Writing Effective Use Cases

Reminders

Write something readable.

Casual, readable use cases are still useful, whereas unreadable use cases won't get read.

Work breadth-first, from lower precision to higher precision.

Precision Level 1: Primary actor's name and goal.
Precision Level 2: The use case brief, or the main success scenario.
Precision Level 3: The extension conditions.
Precision Level 4: The extension handling steps.

For each step:

- Show a goal succeeding.
- Capture the actor's intention, not the user interface details.
- Have an actor pass information, validate a condition, or update state.
- Write between-step commentary to indicate step sequencing (or lack of).
- Ask "why" to find a next-higher level goal.

For data descriptions (only put Precision Level 1 into the use case text):

Precision Level 1: Data nickname.
Precision Level 2: Data fields associated with the nickname.
Precision Level 3: Field types, lengths, and validations.

Identification and Traceability:

- Give each use case a unique id.
- Provide a version number and date.
- Reference specific information, no whole documents.
- Do not turn use cases into an external treasure hunt for requirements or behaviors that belong in the use case.

Design Scope	Goal Level
Organization (black-box)	Very High Summary
Organization (white-box)	Summary
System (black-box)	User-Goal
System (white-box)	Subfunction
Component	Design Details (too low)

The Writing Process

1. **Name the system scope and boundaries.**
   Track changes to this initial context diagram with the in/out list.
2. **Brainstorm and list the primary actors.**
   Find every human and non-human primary actor, over the life of the system.
3. **Brainstorm and exhaustively list use** goals for the system.
   The initial Actor-Goal List is now available.
4. **Capture the outermost use cases to see who really cares.**
   Check for an outermost use case for each primary actor.
5. **Reconsider and revise the summary use cases. Add, subtract, or merge goals.**
   Double-check for time-based triggers and other events at the system boundary.
6. **Select one use case to expand.**
   Consider writing a narrative to learn the material.
7. **Capture stakeholders and interests, preconditions, and guarantees.**
   The system will ensure the preconditions and guarantee the interests.
8. **Write the main success scenario.**
   Use 3 to 9 steps to meet all interests and guarantees.
9. **Brainstorm and exhaustively list the extension conditions.**
   Include all that the system can detect and must handle.
10. **Write the extension-handling steps.**
    Each will end back in the main success scenario, at a separate success exit, or in failure.
11. **Extract complex flows to sub use cases; merge trivial sub use cases.**
    Extracting a sub use cases is easy, but it adds cost to the project.
12. **Readjust the set: add, subtract, merge, as needed.**
    Check for readability, completeness, and meeting stakeholders' interests.

Writing Effective Use Cases

Field	Question
Use Case Title	1. Is it an active-very goal phrase that names the goal of the primary actor?
	2. Can the system deliver that goal?
Scope and Level	3. Are the fields filled in?
Scope	4. Does the use case treat the system mention in Scope as a black box? *The answer must be "yes" if it is a system requirements document, but may be "No" if the use case is a white-box business use case.*
	5. If the system in Scope is the system to be designed, do the designers have to design everything in it and nothing outside it?
Level	6. Does the use case content match the state goal level?
	7. Is the goal really at the stated goal level?
Primary Actor	8. Does he/she/it have behavior?
	9. Does he/she/it have a goal against the system under discussion that is a service promise of the system under discussion?
Preconditions	10. Are they mandatory, and can they be set in place by the system under discussion?
	11. Is it true that they are never checked in the use case?
Stakeholders and Interests	12. Are they named and must the system satisfy their interests as stated? *Usage varies by formality and tolerance.*
Minimal Guarantees	13. Are all the stakeholders' interests protected?
Success Guarantees	14. Are all the stakeholders' interests satisfied?
Main Success Scenario	15. Does it have 3 to 9 steps?
	16. Does it run from trigger to delivery of the success guarantee?
	17. Does it permit the right variations in sequencing?
Each Step in Any Scenario	18. Is it phrased as a goal that succeeds?
	19. Does the process move distinctly forward after its successful completion?
	20. Is it clear which actor is operating the goal – who is "kicking the ball"?
	21. Is the intent of the actor clear?
	22. Is the goal level of the step lower than the goal level of the overall use case? Is it, preferably, just a bit below the use case goal level?
	23. Are you sure the step does not describe the user interface design of the system?
	24. Is it clear what information is being passed in the step?
	25. Does it "verify," as oppose to "check" a condition?
Extension Condition	26. Can and must the system detect and handle it?
	27. Is it what the system actually needs?
Technology and Data Variations List	28. Are you sure this is not an ordinary behavioral extension to the main success scenario?
Overall Use Case Content	29. To the sponsors and users: "Is this what you want?"
	30. To the sponsors and users: "Will you be able to tell, upon delivery, whether or not you got this?"
	31. To the developers: "Can you implement this?"
	32. To the testers: "Can you verify and validate this?"

All answers should be "Yes.

The goal of a use case is to clearly communicate and showcase the requirements.
Be precise and specific. Clarify customer needs. Use cases do not drive implementation.