Application Lifecycle Management Using Visual Studio 2019

- Explore using Azure Boards to planning and tracking work in a sprint.
- Understand workflows and version control.
- Learn to use Visual Studio Live Share to collaborate in real time.
- Get an introduction to Azure Test Plans for use in test case management.

This three-day comprehensive course provides students with the knowledge and skills to effectively use the Application Lifecycle Management (ALM) tools found in Visual Studio and Azure DevOps Server (f.k.a. Team Foundation Server) to plan, track, design, develop, test, and deliver business value in the form of working software. The course demonstrates to developers, testers, product owners, project managers, architects, testers, and release managers the value of the various features and capabilities found throughout Visual Studio and Azure DevOps Server.

Who Should Attend

This course is intended for software development professionals who are involved in building applications with Visual Studio and Azure DevOps. Regardless of the student's role, he or she will be able to learn and get hands-on experience with all of the ALM features found in Visual Studio and Azure DevOps Server.

Prerequisites

Before attending this course, a student should have experience working on a software development team and be familiar with that team's development processes, practices, and tools. Additionally, students should:

- Have familiarity with agile practices and Scrum
- Be able to read and understand C# .NET code (all source code will be provided)
- Have used Visual Studio 2015, 2017, or 2019
- Be able to read and understand requirements
- Understand Microsoft Windows basics

Course Outline

Introduction to Visual Studio ALM

Application Lifecycle Management overview
Visual Studio and Azure DevOps tools and features
Azure DevOps Server vs. Azure DevOps Services
Features and capabilities by edition and role

Team Projects

The various administrator roles
Team project collections and team projects
Creating a team project collection and team project
Configuring a team project (areas, iterations, etc.)
Configuring teams and team membership
Securing a team project

Writing Quality Code

Writing and running unit tests
Using Visual Studio Test Explorer
Leveraging parameterized unit tests
Measuring code coverage while testing
Using IntelliTest to generate unit tests
Using Live Unit Testing to run impacted tests
Test-Driven Development (TDD) overview
Code analysis, code metrics, and code clone analysis
Using application profiling and IntelliTrace

Testing the Application

Introduction to Azure Test Plans
Test case management (test plans, suites, cases)

Planning and Managing Work
Introduction to Azure Boards
Selecting a work item process (e.g. Scrum)
Creating a custom, inherited process
Work item types, categories, and hierarchies
Creating, tagging, finding, and managing work items
Querying and charting work items
Using the agile backlogs, boards, and task boards
Using Excel to query and update work items
Hierarchical backlogs (e.g. epics and features)

Version Control
Introduction to Azure Repos
Git version control system overview
Basic and advanced Git workflows
TFVC version control system overview
Basic and advanced TFVC workflows
Working with Azure Repos from Visual Studio
Associating work items to commits for traceability

Collaborating as a Team
Collaborating effectively as a team
Improving team productivity
Pairing, swarming, and mobbing patterns of work
Creating and maintaining a wiki
Using pull requests to perform code reviews
Requesting and capturing stakeholder feedback
Collaborating in real time with Visual Studio Live Share

Manually testing web and desktop applications
Automated acceptance testing in Visual Studio
Testing through the UI using Selenium and Appium
Load testing using JMeter
Exploratory testing using Test & Feedback extension

Building and Releasing
Introduction to Azure Pipelines
Configuring and using build pipelines
Running tests in the pipeline
Practicing Continuous Integration (CI)
Configuring and using release pipelines
Practicing Continuous Delivery (CD)

Reporting
Agile metrics vs. traditional metrics
Configuring alerts and notifications
Ad-hoc reporting/charting using Excel
Using the Microsoft Analytics extension
Querying data using the REST API

Improving DevOps
What is DevOps?
Principles, challenges, and goals
The Three Ways (flow, feedback, continual learning)
Achieving Continuous Delivery (CD)
Resources