Test Driven Design
Duration: 5 days (9 hours each day with working lunch)

Agile development is not really about being fast, but about sustainable speed. Lack of or inadequate automated testing makes it hard to respond to change. Automated testing provides two significant benefits: regression and better design. But, writing tests mechanically will not automatically give us the design benefits. This course focuses on how to use automated tests to drive the design of applications. We start with strategic design and then dive into tactical design issues influenced by automated tests. We learn about both test and design quality, how to maintain a good code coverage, and also the essential principles for creating lightweight design.

The course has a good balance of interactive lectures and hands-on exercises. The attendees are expected to pair-up and work on the lab exercises. The instructor will assist the attendees as they work on the labs. The objective of the course is for the attendees to gain an in depth practical knowledge of the concepts so they can put them to immediate use on their real projects.

The course content will be customized to meet your teams' specific needs. Please review this detailed outline and suggest changes (additions, deletions, modifications) as you feel fit.

Topics

Influence of TDD on Agility and sustainability
* Reasons to do TDD
* Cost of doing it
* Cost of not doing it
* Software systems, design, complexity, and how to cope
* Importance of Automated tests
* The two benefits of TDD—rapid feedback and design influence
* What's really needed to be effective at TDD?
* Exercises

Test Driving your Design
* Types of Tests: White box and black box testing
* Type of unit tests–positive, negative, and exception tests
* Performance tests
* What is really a unit of code?
* How to test drive a unit of code?
* What is not unit testing?
* Programming by Intention
* Simple code that works
* How to evolve design through unit tests?
* Practices for Unit Testing
* Tenets of Unit Testing

* Detailed example of building a small App test driven
* When to write unit tests?
* When not to write unit tests?
* TDD Patterns and test qualities
* Exercises

Mocking Out Dependencies
* What makes TDD hard?
* How to deal with dependencies?
* First, knock out before you mock out
* What is a Mock object?
* What is not a mock?
* Common pitfalls programmers run into when using mocks
* How to create mock objects?
* Hand tossing mocks
* Using a mocking framework
* Exercises

Test Driving Code for Thread Safety
* Challenges with Multithreaded code
* Is it a lost cause?
* Designing multithreaded code using test cases
* Removing impediments and avoiding design pitfalls
* Let’s test drive a multithreaded code
* Exercises

Functional Testing
* Functional vs. Unit Testing
* Automation tests and feedback loops
* Reasons to do functional testing
* Fit and FitNesse
* Exercises

Behavior Driven Design
* What’s BDD?
* Relationship between TDD and BDD
* Tools for BDD
* Tools for different languages
* Driving UI automation tests
* Exercises

Measuring Code Quality
* Code quality metrics
* Tools to measure code quality
* Code Coverage
* Detecting code duplication and design smells
* Exercises

Refactoring your code

* What is refactoring?
* Why refactor?
* What’s needed for refactoring?
* What to refactor?
* Reconcile differences
* Specific Refactoring recommendations
* …Method level, class level, data layer, abstraction
* Tools that help with refactoring
* Exercise

Continuous Integration

* Why unit test?
* What’s hard about it?
* What tests to run and how often?
* Integrate early and frequently
* Tools for Continuous Integration
* Code Coverage and other metrics measurements
* Exercises

Agile Design

* Measuring Quality of an Abstraction
* Coupling, Cohesion
* Law of Demeter
* Characteristics of a Poor design
* Characteristics of a Good design
* Fundamental and essential design principles
* Exercise

About the Instructor

Dr. Venkat Subramaniam is an award-winning author, founder of Agile Developer, Inc., creator of agilelearner.com, and an instructional professor at the University of Houston.

He has trained and mentored thousands of software developers in the US, Canada, Europe, and Asia, and is a regularly-invited speaker at several international conferences. Venkat helps his clients effectively apply and succeed with sustainable agile practices on their software projects.

Venkat is a (co)author of multiple technical books, including the 2007 Jolt Productivity award winning book Practices of an Agile Developer. You can find a list of his books at agiledeveloper.com. You may read more about Venkat and Agile Developer, Inc. at http://agiledeveloper.com.

Test-Driving JavaScript Applications
Rapid, Confident, Maintainable Code