

## Software Modelling with UML

---

<b>Course category</b>	UML
<b>Training area</b>	Design Techniques
<b>Course code</b>	OO-302
<b>Duration</b>	3 days
<b>Additional information</b>	For price: Contact Feabhas (available for on-site delivery only)

The Unified Modelling Language (UML) has become the de facto standard for modelling software systems. The UML defines a rich set of syntax and semantics that, if used correctly, can greatly improve the quality of software.

This course leads the attendee through the complexities of modelling software using UML. The emphasis is not on producing an exhaustive – and exhausting – portfolio of UML diagrams. The focus is on understanding the core of software modelling – using multiple different models to describe, understand and ask questions of the software design.

### **Course objectives:**

- To provide an understanding of Object Oriented software design
- To show how to model software in a rigorous and systematic manner
- To enable attendees to develop their own practical design skills
- To teach the UML design notation.

### **Delegates will learn:**

- The fundamental concepts and terminology of Object Oriented design
- The diagrammatic and modelling underpinnings provided by UML for OO development methods
- How to apply the design principles in software applications
- The basics of an integrated, traceable and consistent approach in the development of software
- Where and how CASE tools can be used in the development process.

### **Pre-requisites:**

- Some understanding of technical software development methods and some knowledge of a high-level programming language.

### **Who should attend:**

The course is designed for:

- Designers new to the area of software modelling.
- Developers embarking on projects using UML-based techniques for the first time.

**Duration:**

- Three days.

**Course materials:**

- Delegate handbook
- All worked examples and solutions

**Course workshop:**

Approximately 50% of the course involves practical application of the techniques. Delegates work in small groups dealing with problems based on real-world systems.

The course specifically does not make use of a CASE tool. From our experience a CASE tool distracts delegates from learning design issues and UML. However, the workshops clearly demonstrate when CASE tools are helpful and when they are not, thus aiding CASE tool selection.



**Why do we model?**

- Understanding the need for modelling software systems

**Quality:**

- Defining quality
- Understanding the business needs

**Requirements Analysis:**

- Introducing Use Cases

- Understanding stakeholders
- Building the Use Case model
- Use Case description techniques
- Use Case rationalisations and their problems

### **Object Modelling:**

- Understanding objects
- Objects in software
- Quality measures for objects
- Identifying objects
- Scenario-based object modelling
- Responsibility-driven design
- Robustness modelling to improve design

### **Interaction Modelling:**

- Understanding interaction models
- Sequence and Communication diagrams
- Control authority decisions
- Combining and rationalising different scenarios to improve understanding

### **Class modelling:**

- The relationship between objects and classes
- The class diagram
- Attributes and operations
- Class associations

### **Composite Structures:**

- Defining software architectures
- Understanding the relationship between classes, composites and objects
- Improving encapsulation with Ports and Interfaces
- Component-based design
- Active and Passive objects

### **Behavioural Modelling:**

- Modelling modal behaviour
- The State Machine
- Mealy and Moore state machines
- Composite states

- Ensuring internal and external behaviour is consistent
- Modelling algorithmic behaviour
- The Activity diagram
- The relationship between classes, state machines and activity diagrams

### **The software development process:**

- Model Driven Architecture (MDA)
- The relationship between MDA models
- The Computationally Independent Model
- The Platform Independent Model
- The Implementation Model
- How UML can help with MDA

### **Case Study:**

- This is based on developing embedded software for a computer-controlled manufacturing test rig.

**Feabhas Ltd - UK Office** 15-17 Lotmead Business Park, Wanborough, Swindon, SN4 0UY, UK ·

+44 (0) 1793 792909 · [info@feabhas.com](mailto:info@feabhas.com) · [www.feabhas.com](http://www.feabhas.com)