

**Fundamentals of Software Testing**

**Course Number:** BA-114  
**Duration:** 2 days

**Overview**

This private, customizable Fundamentals of Software Testing training teaches attendees the business issues that drive the need for a fully functional testing process. This course walks participants through the testing process and gives them an understanding of the roles, processes, and deliverables associated with testing.

**Prerequisites**

Participants should have basic knowledge of the software life cycle.

**Materials**

All software testing training students receive comprehensive courseware.

**Software Needed on Each Student PC**

For in-person deliveries, attendees do not need computers for this course. We will provide full classroom setup instructions that will include seating in small groups, with supplies such as flipcharts, sticky notes, markers, and pens for the attendees and a projector and Internet connection for the instructor's laptop.

Online deliveries for this interactive training will use an online meeting platform (such as Zoom, WebEx, GoTo, or Teams) to have face-to-face contact online, including use of breakout rooms for group activities.

**Objectives**

* Develop an understanding of basic concepts associated with software testing
* Apply testing to the Software Development Lifecycle (SDLC)
* Recognize benefits of improved quality of deployed software using testing techniques
* Identify the key roles, activities, and deliverables that make up requirements-based testing
* Use a business use case to define scenarios for testing
* Create a test plan and write test cases with test data
* Understand the process for testing functional requirements
* Identify the challenges of testing vendor-supplied applications

**Outline**

* Introduction: Why is Testing Important?
  + Symptoms and sources of quality problems
  + Benefits of early inspections and reviews
  + The Quality Maturity Scale
  + The current state of testing
  + Challenges in improving quality
* The Testing Lifecycle
  + What is a testing lifecycle?
  + Iterative testing principles
  + Sample testing types
* Testing Types
  + Classifying testing types
  + System, Integration, Vendor Validation, Regression, Maintenance, etc.
* Software Testing
  + Typical problems we encounter with software
  + The Cost of Quality (CoQ) and the cost of finding defects too late
  + The four stages of software testing (Unit, Integration, System, User Acceptance Testing (UAT)
  + Testing best practices
* Understanding the Tester’s Terminology
  + What is requirements-based testing and what is the role of the tester?
  + Characteristics of a good tester
  + The tasks of testing
  + The documents
  + Common terms & definitions
    - The three testing techniques
    - Testing visibility (White Box, Gray Box, and Black Box testing)
* The Test Planning Process
  + The importance of test planning
  + What is a Test Plan?
  + The six steps for creating a Test Plan
  + General testing tips
* Test Coverage
  + What is test coverage?
  + Using a Requirements Traceability Matrix
  + Set the testing scope – what to test and what not to test
  + The Test Coverage Matrix
* Creating and Executing the Test Cases
  + What are the goals of testing?
  + What is a Test Case?
  + How do Test Plans and Test Cases relate?
  + The four steps for creating a Test Case
  + Preparing, running, and documenting the Tests
  + General testing tips and techniques
* Verifying the Test Results
  + Documenting test results
  + What is a defect?
  + How to log a defect?
  + The “bug” lifecycle
  + Writing a good problem description
  + Taking screen snapshots
  + 10 tips to avoid writing bad defect reports
  + Signing-off on the product
* Testing Vendor-Supplied Applications
  + Challenges of testing vendor-supplied applications
  + Challenges to the business
  + Eight steps for testing vendor-supplied applications
  + An industry case study for testing vendor products
* Conclusion