Introduction to QA and Testing (2 Days)

This course is designed for users, testers, developers and managers who want to learn how to assure the quality of the software they deliver.

The course lays a foundation in the principles of quality and quality assurance. Then, techniques are presented that can make QA an effective force in your organization. The course concludes with developing your own action plan for quality.

Introduction to QA and Testing will help you understand the larger picture of software quality and you will become more comfortable and confident in testing software applications at just about any level of detail: unit, integration, system, and user acceptance. You will emerge from this two-day session knowing how to develop test cases and test plans. You will learn what makes a quality requirement specification and how to measure software quality. You will also leave with a knowledge of how tools can help you perform testing.

Return on Investment

- Learn how to find costly and embarrassing problems before your customers find them.
- Dramatically improve the effectiveness of your software quality efforts by employing proven principles of quality management.
- Understand the role of QA in a software organization.
- Understand the key issues in testing software applications.
- Learn how to design tests that adequately cover requirements and business events.
- Get the most out of your existing investment in testing and how to leverage that investment.
- Advance your career by reinforcing your testing expertise.

Who Will Benefit

- QA managers
- QA analysts
- Test analysts
- Testers
- End-users
- Project Leaders and Project Managers

Course Topics

Module 1 - Concepts of Quality and Quality Assurance

- What is Quality?
- Why is Quality Important?
- Lessons from the Gurus of Quality
- Terminology
- Deming's 14 Points of Quality
- Two Views of Quality
- The Cost of Quality
- Who is Your Customer?
- Total Quality Management
- Quality Principles
- Productivity Pitfalls
- Steps to Operations Analysis and Improvements
- Deming Workbench Model
- What is the Value of QA?
- Computer System Risks
- Why Are Standards Important?
- Process Improvement Analysis
- Cause and Effect Analysis
- Pareto Analysis
- Flowcharting
- Brainstorming

Module 2 - Exploring the Basics of Testing

- The Top 10 Testing Problems
- Testing Terminology
- When Testing Occurs - The "V" Diagram
- Phases of Testing
- Functional Testing
- Structural Testing
- Economics of Testing
- Basic Testing Principles
- The Importance of Test Strategy and Planning
- How Much Time Should be Spent on Test Planning?
- Basic Test Planning Guidelines
- Tips for Test Planning
- What Must be in Place for Effective Testing?

Module 3 - Regression Testing

- What is Regression Testing?
- Example: No Regression Testing vs. Regression Testing
- The Regression Testing Process
- What's Needed?
- Regression Testing Issues
- How Much Regression Testing is Enough?
- Tips for Performing Regression Testing

Module 4 - Unit Testing

- What is Unit Testing?
- When is Unit Testing Performed?
- How Much Preparation is Necessary?
- Why Create a Unit Test Plan?
- Unit Test Case Design - Functional Tests
- Unit Test Case Design - Structural Tests
- The Unit Test Process
- How to Document Functional Unit Test Cases
- How to Design Structural Tests
- How to Document Unit Interfaces
- Unit Test Execution - Procedural Software
- Unit Test Execution - Event-driven Software
- Unit Test Execution - Case-developed Software
- Automated Methods for Unit Testing
- Unit Test Tools
- Manual Methods of Unit Testing
- Unit Test Defect Reporting
- Unit Test Summary Report

Module 5 - System Testing

- What is System Testing?
- System Test Planning
- Identifying System Test Objectives
- Identifying System Functions to Test
- Identifying Critical Requirements
- Identifying System Interfaces
- Writing System Test Scripts
- Writing System Test Cases
- Profiling for Performance Testing
- Building the System Test Matrix

- Identifying System Test Schedules and Resources
- Finalizing the System Test Plan
- A Representative System Test Plan Outline
- Automated Methods of System Testing
- Manual vs. Automated Testing
- Manual System Test Methods
- Defect Reporting
- The Role of the Defect Administrator
- Evaluating the System Test

Module 6 - Test Tools

- What is a Test Tool?
- The Risks of Not Automating Testing
- The Risks of Automating Testing
- Where Do Tools Fit In?
- The Major Issues
- The Top 10 Test Tools Steps in Selecting a Test Tool
    - Interactive Test/Debug
    - Capture/Playback
    - File and Code Comparison
    - Stress and Load Testing
    - Defect Tracking
    - Test Data Generators
    - Test Management
    - Complexity Analyzers
    - Coverage Analyzers
    - Checklists
- Critical Success Factors

Module 7 - Building an Effective Test Team

- Roles and Responsibilities
- Independent Test Team Benefits and Problems
- Who Can Be On the Test Team?
- What to Look for in Team Members
- Training Team Members
- The Automated Test Organization
- Three Essentials for Test Team Leaders
- Managing Attitudes and Conflict
- User Attitudes
- Developer Attitudes

- Management Attitudes
- Critical Success Factors

Module 8 - Developing Quality Requirements

- The Importance of Requirements
- Quality Requirements
- Testable Requirements
- Identifying Needs
- Ambiguity in Requirements
- The Importance of Inflection
- The Importance of Word Meanings
- Word Games
- The Right People – Who Do We Involve?
- Brainstorming
- Prototyping
- Dealing with Change
- How to Control Change
- Requirements Reviews
- Special Considerations For Existing Software Requirements
- Exercise: Role Playing the Requirements Process