



Object Oriented Analysis & Design using UML & Sparx Systems Enterprise Architect

The application of OOAD techniques has substantial benefits in reducing system development risks and improving the quality of object-oriented software developments. The Unified Modelling Language ([UML](#)) has emerged as the de-facto standard for OO modelling techniques.

The Object Oriented Analysis & Design with [Sparx Systems Enterprise Architect](#) training course is an intensive hands-on workshop in which attendees will gain practical experience of the major techniques of UML modelling using the Enterprise Architect (EA) case tool.

What will I learn on the Object Oriented Analysis and Design with UML & EA training course?

Delegates attending the OOAD with UML & [Sparx Systems Enterprise Architect](#) training course will learn to;

- Undertake use case analysis as part of the analysis and specification process for computer systems
- Use different levels of object models during the specification and design of computer systems – in particular understanding the separation between analysis and design.
- Develop and refine sequence diagrams that guarantee the system specified can be implemented using your object model.
- Develop state models to assist in the specification of systems.
- Use design patterns to assist your designs.
- Use packages to structure large systems.
- Be aware of the elements of technical domain design (e.g. persistence and database interfacing, distribution, user interface, external system interfacing, etc).
- Understand how object models may be implemented directly in OO languages such as C++, Java and VB.
- Be able to use Sparx Systems Enterprise Architect to model systems with the UML notations.

Course Style

40% lecture / 60% practical sessions

Object Oriented Analysis & Design with Enterprise Architect; Course Content:

Introduction

- Brief introduction covering what the UML is, including, diagram types and a brief history.
- Object Oriented Analysis & Design.

Overview of OOAD concepts

- Look at a responsibility pattern approach to OOAD (GRASP patterns).
- Introduction to EA.

What is Enterprise Architect?

- EA Architecture (where is the model stored and high level differences between versions).
- A look at the object browser and how models are structured using views and packages.
- Diagramming techniques.
- EA Options menu.
- EA Settings menu.
- How to search the model.

Focus is on getting new users to a point where they are familiar with how to use the tool and where to find certain features; also trying to instil best practice early on such as using package diagrams to structure views and reuse of elements in the model.

Use Cases

- How to draw the Use Case diagram.
- Identification of use cases and actors.
- Discussion on application within iterative development methodologies.
- Specification of Use Cases (Basic Flow, Alternate Flows, Exception Flows) using text.
- Common issues and their solutions.

Requirements modelling and tracing in Enterprise Architect

- Internal and External requirements in Enterprise Architect
- How to draw the custom diagram for requirements in Enterprise Architect
- Realising and tracing requirements
- Change management using requirements diagrams
- Importing and Exporting requirements

Activity diagram (documenting use cases and business processes diagrammatically)

- How to draw Activity diagrams
- Look at how to document use case specifications diagrammatically with activity diagrams and the benefits it brings
- Look at business process modelling using activity diagrams and the advantage of using structured activities

Class Diagram

- Discuss classes and objects
- How to draw class diagrams
- Differences between domain modelling and implementation modelling
- Object diagram (brief overview)



Object Diagram

- Discuss [UML](#) notation and application of an Object Diagram

Package Diagram

- Discuss UML notation and application of an Package Diagram

State Machine Diagram

- How to draw State Machine Diagrams
- How to Model an Object State in EA so that those states are available for that object throughout the model

Sequence Diagram

- Discuss differences between using sequence diagrams for analysis or design
- How to draw sequence diagrams

MDA

- What is a PIM
- What is a PSM
- PIM to PSM Transformations
- Brief look at the transformation templates in EA

Component Diagram

- How to draw Component Diagrams
- Discuss Applications of Component Diagrams such as analysing existing Architectures to look for possible optimisations
- Show how to link component class design to the component without compromising the structure of the model

Composite Structure Diagram

- How to draw Composite Structure Diagrams
- Using collaborations to model patterns or as a data mapping tool

Code Engineering

- Forward and Reverse Engineering Code including reverse engineering binaries
- Forward and Reverse Engineering Databases (look at Data Modelling using Class Diagrams)
- Creating Sequence diagrams from existing applications through debugging them in EA
- A look at the Code Engineering Templates in EA
- Discuss Pro's and Con's of Code Engineering

Data Modelling

- How to forward & reverse engineer DDL
- How to use MDA transforms to create a data model from a PIM

Documentation in EA

- How to generate RTF documentation in EA
- Customise RTF templates
- How to generate HTML documentation in EA
- Customise HTML templates
- Virtual documents

The Communication Diagram

- A brief overview of UML notation and application for the Communication Diagram

Interaction Overview Diagram

- How to draw Interaction overview diagrams
- When to use interaction overview diagrams

The Deployment Diagram

- A look at using the Deployment Diagram to model the physical distribution of components & other artefacts

The Timing Diagram

- Discuss UML notation and application of the Timing Diagram

Prerequisites

A basic understanding of object oriented principles. Some previous experience in object oriented programming is recommended.

If you require a quote for onsite training please [contact us](#).