Five Days Agile Development Practices Workshop

This workshop is targeted at development teams who want to start transitioning to Agile Software Development methods and want to get a first hand feel for what it means to be working on an eXtreme Programming team. In this week long workshop we'll simulate a real world eXtreme Programming environment. Participants will get to experience the following practices in action

Working on a real eXtreme Programming project for a week will help to understand and experience the core fundamentals of Agile Software Development. Concepts like Self-Organized Teams, Adaptive Planning, Cross-functional Teams & Collaboration, Evolutionary Design, Eliminating Waste, etc will be clear.

Learning Outcomes

- Understand the thought process and the need for XP practices
- Identify poorly designed code by using our elaborate code smell vocabulary.
- Learn various automated refactoring techniques and strategies to carefully clean the identified code smells.
- Hands-on experience in practices like TDD, CI and Pair Programming
- Experience in authoring User Stories
- Clarity about Collective Code Ownership, Informative work spaces, planning game and project retrospectives

Course Outline

- Introduction to eXtreme Programming
- Test Driven Development
  - Introduction to TDD
  - Test First Vs Test Last
  - TDD Rhythm: Red, Green, Refactor
  - Crucial Design Principles
  - TDD and Design
  - Avatars of TDD
- Refactoring and Evolutionary Design
  - Purpose of Refactoring
  - Common Code Smells
  - Refactoring Patterns
  - Using Automated Refactoring
- Continuous Integration
Why CI?
- Principles, Strategies and Techniques
- Setting up a CI server
- Automated Testing
- Applying CI to Large Projects

Pair Programming
- How to Pair?
- Need for Pairing
- Pairing Infrastructure
- Pairing Styles

User Stories
- The Big Picture
- Interaction Design with Low Fidelity Prototyping
- Capture nonfunctional requirements as stories
- Working with user proxies
- Conducting a story-writing workshop
- Planning with User Stories

Planning Game

Project Automation
- Setting up an automated build
- Creating automated unit, acceptance and UI tests
- Setting up a Continuous Integration (CI) process
- Setting up coding standards and adding code coverage or test coverage
- Setting up an integrated project dashboard
- Setting up a build promotion process and team wiki

Collective Code Ownership
- Informative Workspace and information radiators
- Stand-up meetings and Dev Hurdles
- Project Retrospectives

Method of Instruction
- Interactive Dialogues, Programming Exercises, Demos, Online e-Learning and Instructional Games

Transfer %
- Knowledge: 50%, Skill-Building: 50%

Target Audience
• Entire Development Team wanting to experience eXtreme Programming in Action.

Course Level

• Beginner to Intermediate

Course Prerequisites

• Working experience in the project
• Highly Recommended: basic understanding of the life-cycle of software projects

General Requirements

To ensure a successful class, we require the following facilities:

• VGA projector (1024x768 minimum) & Projector screen
• 1 White board & Dry erase markers
• Cluster seating with 5-6 people on each table
• 1 Flip chart with the stand and marker pens for each table
• Notepad and Pen for each participant
• Ample room for students in terms of room size and set up
• For Dev trainings: at least one powerful workstation between two programmers

Development Tools

▼ Java

• Latest Java JDK
• Latest Standard Eclipse OR IntelliJ Idea
• JUnit, Mockito, JBehave, FitNesse or Cucumber JVM

▼ C#

• Latest .Net Framework
• Latest Visual Studio
• Latest ReSharper Plugin
• SpecFlow’s Visual Studio IDE Integration or FitNesse

▼ JavaScript

• Latest WebStorm JavaScript IDE
• Latest Chrome Browser
• Latest Jasmine Framework
• Latest JsTestDriver
- Latest Eclipse CDT OR Visual Studio
- Latest GTest
- Latest Visual Studio
- Latest CLion

- Latest RubyMine
- Latest Cucumber

- Latest PHPStorm
- Latest PHPUnit, BeHat

- Latest Flex SDK
- Latest Flash Builder OR IntelliJ Idea