

OO61
5 days

The UML for Industrial Strength Software Development - Enterprise Architect

The Unified Modelling Language has become one of the industry standard notations for the analysis and design of IT systems. This course introduces the UML 2.0 and its application to modern software development processes using Enterprise Architect. Participants receive a solid grounding in OO technology and are then taken through the software development cycle using the UML. There are extensive exercises and opportunities for discussion. This is an intensive hands-on course using Enterprise Architect to show how to design robust OO systems. The use of the UML in forming a basis for Requirements Gathering through to Testing, Acceptance and Delivery is emphasized, enabling participants to deliver high-quality surprise-free systems. All exercises come with worked solutions and delegates receive a copy of "UML Distilled" by Martin Fowler

Course Objectives

- Explain Object Oriented design and development techniques and terminology
- Show how OO Analysis and Design relates to traditional techniques
- Describe how the use of the UML for modelling fits with OO technology for software development
- Show how the use of Abstraction, Inheritance and Polymorphism can simplify programs.
- Show Enterprise Architect is used throughout the design and build process
- Describe Requirements capture and their management with Use Cases
- Show how static and dynamic attributes can be modelled in the UML
- Explain why Dynamic modelling leads to more complete solutions

- Show how UML is used to model business processes and avoid inconsistent design.
- Show how complex business rules can be captured in Decision Tables and how these are introduced into Use Cases
- Describe the phases and workflows of the unified process and how they are applied with UML
- Show how UML is used to manage a smooth transition from Analysis to Design and Implementation
- Present the Package and Architecture modelling features of UML
- Present Patterns as a way of simplifying programs and maximising reuse.
- Overview of future directions of the UML and how it affects you.
- Show how Testing is implemented throughout a UML-based project

Audience

- Management wanting to understand the project issues of UML
- Technicians wanting to improve their technical performance
- Senior staff needing to examine the potential of UML for their organization
- Anyone concerned with Requirements Gathering and Program Design

Prerequisites

- General knowledge of software development process.
- Keen to provide more professional IT services

Timetable

Register at 09:00 for 09:30 start on Day1, 09:00 the rest of the course.
Finish at 17:00 on all days.

Presentation Style

Lectures, demonstrations, group discussions and exercises

Dates and Venues

Refer to *Course Schedules*.

OO61

5 days

The UML for Industrial Strength Software Development - Enterprise Architect

The course covers:

- Introduction to Software Development Processes
- Process Alternatives
- The Rational Unified Process
- Faults in our Processes
- Iteration as a General Technique
- XP, SCRUM and Spiral

Object Orientation in Business

- What is OO?
- Benefits of Object Technology
- Abstraction
- Objects and Classes
- Objects and State
- Polymorphism and inheritance

Advanced OO Analysis

- Analysis versus Design
- The analysis phase
- Domain modelling
- CRC Techniques
- Concrete and Abstract design
- Inheritance Pitfalls
- Interface versus implementation

UML Overview

- Origins and purpose of UML
- A walkthrough of the UML diagrams
- Using UML diagrams
- What's new in UML2.0 and why
- The UML Meta Language
- OCL

Introduction to Enterprise Architect

- Facilities
- Use in Use Cases, Class Diagrams,
- Use in Sequence Diagrams and
- Use in State Tables
- How to Project Base
- Re-engineering Techniques
- Code Skeleton Generation
- Architecture Modelling
- Tools and Techniques
- Practical Usage

Decision Tables

- Capturing complex requirements
- Completeness Checking
- Business Terminology
- Code Generation from Tables
- Decision Tables in Use Cases

Use Cases

- What are Use Cases
- Constructing Use Case diagrams
- Requirements and Use Cases
- Documenting Use Cases
- Alternates and Exceptions
- Scenarios
- Test Case creation

Static Modelling

- Classes and interfaces
- Class diagrams
- Object diagrams
- Class relationships
- Robustness analysis and Class diagrams
- Association
- Aggregation and composition
- Association classes
- Inheritance
- Abstract classes
- Interfaces
- Stereotypes
- Constraints

Dynamic Modelling

- Introduction to UML dynamic modelling
- Activity diagrams
- Sequence diagrams
- Collaboration diagrams
- Statecharts and Statetables
- Generating code from tables

Objects and Databases

- Object Persistence
- Mapping Classes to Tables
- Proxies
- Factory Classes
- OO Databases
- Relational Databases
- Object-Relational
- JDO, Hibernate and related mapping tools

Component and Deployment Models

- Using Packages to organise your system
- Using components to model physical organisation
- Modelling hardware architecture

What are Patterns?

- Designing with Patterns
- Where to find Patterns
- Frameworks
- Some Core Patterns
- Delegation
- Factory
- Singleton
- Decorator
- Pros and Cons of Patterns

Testing

- From Use Cases to Acceptance tests
- Class-oriented testing
- Test Frameworks
- Test-driven development
- Test metrics

Summary and Conclusion

- A strategy for Beginning
- Where to go for Information
- Avoiding mistakes
- Commonsense approach to introducing the UML