Benjamin Day Consulting, Inc.
17 Roberts St #2
Brookline, MA 02445
617-645-0188
http://benday.com

Testing with Visual Studio 2015 & Microsoft Test Manager 2015

Visual Studio 2015 and Team Foundation Server 2015 both have a major focus on testing and quality. For developers, there are the unit testing features that help you find problems before you check in and help you validate automated builds, Web Tests and Load Tests that help you validate and stress your web applications, and Coded UI Tests to help you test running applications. For QA Testers there are the features for performing QA tests – aka “manual testing” – on your applications. Not only do the new manual test features allow you to track the work of a QA person and help your QA staff to record bugs but you can also record and automate tests against your Windows and browser-based applications. As if that wasn’t enough, Team Foundation Server 2015 allows you to manage virtual machine configurations to help streamline and automate your testing activities.

If you are focused on software quality and testing, Visual Studio 2015, Team Foundation Server 2015, and Microsoft Test Manager can really help you out. In this course, you will learn the testing features you need to know including unit testing, Test Impact Analysis, Coded UI Tests, Web Performance Tests, Load Tests, Code Profiling, Test Case Management, Automated Builds, and Hyper-V Test Lab Management. Throughout the course you’ll hear about how you can use TFS to get a clear understanding of what’s happening on your project and identify quality issues before you ship. This session is aimed at both developers and QA testers.

What you’ll learn:

- Unit Testing & Test-Driven Development
  - What is a unit test?
  - What is Test-Driven Development (TDD)? Why is TDD important?
  - What is the TDD process?
  - Common (bogus) excuses for NOT doing Unit Testing & TDD
  - How do I sell my team on Unit Testing & TDD?
  - Create and write unit tests
  - Test types in Visual Studio 2015
  - What is Code Coverage and why do I care?
  - Use the NUnit Framework instead of MSTest from within Visual Studio
  - Continuous Testing
- Application Design for Unit Testability
  - Architect your application for testability
  - Testing user interfaces
  - Test non-public methods

    - Strategies for unit testing legacy code
    - Fakes, Mocks, & Stubs
        - Using the Visual Studio 2015 Fakes Framework
        - Using Mocks & Stubs to avoid The Huge Integration Test pitfall
        - Dealing with databases and test data in your unit tests
        - Mocking web services, back-end systems, and database calls
        - Use Mocks & Stubs to reach those hard to test cases
    - Design Patterns for testability: Repository, Adapter, N-Tier Architecture, Model-View-Presenter (MVP), Model-View-Controller (MVC), and Model-View-ViewModel (MVVM)
    - Best Practices for fixing bugs & defects using TDD
    - Refactor for testability
- Coded UI Testing
    - Unit Tests vs. Coded UI Tests
    - Structure of a Coded UI Test
    - Recorded Coded UI Tests
    - Customizing and Editing the UIMap
    - Checking UI values with Asserts
    - Application Under Test & Environment Variables
    - Coded UI vs. MTM Action Recordings
- QA Testing with Microsoft Test Manager & the TFS Test Hub
    - What is Microsoft Test Manager?
    - What is the TFS Test Hub?
    - Managing Test Suites, Requirements, and Test Cases
    - Use Action Recordings to minimize the tedium
    - Create actionable bugs using video and IntelliTrace
    - Fully automated test cases with Associated Automations
    - Test Plans & Automated Builds
    - Query-based vs. Requirements-based Test Suites
    - Shared Steps
    - Use Parameters in your Test Cases
- Bug / Defect Management with MTM
    - Test Settings
    - Video Recordings
    - Test Impact & IntelliTrace
    - Create a Bug with MTM & Team Foundation server
    - Exploratory Testing
    - Assigning work to multiple testers
    - Tracking Test Results & Progress
- Virtual Lab Management using Microsoft Test Manager
    - What is Lab Management?
    - Create a virtual test environment

    - Run QA tests and capture bugs in a virtual test environment
    - Lab Management Builds, automatic deployment, and Coded UI tests
- Web Performance Tests
    - What is a Web Performance Test (WPT)?
    - Organize your WPTs in preparation for Load Testing
    - Validation Rules
    - Extraction Rules
    - Parameterization & Test Context
    - Custom Validation & Extraction Rules
    - Data Sources & Data-driven WPTs
    - Conditional & Loop Steps
    - WPT Plug-ins
    - Coded vs. Recorded Web Performance Tests
    - WPT Security
- Load Tests
    - What is a Load Test?
    - Pieces of a Load Test
    - Load Test Scenarios
    - Test Mix Models
    - User Load Patterns
    - Counter Sets & Test Settings
    - SQL Server Tracing/Profiling from a Load Test
    - Custom Performance Counters in Load Tests
    - Coded UI Tests in a Load Test
    - Load Test Rigs
    - Load Test Controllers & Agents

Details

Technologies: Visual Studio 2015
Programming language: C#
Duration: 2 days, 9am to 5pm

Lab Machine Requirements

Operating System: Windows 8, Windows Server 2015
Memory: 8GB or more
Required applications: Visual Studio Ultimate 2015 with latest service packs applied
Other: Lab machines should have the Hyper-V role installed and be up to date with patches from Windows Update. Students should have administrator rights on their laptop.