

# Business Requirements Analysis in the Unified Process

---

## Synopsis:

Business requirements analysis is the process of discovering requirements, analyzing requirements for incomplete, inconsistency, relevance and practicality and negotiating the final requirements for the systems. Whether the computer systems developed are what users want depends on how well requirements are captured at the earlier stage of system development. This course teaches Business Requirements Analysis in the Unified Process, particularly, the process of requirements elicitation, analyses, documentation, communication, and change management. The course will also teach UML (Unified Modelling Language) as a notational tool for expressing requirements and Use Case Modelling for capturing and representing user requirements. The course will also highlight how to transition from Requirements to Design.

## Who should attend?

Business Analysts, Systems Analysts, Requirements Analysts, Managers, Subject Matter Experts, Users, Project Leaders, Software Developers and Systems Architect.

## Objectives

1. Learn about requirements elicitation and its techniques
2. Learn how to use Use Cases to gather user requirements
3. Learn how to capture non-functional requirements
4. Understand and practise Use Case Modelling and UML
5. Learn how to produce Use Case Specification
6. Learn how to manage changing requirements

## Prerequisites

None

## Duration

3 days

## Course Outline:

1. **SYSTEM ANALYSIS AND DESIGN**  
Describe analysis and design  
Discuss key steps in Analysis and Design:
  - Define Use Cases
  - Define Domain Model
  - Define Interaction Diagrams
  - Define Design Class DiagramExplain UML (Unified Modelling Language)  
Define the Unified Process, an iterative and evolutionary development process  
Describe how to manage change in an iterative project
2. **THE UNIFIED PROCESS**  
Describe the Unified Process Disciplines: Business Modelling, Requirements, Design, Implementation, Test, Deployment, Configuration and Change Management, Project Management, Environment  
Describe the Unified Process Phases: Inception, Elaboration, Construction, Transition  
Explain the relationship between the disciplines and phases
3. **REQUIREMENTS ELICITATION**  
Define requirements  
Explain Evolutionary vs Waterfall Requirements  
Describe the types and categories of requirements  
Discuss requirements elicitation issues  
Discuss various requirements elicitation techniques  
Describe benefits of each technique

- Describe the skillful means to finding requirements
- Discuss how to organize requirements as Unified Process artifacts
- Discuss other essential requirements in System Development
  
- 4. USE CASES
  - Define and Discuss Actors, Scenarios, and Use Cases
  - Explain Use Case Modelling
  - Explain why use Use Cases
  - Explain Use Case Writing Formats and Formalities
  - Explain how to find Use Cases
  - Discuss Use Case Diagrams
  - Discuss Activity Diagrams
  - Discuss the benefits of Use Cases
  - Discuss how to work with Use Cases in Iterative Methods
  - Practice: Writing Use Cases and Drawing Activity Diagrams*
  
- 5. THE INCEPTION PHASE
  - Define Inception
  - Explain what happens in the Inception Phase
  - Discuss how to write Use Cases in the Inception Phase
  - Explain how long is the Inception Phase
  - Explain what artifacts may start in the Inception Phase
  - Explain what is not in the Inception Phase
  
- 6. THE ELABORATION PHASE
  - Define Elaboration
  - Explain what happens in the Elaboration Phase
  - Explain what artifacts may start in the Elaboration Phase
  - Explain what is not in the Elaboration Phase
  
- 7. DOMAIN MODEL
  - Define Domain Model and Conceptual Classes
  - Explain the motivation for creating a domain model
  - Describe how to create a domain model
  - Explain how to distinguish Attributes from Classes
  - Describe Data Types
  - Describe UML notations for representing domain models
  - Define and Explain Associations
  - Define and Explain Compositions (or commonly known as Aggregations)
  - Describe Roles and Multiplicity
  - Explain Iterative and Evolutionary Domain Modelling Process
  - Practice: Creating Domain Models*
  
- 8. SYSTEM SEQUENCE DIAGRAM
  - Define System Sequence Diagram and Conceptual Classes
  - Explain the motivation for drawing System Sequence Diagrams
  - Describe the relationship between System Sequence Diagrams and Use Cases
  - Explain how to name System Events and Operations
  - Explain how to model System Sequence Diagrams with other External Systems
  - Describe what information in System Sequence Diagrams to place in the Glossary
  - Define System Operation and Operation Contract
  - Explain how to define Operation Contracts for System Operations
  - Explain what is a Post-Condition
  - Explain how are Post- Conditions related to Domain Model
  - Explain the motivation for Post-Conditions
  - Explain how to create and write Operation Contracts
  - Discuss Operations Contracts within the Unified Process
  - Practice: Drawing System Sequence Diagrams*
  
- 9. REQUIREMENTS TO DESIGN
  - Define and describe Logical Architecture and Layers
  - Define Software Architecture
  - Describe UML Package Diagram
  - Explain how to design system with layers

- Explain the Model-View Separation principle
- Explain Static and Dynamic Modelling in designing objects
- Describe Class Responsibility Collaboration (CRC) Card as an Object Design technique
- Describe UML Interaction Diagrams: Sequence Diagrams and Communication Diagrams
- Describe UML Interaction Diagram Notation
- Describe UML Class Diagrams
- Describe UML Class Diagram Notation
- Explain how to show UML Attribute Text and Association Lines
- Explain Operations and Methods
- Explain how to show Methods in Class Diagrams

10. SUMMARY
  - Summarize important points taught in course
  - Review course objectives and how they are met

### **Course Leader**

**Dr. Danny Poo** graduated with a BSc (Hons), MSc and PhD in Computer Science from the University of Manchester Institute of Science and Technology (UMIST), England. He is currently a tenured Associate Professor in the Department of Information Systems, National University of Singapore and teaches courses on Object-Oriented Software Engineering and Enterprise JavaBeans at the undergraduate level. He is presently the Vice-Chairman, Steering Committee for the Asia-Pacific Software Engineering Conference and is actively involved in teaching professionals on Object-Oriented Analysis, Design, and Programming. He is the founder and director of Cicada Cube Pte Ltd, an NUS spin-off company specializing in Enterprise-level Search and Retrieval Solutions. **Dr. Poo is author of 4 books: “Object-Oriented Programming and Java”, Springer-Verlag, 1998; “Developing Systems Using J2EE”, Prentice-Hall, 2004, “Learn To Program Java”, Prentice-Hall, 2004; and “Learn To Program Enterprise JavaBeans”, Thomson Learning, 2005.**