Mastering the Requirements Process

Second Edition

Suzanne Robertson
James Robertson

Mastering the Requirements Process

Second Edition

of this determination is the product use case. Naturally, we suggest you describe it using a product use case scenario.

Earlier in this chapter we introduced this business use case scenario:

Business Event Name: Passenger decides to check in.

Business Use Case Name: Check passenger onto flight.

Trigger: Ticket, record locator, or identity and flight.

Preconditions: The passenger must have a reservation.

Interested Stakeholders: Check-in agent, marketing, baggage handling, reservations, flight manifest system, work flow, security, destination country's immigration.

Active Stakeholders: Passenger (trigger), check-in agent.

1. Locate the passenger's reservation.
2. Ensure the passenger is correctly identified and connected to the right reservation.
3. Check the passport is valid and belongs to the passenger. See procedure guidelines EU-175.
4. Attach the frequent-flyer number to the reservation.
5. Allocate a seat.
6. Get correct responses to security questions.
7. Check the baggage onto the flight.
8. Print and convey to the passenger the boarding pass and bag tags.
9. Wish the passenger a pleasant flight.

Outcome: The passenger is recorded as checked onto the flight, the bags are assigned to the flight, a seat is allocated, and the passenger is in possession of a boarding pass and bag claim stubs.

To make decisions about the product boundary for this business use case, we need to define the constraints. We also need input from the stakeholders who understand the technical and business implications and the possibilities for the product boundary along with the business goals for the project.

Let's suppose that you have that information. That is, you and the stakeholders have decided what product to build. The product use case scenario shows what the product is to do:

Product Use Case Name: Check passenger onto flight.

Trigger: Passenger's name + passenger's passport number.

Preconditions: The passenger must have a reservation.

Interested Stakeholders: Check-in agent, marketing, baggage handling, reservations, flight manifest system, work flow, security, destination country's immigration.

Actor: Check-in agent.

1. Locate the passenger's reservation.
4. Attach the frequent-flyer number to the reservation.
5. Allocate a seat.
7. Check the baggage onto the flight.
8. Print the boarding pass and bag tags.

Outcome: The passenger is recorded as checked onto the flight, the bags are assigned to the flight, a seat is allocated, and the passenger's boarding pass and bag claim stubs are printed.

Note the differences between the business use case scenario and its corresponding product use case scenario. For example, the appropriate stakeholders decided that ensuring the correct passenger is attached to the reservation (step 2) is outside the scope of the product. Thus step 2 does not show up on the product use case scenario. Similarly, steps 3 and 6 are defined as being outside the product scope, so they are also omitted from the product use case scenario. (We have retained the same step numbers so that you can compare the business use case and the product use case. In practice, you would renumber the steps of the product use case so they consecutive.)

Note also that, instead of a list of active stakeholders, we identified the check-in agent as an actor. An actor is a person or system that has a direct interface with the product. In other words, the check-in agent will be the hands-on user for this product use case.

Bear in mind that our example product use case scenario reflects a particular set of decisions about the product scope. If the stakeholders had decided on a different product, then naturally the product use case scenario would be different.

As you will see in the next chapters, the product use case is the basis for writing the functional and nonfunctional requirements.

Summary

A scenario is a tool for telling a story. We have discussed how to write scenarios for both the story of the business use case and the story of the product use case.

The business use case scenario is intended to help you and your stakeholders come to a coherent understanding of the business rules that apply to a specific business event. Writing the scenario tests whether sufficient trawling

has been done, and whether the requirements analyst needs to ask more questions and investigate further.

The product use case scenario specifies the boundary or scope of the product for a specific business use case. Constraints, goals, and technical and business guidance are necessary input for defining the product use case boundary.

You can use a very similar form of scenario for the business use case and the product use case, providing (for the purposes of communication and traceability) you make it clear which one is which.

This page intentionally left blank

7 Functional Requirements

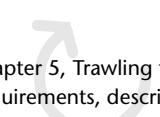
in which we look at those requirements that cause the product to do something

The functional requirements specify what the product must do. They describe the actions the product must carry out to satisfy the fundamental reasons for its existence. For example, the functional requirement

> ***The product shall predict which road sections will freeze within the selected time parameters.***

describes an action the product must take if it is to carry out the work for which it is intended. The intention is to understand the functional requirements and so convey to the developers what the product is required to do for its intended operator.

In Chapter 5, we described how to gather the requirements. In Chapter 6, we described how the requirements analyst uses business use case scenarios to illustrate the functionality for the interested stakeholders, and product use case scenarios to define ideas for the product boundary. When the scenarios have been agreed upon, the requirements analyst writes the functional requirements for the product use cases. This process is illustrated in Figure 7.1.



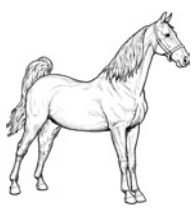
Chapter 5, Trawling for Requirements, describes how to gather requirements. Chapter 6, Scenarios and Requirements, explores how to use scenarios to describe business use cases and product use cases.

Agility Guide

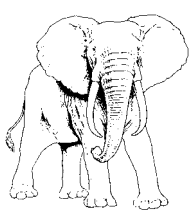
To get the most out of this chapter, it is necessary to understand the difference between a requirement and a solution. It is also necessary to understand that while we are describing how to write requirements, the most

important thing is to understand and communicate them in the way that works for your project.

Rabbit projects have short durations between releases, and as much as possible, they avoid writing the requirements before starting to build a product to meet them. The advice in this chapter that applies to rabbit projects is simple: Rushing to a solution before understanding the requirement—note that we did not say "writing the requirement"—generally wastes time. Without knowing the underlying reason for the solution (in other words, the essential requirement), the solution is likely to solve the wrong problem. But, armed with an understanding of the requirements, rabbit projects for the most part bypass writing the atomic requirements and use their scenarios to communicate the functional requirements.



Horse projects usually have a need to write their requirements. Compared to rabbits, horses have longer release cycles and geographically scattered stakeholders. This wider distribution of project participants puts greater emphasis on communicating requirements in a more precise and consistent form. It is vital that team members have a solid understanding of what a functional requirement is and what the functional requirements do for the eventual product. That said, horse projects should maximize the roles that their scenarios and a business class model play in communicating the functional requirements.



Elephant projects need a complete and correct requirements specification. All of the information in this chapter is relevant to them, but the discussion on level of detail is particularly pertinent.

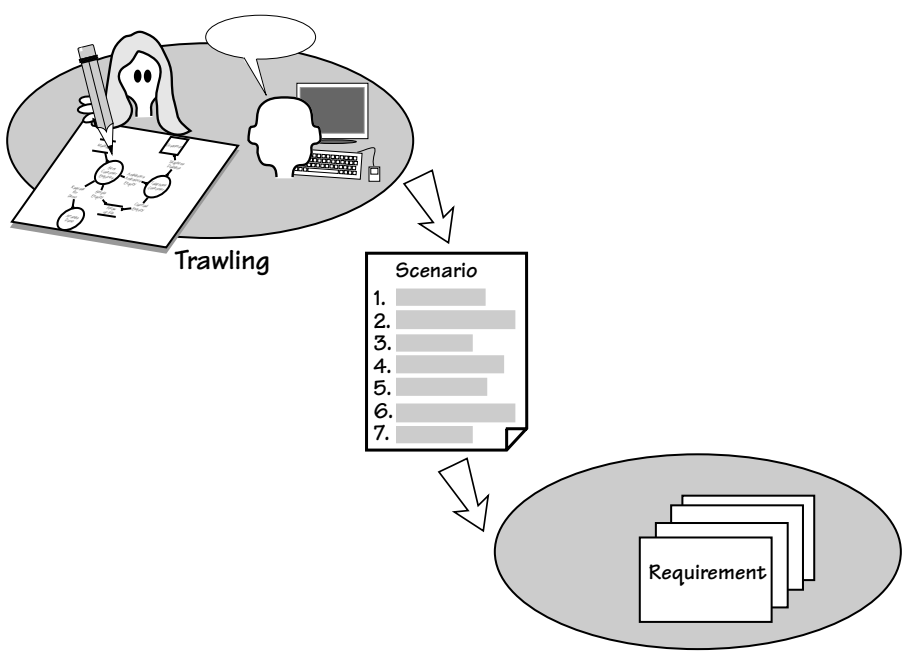


Figure 7.1

The functionality of the work is described during the trawling activity. You usually communicate this functionality back to the stakeholders by writing a scenario. You then write functional requirements by referring to the scenario. The end result is a collection of functional requirements that, taken together, describe the product's contribution to the work.