not a pre-requisite, although a familiarity with coding concepts is helpful.

Course Topics

Module 1 - Introduction to Unit Testing

- What is Testing?
- The Economics of Testing
- The 1:10:100 Rule
- When Should Testing be Performed?
- Testing and Risk
- Key Testing Concepts

Module 2 - Unit Testing Prerequisites and Terminology

- Test Phases
- Unit, Integration, System and User Acceptance Testing
- Defects
- Requirements
- Verification and Validation
- Functional (Black box) testing
- Structural (White box) testing
- Independent Testing
- Developer Testing
- What Unit Testing Should Cover
- What Unit Testing Should Include
- Unit Test Approach
- Unit Test Techniques
- Unit Testing Responsibilities
- GUI Challenges and Considerations
- Legacy Challenges and Considerations
- Web Considerations and Challenges

Module 3 - Unit Testing Process

- Step 1 - Planning
    - Functional tests
    - Structural tests
- Step 2 - Define Tests
    - Boundary cases
    - Equivalence classes
    - Decision tables
    - Requirement-based cases
    - Error forcing
    - Structural test coverage measures - statement, branch, condition, multi-condition and path
    - Test scripts
    - Batch tests
- Step 3 - Create and Maintain Test Data
- Step 4 - Execute Tests
    - Drivers and stubs
    - Functional tests
    - Regression testing
    - Structural tests
- Step 5 - Check Results

    - Compare actual to expected results
    - Report defects
- Step 6 - Evaluate Results
    - Summarize
    - Evaluate
    - Recommend

Module 4 - Unit Test Tools

- What is a Test Tool?
- Automated Tools
- Manual vs. Automated Testing
- Categories of Test Tools
- Bounds Checkers
- Capture/Playback
- Code Coverage
- Tool Information Sources

Module 5 - Unit Test Management and Summary

- Test Management Considerations
- Maintenance Considerations
- Test Measurement
- Key Seminar Points