The Learning House

427 S 4th Street #300

Louisville KY 40202

SOFTWARE GUILD

Unit Testing and TDD

.NET Cohort

Coding Bootcamp

SOFTWARE GUILD

Lesson Goals

- Become familiar with how a developer can use unit tests to create higher-quality code and safely change (re-factor) existing code
- Introduction to the nUnit unit test framework
- Learn how resharper interacts with unit tests in code

What is Developer Testing?

- As a software craftsman, we are expected to release high quality code.
  - It costs up to 500% more to fix bugs after release into production.
- In the old days (and still today in many shops), developers created test executables and stepped through debugging code to test.
  - This was brittle because the executable code would change to test different features.
  - Also, since code was changing in the test, some test cases would be lost, removed, or outright missed.

SOFTWARE GUILD

Why Unit Testing

- Higher quality / fewer defects
- Unit tests serve as living documentation of requirements
- Unit testing forces developers to think about isolation, encapsulation, and *loosely coupled* code
- Automatically creates *regression tests*

What is a Unit Test?

- A unit test is designed to confirm functionality of a component in a larger system.
- These tests are typically written by developers.
- The goal is to isolate individual parts of a system and assert they are working properly.

Organizational Resistance

The biggest resistance to unit testing is the perception that the amount of time it takes to write unit tests is at odds with deadlines.

However, since the amount of time and resources spent supporting code after creation dwarfs the amount of time spent on initial creation, making sure a proper test suite is created offers a payback much higher than initial creation.

Unit Testing Frameworks

- There are many unit testing frameworks out there. Some of the common ones in .NET are nUnit, xUnit, MS-Test (Visual Studio), and MBUnit.
  - We are going to use nUnit, which is one of the most established frameworks.
  - We will also look at MS-Test.
- Just about every language/platform has unit testing frameworks.

How To Unit Test

- Create a *Class Library* for your tests
  - Reference the unit testing framework
  - Reference the class library for your production code
  - Add a testing framework (ex: nUnit) via nuGet or use the built-in Visual Studio ones.
- Determine and document the assertions that define correctness.
  - A good test only tests one assertion at a time.
  - TDD (Test-Driven Development): write the test first, then the code, then refactor (red-green-refactor).
  - Test-After development is also viable.

SOFTWARE GUILD

String Calculator Kata

TIME TO WRITE SOME CODE