Agile Testing (ICP-TST) Training

Duration: **3 Day**

BILGINC IT Academy
info@bilginc.com
bilginc.com

Overview

Transition sound traditional test practices into an Agile development environment. Just enough, just in time, with responsive high levels of quality.

Building on traditional risk-based test practice we are now challenged by incremental delivery. To address but not inhibit Agility, we examine concepts like pair testing and strengthening relationships in a team setting to build a sense of common purpose. Working in parallel with developers eliminates hand-offs and late stage testing, and enables incremental confirmed component delivery. This program builds on what we know and adapts to what we need ... Agile delivery. Agile testing opens new areas of opportunity to build strength in product quality, process improvement and test confidence.

When implementing changes in methods and approaches, we may slide back into old test habits, or leave methods behind rather than transition from them. We may throw everything away and do a wholesale replacement without appreciating what really needs to change. Most testing practices work, but only within appropriate development contexts. This program will contrast traditional against Agile testing, highlighting what changes we must make to apply the best testing principles in the new role that we will play. Agile testing is not a separate service, but an integrated part of the development team. This 2-day course will introduce you to the Agile Testing Process, and explore Agile testing practices that you can immediately take from the classroom into the office with newfound confidence.

Upon successful completion of this course, attendees will be awarded with the ICAgile Certification in Agile Testing. The ICP-TST certification can only be earned by completing a live class.

Who Should Attend?

- Quality Analysts & Engineers
- Software Test Leads & Testers
- Software Quality & Testing Managers
- Software Project Managers
- Software Engineering Managers
- Business Analysts
- Software Development Managers
- IT Managers
- Programmers/Developers

Agile Testing (ICP-TST) Training

Duration: **3 Day**

BİLGİNÇ IT Academy
info@bilginc.com
bilginc.com

What You Will Learn

- Improve software quality delivered by your Agile development teams.
- Apply all four quadrants of software testing to ensure the effectiveness of both business facing and technology facing testing.
- Accelerate defect detection and correction and lower integration costs through continuous integration.
- “Build the product right” by applying techniques like Test-Driven Development (TDD)
- Capitalize on your investment in traditional testing methods by transitioning them to an Agile approach.
- Address product and technical risk using Risk-Based Testing to better focus time and resources.
- Enhance the skills of your testers and development team to effectively conduct adequate testing.
- Learn from defects so you can optimize your Agile development process.
- Achieve higher product confidence by using Pair Testing.
- Optimize the performance of your Agile teams through proper testing roles.

Outline

Part 1: Agile Testing Mindset

Much like Agile itself, many of the Agile testing techniques where espoused well before the Agile Manifesto was created. But Agile testing is much different from testing performed during traditional software development approaches. This topic anchors the ideas of Agile testing in earlier work, while also providing insight into the major differences between Agile testing and testing performed as part of traditional (phased-based) software development.

The 12 Principles of the Agile Manifesto establish guiding principles for not only the Agile movement but Agile testing as a discipline. The Agile mindset includes: Quality is not “owned” by a particular role in Agile; Testers become facilitators of the team’s quality efforts; Agile testing provides critical insights and feedback into the software process. This topic helps learners understand how the Agile Manifesto is realized within an Agile testing process and approach, and to adopt the requisite Agile mindset.

1. Overview of Agile Testing
    - Origins of Agile Testing
    - Agile Testing vs. Traditional Approaches
2. Mindset & Culture
    - Agile Testing Principles
    - Whole Team Approach
    - Building Quality In
    - Continuous Improvement and Feedback
    - Ingraining The Agile Testing Mindset (Hands-on Exercise)

Part 2: Testing Techniques

Testing activities can be broken into various categories (or Quadrants) of testing based on their purpose and value. Automated testing can be performed at various levels (the automation pyramid) within a software application, and appropriate testing techniques must be applied to each. This topic provides the learner with a sound understanding of the purpose of various categories of testing,

opportunities for automation, and testing techniques so they can be applied appropriately and at the right time within an Agile environment.

Developer testing of individual software units and associated components is critical to detecting implementation defects within the software. Unit and component tests are leveraged within TDD as well. This topic helps the learner to thoroughly understand the purpose and approach to successfully implementing unit and component testing on Agile projects and how testers support developer testing during development cycles.

Test-Driven Development (TDD) and its derivatives, Acceptance Test-Driven Development (ATDD), Behavior-Driven Development (BDD) and Spec by Example are techniques for assuring that Stories are implemented in a manner that satisfies the customers' needs. This topic helps the learner to thoroughly understand the purpose and approach to successfully implementing these techniques on Agile projects.
Testing of User Stories is critical to successful development of software within an Agile project. This testing is often performed using the techniques above but can be done in other ways as is appropriate or necessary. This topic enables the learner to thoroughly understand the various options for testing User Stories during software development; this is an extension to ATDD to include boundary conditions and other types of testing such as exploratory testing.

1. Categories of Testing
    - Agile Testing Quadrants of Categories
    - Automation Pyramid – Introduction
    - Testing Techniques
2. Collaborating with Developers
    - Unit and Component Testing
    - Pairing Between Developer and Tester
3. Example Driven Development
    - Acceptance Test-Driven Development (ATDD)
    - Behavior-Driven Development (BDD)
    - Spec by Example
4. Feature and Story Testing
    - User Story Testing
    - Feature Testing
    - Exploratory Testing
    - Non-Functional Testing

Part 3: Agile Testing Process

Testing during an Agile project is team-oriented, so it is common for every member of the team to provide some level of testing support. This includes the Product Owner and other Business Representatives and the programmers in addition to the testers. This topic provides the learner with an understanding that within an Agile project, the entire project team is responsible for testing activities, with a specific focus on how this affects specific roles.

Lightweight planning and documentation are typical of Agile projects. The best Agile projects do just enough planning and documenting to support project activities and the needs of the end users. This topic provides the learner with an understanding of how lightweight test strategy and planning is performed on Agile projects, and how decisions are made regarding what type of test documentation, records, metrics, and reports are needed and how much is enough.

Agile Testing (ICP-TST) Training

Duration: **3 Day**

BİLGİNÇ IT Academy
info@bilginc.com
bilginc.com

Agile projects employ a variety of techniques around the delivery of the product, including Time-boxed Iterations and Continuous Delivery, that all have a very strong focus on testing. This topic helps the learner to appreciate all of the various ways that testing is used in the “End-Game” activities (which don’t just happen at the end of the Agile project!)

Multiple environments are often necessary to support testing activities during iterations and the release process. This topic provides the learner with an understanding of the typical test environments that must be set up and maintained to support testing activities during iterations and releases and how the product must be managed as it progresses through those environments.

Distributed teams are a fact of life in most organizations and must be dealt with to make Agile testing initiatives successful. This topic provides the learner with an understanding of how communication and coordination of testing activities can be most effective on distributed teams.

1. Roles and Responsibilities
    - Team-Based Testing Approach
    - Typical Business Representative Role in Testing
    - Typical Programmer Role in Testing
    - Typical Tester Role in Testing
    - Role of Test Managers in Agile
2. Test Strategy and Planning
    - Different Strategies Based on Levels of Precision
    - During Iteration Planning/Kickoff
    - Lightweight Test Plan Documentation
    - Defect Tracking and Management
    - Results Reporting
    - Test Metrics
    - Regression Tests
3. Successful Delivery
    - Time-Boxed Delivery
    - Continuous Delivery
    - Post-Development Test Cycles
    - Iteration Wrap-Up
    - Definition of a Release/End Game
    - User Acceptance Test (UAT)
    - System-Wide and Cross-Team Testing
    - Post-Release Testing
    - Documentation for Regulatory Requirements
4. Test Environments and Infrastructure
    - Typical Environments for Test
    - Build Pipeline
    - Automated Builds
    - Testing the Proper Build
    - Test Data Management
5. Working on Distributed Teams
    - Distributed Team Communication
    - Distributed Team Coordination