# Use-Case Foundation

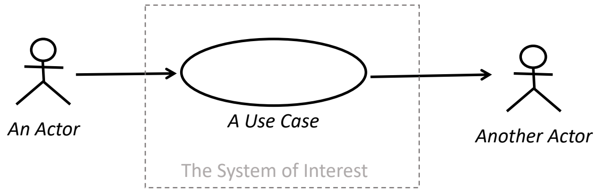
*Ivar Jacobson and Alistair Cockburn*

To get to the heart of what a system must do, focus on who or what will use it, and then look at what the system must do for them to help them achieve their goals.

A *use case* is all the ways of using a system to achieve a goal of a particular user.

## Core Concepts

1. A system of interest
2. A primary actor with a goal
3. A flow of events (there will be several)
4. A use case to collect those flows.



**A use case** – A use case is all the ways of using a system to achieve a goal of a particular user.

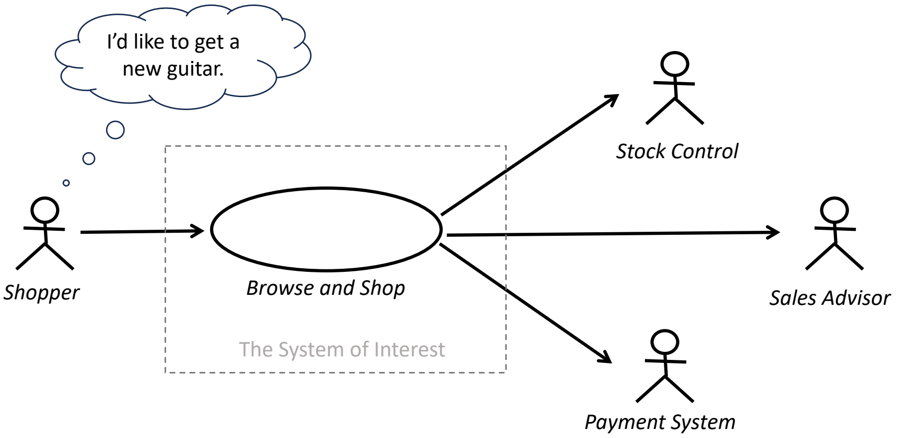
Notes:

- This includes all the successful, challenged and failure paths.
- It may be described textually or visually.
- It is independent of implementation, technology, and platform.

**The System of Interest** – The system used to achieve the goal.

**An Actor** – "Actor" is intended to cover anything with behaviour. It can be a person, an organization, a piece of software, or any combination.
An actor identifies a role played when interacting with the system. A use case might involve many actors: The actor that initiates a use case is known as the "primary actor" and the actors called upon by the system are known as "supporting actors".

**The Goal** – The reason that the user will use the system and the value that they will receive when successfully using the system.
A simple example:



Primary Actor

In this case a shopper with the goal of selecting and purchasing a product.

The System of Interest

In this case an on-line portal providing advice on all things musical.
One of this system's use cases is 'Browse and Shop'

Supporting Actors
Other Actors that can be involved in the successful completion of the use case. These can be other systems or other people.
In this case the system of interest needs to interact with a Stock Control System, a Payment System, and for specialist, high value products a Sales Advisor.

## Underlying Principles

1. Use Cases apply to systems of all types and sizes: businesses, IT systems, physical systems or any combinations thereof.

2. Use cases help you understand the big picture: the system's purpose and how it will be used.

3. Use cases focus on value: the users' goals and how best to achieve them.

4. Stakeholder involvement is essential: bring all the involved parties together to establish the intent and scope of the system.

5. A use case tells the whole story, as a story, from the initial event to the realization of the value it provides or the eventual failure if it can't be met. It includes how to handle any problems and alternatives that may occur on the way.

6. Use cases trigger conversations: While discussing the possible alternate flows, you and your co-writers will think of missing steps and missing alternatives. These conversations help you find situations that often get overlooked.

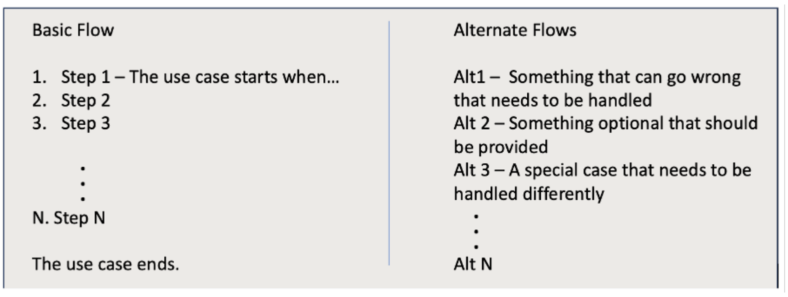
7. Prioritize readability: the goal is to communicate the big picture to everyone involved, generating comments, spotting any gaps, and getting their buy-in.

8. The amount of detail and the format used will vary to match your circumstances: You can start with a sketch of the flow of events and add detail as needed.

9. A use case can be implemented in stages: develop and put into place some key flows of a use case early to capture value and feedback, add less used or less critical flows over time strategically.

# A Sample Use Case

The use case is presented as a network of flows, each describing a path to value.



Basic Flow

1. Step 1 – The use case starts when…
2. Step 2
3. Step 3

⋮

N. Step N

The use case ends.

Alternate Flows

Alt1 – Something that can go wrong that needs to be handled
Alt 2 – Something optional that should be provided
Alt 3 – A special case that needs to be handled differently
⋮
Alt N

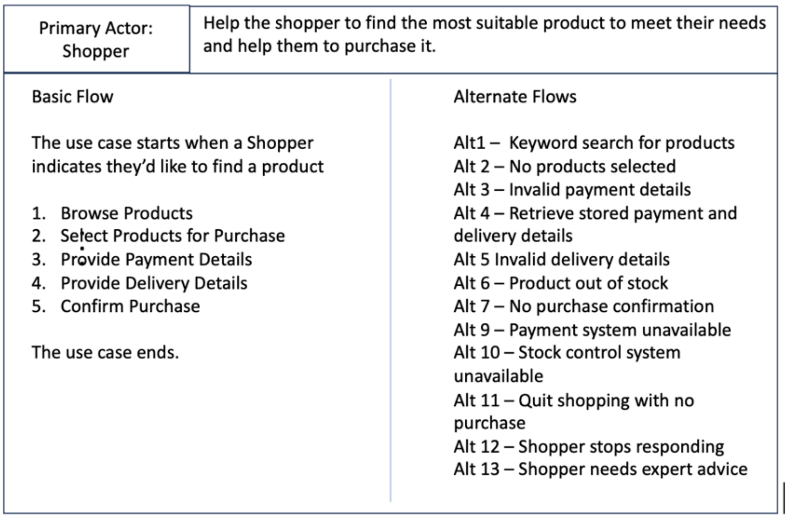
**The Basic Flow** – The normal, happy path to value often referred to as the 'main scenario' or the 'happy path.' This is described as a simple sequence of steps each of which involves the system and / or one of the actors doing something.

**Alternate Flows** – A list of all the special cases, alternative paths, optional steps, and errors that need to be handled.

The key aspect of a use case is its structure: the way it identifies the basic and alternate flows – this acts as a map of how the system will be used. The flow of events can be described as simply as a bulleted list of steps and alternatives, or elaborated to fully describe what should happen at each step or within each alternative. It can be described in text, as above, or in some graphical form.

What is important is the *accuracy* of the flow of events and not *how detailed* you write out the steps and alternatives.

A simple example:



Primary Actor: Shopper

Help the shopper to find the most suitable product to meet their needs and help them to purchase it.

Basic Flow

The use case starts when a Shopper indicates they'd like to find a product

1. Browse Products
2. Select Products for Purchase
3. Provide Payment Details
4. Provide Delivery Details
5. Confirm Purchase

The use case ends.

Alternate Flows

Alt1 – Keyword search for products
Alt 2 – No products selected
Alt 3 – Invalid payment details
Alt 4 – Retrieve stored payment and delivery details
Alt 5 Invalid delivery details
Alt 6 – Product out of stock
Alt 7 – No purchase confirmation
Alt 9 – Payment system unavailable
Alt 10 – Stock control system unavailable
Alt 11 – Quit shopping with no purchase
Alt 12 – Shopper stops responding
Alt 13 – Shopper needs expert advice

Note – If you look closely you might find issues and problems, missing steps and missing alternatives, with this use case,. This is deliberate. This is exactly the sort of conversation that we want the use-case to start.