This 2-day course will introduce you to Agile testing methods, and explore their use so that you can immediately step from the classroom into the office with new found confidence. We will discuss the roles, methods, tools and technologies that can be relied upon to deliver speed and optimum flexibility. You will start to feel a new sense of flexibility, confidence and enthusiasm (maybe for the first time in your entire development career). This course will demonstrate how traditional test practices impede the speed of software delivery and how an Agile testing approach enhances delivery speed and improves overall quality. Learning the goals of Agile will help you transition, implement and monitor testing in an Agile testing environment.

- NOTE: Live Virtual Classroom course length is 3 days.

**Skills Gained**
- Understand the key differences between traditional and Agile testing practices
- Learn about the different quadrants of Agile testing and how they are used to support the team and critique the product
- Get exposed to the different levels of test automation and understand what the right mix is to accelerate testing
- Operate in a time constrained development cycle without losing testable value
- Capitalize on test development through use & reuse management
- Integrate team testing into Agile projects
- Engage stakeholders in quality trade-off decision-making
- Coach story card contributors in test case construction
- Gain exposure to automation support opportunities

**Who Can Benefit**
- 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

**Course Details**

*Agile principles add value to your organization.*

Building on traditional risk-based test practice we are now challenged by incremental delivery. To address, and not inhibit, we
examine the concept of pair testing. Strengthening relationships in a team setting, helps to build a sense of common purpose. Working in parallel eliminates hand-offs, late stage testing, and an opportunity for incremental confirmed component delivery. This program builds on what we know, and adapts to what we have… Agile delivery. Traditional methods become a barrier under new development methods and place further pressure on the test team to feel a sense of value compromise. Agile testing opens new areas of opportunity to build strength in product quality, process improvement and test confidence.

Learn to transition to Agile testing practices across the entire release cycle.
Despite changes in methods and approaches, we continue to slide back into old test habits. Often we leave methods behind, rather than transition from them. We throw everything away and do a wholesale replacement without putting to bed the reasons for change. Most testing works, but only within the development context that it was framed. The program will reflect traditional against Agile testing, what changes we must make and the role that we will play. Testing is not a service, but an integrated part of the development team.

Agile Testing
We will discuss the testing and it's role in software quality. Quality is the collective responsibility of the team from business analyst to developer to tester to customer. Traditional waterfall "over-the-wall testing" can be inefficient and frustrating. We will discuss typical challenges and pitfalls in this traditional approach and start to contrast how Agile Teams test differently.

- Poor Quality creates Drag
- Integrating the Team into an Agile Testing mindset
- Understand hard & soft constraints to adopting Agile Testing
- Getting the Customer to participate in Quality decisions

Testing Practices
The benefits that various types of testing provide to the team will be reviewed. Additional discussion will focus on the how and what to automate to shorten feedback cycles.

- Testing Quadrants
- Automation
- Unit Tests
- Integration Tests
- Acceptance Tests
- Functional Tests

Quality Practices
Understanding that getting feedback is as important as testing. We will discuss techniques that provide feedback on the quality of software and the effectiveness of the process.

- Pairing & Collaboration
- Inspections
- Reviews
- Demos

Unit Testing & Test Driven Development (TDD)
We will introduce Unit Testing and Test Driven Development. The benefits and process of TDD and how it can lead to better overall design and simplicity and engage the Developer in the test processing will be discussed.

- Unit Testing Principles
- Test First vs. Test Last
Continuous Integration
The concept of Continuous Integration and the CI Attitude will be discussed. Continuous Integration provides an essential role in maintaining a continuous process for providing feedback to the team.

- Discuss the Attitude of Continuous Integration
- Benefits & Practices of Continuous Integration
- Continuous Feedback
- Continuous Builds
- Continuous Inspections
- Continuous Testing
- Continuous Deployments

Acceptance Testing
The discipline of Acceptance Testing can lead to better collaboration with both the customer and the team. Automating Acceptance Tests can provide an invaluable tool to support the creation higher quality software and continue to support the team from story to story and sprint to sprint.

- Acceptance Criteria
- Writing Acceptance Tests
- Acceptance Test Driven Development
- Automating Acceptance Tests
- Behavior Driven Development

Functional Testing Web Applications & Web Services
As we develop a functioning application we can perform higher-level and coarser grained functional tests. Functional testing software, web applications and web services will be explored.

- Functional Testing Applications
- Testing Web Applications
- Testing Web Services

Hands-on Critiquing the Product
Everything can't be automated, nor should it. We will discuss manual technique that will help us critique the product and provide valuable feedback. We will discuss when and how these testing techniques should be used effectively.

- Exploratory Testing
- Scenario Testing
- Usability Testing
- User Acceptance Testing
Using Tools to Test Complexity and Critique the Product
Tools can be used to testing complex, critical attributes of the software. We will discuss when and tools should be used to test the complex, critical qualities of software.

- Performance & Load Testing
- "ility" Testing
- Security Testing

High-Speed Testing Techniques
We’ll introduce some techniques that can speed the testing process and provide faster feedback to the team and customer.

- Risk Based Testing
- Pairwise Testing
- Pareto Technique

Iterating to Testing Agility
How do we ever get there? We will discuss pragmatic techniques to iterate your team and organization to Testing Agility. We will discuss and craft a roadmap for your team and organization based off the practices and techniques discussed.

- Prioritize regularly
- Realize Constraints
- Challenge Constraints
- Keep moving forward
- Automate, Automate, Automate
- Roadmap & Planning

Schedule (as of 4 )

<table>
<thead>
<tr>
<th>Date</th>
<th>Location</th>
<th>Enroll</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nov 6, 2019 – Nov 7, 2019</td>
<td>Kirkland</td>
<td></td>
</tr>
<tr>
<td>Nov 13, 2019 – Nov 15, 2019</td>
<td>Virtual</td>
<td></td>
</tr>
<tr>
<td>Dec 11, 2019 – Dec 12, 2019</td>
<td>Herndon</td>
<td></td>
</tr>
<tr>
<td>Dec 11, 2019 – Dec 13, 2019</td>
<td>Virtual</td>
<td></td>
</tr>
<tr>
<td>Jan 29, 2020 – Jan 31, 2020</td>
<td>Virtual</td>
<td></td>
</tr>
<tr>
<td>Feb 26, 2020 – Feb 28, 2020</td>
<td>Virtual</td>
<td></td>
</tr>
<tr>
<td>Mar 25, 2020 – Mar 27, 2020</td>
<td>Virtual</td>
<td></td>
</tr>
</tbody>
</table>