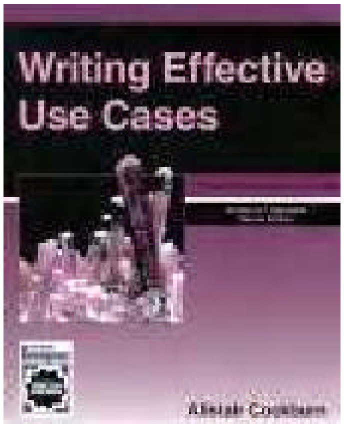
WRITING EFFECTIVE USE CASES

Alistair Cockburn
Humans and Technology

pre-publication draft #3, edit date: 2000.02.21

published by Addison-Wesley, c. 2001.



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.
Highlight 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

Icons

Design Scope	**Goal Level**
🏠 Organization (black-box)	☁ Very high summary
⌂ Organization (white-box)	🪁 Summary
■ System (black box)	〰 User-goal
□ System (white box)	🐟 Subfunction
🔩 Component	🐚 too low

For Goal Level, alternatively, append one of these characters to the use case name:

Append "+" to summary use case names.
Append "!" or nothing to user-goal use case names.
Append "-" to subfunction use case names.

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 user goals for the system.

   The initial Actor-Goal List is now available.
4. Capture the outermost summary 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 (MSS).

   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 ends back in the MSS, 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 case 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.