Hands-On Software Quality and Design Patterns Workshop for Effective Agile Development

Practical, Hands-on Training for Agile Teams

This three day course gives the participants actual hands on experience with the concepts and practices of Agile Software Development. For a team to reach the productivity of a hyper-productive team not only does their process need to be Agile, but their product needs to be Agile. This means that the code must be easily adaptable to reasonable change. This means that the code must be testable, and tested. This means that the team must truly understand how to create maintainable software.

The original Software Design Patterns book is almost not comprehensible to even the most experienced developers, yet it offers an enormous insight into how to write better, more maintainable software. This course not only explains what the Design Patterns book is trying to say, but it also gives the students hands-on practice finding and using some of the most common software design patterns.

This course, inspired by the book "Design Patterns Explained" by Alan Shalloway and Jim Trott, will offer a fresh look at software quality, design patterns and the practical issues of Agile software from the developer's point of view. This course also draws heavily on the popular book "Head First Design Patterns" by Eric and Elisabeth Freeman for its code examples.

Course Objectives

- An understanding of the attributes of Agile software
- An example based exposure of some common software design and quality problems
- A review of the important aspects of Agile quality code
- A focus on Agile Architecture and Design for Maintainable Software
- Practical, skill and experience building practice in Agile Design
- Instruction and practice to develop your skills with Design Patterns
- Discussion and practice about Emergent Design

Design Patterns Covered

▪ Strategy	▪ Adapter	▪ Decorator
▪ Bridge	▪ Facade	▪ Chain of Responsibility
▪ Analysis Matrix	▪ Singleton	▪ Template Method
▪ Abstract Factory	▪ Proxy (Several)	▪ Iterator