

Agile Quality Assurance and Testing

Course Overview:

Agile development processes place both added importance as well as special demands on your software quality assurance (SQA) practices. As an integrated part of the agile team, testers participate in the full life-cycle from requirements through release. While many of the principles and practices of software quality assurance apply, an agile approach requires some new ways of viewing testing activities in the development process.

Through a combination of lecture, group discussions and hands-on exercises, this course provides your team with the knowledge and expertise needed to integrate QA as an effective, high-performance part of your agile process.

Who Should Attend?

This course is designed for agile team members who will be adopting agile processes and working as part of an agile development team, including software quality assurance managers, engineers, analysts and testers. A basic familiarity with software quality assurance is assumed.

In This Course, You Will Learn:

- How agile principles and practices apply to software quality assurance.
- How to integrate software quality assurance into the agile process cycles.
- The role of SQA personnel in an agile team, and the role of non-testing personnel in the agile testing efforts.
- Specific practices and techniques for agile testing.
- The role of tools and automation in agile testing.
- Strategies for addressing specific challenges encountered in agile testing.

Logistics and Pricing:

This course is available in a one-day seminar format. Due to the highly interactive nature of this course, course sizes must be limited.

Please contact Agile Logic for pricing and availability.

Course Content:

- A Brief Tour of Agile Processes and Practices
- The Contribution of Testing in Agile Processes
 - Conditions of Satisfaction
 - Tests as Unambiguous Specifications
 - Tests as Supporting Design Quality
 - Avoiding Quality Debt in Projects
- Testing in the Agile Process Cycles
 - Release Cycles
 - Iteration Cycles
 - Development Cycles
 - Iterative, Incremental Testing
- The Role of Testers on the Agile Team
 - Integrated “Whole Teams”
 - Full Lifecycle, Continuous Involvement
 - Testing as a Cooperative Effort
- Types of Testing in an Agile Environment
 - Engineering Testing (Unit and Integration)
 - User Functionality Testing
 - Acceptance and Capability Testing
 - Regression Testing
 - Exploratory Testing
 - Performance and Scalability Testing
 - Regulatory and Conformance Testing
- Configuration Management for Testing
 - Managing Test Plans, Scripts and Data
 - Environments and Promotional Strategies
- Testing Strategies for Architectural Components
 - Testing Databases and Data Access Tiers
 - Testing Service Oriented Architectures
 - Testing Business Logic and Workflows
 - Testing User Interfaces
- Test Metrics for Agile Processes
 - Why Failing Tests are Not an Option
 - Bug Discovery as a Measure of Quality Debt
- Test Automation and Tools
 - Why Automation is Not Optional
 - The Continuous Integration Cycle
 - Tools for Test Automation
 - Make Your Bug Tracking Tool Obsolete
- Test-Driven Development
 - Unit Test-Driven Development
 - Story Test-Driven Development
- Agile Testing for Legacy Systems
 - Bootstrapping Testing on Legacy Systems
 - Incremental Automation for Legacy Systems
- Addressing Challenges in Agile Testing
 - Not Enough Time to Continuously Test
 - Only Testers Can Test
 - Not All Testing is Verification and Validation
 - The Post-Development Regression Spiral
 - When Automating Tests Is Difficult
 - Root Causes of Agile Testing Challenges
 - Introducing Agile Testing to Teams