Course Outline

Assuring Quality Using Azure Test Plans

Course Summary

Description

This one-day course demonstrates how an agile team can configure and use Azure Pipelines to effectively build, test, and deploy software while practicing Continuous Integration (CI) and Continuous Delivery (CD). To maximize learning, students will work in teams, in a common Azure DevOps project, on a common case study.

Topics

- Continuous Integration
- Continuous Delivery
- Continuous Deployment

Audience

This course is appropriate for all members of a software development team, especially those who are actively involved in automating the building, testing, and deploying of software. This course will also provide value for individuals outside the development team (managers, IT, operations, and other stakeholders) who want hands-on exposure to the capabilities of Azure Pipelines. Ideally, attendees should have some experience with Azure DevOps. Understanding of C# and .NET 6.0 will be beneficial but is not required.

Duration

One day

Course Outline

Assuring Quality Using Azure Test Plans

Course Outline

I. CONTINUOUS INTEGRATION

A. Introduction to Azure Pipelines
B. Creating a build pipeline
C. Pipeline tasks, variables, and triggers
D. Agents (hosted vs. on-premises)
E. Configuring self-hosted agents
F. Creating a YAML-based pipeline
G. Running tests and computing code coverage
H. Using Coverlet to collect/report code coverage
I. Configuring CI using triggers
J. Practicing Continuous Integration
K. Configuring and using Test Impact Analysis
L. Exploring pipeline analytics
M. Continuous Integration+
N. Hands-on lab

II. CONTINUOUS DELIVERY

A. Creating a release pipeline
B. Pipeline stages, jobs, tasks, and targets
C. Multi-stage YAML pipelines
D. Configuring environments and resources
E. Creating, deploying, and managing a release
F. Configuring/practicing Continuous Delivery
G. Hands-on lab

III. CONTINUOUS DEPLOYMENT

A. Continuous Delivery vs. Continuous Deployment
B. Build-measure-learn explained
C. Using feature flags for CD and feedback
D. Hypothesis-Driven Development
E. Using LaunchDarkly for feature management
F. Creating and using feature flags
G. Experimenting in production, A/B testing
H. Hands-on lab