Evolutionary Architecture and Design
Duration: 5 days (9 hours each day with working lunch)

Big Up-front Design is a bad idea, as an industry we've learned the consequence of this over the past few decades. Creating design just in time can be a disaster as well. The risk is creating a design that is not extensible and a software system that can get quite hard to evolve and maintain. This is where the principles and the practices we can learn and follow for evolutionary architecture and design come in. This course will help us avoid the perils of up-front design, and at the same time, keep us away from the traps of inadequate or insufficient design. Learn how to create practical design that can efficiently balance the tradeoffs we face in architecting and designing software.

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

Why Evolutionary?
* What's Architecture and Design
* Good Architecture
* Lessons from the past
* When to create architecture?
* Strategic vs. Tactical design
* Risk of up-front design
* Risk of insufficient design
* Risk of acquiring technical debt
* Exercise

How to Create Evolutionary Architecture
* How much to architect and when?
* Role of architects and lead programmers
* Inherent vs. accidental complexities
* Refactoring
* Principles and practices for evolutionary architecture
* Applying the principles and practices
* Exercise

Refactoring Design and Architecture
* How to approach refactoring
* When to refactor
* What's needed before refactoring
* Deciding to refactor or not
* Selecting refactoring techniques
* Mitigating the risks of refactoring
* Exercise

Code Quality and Impact on Evolutionary Design
* Why care about code quality?
* Creating good quality code
* Identifying and removing code smells
* Tools to assist with code quality
* Creating readable and maintainable code
* Exercise

Creating Evolutionary Agile Design
* Perils of poor design
* Design qualities
* Measuring quality of design
* Principles and practices for better, lightweight design
* Cohesion, coupling
* Law of Demeter
* YAGNI
* TDA
* DRY
* SOLID principles
* Other useful principles
* When and how to apply these principles
* Evolving design using the principles
* Exercise

Common Patterns Useful in Evolutionary Design
* Relevance of Design Pattern in Evolutionary Design!
* Pattern that require forethought!
* Patterns that help with afterthought!
* Patterns that play critical role in evolutionary design
* Language influence on realizing patterns!
* Exercise

Test Driven Design
* Driving design through tests
* Programming with intention
* Minimalism and simple design
* Making it work and making it better
* TDD and evolutionary design
* Avoiding brittle tests
* Effective ways to create maintainable tests

* Exercise

Practices to Sustain and Succeed in Evolutionary Design Collective Ownership
* Continuous review
* Effective design and code reviews
* Automation testing
* Continuous integration
* Measuring Design and Code quality metrics
* Benefiting from visible metrics
* Exercise

Creating Evolutionary Lightweight Design
* A practical example of creating lightweight design
* 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
Venkat Subramaniam

Functional Programming in Java
Harnessing the Power of Java 8 Lambda Expressions
Venkat Subramaniam