

TDD & BDD - Design Through Testing Course Prospectus

Purpose

Learn how TDD and BDD combine to strengthen the design of your code, improve your programming skills and significantly reduce bugs in production. A course for software technicians, with exercises to practise the techniques, and demos by Dave Farley.

Objectives

- To understand the benefits of TDD and BDD, and the design principles that underpin the effective use of TDD as a practice to create high-quality software
- Develop the skills and techniques to write long-lasting, resilient tests, and significantly reduce defect-rates in your code
- Adopt TDD as an approach to software Design rather than focus on “after-the-fact” unit-testing
- Adopt a Behaviour Driven Development (BDD) approach to the use of TDD to drive the creation of higher-quality software.
- Create better tests and learn how to “listen to the code” to modify designs for the better.

Content

Test Driven Development (TDD) is really more about Test Driven Design than it is about testing, and is one of those very rare software engineering practices that can make a real difference to the quality of your code.

The course starts with an introduction to the benefits of Continuous Delivery and TDD, the principles of “How Test First Improves Design”, and “Red, Green, Refactor”, and progresses through a series of practical exercises to practise the techniques, followed by advice and demos by Dave Farley to show how he would approach these problems.

The lessons include:

- How to Write Great Tests*
- Use of Dependency Injection and Mocking*
- TDD Patterns and Anti-Patterns*
- BDD and the Importance of Naming*
- Design Problems and Solutions*
- Refactoring*
- How to ‘listen to the code’*
- Acceptance Test Driven Development*
- How Tests Improve Design*