

Microsoft - End-to-End Testing for Azure-Native Applications

Download Whitepaper: Accelerate Your Modernization Efforts with a Cloud-Native Strategy
Get Your Free Copy Now

Course Number: WA3713

Duration: 2 days

Overview

Course Description

In this Azure course, participants learn how to deliver high-quality Azure-native applications by implementing testing through Visual Studio, Azure Test Plans, and Visual Studio App Center. This training is hands-on and guides learners through the design, development, and execution of automated tests. The course also explores the seamless integration of testing into continuous integration and delivery pipelines, fostering a DevOps culture for efficient and reliable software deployment. Participants delve into monitoring and analyzing test results using modern dashboards and tools to interpret metrics and make data-driven decisions for the continual enhancement of application quality. This comprehensive training teaches learners to navigate testing challenges specific to Azure environments, ensuring the success of their software development projects.

Skills Gained

- Understand different types of tests and their purposes
- Implement unit tests with modern frameworks

- Develop effective mock tests using popular mocking frameworks (e.g., Moq, NSubstitute)
- Create robust UI tests with modern tools (e.g., Selenium, Appium)
- Execute security tests and performance/load tests
- Automate tests and integrate them into CI/CD pipelines
- Monitor and analyze test results for continuous quality improvement

Prerequisites

- Familiarity with .NET development (coding experience is recommended)
- Basic understanding of software testing principles
- Some exposure to Azure environments is helpful

Audience

Course Details

Introduction to Azure-native Application Testing

- Understanding the Azure-native architecture
- Exploring Azure-native applications that require testing (e.g., Azure App Service, Azure Functions, Azure Kubernetes Service)
- The importance of testing in modern development cycles
- Overview of different types of tests (unit, integration, UI, security, performance, and load)
- Introduction to testing tools: Visual Studio, Azure Test Plans, and Visual Studio App Center

Unit Testing in Visual Studio

- Fundamentals of unit tests and their importance
- Test-driven Development (TDD) principles
- Exploring popular unit testing frameworks (MSTest, NUnit, xUnit)
- Writing and organizing tests into test suites
- Implementing setup and teardown methods
- Utilizing Test Explorer for test execution
- Enhancing unit testing with concurrent testing tools like Live Unit Testing and NCrunch

Data-driven Unit Testing

- Implementing data-driven tests using parameterized test methods
- Feeding test data from various sources (e.g., CSV files, Excel sheets)
- Validating multiple input scenarios efficiently

Mock Testing in Visual Studio

- Understanding the role of mock tests in isolating components
- Exploring popular mocking frameworks (e.g., Moq, NSubstitute)
- Creating mock objects for dependencies
- Simulating method responses and properties
- Mocking database interactions and external services

Code Coverage Analysis in Visual Studio

- Overview of code coverage analysis
- Configuring and running code coverage tools in Visual Studio
- Interpreting code coverage results to improve test quality

UI Testing with Modern Tools

- Overview of UI testing and its challenges
- Creating UI tests using modern frameworks (e.g., Selenium, Appium) integrated with Visual Studio
- Best practices for developing maintainable and robust UI tests
- Strategies for automating UI interactions and validations
- Utilizing cloud-based device testing options for real-world scenarios

Test Planning Using Azure Test Plans

- Introduction to Azure Test Plans within Azure DevOps
- Creating comprehensive test plans and organizing test suites
- Associating automated tests with test cases
- Integrating Visual Studio tests with Azure Test Plans
- Configuring release pipelines to run automated tests from test plans

Test Execution and Analysis in Azure Test Plans

- Running selected tests on demand directly from Azure Test Plans
- Viewing detailed test results, logs, and analytics within the Azure Test Plans environment
- Analyzing test outcomes to identify areas for improvement

Exploratory Testing and Feedback Management

- Overview of exploratory testing and its benefits
- Installing and configuring the Test & Feedback extension for Visual Studio and browsers

- Conducting exploratory testing sessions and capturing rich data
- Creating bugs, tasks, and test cases directly from exploratory testing sessions
- Managing and tracking feedback efficiently within Azure Test Plans

Performance and Load Testing in Azure

- Understanding the importance of performance and load testing
- Configuring and executing performance tests using Azure tools
- Conducting load testing with Visual Studio and Microsoft Azure Load Testing
- Monitoring application performance and resource utilization
- Interpreting performance test results for optimization

Security Testing in Azure

- Identifying and addressing security vulnerabilities in Azure-native applications
- Leveraging Microsoft Defender for Cloud (formerly Azure Security Center) for security testing
- Integrating security tests into the development lifecycle
- Best practices for securing Azure-native applications

Continuous Testing with Azure DevOps

- Implementing continuous integration and continuous testing strategies
- Integrating automated tests into Azure DevOps pipelines
- Monitoring test execution within CI/CD workflows
- Ensuring rapid feedback and quality assurance throughout the development cycle

Introduction to Visual Studio App Center

- Overview of Visual Studio App Center and its role in mobile and desktop testing
- Key features of Visual Studio App Center for build, test, and distribution
- Understanding the end-to-end testing process in App Center

Working with Visual Studio App Center

- Setting up a build pipeline for generating test-ready artifacts
- Automating UI testing with App Center Test
- Deploying applications for testing as part of the build and release workflow
- Running automated UI tests on real devices in the App Center cloud

Analyzing Test Results in App Center

- Examining test results, logs, and screenshots within Visual Studio App Center

- Leveraging App Center dashboards for efficient test result management
- Utilizing insights from test analytics to drive quality improvements