

# Zero to 60 Unit Testing

with Roy Osherove

---

## The Problem

Unit testing, done badly, can hurt your project more than it helps. In some cases, it may lead to project schedule delays, and unhappy developers due to huge maintenance problems. But when unit testing is done right, it can help the project and developers' confidence immensely.

The challenge is - how do you do it in a way that is helping, rather than hurting your project?

---

## Objective

In this workshop we will learn essential unit testing and test driven skills and techniques for writing unit tests in a readable, maintainable and trustworthy fashion, so that we can feel more confident in our code, make it simpler, and save ourselves time in debugging and maintaining our code base.

1. Learn how to write readable, maintainable, trustworthy unit tests using C#, NUnit and other unit testing tools and frameworks.
2. Learn how to refactor and test legacy code, and what to do when we cannot change existing code.
3. Learn the principles of Test Driven Development, and develop new features using TDD.
4. Learn how to Pair Program, and experience it throughout the course
5. Learn about mocks, stubs fakes, and the difference between them
6. Learn about Isolation(mocking) frameworks, how they work, and why some frameworks are more capable than others

---

## Materials

All students get a PDF of all the slides in the course, in downloadable form.

---

## Agenda - Day 1: Beginnings (09:00-16:00)

1. What is the "unit" in "unit testing"? A discussion about unit of work.
2. The first unit test
3. Basic NUnit APIs, test runners, tips and tricks
4. Writing readable tests
5. Test Driven Development Introduction
6. String Calculator Kata Exercises
7. Value based, State Based and Interaction Based Tests
8. Understanding Fakes, Mocks and Stubs

9. Hand Written Fakes, Stubs and Mocks
10. Interaction Test Exercises with Hand Written Fakes
11. Isolation Frameworks: Constrained and Unconstrained
12. Isolation Framework Comparisons (Rhino Mocks, Typemock, Moq, FakeItEasy, NSubstitute, NUnitMocks, MsFakes)
13. Unit Testing Best Practices
14. Legacy Code