



Agile Software Testing

This course covers the concepts, practices and implementation of agile software testing. After knowing the standard artifacts needed in testing software projects, participants will learn how to strategize, plan, design and execute tests in short development iterations and with incomplete specifications.

Training Objectives

At the end of the course, the participants will be able to:

1. Apply the concepts of Agile development.
2. Test effectively in agile project environments.
3. Describe the agile life cycle.
4. Describe the agile development practices.
5. Perform exploratory testing.
6. Strategize testing in an agile environment.

Duration 2 days

Topics

- I. Understanding the Agile Mindset
 - a) The Agile Manifesto
- II. Agile Values
- III. Agile Principles
- IV. Scrum Roles
- V. Scrum Techniques
- VI. Commonly Used Agile Artifacts
- VII. Agile Testing Strategies
 - a) The goals of testing in an Agile environment
 - b) The Role of Agile Testers
 - c) Agile Testers as part of the project team
 - d) Testing in an Agile Environment
- VIII. Agile Techniques & Concepts
 - a) Refactoring
 - b) Test-driven Development (TDD)
 - c) Agile Model-driven Development (AMDD)
- IX. Testing in the Agile SDLC
- X. Testing within an Iteration
- XI. Agile Testing Quadrants
- XII. Testing during Feature Development
 - a) Participation in design reviews
 - b) Sequencing of implementation tasks
 - c) Developer testing versus user testing
 - d) Communication of code changes
 - e) Feature unit testing strategies
- XIII. Exploratory Testing
 - a) Dealing with incomplete specs
 - b) Session-Based Exploratory Testing
 - c) Ad-hoc vs. Exploratory Testing
 - d) Exploratory test objectives: definition and prioritization
 - e) What is Chartering?
 - f) Elements of a Charter
 - g) Recommendations for Good Chartering
 - h) Mnemonics for Risk and Coverage Ideas
 - i) Risk and Coverage Knowledge Base