
<Company Name>

<Project Name>
Use Case Specification: <Use-Case Name>

Version <1.0>

[Note: The following template is provided for use with the Rational Unified Process. Text enclosed in square brackets and displayed in blue italics (style=InfoBlue) is included to provide guidance to the author and should be deleted before publishing the document. A paragraph entered following this style will automatically be set to normal (style=Body Text).]

[To customize automatic fields (which display a gray background when selected), select File>Properties and replace the Title, Subject and Company fields with the appropriate information for this document. After closing the dialog, automatic fields may be updated throughout the document by selecting Edit>Select All (or Ctrl-A) and pressing F9, or simply click on the field and press F9. This must be done separately for Headers and Footers. Alt-F9 will toggle between displaying the field names and the field contents. See Word help for more information on working with fields.]

<Project Name>	Version: <1.0>
Use Case Specification: <Use-Case Name>	Date: <dd/mmm/yy>
<document identifier>	

Revision History

Date	Version	Description	Author
<dd/mmm/yy>	<x.x>	<details>	<name>
19 Jan 2010	1.1	Added Subflow sections as prescribed by UCM Guidelines Document. Added Scenario section accordingly	Lars Pareto

<Project Name>	Version: <1.0>
Use Case Specification: <Use-Case Name>	Date: <dd/mmm/yy>
<document identifier>	

Table of Contents

1.	Use Case Name	1
1.1	Brief Description	1
2.	Flow of Events	1
2.1	Basic Flow	1
2.2	Alternative Flows	1
2.2.1	< First Alternative Flow >	1
2.2.2	< Second Alternative Flow >	1
2.3	Subflows	2
2.3.1	< First Subflow >	2
2.3.2	< Second Subflow >	2
3.	Special Requirements	2
3.1	< First special requirement >	2
4.	Preconditions	2
4.1	< Precondition One >	2
5.	Post Conditions	2
5.1	< Post condition One >	2
6.	Extension Points	2
6.1	<name of extension point>	2
7.	Scenarios	2
7.1	<name of first scenario>	2

<Project Name>	Version: <1.0>
Use Case Specification: <Use-Case Name>	Date: <dd/mmm/yy>
<document identifier>	

Use Case Specification: <Use-Case Name>

1. Use Case Name

1.1 Brief Description

[The description should briefly convey the role and purpose of the use case. A single paragraph should suffice for this description.]

2. Flow of Events

2.1 Basic Flow

[This use case starts when the actor does something. An actor always initiates use Cases. The use case should describe what the actor does and what the system does in response. It should be phrased in the form of a dialog between the actor and the system.]

The use case should describe what happens inside the system, but not how or why. If information is exchanged, be specific about what is passed back and forth. For example, it is not very illuminating to say that the Actor enters customer information; it is better to say the Actor enters the customer's name and address. A Glossary of Terms is often useful to keep the complexity of the use case manageable; you may want to define things like customer information there, to keep the use case from drowning in details.

Simple alternatives may be presented within the text of the use case. If it only takes a few sentences to describe what happens when there is an alternative, do it directly within the flow of events section. If the alternative flows are more complex, use a separate section to describe it. For example An Alternative Flow describes how to describe more complex alternatives.

A picture is sometimes worth a thousand words (though there is no substitute for clean, clear prose). If it improves clarity, feel free to paste graphical depictions of user interfaces, process flows, or other figures into the use case to improve its clarity. If a flow chart is useful to present a complex decision process, by all means use it! Similarly for state-dependent behavior, a state-transition diagram often clarifies the behavior of a system better than pages upon pages of text. Use the right presentation medium for your problem, but be wary of using terminology, notation or figures that your audience may not understand. Remember that your purpose is to clarify, not obscure.]

2.2 Alternative Flows

2.2.1 < First Alternative Flow >

[More complex alternatives should be described in a separate section, which is referred to in the basic flow of events section. Think of the alternative flow sections like alternative behavior – each alternative flow represents alternative behavior (many times, because of exceptions that occur in the main flow). They may be as long as necessary to describe the events associated with the alternative behavior. When an alternative flow ends, the events of the main flow of events are resumed unless otherwise stated.]

2.2.1.1 < An alternative sub-flow >

[Alternative flows may in turn be broken down into sub-sections if it improves clarity.]

2.2.2 < Second Alternative Flow >

[There may be, and most likely will be, a number of alternative flows in a use case. Keep each alternative separate to improve clarity. Using alternative flows improves the readability of the use case, as well as preventing use cases from being decomposed into hierarchies of use cases. Keep in mind that use cases are just textual descriptions, and their main purpose is to document the behavior of a system in a clear, concise and understandable way.]

<Project Name>	Version: <1.0>
Use Case Specification: <Use-Case Name>	Date: <dd/mm/yy>
<document identifier>	

2.3 Subflows

2.3.1 < First Subflow >

[Sub-flows are described in their own section of the use case. Sub-flows have an identifier in front of them (S1, S2, and so on) to identify them as sub-flows.]

A sub-flow is explicitly called from a flow. When a sub-flow completes, it always returns to the line after it was called, therefore, sub-flows do not need a line that says where they return. It is similar in concept to a subroutine call in some programming languages. Each sub-flow will explicitly and clearly define all of the possible entry points into the flow.

Sub-flows called out in descriptions will be identified in bold, italics text.]

2.3.2 < Second Subflow >

3. Special Requirements

[A Special Requirement is typically a non-functional requirement that is specific to a use case but is not easily or naturally specified in the text of the use case's event flow. Examples of special requirements include legal and regulatory requirements, application standards, and quality attributes of the system to be built, including usability, reliability, performance or supportability requirements. Additionally, other requirements such as operating systems and environments, compatibility requirements, and design constraints should be captured in this section.]

3.1 < First special requirement >

4. Preconditions

[A precondition (of a use case) is the state of the system that must be present prior to a use case being performed.]

4.1 < Precondition One >

5. Post Conditions

[A post condition (of a use case) is a list of possible states the system can be in immediately after a use case has finished.]

5.1 < Post condition One >

6. Extension Points

[Extension points of the use case.]

6.1 <name of extension point>

[Definition of the location of the extension point in the flow of events.]

7. Scenarios

[Scenarios represent an instance of a use case. It is one flow through a use case. Document as many scenarios as are required to understand the system being developed, but architecturally significant, and high-risk use cases must be documented.]

7.1 <name of first scenario>

[List the flows included in the scenario]