Real-time System Requirements Definition and System Analysis using SysML - Training Course - 3 Days

This SysML training course is aimed at real-time system architects, engineers and developers who want to produce a detailed outside-in model of real-time and embedded system requirements using use cases and a detailed structural and behavioural model of real-time and embedded systems from the requirements. The strict approach to writing effective use cases ensures that the results satisfy the needs of both non-technical and technical stakeholders. Industry best practice system modelling techniques are based on the Systems Modelling Language SysML v1.1 and are taught within the context of a model-driven development process. The models produced are sufficiently detailed to form the basis for the design of systems using a variety of different real-time and embedded systems architectures. The advantages that using these techniques has for estimation, traceability, test development and project management is discussed. Each technique is taught to the level required for competence on a real project. Understanding is tested and improved with exercises based on a real-world project example and using a suitable case tool.

Delegates will learn:

- The basics and the necessary detail of the Systems Modelling Language SysML
- The basics and the detail of Object Orientation
- How to create a first cut overview of functional requirements for real-time and embedded systems with actors and use cases on a use case diagram
- How to write an effective use case description in a way that satisfies both non-technical and technical stakeholders
- How to restructure the use case diagram to handle complex relationships between use cases without bloating the use case model
- How to model high-level system structure and decomposition using block diagrams, blocks, parts, ports, interfaces and flow specifications
- How to create a detailed model of software structure and data using classes and their relationships on class diagrams
- How to map the functionality of the system requirements onto the object model using sequence diagrams
- How to structure the modelling in the form of a use case implementation
- How to model the dynamics of system functionality and data using statecharts
- How to model at a consistent level of abstraction
- How the modelling performed fits into an incremental model-driven development process

Suitable for:

System Requirements Gatherers, System Architects, System Engineers, System Analysts and Developers with at least 2 years experience. This course is not suitable for those seeking certification as a step towards a qualification. See UML Certification for a detailed discussion.

Course Logistics:

Course attendance is limited to 12 students. Courses start at 9.30am on the first day, 9.00am on subsequent days and finish at 5.00pm each day. Students use a computer for the exercises. For a discussion on case tool use please see Case Tool Use on Courses. Printed course manuals for each student with copies of all presentations, exercises and solutions are provided.

On-Site (In-House) Courses:

The client is expected to provide an appropriate venue, refreshments, SVGA/XGA projector and screen, whiteboard or flipchart and at least one computer per two students loaded with a SysML case tool. For a full discussion of on-site course issues please see On-Site Course Logistics.

Scheduled Public Courses:

This course is currently not available as scheduled public training. Please see the Scheduled Public Courses page for available courses.

Pricing:

On-site (in-house) course pricing is available from the On-Site Course Price Calculator page. Public course pricing is available on the Scheduled Public Courses page. Consultancy pricing is available on the Consultancy page.

Training Course Outline

Day 1

Introduction

People - Course Structure - Object Orientation - System Modelling Language SysML - Use Cases and System Analysis - A Process for Modelling

Specifying Functional Requirements with Use Cases

System Use Cases and Actors - Primitive Use Cases and the Basic Course - Writing Effective Use Case Descriptions - Writing Sub-flows and Alternate Flows - 'Include' and 'Extend' Relationships - Modelling Timing Constraints

System Use Case Workshop

Gathering Requirements

Collecting Requirements Information - Mapping from the Business Model - Proof of Concept Prototypes - Requirements Documents - Estimating and Traceability - Incremental Development

Day 2

Gathering Requirements Workshop

Objects and Classes

What is an Object? - Classes and Objects - Attributes - Operations and Methods - Designing Good Classes - Choosing the Right Classes

Object and Class Workshop

System Structure Diagrams

Blocks - Composition - Ports - Block Diagrams - Instances - Links - Part Diagrams - Interfaces - Flow Specifications

System Structure Workshop

Software Class Diagrams

Associations and Links - Navigability and Naming - Multiplicity and Other Adornments - Aggregation and Composition

Object Relationship Workshop

Day 3

Interaction Modelling

Interactions, Messages, Operations and Methods - Sequence Diagrams - Selection and Iteration - Activation - Collaboration Diagrams

Interaction Modelling Workshop

State Modelling

The Meaning of the State Model - States and Transitions - Events and Conditions - Actions and Activities - Consistency with Other Diagrams - Advanced Syntax - Implementation

State Modelling Workshop

System Analysis

Creating the Initial Object Model - The Analysis Cycle - Iterative Modelling - Interface Prototyping - Completing the Model

System Analysis Workshop