

Refactoring Skills for TDD in .NET and Java

with Roy Osherove

The Problem

You have been trying to learn TDD, and it's going pretty well, but your code looks horrible. Your design skills are lacking, and trying to let TDD drive the design without having design skills can lead to a very problematic, although testable, design of your code.

Objectives

I usually do not focus about design in my TDD courses. I believe that separating the learning about the core skills of Unit Testing, TDD, and Design should be separated so that the learner will not feel overwhelmed. This course is a complement to the Art of Unit Testing course I am teaching. This course is specifically dealing with design skills, and assumes that you already know how to write unit tests.

Assumed: You already have some experience writing unit tests.

1. Clean Code and SOLID Design Principles
2. Design for Testability
3. Refactoring patterns on hard to test legacy code
4. Writing tests against refactored code
5. When it does and does not make sense to refactor
6. We will also take a look at open source projects and try to see how we would refactor them.

Materials

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

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

7. From TDD Kata to Design Skills
 8. Single Responsibility Functions
 9. Single Responsibility Classes
 10. SOLID Principles
 11. Drills throughout the day
-

Agenda - Day 2:(09:00-16:00)

- 12.** Working with Legacy Code
- 13.** Refactoring for Inversion of Control
- 14.** Refactoring for Open Closed
- 15.** Liskov Substitution Problems
- 16.** Refactoring for Dependency Injection
- 17.** Drills throughout the day

About Roy Osherove

Roy Osherove is the chief scientist at Bouvet.no, and the author of [The Art Of Unit Testing](#) and [Notes to a software team leader](#). He is also one of the original ALT.NET organizers. He is now in the process of writing a book on Beautiful Builds.

He consults and trains teams worldwide on the gentle art of unit testing, test-driven development and how to lead software teams. He frequently speaks at international conferences on these topics and others.