

TRAINING COURSE SUMMARY

Agile

The Best of Agile Methodologies (2 days)

Examines the primary agile methodologies, revealing the differences in techniques, outputs and workflow. A closer examination of agile techniques considered to be the best is undertaken, through six real-world case-studies and five workshop exercises.

Agile 101 (1 day) (cut-down version of The Best of Agile Methodologies)

Provides an understanding of the fundamental principles and practices of the three main-stream agile methodologies (XP, Scrum and FDD). These are reviewed and compared to exemplify the agile principles and to understand key agile techniques. Five real-world case-studies and three workshop exercises provide further insight and practical appreciation of applied use.

Scrum (1 day)

The principles, output and team structure of Scrum are discussed and then the top five Scrum practices (Product Backlog, Sprints, etc) are explored in more detail. Two hands-on group activities plus five real-world case-studies aim to cement this understanding of Scrum.

eXtreme Programming (1 day)

Provides a pragmatic insight into the rules and practices of XP, covering all development aspects (manage, design, test & code) the XP way. Three real-world case studies and five workshop exercises ensure a practical appreciation of XP.

Requirements Analysis: JAD to Prototype (one day)

This course focuses on five of the key techniques that ensure the scope of a system actually meets the user's requirements. A practical mini-JAD (Joint Application Development) session is conducted with the class, as well as four other workshop exercises in the key techniques.

Code Inspection (half day)

A very hands-on training workshop, with review of the different forms of peer review, followed by a practical code inspection (in pairs) of the attendees' code they've brought to the workshop.

Object-Oriented

OO Principles and Patterns (1 day)

Two fundamental aspects of designing object-oriented software are explained – OO Design Principles are the underlying guidelines for all OO designs; and Design Patterns are the essence of reuse. A range of principles and patterns are presented with two workshop exercises undertaken to cement this understanding.

Software Architecture (2 days)

An advanced workshop giving practical insights into “architecturally centric” software development. Real world and analogous examples are reviewed as well as eight group-based workshop exercises undertaken.

UML in a Nutshell (1 day)

A comprehensive introduction to the Unified Modelling Language (UML). Several real-world examples are examined and there are four workshop exercises in the key UML notations: Use Case, Class, Sequence and State Machine.

OO Software Engineering (3 days)

Provides a comprehensive understanding of Object-Oriented technology. A range of OO techniques, UML notation, principles, patterns and architectural focus are all presented. A total of eight workshop exercises are used to ensure the most practical understanding.

OO Project Management (2 days)

Intended for project managers, software managers, team leaders and the like. Covers the “what” to expect with OO technology and then goes into the details of how to manage OO developments and how to migrate to OO technology.

