

Software Testing using Visual Studio TFS 2015

Course ISI-1464 2 Days Instructor-led, Hands on

Introduction

Testing is one of the main phases in the software development lifecycle. Managing the test scenarios, test cases, defects, and linking each one of these is bit of a challenge without using any tools. For frequently changing businesses, it is essential to address testing requirements by matching the pace of the business. This can only be achieved through automation of the possible testing scenarios to reduce the turnaround time of testing.

Software Testing Using Visual Studio Team Foundation Server 2015 is a practical, hands-on, instructor-led course that provides a number of clear, step-by-step exercises, which will help you to take advantage of the new features and real power of Visual Studio 2015.

Course Objectives

- Software Testing Using Visual Studio 2015 starts with the basics of testing types and managing the tests using Test Explorer and a few other tools. Practical examples are added to help you understand the usage of various tools and features in a better way.
- Software Testing Using Visual Studio 2015 is written from a developer point of view and helps you take advantage of the unit testing features and customize the tests by generating the code and fine-tuning it as per your needs.
- The Test Manager is a standalone tool which is part of the Visual Studio tools family and takes care of test management. Software Testing Using Visual Studio 2015 covers the usage of the test plan, test suite, manual testing, and exploratory testing using Test Manger and managing these aspects using these tools.
- Software Testing Using Visual Studio 2015 also covers the recording of user actions and creating automated tests out of it. This book covers generating and adding parameters to the recording and replacing it at runtime, adding validation and extraction rules to the tests, running the tests using command line commands, looking at the details of test results, and configuring the running the load test out of web performance or unit tests.
- This course helps you to understand the complete testing features and how to make use of automation features as well. It will appeal to both developers and testers.

Prerequisites

Users must have knowledge in using Visual Studio and Team Foundation Server.

Course Outline

Module 1: Visual Studio 2015 Test Types

- Software testing in Visual Studio 2015
- Testing as part of software development life cycle
- Types of testing
- Unit testing
- Manual testing
- Exploratory testing
- Web performance tests
- Coded UI Test
- Load testing
- Ordered test
- Generic test
- Test management in Visual Studio 2015
- Introduction to testing tools
- Test Explorer
- Code coverage results
- Microsoft Test Manager
- Connecting to Team Project
- Test Plans, suites, and test cases
- Defining test cases
- Lab Center

Module 2: Test Plan, Test Suite, and Manual Testing

- Test Plan
- Test Suite and its types
- Static Test Suites
- Query-based Test Suites
- Requirement-based Test Suites
- Running manual tests
- Action recording
- Shared steps and action recording for shared steps
- Creating shared steps
- Action recording for shared steps
- Adding parameters to manual tests

Module 3: Automated Tests

- Coded UI tests from action recordings
- Files generated for coded UI test
- CodedUITest1.cs
- UIMap.Designer.cs

- UIMap.cs
- UiMap.uitest
- Data-driven coded UI test
- Adding controls and validation to coded UI test

Module 4: Unit Testing

- Creating unit tests
- Assert statements
- Types of Asserts
- Assert
- Assert.AreEqual
- Assert.AreNotEqual
- Assert.AreSame
- Assert.AreNotSame
- Assert.Fail
- Assert.Inconclusive
- Assert.IsTrue
- Assert.IsFalse
- Assert.IsNull
- Assert.IsNotNull
- Assert.IsInstanceOfType
- StringAsserts
- StringAssert.Contains
- StringAssert.Matches
- StringAssert.DoesNotMatch
- StringAssert.StartsWith
- StringAssert.EndsWith
- Downloading the example code
- CollectionAssert
- CollectionAssert.AllItemsAreNotNull
- CollectionAssert.AreEqual
- CollectionAssert.AreNotEquivalent
- CollectionAssert.AllItemsAreInstancesOfType
- CollectionAssert.IsSubsetOf
- CollectionAssert.IsNotSubsetOf
- CollectionAssert.AllItemsAreUnique
- CollectionAssert.Contains
- CollectionAssert.DoesNotContain
- CollectionAssert.AreEqual
- CollectionAssert.AreNotEqual
- AssertFailedException
- UnitTestAssertionException
- ExpectedExceptionAttribute



- Unit Tests and Generics
- Data-driven unit testing
- Unit Testing using Fakes
- Stubs
- Shims
- Difference between Stubs and Shims
- Code coverage unit test
- Blocks and lines
- Excluding elements

Module 5: Web Performance Test

- Creating the web performance test
- Recording a test
- Adding comments
- Cleaning the recorded tests
- Copying the requests
- Adding Loops
- Web performance test editor
- Web test properties
- Web test request properties
- Other request properties
- Form POST Parameters
- QueryString parameters
- Extraction rules
- Validation rules
- Transactions
- Conditional rules
- Toolbar properties
- Add data source
- Setting credentials
- Add recording
- Parameterize web server
- Context parameters
- Adding a web test plugin
- Debugging and running the web test
- Settings in the .testsettings file
- General
- Roles
- Data and Diagnostics
- Deployment
- Setup and Cleanup Scripts
- Hosts
- Test Timeouts

Contact ISInc for more information at 916.920.1700 or by visiting our website at <http://www.isinc.com>

Module 6: Advanced Web Testing

- Dynamic parameters in web testing
- Coded web test
- Generating code from a recorded test
- Transactions in coded tests
- Custom code
- Adding a comment
- Running the coded web test
- Debugging coded web test
- Custom rules
- Extraction rules
- Validation rules

Module 7: Load Testing

- Creating a Load Test
- Load Test Wizard
- Specifying a scenario
- Knowing about think time
- Defining the Load Pattern
- Constant load
- Step load
- Defining the Test Mix Model
- Test Mix Model based on total number of tests
- Test Mix Model based on number of virtual users
- Test Mix Model based on user pace
- Test Mix Model based on sequential test order
- Defining the Network Mix
- Defining browser mix
- Counter sets
- Run Settings
- Threshold rules
- Editing Load Tests
- Adding context parameters
- Storing results in central result store
- Running the Load Test
- Analyzing and exporting Test Results
- Graphical view
- Summary view
- Tables view
- Detail view
- Exporting to Microsoft Excel
- Using Test Controller and Test Agents

- Test Controller and Test Agent Configuration

Module 8: Ordered and Generic Tests

- Ordered tests
- Creating an ordered test
- Executing an ordered test
- Generic tests
- Creating a generic test
- The summary results file

Module 9: Managing and Configuring Tests

- Using Test settings
- The General option
- The Roles option
- Data and Diagnostics
- The Deployment section
- Setup and Cleanup Scripts
- The Hosts option
- The Test Timeouts option
- The Unit Test option
- Editing the Test Run configuration file
- Editing the deployment section
- The Web Test option
- Configuring unit tests using the .runsettings file

Module 10: The Command Line

- VSTest.Console utility
- Running tests using VSTest.Console
- The /Tests option
- The /ListTests option
- MSTest utility
- Running a test from the command line
- The /testcontainer option
- The /testmetadata option
- The /test option
- The /unique option
- The /noisolation option
- The /testsettings option
- The /resultsfile option
- The /noresults option
- The /nologo option
- The /detail option

- Publishing Test Results
- The /publish option
- The /publishbuild option
- The /flavour option
- The /platform option
- The /publishresultsfile option
- Step 1 – create/use existing Test Project
- Step 2 – running the test
- Step 3 – creating a build
- Step 4 – building the project
- Step 5 – publishing the result
- TCM command line utility
- Importing tests to a Test Plan
- Running tests in a Test Plan

Module 11: Working with Test Results

- Test Runs and Test Results
- Test as part of the Team Foundation Server build
- Building report and Test Result
- Creating a work item from the result

Module 12: Exploratory Testing and Reporting

- Exploratory testing
- Reports using Team Foundation Server
- Bug status report
- Test case readiness report
- Status on all iterations
- Other out-of-the-box reports
- Creating a report definition using Visual Studio 2012

Module 13: Test and Lab Center

- Connecting to Team Project
- Testing Center
- Testing Center – Plan tab
- Testing Center – Test tab
- Testing Center – Track tab
- Testing Center – Organize tab
- Lab Center
- Environments
- Deployed environments