Basic Training in Software Testing (2 Days)

This is a practical hands-on seminar to cover the critical path of testing. Your instructor will be an experienced practitioner in the QA and testing field. You will learn the terminology, process, and challenges of testing in the real world. As a result of attending this workshop, you should have a good working knowledge of software testing and what it takes to design and conduct an effective test of software, regardless of the technology.

Basic Training in Software Testing will help you 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 session knowing how to develop test cases and test plans. You will also leave with the knowledge of how tools can help you perform testing and how to deal with special testing challenges.

Sometimes people feel intimidated by the technical aspects of software testing and lack the confidence they need to be credible test leaders in their organization. Learn the issues and processes for effectively testing software by attending this hands-on course.

Return on Investment

- Learn how to find costly and embarrassing problems before your customers find them.
- 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

- Test analysts
- Testers
- Developers
- Test and Developers

Course Topics

Module STBA – Surviving The Top 10 Challenges of Software Testing

- The Top 10 Challenges
- Solutions to Each Challenge

Module STBB – Terminology

- The Deming Workbench Model
- The Software Life Cycle
- Test Terminology
    - Software Testing
    - Quality Assurance
    - Quality Control
    - Verification
    - Validation
    - Defect
    - Requirement
    - Static Testing
    - Dynamic Testing
    - Black Box (Functional) Testing
    - White Box (Structural) Testing
    - When Testing Occurs - “V” Diagram
    - A Comparison of Project Methodologies and Where Testing is Performed
        - Waterfall
        - Rapid Application Development
        - Rational Unified Process
        - Spiral Model
        - Extreme Programming
        - Commercial Off-the-shelf Applications
    - Process Maturity and the Capability Maturity Model

Module STBC - The Economics of Testing

- Where Defects Originate
- Where Testing Resources are Used
- The Relative Cost of Fixing Defects

Module STBD - Basic Test Planning

- Basic Testing Principles
- The Basic Testing Process
- What is a Test Strategy?
- The Components and Format of a Test Strategy
- Identifying Critical Success Factors
- Defining Test Objectives
- The Components and Format of a System-level Test Plan
- Major Elements of a Test Plan
- How Much Time Should be Spent on Test Planning?
- Planning Time Guidelines

- Tips for Test Planning

Module STBE - The Test Planning Process

- Key Test Planning Components
- Major Test Planning Tasks
    - Task 1 - Develop Test Strategy
        - Critical Success Factors
    - Task 2 - Define Test Objectives
    - Task 3 - Identify Needed Test Resources
    - Task 4 - Plan Test Environment
    - Task 5 - Define Test Procedures
    - Task 6 - Identify Functions To Be Tested
    - Task 7 - Identify Interfaces With Other Systems or Components
    - Task 8 - Write Test Scripts
    - Task 9 - Define Test Cases
    - Task 10 - Design Test Data
    - Task 11 - Build Test Matrix
    - Task 12 - Determine Test Schedules
    - Task 13 - Assemble Information
    - Task 14 - Finalize the Plan

Module STBF - Test Case Development

- Key Test Planning Components
- Test Scripts
- Test Cases
- How to Document Test Cases
- Types of Test Cases
    - Functional Test Cases - Boundary Value Analysis
    - Functional Test Cases - Equivalence Cases
    - Functional Test Cases - Requirements-based Cases
    - Decision Tables
    - Orthogonal Arrays
    - Functional Test Cases - Error Guessing
    - Functional Test Cases - Business-oriented Cases
    - Transaction Threads Based on Business Scenarios
- Test/Cycle Matrix
    - What is a Test Cycle?
    - Why Use Test Cycles?
    - Traditional Test Cycle
- Use Cases
    - Use Case Components
- Structural Test Cases - Logic-based
- Structural Test Coverage Levels
- Structural Test Cases – Behavioral
- Regression Test Cases

Module IQAC – 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
- Example: Unit Test Overview
- 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
- Example: Unit Test Summary Report

Module IQAD – System Testing

- What is System Testing?
- System Test Planning
- Identifying System Test Objectives
- Identifying System Functions to Test
- Example: System Function Worksheet
- Identifying Critical Requirements
- Example: Critical Requirements Worksheet
- Identifying System Interfaces
- Example: System Interface Worksheet
- Profiling for Performance Testing
- Example: System Performance Profile Worksheet
- Building the System Test Matrix
- Example: System Test Matrix
- Identifying System Test Schedules and Resources
- Example: System Test Schedule
- Finalizing the System Test Plan
- A Representative System Test Plan Outline

Module IQAE – User Acceptance Testing

- User Acceptance Testing
- What is User Acceptance Testing?
- Methods of User Acceptance Testing
- The User’s Role in Testing
- An Overview of the Structured User Acceptance Testing Process

- Definitions
- The Structured User Acceptance Testing Process
  - Phase 1 - Plan Tests
  - Phase 2 - Execute Tests
  - Phase 3 - Evaluate Tests
- Setting Acceptance Test Objectives
- Organizational responsibilities
- Defining Acceptance Criteria
- Recording of Test Results
- Phase 2 - Execute Tests
  - Step 1 - Select tools
  - Step 2 - Train team members
  - Step 3 - Execute test plan
  - Step 4 - Track progress
  - Step 5 - Perform regression testing
  - Step 6 - Document test results
- Manual Methods
- Automated Methods
- Pitfalls to Avoid
- Managing Attitudes and Conflict
- Phase 3 - Evaluate Tests
  - Step 1 - Record Defects
  - Step 2 - Perform Data Reduction
  - Step 3 - Develop Findings and Recommendations
  - Step 4 - Formalize Test Report

Module UATF - Test Cases From Use Cases

- What is a Use Case?
- Use Case Components
- Use Case Model
- Sample Use Case
- Translating Use Cases into Test Cases
- Test Cases and Test Scripts
- Sample Test Cases Derived From Use Cases
- The Process

Module STBG - Test Execution Strategies

- Automated Testing
- Risks of Not Automating Testing
- Risks of Automating Testing
- Where Do Tools Fit In?
- The Major Issues
- “Top 10” Test Tools
- Critical Success Factors
- Test Execution - Manual Methods
- Test Folders
- Test/Function Matrix
- Building the Test Environment

- How to Create and Maintain Test Data
- Test Data Cycle
- The Process for Configuration/Release Management
- Pitfalls to Avoid

Module STBH - Regression Testing

- What is Regression Testing?
- No Regression Testing: Hidden Defects
- Regression Testing: No Hidden Defects
- Regression Testing - The Process
- Regression Testing - What’s Needed?
- Regression Testing Issues
- Regression Testing - How Much is Enough?
- Tips for Performing Regression Testing

Module IQAH – 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
- 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
- Steps in Selecting a Test Tool
- Critical Success Factors

How to Build a Robust Testing Baseline

- Regression Testing – Comparing to a Baseline
- How to Create and Maintain Baseline Test Data
- Test Baseline Data Cycle
- Regression Testing – Maintaining the Baseline
- How to select regression test cases based on risk and other factors
- How to perform regression testing in highly complex systems

Test Case Efficiency (pair-wise, classification trees, etc.)

- Why test efficiency is important
- The fault model behind pairwise testing

- The research behind pairwise testing
- How to create test cases using pairwise techniques
- Using optimized test cases for regression testing
- Experience: Reducing the number of test cases from thousands to less than 100

Module STBI - Test Evaluation and Reporting

- Prerequisites for Test Evaluation
- Test Evaluation and Reporting Process
- Test Reporting Attributes
- Types of Test Reporting
    - Defect reports
    - Status reports
    - Final report
- System Test Evaluation - Defect Reporting
    - Paper-based
    - Defect tracking tools
    - The role of the defect administrator
- The Defect Life Cycle
- Defect Tracking - Things You Need to Know
    - What is a defect?
    - How is a defect reported?
    - What are the defect categories?
    - How much detail should be reported?
    - What does the tester do when the defect is fixed?
- Sample Defect Categories
- Sample Defect Priorities
- Status Reporting
- Final Reporting
- Test Summary Report
- How Can This Data be Used?

Module STBJ – Test Project Management

- Keeping the Test on Track
- Facts About Change
- Dealing with Change During Testing
- Ways to Deal with Change
- The Sources of Change
- Six Tips for Dealing With Change During Testing
- Levels of Change Severity
- How to Estimate Test Time
    - Ratios
    - Comparison to Scope Baseline
    - Historical Models
    - Defect Prediction Models
    - Time Boxes
    - Testware
- How to Track Progress

Resources

- Checklists and Templates
- Glossary
- Bibliography

Deliverables

- Course notebook with slides, worksheets, checklists, complete examples and supporting text
- Certificate of course completion
- You will have the basic information needed to plan and execute a test of a software application.