

WRITING USE CASES

SOURCE: «USE CASES. PATTERNS AND BLUEPRINTS.» BY GUNNAR ÖVERGAARD, KARIN PALMKVIST

MAIN QUESTIONS

What is the purpose of the use case?

Which is the audience of the description?

How many description will there be of the use case?

Which level of abstraction is to be chosen?

Is a black-box, gray-box, or a white-box description to be produced?

How formal should the description be?

RULES OF WRITING USE CASE

USE A CONSISTENT STYLE — Use common words and phrases in short sentences when describing the flow. The text will be easier to read and understand.

PREPARE A TEMPLATE AND AN EXAMPLE — It will be easier to produce uniform descriptions.

USE SUBSECTIONS — When organizing the text, use subsections in the same way as you do when writing other text. The text will be much easier to read.

DESCRIBE THE FLOW AS SOMETHING THAT ACTUALLY HAPPENS — Even if the system has not been built yet, describe the flow as something that actually happens to make it easier for the reader to get an idea of the system.

USE THE ACTIVE VOICE — To make it clear who initiates the interaction, the system or the actor, use an active voice writing style.

ALWAYS USE THE NAME OF THE ACTOR IN THE INTERACTION — Do not just write «the actor.» This makes it clear which actor is involved.

CAPITALIZE THE NAMES OF THE ACTORS — This will make it clear that you are referring to an actor and not to some information about the actor inside the system.

USE SHOT PARAGRAPHS WITH ONE INPUT — A paragraph should contain no more than one actor input, which should be given as the first sentence. This makes the text more structured and readable.

BE EXPLICIT ABOUT WHAT INFORMATION IS SENT, RECEIVED, STORED, AND SO ON — Never use phrases such as *store the information about X* unless you are referring to information explicitly state one or two sentences back. Otherwise, the reader will have no clue what information you are referring to.

STATE BUSINESS RULES EXPLICITLY — In a separate section in the use-case description or in a separate document to be used in all use-case descriptions, state business rules explicitly. Reference these business rules in the description of the flow. This will make the text much easier to read, and the business rules can be maintained and enhanced separately.

CREATE SEPARATE SECTIONS FOR DETAILS — If there are many information attributes, list all the details in a separate section of the use-case description that can be referenced from the description of the flow. In this way, the description of the flow will not be cluttered with a lot of details that disturb the understanding of the flow. If the same information attributes are used in several use case, put them in a separate document and treat them in the same way as business rules.

IDENTIFY NONFUNCTIONAL REQUIREMENTS — State nonfunctional requirements explicitly in a separate section of the use-case description or in a separate document to be used in all use-case description.

DO NOT MENTION GUI DESIGN DETAILS IN THE FLOW — For example, do not identify characteristics of the GUI such as drop-down lists, lists of radio buttons or check buttons, or text fields. The GUI design will most likely be changed in a later version of the system, and if no mention is made, the use-case descriptions need not be modified.

DO NOT MENTION DESIGN OR IMPLEMENTATION DETAILS IN THE FLOW — Avoid using terms such as *applet*, *database*, and *process*. This would restrain the designers using the use-case model as input to making design decisions. Changing the implementation technology should not imply changes in the descriptions of the use-cases. Furthermore, the use-case model will become more general this way. For example, the same use-case can be used for presenting information on a web page as well as on a mobile phone screen.

SPECIFY INFORMATION RETRIEVAL AND CALCULATION MEANS — Be explicit about how information handled in the use case is retrieved or calculated.

CONFIRM ENOUGH DETAILS ARE INCLUDED — Make sure the description contains enough details so it is possible to produce a use-case realization based on it, to estimate time and resources, and to identify all major architecture components.

DESCRIBE LONG LOOPS SEPARATELY — Describe the body of a loop directly in the flow only if it is short (less than half a page). Otherwise, put it in a separate subsection. If not, the overview of the flow will be lost.

DEFINE LONG ALTERNATIVES SEPARATELY — Describe the alternatives in a condition in the flow only if they are short (less than a third of a page). Otherwise, put them in separate subsections.

SPECIFY WHERE USE CASE ENDS — Explicitly state that the use case ends at the appropriate place(s) in the flow description.

AVOID CROSS-REFERENCE — Never reference (sub-) sections in other use-case descriptions. These may be modified later, which will make references outdated.

REFERENCE EXTENSION POINTS OF OTHER USE CASES — When defining an extend relationship to another use case, only reference extension points of that use case. The flow or the description of the flow of that use case might be reorganized later, which will make other references obsolete.

REFERENCE LOGICAL LOCATIONS WITHIN A USE CASE — Never reference paragraph numbers or line numbers. These will be modified. Reference locations defined by the state of the use case or (sub-) sections within the use-case description instead; these are more stable.

DO NOT USE EXPLICITLY TAGGED KEYWORDS — The text will be harder to maintain, and the reader is most likely not a programmer and therefore not used to the notion of keywords.

PREPARE A GLOSSARY OF WORDS AND TERMS — This glossary is to be used by persons producing use-case descriptions, which means that the same terminology will be used in all the descriptions. Furthermore, the reader can look up the precise meaning of a word in this particular model and project.

DIAGRAM COMPLEX FLOWS — If the flow is complicated, prepare a state diagram or activity diagram to make it easier to follow the text.

DIAGRAM NONTRIVIAL CASES — In nontrivial cases, prepare one diagram for each use case, showing the use case in the middle of the diagram and all the elements it has relationship with in a circle around it together with the relationships. Put the actor initiating the use case on the same side of the use case from which you start to read. Included and parent use cases are normally put at the top of the circle.