

## **Code Reviews, Clean Code, and Refactoring**

### **Course Summary**

#### **Description**

For most IT organizations maintaining existing software systems is major challenge and expense. This course is designed to provide students with proven best practices to keep code bases, clean, and agile thus making maintenance tasks easier, cheaper, and faster complete. The course begins with code review techniques that are highly effective, pragmatic and lightweight. These practices can be incorporated into existing software processes by leveraging existing tools already in use by the software teams. With the process aspects covered the course then moves to the principles of clean code providing students with clear path to writing clean code, as well as cleaning up existing code bases to make the code more agile. With the process, and principles laid out, the course then covers automated refactoring tools that can be used to quickly cleanup code bases. Upon completion of the course, students will be able to quickly and efficiently cleanup existing code bases to make the code bases cheaper and easier to maintain.

#### **Topics**

- Peer Code Reviews
- Clean Code
- Automated refactoring
- Mini project

#### **Audience**

This course is designed for programmers and team leads who want to keep code bases clean and agile in order to make maintenance tasks quick and easy.

#### **Prerequisites**

Students should have experience with Java or C# programming experience.

#### **Duration**

Three days

## **Code Reviews, Clean Code, and Refactoring**

### **Course Outline**

#### **I. Peer Code Reviews**

- A. Peer Code Reviews
- B. Types of Code Reviews
- C. Pragmatic Lightweight code review process
- D. Overcoming obstacles to incorporating code reviews into existing projects.
- E. Applying code reviews in the real world

#### **II. Clean Code**

- A. What is clean code
- B. Clean names
- C. Clean Functions
- D. Clean Comments
- E. Clean Formatting
- F. Clean Objects
- G. Clean Data Structures
- H. Clean Error Handling
- I. Clean Boundaries
- J. Clean Unit Tests
- K. Clean Classes
- L. Clean Systems
- M. Clean concurrency
- N. Guide to keeping code clean

#### **III. Refactoring**

- A. What is refactoring
- B. Code Smells
- C. Systematic exploration of all automated refactoring features included with eclipse
- D. Visual Studio Refactoring tools (Optional)
- E. Refactoring towards patterns
- F. Catalog of common refactoring
- G. Planning refactoring
- H. Integrating refactoring into the development process
- I. Optimizing refactoring

#### **IV. Mini Project**

- A. During the mini project students will apply what they have learned during the class to copy of an existing code base that they are currently working on.
- B. If the students don't have access to a code base a sample code base will be provided.
- C. Through a structured series of individual and group exercises students will apply all the techniques that they learn in the course and be able to bring those practices and techniques to the codebases that they work with on a daily basis.