

Technical Training Solutions



.NET Unit Testing with Visual Studio and NUnit

Technical
Course
Outline

(2 Day Course)

Course Summary

The goal of unit testing is to isolate individual units of code within a software system and verify that the individual parts work as expected. But this is easier said than done. This course not only covers the mechanics of writing and executing a unit test, but it dives much deeper; covering how to create quality tests that are repeatable, understandable, run quickly, and can stand up to changes in the code being tested. Testing methodologies, such as Test-Driven Development (TDD) are covered as well as tangential topics, such as “mock” and “stub” objects and how they are necessary to the unit test developer.

Several frameworks for testing and mocking of test related objects have emerged in recent years and this course will focus on the use of the very popular NUnit test framework, which is based on the very popular JUnit for Java. Developers will learn how to obtain, install and configure NUnit for integrated use within Visual Studio as well as begin to explore its rich API and different forms of syntax. Additionally, students will learn about “mock” and “stub” objects and what their role is in unit testing. The Moq Framework will be utilized for this portion of the course.

Intended Audience:

While this course does cover some generalized concepts and techniques, these concepts and techniques will be applied to .NET application development. As such, the audience for this course is current .NET application developers who have existing knowledge and experience in VB .NET or C#.

Prerequisites:

To ensure the best learning experience for all participants, the following pre-requisites *must be met* in order to participate in this course:

- At least 6 months prior experience with VB .NET or C#.

Course Contents:

Unit Tests Basics

- What is Unit Testing?
 - Unit Testing vs. Integration Testing
 - Regression Testing
 - The “System Under Test” (SUT)
 - The Test “Context”
- “Quality” Unit Tests
 - Definition & Characteristics
 - “Arrange-Act-Assert” (AAA)
 - Trustworthy Tests
 - Testing Only One Thing
 - Writing Readable Tests
 - Avoiding Logic In Tests
 - Test Project Organization & Naming Guidelines
 - What Not To Test
 - InternalsVisibleTo

Testing & Software Development

- Test-Driven Development & Test-Driven Design
- Red|Green|Refactor

Introducing NUnit

- The GUI Test Runner
 - Setting Up & Configuring the NUnit GUI
 - Enabling Visual Studio Debugging of NUnit Tests
- Compiler Attributes
 - TestFixture & Test
 - SetUp & TearDown
 - Ignore & Inconclusive
 - Description & Category
 - Values & Range

Introducing NUnit (continued)

- Assert
 - Classic vs. Constraint Syntax
 - The AssertionHelper Class
 - Multiple Asserts In One Test
 - Assert Messages
 - Pass, Fail, Ignore, Inconclusive
- Expected & Unexpected Exceptions In Tests
 - Correct Usage of Try/Catch in Tests
 - The AssertionException Class
 - Exception Assertions
 - The ExpectedException Attribute

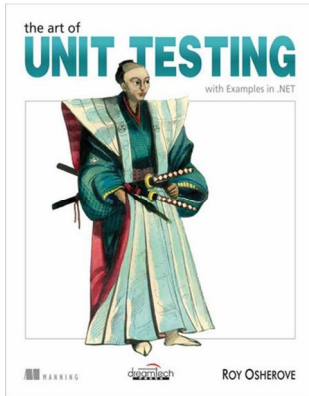
External Dependencies & Unit Tests

- What Is An External Dependency?
- “Seams”
- Better Software Design: Breaking of External Dependencies

Interaction Testing using Mocks & Stubs

- What Is Interaction Testing?
- Mocks vs. Stubs
 - Providing Fake State Information
 - Indirect Testing of Behavior via Side Effects
 - Manual Mocks & Stubs
- Introduction to Moq
 - Obtaining and Using Moq
 - Strict vs. Loose Mocks
 - Moq and AAA
 - Arranging Mocks and Stubs
 - Acting with Stub Data and Mock Behavior
 - Asserting Mock Behavior

Recommended Courseware:



The Art of Unit Testing: With Examples in .NET

by: Roy Osherove

- Paperback: 320 pages
- Publisher: Manning Publications; 1 edition (July 5, 2009)
- Language: English
- ISBN-10: 1933988274
- ISBN-13: 978-1933988276

Amazon.com link:

http://www.amazon.com/The-Art-Unit-Testing-Examples/dp/1933988274/ref=sr_1_1?ie=UTF8&qid=1361808487&sr=8-1&keywords=the+art+of+unit+testing#