Agile Unit Testing Workshop

- Practice designing, implementing, and executing unit tests
- Discover agile unit testing best practices
- Explore key unit testing approaches like TDD
- Find out how to incorporate non-functional tests such as security and performance
- Understand how to use stubs and mocks to eliminate dependencies
- Learn strategies for increasing unit test coverage

Gain hands-on experience with unit test design, implementation, and execution in this two-day course for development team members who are comfortable with their programming language and the basics of object-oriented design. Attendees learn the techniques of design and developing unit tests, test-driven development (TDD), unit testing best practices, and ways to increase unit test coverage. Practice common unit testing and experience how these techniques reduce defect escapes downstream. This set of practices for developers is essential core competencies for ensuring what they build is effectively tested.

This workshop-style course provides hands-on exercises to allow developers to immediately apply unit testing techniques in an informal and interactive setting. Attendees will leave the workshop with the ability to apply unit testing best practices in their own environment.

Who Should Attend?
This is a novice to intermediate level course intended for software developers. Attendees should have competence with common programming languages such as Java, JavaScript, or C#.

Prerequisites/Requirements
A working knowledge of software development processes, familiarity with basic object-oriented principles of design, and a basic familiarity with an agile framework such as Scrum or Kanban are necessary for this course.

Laptop Required
This class involves hands-on activities using sample software to better facilitate learning. Each student should bring a laptop with a remote desktop protocol (RDP) client preinstalled. Connection specifics and credentials will be supplied during class. Please verify permissions with your IT Admin before class. If you or your Admin have questions about the specific applications involved, contact our Client Support team [1].

Course Outline

Introduction to TDD
Origin
Why do TDD?
What are unit tests?
Comparison of test levels

Unit Test details
Drivers
Annotations
Handling exceptions

Practice
TDD coding cycle
Best practices
Code coverage
- Statement
- Decision
- Condition
- Notes on technique
- Mutation testing

General notes

Mocks
Assertions

Concepts
Black box thinking
Domain
Related concepts

- Agile
- MVP
- BDD
- Pair programming
- Continuous integration

- Duration
- Test orbits and VCS hooks

Transition
What's standing in our way?
Next steps