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 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

Course Outline

1. Introduction to 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. Azure DevOps Projects
    1. The various administrator roles
    2. Project collections and projects
    3. Creating a project collection and project
    4. Configuring a project (areas, iterations, etc.)
    5. Configuring teams and team membership
    6. Securing a project
3. Azure Boards
    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. Azure Repos
   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. Azure Test Plans
   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. Azure Pipelines
   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 Analytics Service
      5. Using Power BI to query analytics
      6. 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:

- Familiarity with agile practices and Scrum.
- Ability to read and understand C# .NET code (all source code will be provided).
- Experience with Visual Studio.
- Ability to read and understand requirements.
- Understanding of Microsoft Windows basics.