

## Professional Software Testing Using Visual Studio 2019

---

### Course Summary

#### Description

This course will introduce you to the contemporary testing principles and practices used by agile teams to deliver high-quality increments of software on regular iterations. Through a combination of lecture, demonstrations, and team-based exercises, students will experience how to do this by leveraging the tools found in Visual Studio, Azure DevOps Services, and the community marketplace.

#### Objectives

By the end of this course, students will be able to:

- Agile software development and testing
- The role of the agile tester
- Developer and tester collaboration
- Agile software requirements
- Introduction to Azure DevOps Services
- Using Azure Boards to plan and track work
- Creating, managing, and refining a product backlog
- Defining and planning for quality software
- Using Azure Test Plans for test case management
- Creating and managing test plans
- Organizing test cases into test suites
- Test configurations and configuration variables
- Creating and managing test cases
- Creating parameterized test cases
- Leveraging shared steps
- Importing and exporting test artifacts
- Triaging and reporting bugs
- Extending Azure Test Plans
- Introduction to development tests
- Writing and running unit tests
- Data-driven unit tests
- Analyzing code coverage
- Customizing code coverage
- Test Explorer, CodeLens, and other tools
- Practicing Test-Driven Development (TDD)
- Concurrent testing (Live Unit Testing and NCrunch)
- Introduction to acceptance tests
- Acceptance criteria and definition of “done”
- Acceptance Test-Driven Development (ATDD)
- Creating automated acceptance tests in Visual Studio
- Using SpecFlow to automate acceptance testing
- Using Microsoft Test Runner
- Testing web and desktop applications
- Capturing screenshots and video while testing
- Viewing and charting test run results
- Using Selenium for automated web UI testing
- Using Appium for automated desktop UI testing
- Performance and load testing
- Introduction to exploratory testing
- Using the Microsoft Test & Feedback extension
- Creating a work item during a testing session
- Exploratory testing tours
- Requesting and providing stakeholder feedback
- Introduction to Azure Pipelines
- Building, testing, & releasing code using Azure Pipelines
- Hosted vs. on-premises agents
- Running automated tests in the pipeline
- Practicing Continuous Integration (CI)
- Improving performance with Test Impact Analysis
- Agile metrics vs. traditional project metrics

## Professional Software Testing Using Visual Studio 2019

---

### Course Summary (cont.)

#### Objective (Cont.)

- Configuring project alerts and notifications
- Using Excel for reporting and charting
- Using the Analytics Service and related widgets
- Using Power BI and the REST API for reporting
- Understanding and avoiding technical debt
- Becoming a high-performance agile development team

#### Topics

- Agile Software Testing
- Planning and Tracking Quality
- Development Tests
- Acceptance Tests
- Exploratory Tests
- Build and Release Testing
- Reporting
- Delivering Quality Software

#### Audience

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.

You should take this class if any of these issues sound familiar:

- Release dates and budgets are missed due to low quality and bugs
- Testing activities are performed at the end of the sprint/iteration or release
- No collective ownership or collaboration exists between the developers and testers
- The team tests the wrong things at the wrong time
- No automated tests, no regression tests, and no idea of the quality of your software

#### Prerequisites

There are no prerequisites for this course.

#### Duration

Three Days

## Professional Software Testing Using Visual Studio 2019

---

### Course Outline

#### I. *Agile Software Testing*

- A. Overview of agile software development
- B. The agile tester and agile testing practices
- C. Different types of testing
- D. Introduction to Azure DevOps Services
- E. Agile requirements and acceptance criteria
- F. Creating, organizing, and managing a backlog

#### II. *Planning and Tracking Quality*

- A. Defining quality software
- B. Introduction to Azure Boards
- C. Forecasting and planning a sprint
- D. Introduction to Azure Test Plans
- E. Organizing testing using test plans and suites
- F. Creating and managing test cases
- G. Leveraging parameters and shared steps
- H. Importing and exporting test artifacts
- I. Triaging and reporting bugs

#### III. *Development Tests*

- A. Introduction to development tests
- B. Unit testing in Visual Studio
- C. Data-driven unit tests
- D. Analyzing code coverage
- E. Practicing Test-Driven Development (TDD)
- F. Concurrent testing (Live Unit Testing and NCrunch)

#### IV. *Acceptance Tests*

- A. Introduction to acceptance tests
- B. Acceptance criteria and definition of "done"
- C. Acceptance Test-Driven Development (ATDD)
- D. Using SpecFlow to automate acceptance testing
- E. Using Selenium for web UI testing
- F. Using Appium for desktop UI testing
- G. Manually testing web and desktop applications
- H. Performance testing and load testing

#### V. *Exploratory Tests*

- A. Introduction to exploratory tests
- B. Using the Microsoft Test & Feedback extension
- C. Connected mode vs. standalone mode
- D. Exploring work items
- E. Capturing rich data during an exploratory session
- F. Exploratory testing "tours"
- G. Requesting and providing stakeholder feedback

#### VI. *Build and Release Testing*

- A. Introduction to Azure Pipelines
- B. Automated builds using build pipelines
- C. Running automated tests in the pipeline
- D. Practicing Continuous Integration (CI)
- E. Leveraging Test Impact Analysis
- F. Automated releases using release pipelines
- G. Creating, deploying, and testing a release
- H. Viewing and managing a deployment

#### VII. *Reporting*

- A. Agile metrics that matter
- B. Configuring alerts and notifications
- C. Using the Microsoft Analytics extension
- D. Ad-hoc reporting using Excel and Power BI
- E. Querying data using the REST API

#### VIII. *Delivering Quality Software*

- A. Understanding and avoiding technical debt
- B. Detecting and measuring technical debt
- C. Defining and obeying a definition of "done"
- D. Overcoming dysfunctional team behaviors
- E. Becoming a high-performance team
- F. Case studies