

Requirements Management With UML

Duration:	2 days
Type:	intermediate

Description

This course provides a comprehensive introduction to requirements management using the UML and Use Cases. The target audience is requirements analysts, business analysts and managers but the course can also be useful for developers seeking a more in-depth understanding of how requirements are captured.

The course begins with an introduction to how requirements management is used in modern software processes, including Agile methodologies. The process of capturing and developing Use Cases is then covered in detail, with special attention paid to supplementing Uses Cases with additional documents as required. Finally the course provides an overview of how Use Cases are realized by the development team, and how the analyst can best facilitate the work of architects, designers and coders.

All of the UML diagrams are introduced during the course, including the new features of UML 2.0. By default the course does not use a specific UML editor, but any modeling tool can be covered if required.

Prerequisites

Delegates should have several years experience in software development

List of Modules

Introduction to the UML

- The origins and uses of UML
- The 4 + 1 views of your system
- An overview of each UML diagram
- Review of popular UML tools
- Using UML within Agile development
- Using UML within the Unified Process

Introduction to Requirements Analysis

- The most important job in software development
- Identifying project stakeholders
- Finding sources of requirements
- Interviewing sources successfully
- Identifying and avoiding problems
- Correcting mistaken beliefs about coding
- Managing scope and controlling change
- Tracing requirements into implementation

Working With Use Cases

- Identifying Actors and Use Cases
- Common mistakes when identifying actors
- Treating the system as a black box
- Working with CRUD based Use Cases
- Scoping and specifying more complex Use Cases
- Reviewing stories via a Use Case Diagram
- Modeling Use Cases initiated by the system
- Modeling Use Cases that consult human actors

Writing Use Case Reports

- The purpose of a Use Case Report
- Deciding on the correct level of detail
- Specifying flows of events as a dialog
- Identifying the basic flow (happy path)
- Identifying alternate and exceptional flows
- Coping with conditionality in flows of events

Advanced Use Case Reports

- When one Use Case should include another
- When one Use Case should extend another
- Generalization relationships between Use Cases
- Generalization relationships between Actors

Supplementing Use Case Reports

- Using Activity Diagrams to describe events in detail
- Options for creating User Interface Prototypes
- Mistakes to avoid when working with screenshots
- Advantages and disadvantages of using pseudo-code

Capturing Non Functional Requirements

- Problems capturing non functional requirements
- Special considerations for Web Applications
- Performance, usability and reliability requirements

Moving From Requirements to Implementation

- Understanding the design/code/test process
- The importance of Object Oriented Programming
- How classes are extracted from Use Cases

Understanding Use Case Realizations

- How a design is created by realizing Use Cases
- Describing classes using VOPC Diagrams
- Defining behavior using Sequence Diagrams
- Supplementing Class Diagrams with Statecharts
- Grouping classes into components
- Documenting deployment scenarios