## Course duration

- 3 days

## Course Benefits

- Visual Studio and Azure DevOps ALM tools
- Editions, capabilities, and version compatibilities
- Azure DevOps Server vs. Azure DevOps Services
- Administrator types and related tools (Team Explorer)
- Planning/creating team projects and collections
- Configuring services, security, teams, areas, iterations
- Introduction to Azure Boards
- Selecting an appropriate work item process
- Work item types and work item characteristics
- Using the agile tools (backlogs/boards) to visualize work
- Creating and refining a product backlog
- Tagging, finding, querying, and removing work items
- Using hierarchical (epics and features) backlogs
- Planning and tracking work in a sprint
- Introduction to Azure Repos
- Basic and advanced Git workflows
- Basic and advanced TFVC workflows
- Working with Azure Repos from Visual Studio
- Associating work items to commits for traceability
- Collaborating as a team and improving productivity
- Pairing, swarming, and mobbing patterns of work
- Creating and maintaining a wiki
- Performing code reviews using pull requests
- Requesting and capturing stakeholder feedback
- Using the Test and Feedback browser extension
- Using Visual Studio Live Share to collaborate in real time
- Writing and executing .NET unit tests
- Using IntelliTest to generate unit tests
- Using Live Unit Testing to run only impacted tests
- Using FxCop Analyzers and code metrics
- Using code clone analysis to find duplicate code
- Using IntelliTrace to troubleshoot and diagnose
- Using Performance Profiler to find problems in code
- Introduction to Azure Test Plans
- Test Case Management using test plans, suites, cases
- Testing web and desktop applications
- Capturing screenshots and video while testing
- Viewing and charting test run results
- Creating automated acceptance tests in Visual Studio

- Using Selenium and Appium for automated UI testing
- Using JMeter for load testing applications
- Practicing exploratory testing by taking testing tours
- Introduction to Azure Pipelines
- Creating and using build and release pipelines
- Running automated tests in the pipeline
- Configuring on-premises agent for build/release
- Practicing Continuous Integration (CI) and Delivery (CD)
- 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 the REST API for reporting
- Relevant Visual Studio Marketplace extensions
- DevOps principles, challenges, and goals
- DevOps practices and related tools in Azure DevOps

## Course Outline

1. Introduction to Visual Studio ALM
    1. Application Lifecycle Management overview
    2. Visual Studio and Azure DevOps tools and features
    3. Azure DevOps Server vs. Azure DevOps Services
    4. Features and capabilities by edition and role
2. Team Projects
    1. The various administrator roles
    2. Team project collections and team projects
    3. Creating a team project collection and team project
    4. Configuring a team project (areas, iterations, etc.)
    5. Configuring teams and team membership
    6. Securing a team project
3. Planning and Managing Work
    1. Introduction to Azure Boards
    2. Selecting a work item process (e.g. Scrum)
    3. Creating a custom, inherited process
    4. Work item types, categories, and hierarchies
    5. Creating, tagging, finding, and managing work items
    6. Querying and charting work items
    7. Using the agile backlogs, boards, and task boards
    8. Using Excel to query and update work items
    9. Hierarchical backlogs (e.g. epics and features)
4. Version Control

    1. Introduction to Azure Repos
    2. Git version control system overview
    3. Basic and advanced Git workflows
    4. TFVC version control system overview
    5. Basic and advanced TFVC workflows
    6. Working with Azure Repos from Visual Studio
    7. Associating work items to commits for traceability
5. Collaborating as a Team
    1. Collaborating effectively as a team
    2. Improving team productivity
    3. Pairing, swarming, and mobbing patterns of work
    4. Creating and maintaining a wiki
    5. Using pull requests to perform code reviews
    6. Requesting and capturing stakeholder feedback
    7. Collaborating in real time with Visual Studio Live Share
6. Writing Quality Code
    1. Writing and running unit tests
    2. Using Visual Studio Test Explorer
    3. Leveraging parameterized unit tests
    4. Measuring code coverage while testing
    5. Using IntelliTest to generate unit tests
    6. Using Live Unit Testing to run impacted tests
    7. Test-Driven Development (TDD) overview
    8. Code analysis, code metrics, and code clone analysis
    9. Using application profiling and IntelliTrace
7. Testing the Application
    1. Introduction to Azure Test Plans
    2. Test case management (test plans, suites, cases)
    3. Manually testing web and desktop applications
    4. Automated acceptance testing in Visual Studio
    5. Testing through the UI using Selenium and Appium
    6. Load testing using JMeter
    7. Exploratory testing using Test & Feedback extension
8. Building and Releasing
    1. Introduction to Azure Pipelines
    2. Configuring and using build pipelines
    3. Running tests in the pipeline
    4. Practicing Continuous Integration (CI)
    5. Configuring and using release pipelines
    6. Practicing Continuous Delivery (CD)
9. Reporting
    1. Agile metrics vs. traditional metrics
    2. Configuring alerts and notifications
    3. Ad-hoc reporting/charting using Excel
    4. Using the Microsoft Analytics extension
    5. Querying data using the REST API
10. Improving DevOps

1. What is DevOps?
2. Principles, challenges, and goals
3. The Three Ways (flow, feedback, continual learning)
4. Achieving Continuous Delivery (CD)
5. Resources

## Class Materials

Each student will receive a comprehensive set of materials, including course notes and all the class examples.

Class Prerequisites

Experience in the following *is required* for this Azure DevOps Services class:

- Experience working on a software development team and be familiar.
- with that team's development processes, practices, and tools. Familiarity with agile practices and Scrum.
- Ability to read and understand C# .NET code (all source.
- code will be provided).
- Experience using Visual Studio 2015, 2017, or 2019.
- Ability to read and understand requirements.
- Understanding of Microsoft Windows basics.