Professional Software Testing Using Visual Studio 2019

- Learn practical steps to reduce quality issues and meet release dates and budgets
- Shift your testing activities left to find bugs sooner
- Improve collaboration between the developers and testers
- Get hands-on experience with Visual Studio 2019, Azure DevOps Services, and common marketplace tools.

Through a combination of lecture, demonstrations, and team-based exercises, learn to deliver high-quality increments of software on regular iterations using the tools found in Visual Studio, Azure DevOps Services, and the community marketplace. This three-day course provides students with practical software testing techniques and technical skills, as well as exposure to Test-Driven Development (TDD), Acceptance Test-Driven Development (ATDD), Test Impact Analysis, Continuous Integration (CI), and more.

Understand and practice development testing acceptance testing, and exploratory testing. Create automated acceptance tests in Visual Studio, use SpecFlow to automate acceptance testing, and learn to use Microsoft Test Runner. Learn to plan and track work, manage test cases, and more utilizing Azure DevOps Services and associated tools.

Who Should Attend
This course is appropriate for all members of a software development team, especially those performing testing activities. This course also provides value for non-testers (developers, designers, managers, etc.) who want a better understanding of what agile software testing involves.

Course Outline

Agile Software Testing
Overview of agile software development
The agile tester and agile testing practices
Different types of testing
Introduction to Azure DevOps Services
Agile requirements and acceptance criteria
Creating, organizing, and managing a backlog

Planning and Tracking Quality
Defining quality software
Introduction to Azure Boards
Forecasting and planning a sprint
Introduction to Azure Test Plans
Organizing testing using test plans and suites
Creating and managing test cases
Leveraging parameters and shared steps
Importing and exporting test artifacts
Triaging and reporting bugs

Exploratory Tests
Introduction to exploratory tests
Using the Microsoft Test & Feedback extension
Connected mode vs. standalone mode
Exploring work items
Capturing rich data during an exploratory session
Exploratory testing “tours”
Requesting and providing stakeholder feedback

Build and Release Testing
Introduction to Azure Pipelines
Automated builds using build pipelines
Running automated tests in the pipeline
Practicing Continuous Integration (CI)
Leveraging Test Impact Analysis
Automated releases using release pipelines
Creating, deploying, and testing a release
Viewing and managing a deployment

Development Tests
Introduction to development tests
Unit testing in Visual Studio
Data-driven unit tests
Analyzing code coverage
Practicing Test-Driven Development (TDD)
Concurrent testing (Live Unit Testing and NCrunch)

Acceptance Tests
Introduction to acceptance tests
Acceptance criteria and definition of “done”
Acceptance Test-Driven Development (ATDD)
Using SpecFlow to automate acceptance testing
Using Selenium for web UI testing
Using Appium for desktop UI testing
Manually testing web and desktop applications
Performance testing and load testing

Reporting
Agile metrics that matter
Configuring alerts and notifications
Using the Microsoft Analytics extension
Ad-hoc reporting using Excel and Power BI
Querying data using the REST API

Delivering Quality Software
Understanding and avoiding technical debt
Detecting and measuring technical debt
Defining and obeying a definition of "done"
Overcoming dysfunctional team behaviors
Becoming a high-performance team
Case studies