



# Use Case Modeling and Solution Requirements

3 Days | Virtual and Face-to-Face

Use case models can serve as a structured tool for eliciting and organizing functional and non-functional requirements for a proposed solution. Once the overall scope of an effort has been determined, a use case diagram can depict the scope of near-term analysis work. Each use case on the diagram is subsequently refined so that all desired behavior is well-understood: the happy path, alternate paths, errors, exceptions, and edge cases.

Use cases and user stories are very different tools but can be used in conjunction with each other quite effectively. Participants will learn to “paint the big picture” with use cases, then decompose each use case into user stories. The user stories describe smaller subsets of needed functionality; this supports many agile development approaches.

Using a provided case study or a real-life project, participants will establish the scope for an overall effort. Use cases will be identified, and participants will practice developing functional and non-functional requirements to describe how the solution should behave. In addition to use case diagrams and use case descriptions, participants will practice developing business rule models, data dictionaries, Gherkin scenarios, user interface specifications, data mapping/conversion requirements, and non-functional requirements. Approximately 50% of class time will be spent in hands-on activities, with review and coaching from the facilitator. Participants will also leave with numerous job aids that will help them implement the learned concepts on their own projects.

## Learning Objectives

- Distinguish between use cases, user stories, and test cases
- Clarify the scope for a work effort
- Prioritize requirements for inclusion in solution development using plan-driven (traditional) and change-driven (iterative and agile) techniques
- Create a use case diagram to clarify near-term analysis scope
- Elicit, analyze, and communicate functional requirements that specify how users will interact with the software and how the software will respond
  - Deliver consistent, detailed use case descriptions
  - Model business rules and business data
  - Specify functionality using Gherkin scenarios
  - Incorporate usability principles when developing prototypes
  - Determine the impact of interfaces and develop interface requirements
- Identify non-functional requirements appropriate for each project
- Combine use cases and user stories to leverage the strengths of each tool

## Intended Audience

This course is designed for business analysts, systems analysts, or any other project team members responsible for developing functional and non-functional requirements. Participants are encouraged to bring examples of their requirements to class for review and feedback. This course may also be appropriate for individuals who manage business analysts. Developers and solution implementers will benefit from an understanding of how functional and non-functional requirements are elicited and analyzed.

## Prerequisites

We recommend that students first attend our [Essential Skills for Business Analysis](#) class or have experience in project scope definition, eliciting requirements from stakeholders, and understanding how business requirements fit into the entire systems development effort. We also recommend that students attend [Business Process Analysis](#) before attending this class.

## Learning Topics

Topics
<b>Key Concept Review</b>
<ul style="list-style-type: none"><li>• Define solution and transition requirements</li><li>• Review requirements categories and classifications</li><li>• Discuss the differences between business and functional requirements</li><li>• Discuss requirements implications based on the type of solution being developed (vendor package purchase, in house development, maintenance, BI)</li><li>• Learn about the software development approaches used by the team (change-driven vs. plan-driven) as it relates to solution requirements</li></ul> <p>Exercise: Complete a concept review quiz</p>
<b>Introduction to Use Cases</b>
<ul style="list-style-type: none"><li>• Define use case, user story, and test case</li><li>• Review the format and purpose of a user story</li><li>• Introduce the concept of a use case</li><li>• Discuss how test cases are formatted and developed in the participants' environment</li></ul> <p>Exercise: Define key terms</p>
<b>Determine and Visualize Scope</b>
<ul style="list-style-type: none"><li>• Define the solution scope model</li><li>• Identify high-level data and stakeholders</li><li>• Identify processes that will be included in the scope of the effort</li><li>• Develop visualizations of scope to communicate and confirm the overall effort</li></ul> <p>Exercises: Develop a context diagram and a partial functional decomposition diagram</p>
<b>Scope the Analysis Work</b>
<ul style="list-style-type: none"><li>• Derive use cases from business requirements (scope)</li><li>• Work with stakeholders to prioritize the use cases<ul style="list-style-type: none"><li>○ Learn about common approaches to prioritizing</li></ul></li><li>• Determine use cases that will belong to each phase or iteration</li></ul>

- Model the scope using a use case diagram:
  - Define actors involved with the application
  - Identify actor interactions

Exercises: Identify actors and their goals, determine relevant use cases for a solution, develop a use case diagram.

#### Developing Gherkin Scenarios

- Learn to write effective Gherkin scenarios
- Simplify complex scenarios using Gherkin backgrounds and scenario outlines
- Review common pitfalls in Gherkin scenario development
- Apply techniques to ensure that the set of Gherkin scenarios created for a use case or user story are complete
- Leverage the power of collaboration by using the Three Amigos model to develop and review functional requirements

Exercises: Develop and simplify Gherkin scenarios, identify needed scenarios with a mind map.

#### Defining Business Data and Business Rules

- Discuss the importance of understanding requirements for business data
- Review tools for documenting data entities, attributes, and attribute metadata
- Define the concept of a business rule and explore techniques for representing them

Exercises: Detail business data and business rules using various techniques

#### Designing User Interfaces

- Identify where interfaces are necessary
- Create and document prototypes
- Document report requirements, including ad-hoc and predefined
- Detail a user interface specification including screen field edits and screen functionality
- Learn about advanced user interface requirements used in business intelligence (BI)
- Incorporate usability principals into user interfaces

Exercises: Identify needed interfaces, develop a user interface specification, elicit requirements for an executive dashboard

#### Creating Use Case Descriptions

- Outline each use case to confirm a high-level understanding of the broad behavior
- Identify the primary path, alternate paths, and exception paths
- Decompose large use cases into smaller subsets, identifying reusable use cases wherever possible
- Identify “misusers” and “misuse” cases
- Simplify complex scenarios using Gherkin backgrounds and scenario outlines
- Review common pitfalls in Gherkin scenario development
- Learn to create detailed use case descriptions that link other requirements artifacts
- Learn how and where to document system user messages

Exercises: Outline a use case, develop a detailed use case description, break up a large use case, identify “misusers” and “misuse” cases

#### Analyze System Interface Requirements

- Identify required system interfaces
- Understand the most effective interface strategy for each design solution

- Write interface requirements for each interface
- Detail data conversion and mapping needs

Exercises: Choose an appropriate system interface strategy, map information coming from an external source onto internal data structures

#### Identifying Non-Functional Requirements

- Identify requirements not previously addressed by business, functional, or technical requirement categories, such as:
  - Performance requirements
  - Security requirements
  - Quality requirements
  - Scalability

- Consider which non-functional requirement types are important for your project
- Discuss the business analyst role in the development of these requirements

Exercises: Develop sample non-functional requirements, review and provide feedback on requirements developed by another team

#### Use Cases and User Stories

- Understand how use cases and user stories can complement each other
- Learn approaches for decomposing use cases into user stories
- Use a use case to organize disparate user stories and confirm the completeness of the backlog

Exercise: Decompose a use case into a full set of supporting user stories

#### Course Summary

- Course retrospective
- Develop a Post Class “Go Do It!” Plan with next steps for the student's current project.



This class is a part of the **B2T Training Business Analyst Certification Program**. For more information on the program, please see our [Certification](#) page.