

Integrated Systems Design and Requirements Analysis for Software Development Course

11 – 15 October 2027
Abu Dhabi

AGILE LEADERS
Training Center

AGILE LEADERS
Training Center

Integrated Systems Design and Requirements Analysis for Software Development Course

Ref: 118_111837 **Date:** 11 – 15 October 2027 **Location:** Abu Dhabi **Fees:** 4700 Euro

Overview

Integrated Systems Design and Requirements Analysis for Software Development is a 1-week course for business analysts, systems analysts, software developers and IT project staff who leave with a traceable requirements specification and a design handover package. The course addresses the gap between what stakeholders ask for and what development teams build, where vague requirements cause rework, missed scope and disputed acceptance. Participants elicit needs, model them with use cases and UML, shape architecture and data, and validate the result before handover. Systems design and requirements analysis practice is delivered by Agile Leaders Training Center.

Who Should Attend

- Business analysis teams responsible for eliciting stakeholder needs and documenting business rules
- Systems analysis teams responsible for translating needs into functional and non-functional specifications
- Software development teams responsible for turning models and specifications into working components
- Solution and data design staff responsible for system structure, interfaces and data models
- IT project and quality staff responsible for scope control, traceability and acceptance testing

The course assumes participants already take part in software projects at work, and it leaves out programming language instruction and tool vendor certification.

Departments and Industries

The course serves departments and industries that commission, specify, build or maintain business software.

- IT and digital transformation departments
- Software houses and systems integrators
- Banking, insurance and financial services
- Healthcare and hospital information systems
- Government service delivery and e-services units
- Telecommunications, retail and logistics operations

Learning Objectives

By the end of this course, participants will be able to:

- Apply structured elicitation techniques to capture stakeholder needs
- Build use case and UML models of system behavior and structure
- Analyze requirements for completeness, conflicts and testability
- Evaluate architecture options against quality attributes
- Build logical and physical data models from domain classes
- Prioritize and baseline requirements for a development handover

Course Agenda

Day 1: Requirements Foundations and Stakeholder Analysis

- Software Development Life Cycle Models and the Place of Analysis and Design
- Requirements Engineering Processes in ISO/IEC/IEEE 29148 and IIBA BABOK v3
- Stakeholder Map and RACI Matrix for Software Projects
- Elicitation Techniques: Interviews, Workshops, Observation and Document Analysis
- Business Process Modeling with BPMN for As-Is and To-Be Flows

Day 2: Documenting and Modeling Requirements

- Functional and Non-Functional Requirement Statements with Acceptance Criteria
- Use Case Diagrams and Fully Dressed Use Case Specifications
- User Stories, Story Maps and the INVEST Criteria
- UML Activity Diagrams for Workflow and Business Rules
- Wireframes and Low-Fidelity Prototypes for Requirement Confirmation

Day 3: System Architecture and Behavior Design

- UML Class Diagrams for the Domain Model
- UML Sequence and State Machine Diagrams for Interaction Design
- Architecture Styles: Layered, Client-Server, Microservices and Event-Driven
- Quality Attribute Scenarios for Performance, Security and Availability
- Interface and API Specification Between System Components

Day 4: Data Design, Validation and Handover

- Entity-Relationship Modeling and Normalization to Third Normal Form
- Mapping Domain Classes to a Physical Database Schema
- Requirements Reviews, Inspections and Prototype Walkthroughs
- Requirements Traceability Matrix from Need to Test Case
- MoSCoW Prioritization, Baselining and Change Control

Day 5: Applied Analysis and Design Practice

- Exercise: Running an Elicitation Workshop for a Service Request System
- Exercise: Writing Use Case Specifications and Activity Diagrams for the Case
- Exercise: Designing Class, Sequence and Entity-Relationship Models for the Case
- Exercise: Reviewing a Specification for Ambiguity and Testability
- Capstone: Presenting a Traceable Specification and Design Handover Package

Practical Exercises

The course includes suggested activities based on software cases from several sectors.

- Suggested activity: map stakeholders and plan elicitation for a hospital appointment system
- Suggested activity: model an online loan application with use cases and UML diagrams
- Suggested activity: derive a normalized data model for a retail order process
- Suggested activity: build a traceability matrix linking requirements to test cases

FAQs

Who is systems design and requirements analysis training suited to, and what does it assume?

It suits business analysts, systems analysts, developers and IT project staff at intermediate level. It assumes participants already work on software projects and are familiar with basic project terms, but it requires no specific programming language.

How does systems design and requirements analysis differ from a pure software architecture course?

Systems design and requirements analysis runs from stakeholder needs through models, architecture and data to validated handover. A pure software architecture course concentrates on technical structure, patterns and infrastructure decisions, with less attention to eliciting and documenting requirements.

Which UML diagrams are used most in requirements analysis?

Use case, activity, class, sequence and state machine diagrams are the most used. Use case and activity diagrams capture behavior and workflows, while class, sequence and state diagrams describe structure and interactions for designers and developers.

What makes a software requirement testable in systems analysis?

A testable software requirement is specific, measurable, unambiguous and linked to acceptance criteria. Testers can then design a test case that clearly passes or fails, and the requirement can be traced from its source need to its verification.

AGILE LEADERS
Training Center

How is a requirements traceability matrix used in software development?

A requirements traceability matrix links each requirement to its source, design elements and test cases. Teams use it to confirm coverage, assess the impact of changes and show that every agreed requirement was built and verified.

Conclusion

Participants take back a traceable requirements specification and design handover package covering use cases, UML models, architecture decisions and a data model. The package gives development and testing teams a shared, validated reference instead of scattered notes. It reduces rework and disputes at acceptance by linking every design element to an agreed stakeholder need.