

Test-Driven Development

The developer's safety net: Enable Agility through quality first.

COURSE DESCRIPTION

Unit testing, test-first and refactoring practices lay the foundation for sustainable code quality and Agile development. Test quality helps drives quality design and code. This course teaches participants how to build quality software through effective use of test-driven development practices.

The course consists of 6 modules:

- 1 Unit Testing Basics**
Unit Testing Principles, Anatomy of the Unit Testing Framework, Crucial Design Principles
- 2 Test-Driven Development (TDD)**
Test First vs. Test Last, TDD Rhythm: Red, Green, Refactor, TDD Influence on Design, Supporting Practices, TDD Example with xUnit
- 3 Refactoring**
Safety Net of Tests, Refactoring Patterns, Refactoring for Maintainability, Refactoring Example
- 4 Beyond the Basics**
Test Doubles (Stubs, Fakes & Mocks), Unit Testing Legacy Apps, Unit Testing with External Dependencies, Unit Testing the Database
- 5 Acceptance Testing & Behavior-Driven Development**
Acceptance Testing, Fitnesse Example, Acceptance Test-Driven Development (ATDD), Behavior-Driven Development (BDD)
- 6 Building the Application**
A multi-part exercise using TDD to guide the process of building an application from a set of high-level requirements (user stories) using ATDD or BDD in combination with TDD practices

AUDIENCE

Developers who want to learn how to enhance their design and development skills to build better quality applications. Participants should have a strong understanding of object-oriented development using C#, Java or equivalent language. This course can be delivered in Java, C# or other language with prior arrangement.

LEARNING OBJECTIVES

By the end of this course, participants will:

- Understand how to write effective unit tests.
- Characterize the properties of effective unit tests.
- Use mock objects to isolate the “system under test.”
- Refactor the code base effectively.
- Articulate the benefits of test-first and test-driven development.
- Use acceptance test and behavior-driven development to further advance test-driven development.

DURATION

3 days