

# Agile Software Development with Azure DevOps Training

## Course Outline

### **1. Application Lifecycle Management with Azure DevOps**

Overview of Agile methodologies

- Agile
- Scrum

### **2. Introducing Azure DevOps**

Surveying Azure DevOps features

- Team collaboration
- Process templates
- Version control
- Automated builds
- Integrated tests

Evaluating Azure DevOps editions

- Choosing Azure DevOps Server vs. Azure DevOps Express
- Comparing Azure DevOps Service vs. Azure DevOps Server

### **3. Administering Azure DevOps**

Installing Azure DevOps

- Planning an Azure DevOps installation
- Creating collections with the administrative console
- Managing project security and group membership with the Web control panel

Securing Azure DevOps projects

- Assigning permissions to security groups
- Adding users to teams and projects
- Restricting user rights
- Configuring user alerts

Creating projects for Azure DevOps

- Comparing the built-in Scrum, CMMI and Agile process templates
- Specifying source control
- Customising process templates

#### **4. Managing Work with Azure DevOps**

Mapping Azure DevOps and Agile process terms

- Defining epics and user stories as Azure DevOps work items
- Inputting features and backlog items
- Estimating and prioritising work items
- Breaking backlog items into tasks

Organising backlogs

- Scheduling start and end dates for releases and Sprints
- Assigning backlog items to iterations and team members
- Recording work remaining and status
- Monitoring team progress

#### **5. Maintaining Source Code with Version Control**

Collaborating on code with Team Foundation Version Control (TFVC) and Git

- Selecting centralised vs. distributed version control
- Synchronising local repositories with the master repository
- Viewing project history with changesets

Connecting to Azure DevOps from Visual Studio Team Explorer

- Checking out and checking in source code
- Merging changesets from multiple developers
- Branching for release management and maintenance
- Resolving multi-user conflicts
- Adding existing code to Azure DevOps source control

Integrating version control with client applications

- Plugging TFVC into Windows Explorer

## **6. Achieving Continuous Integration**

Creating, running and automating builds

- Defining and scheduling builds
- Managing releases and deployments
- Controlling build output

Crafting quality software through continuous testing

- Integrating unit testing into builds
- Leveraging Visual Studio code analysis and testing tools
- Creating TFS test scripts for manual testing
- Documenting and tracking bugs

## **7. Scheduling and Monitoring Team Progress**

Managing backlogs and querying Azure DevOps content

- Assigning work to iterations
- Querying and defining iteration backlogs
- Visualising work-in-progress with Kanban boards
- Monitoring progress with burndown charts