Professional Software Testing Using Visual Studio 2019

PTVS2019 | 3 Days

This three-day 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.

Course Objectives

At course completion, attendees will have had exposure 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
- ✓ 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

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.

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!

Accentient™

Professional Software Testing Using Visual Studio 2019

PTVS2019 | 3 Days

Modules

Module 1: 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

Module 2: 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

Module 3: 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)

Module 4: 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

Module 5: 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

Module 6: 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

Module 7: 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

Module 8: 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

Course Designer

This course was designed by Richard Hundhausen, a Microsoft Developer Technologies MVP, Professional Scrum Trainer, co-creator of the Nexus Scaled Professional Scrum framework, and an experienced software developer and trainer. To see other developer courses, visit www.accentient.com.

Accentient™