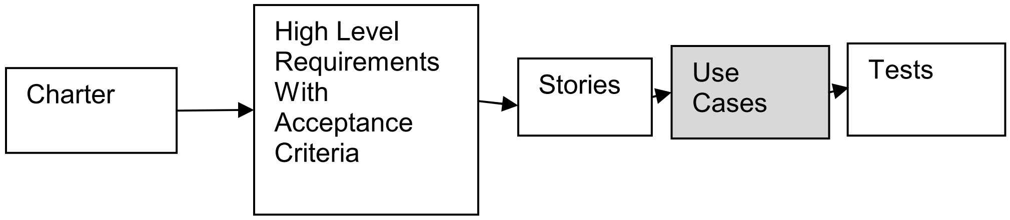
7

Collaborating On Use Cases

"What we've got here is a failure to communicate"
Cool Hand Luke

Debbie explains to Cathy how to create use cases. The triad constructs a use case for the user story check out a CD. Issues in collaboration are discussed.



Use cases from user stories

Debbie explains to Cathy about how they discover the details of a user story. We use a common technique called a use case. (**Footnote – Event/Response is shown in another chapter **). "The use case describes a sequence of actions and reactions between the user and the software. There are several formal templates for a use case, but Tom and I prefer a simple one. It's adapted from Alistair Coburn's book on use cases (**Reference**). The first story we're going to work on is Checkout CD."

Story ID:	1	**Name:**	Checkout CD	**Related Stories:**	None		
Description:	As the clerk, I want to check out a CD for a customer.						
Acceptance Criteria:	Check out CD. See if recorded as rented and rental info is correct.						
Details:	Cary	**Business Value Estimate:**	100	**Story Point Estimate:**	13	**Bang For The Buck:**	8
Notes:	Make sure user interface also works for Larry, the Part-Time Clerk						

Debbie continues, "Often the use case is part of a workflow that either involves other use cases or actions that occur outside of the software system. Let's track the steps which occur when one of your customers rents a CD with the manual process. Cathy, can you describe the current flow for checking out a CD?"

"Sure", Cathy replies. She writes on a whiteboard the steps. After a few additions and corrections, the steps look like:

***Start Background

Customer selects a CD from the cases on the wall.
    Cases are empty, but just have cover page
Customer brings CD case to the Clerk
Clerk gets actual CD in a case from shelf behind counter
Customer presents his driver's license
Clerk pulls out the rental card from the CD case
Clerk writes down the customer's name and the current date on the rental card
Customer signs the rental card
Clerk files the rental card in the CD case with the title and stores the card in a box on the back shelf.

***End Background

Debbie starts, "The software system will not replace all of these steps. A bigger system, like those red DVD rental kiosks might, but not the system we're replacing. So we only need to concentrate on the steps involved with recording the rental itself. Based on your current workflow, in the new system, these might be:"

***Start Background

Clerk enters customer identification and CD identifier into the system
System records the information
System prints a form that the customer signs

***End Background

"These steps form the main course for a use case. Some people call this the happy path, since it assumes that nothing goes wrong. The template for a basic use case looks like this:"

***Start Background

Name: Identifier to easily reference it by
Description: Brief note
Actor: Who initiates the use case
Pre-conditions: What must be true before use case is initiated
Post-conditions: What is true if use case successfully executes
Main Course: Steps that show sequence of interactions

***End Background

"The Actor is almost always the role (the user) in the user story. The name of the use case can be the name of the user story. The brief description can be the same as the description on the user story. The pre-conditions describe the required state of the system prior to starting the main course. The post-conditions are how the state of the system has changed. They describe the tests we need to run to ensure that the implementation successfully implemented the use case. The pre-conditions represent the setup required for those tests. So based on the story and the steps, the basic use case looks like":

***Start Background

Name: Checkout CD
Description: Check out a CD for a customer
Actor: Clerk
Pre-conditions: Customer has identification. CD has identity.
Post-conditions: CD recorded as rented. Rental form printed.
Main Course:
1. Clerk enters customer identification and CD identifier into the system
2. System records the information
3. System prints a form that the customer signs

***End Background

Debbie states, “Now that we’ve identified the main course, we can add additional information to the use case. During the use case, conditions can occur that do not allow it to reach its post-conditions. We call these conditions exceptions. There are exceptions that can happen in almost any use case. For example, you could have a power failure and the computer could go down. Or there could be some inexplicable software failure.” Tom interrupts, “But that never ever happens with Debbie’s code”.

After a nod to Tom, Debbie resumes, “We deal with those sorts of exceptions with an overall response scheme, such as filling out the rental form manually. There are specific exceptions that can occur during the main course. For example, it’s possible that the customer identification is not recognized when the clerk enters it. We identify this exception with an item that is numbered with the step in the main course where it could occur. We add a letter to denote it as an exception. So this might look like:”

***Start Background

Exceptions:
1a. Customer identification not recognized.

***End Background

“We could have the clerk enter it again. So we make note of that action with the exception. We put that beneath the exception, such as:”

***Start Background

Exceptions:
1a. Customer identification not recognized.
    Repeat step 1.

***End Background

“But suppose that this step is repeated and the customer identification is still not recognized. Now this is a decision for you Cathy. It could be that the customer identification is not very readable or it could be a fake customer identification. It’s not up for the developer unit to determine how to handle this exception. It’s the business unit’s responsibility. What should the system do?” Debbie asked.

Cathy replies, “I suppose the clerk could take down additional information from the customer and rent it anyway. We might lose a CD or two for fake ids, but we would avoid making real customers unhappy. I’ll check with Sam, but for now, let’s do that.”

Debbie says, “Okay, so let’s call those steps ‘record customer id’ and ‘checkout manually’. You can come up later with the exact details. Let’s put that down. Since it is a different exception, we give it a different letter. So the two exceptions that can occur during step 1 are:”

***Start Background

Exceptions:
1a. Customer identification not recognized on first try
    Repeat step 1.
1b. Customer identification not recognized on second try
    Clerk performs “record customer id” and “checkout manually”
    Use case exits

***End Background

Debbie asks, “Do you have any business rules that apply to the rental process? Our definition of a business rule is something that is true, regardless of the technology.”

“We do have one that is hard to enforce, given the way we do things now,” Cathy replies. “Sam and I agreed that a customer should not be able to rent more than three CDs at any time. The rule limits our losses in case the customer skips out on us. It also keeps more CDs in stock for other customers.”

Debbie responds, “So you want the checkout abandoned in that case. Let’s get that one down. Later on you can change your mind, such as increasing the limit for a particularly responsible customer. But that would involve a little more coding.”

***Start Background

Exceptions:
1c. Customer violates CD Rental Limit

Clerk notifies customer of violation

Use case abandoned

Business Rule:

CD Rental Limit

A customer can only have three CDs rented at any one time

***End Background

“One other facet of use cases is the alternative. An alternative is a flow that allows the use case to be successful even if the some condition occurs.   For example, the printer might jam when printing out the rental form. In this case, the clerk could fill out the form manually. So we’d add an alternative to step 3:”

***Start Background

Alternatives

3a. Printer jams.

Clerk fills out form

Use case ends successfully.

***End Background

“This use case is fairly straightforward. If there were several alternatives, we’d make up separate use cases to keep each one simple. We know from experience with testing that each alternative always requires more tests. If the number of tests for an alternative seemed large, then we definitely would split up the use case. If it took me a while to implement an exception or you could use the system without the exception being handled, then we’d make up separate stories for either an individual or a group of exceptions. Those stories would be related back to the one for the main use case.”

The Tests

Debbie continues, “Now that we have the use case for this story, it’s time to outline the tests to write against it. We need at least one test for the main course, each exception, and each alternative.   Later we will make up specific examples for each of these tests. The use case suggests these tests:”

***Start Background

Rent a CD – main course

One Bad Customer ID – enter customer id wrong once

Two Bad Customer IDs – enter customer id wrong twice

CD Rental Limit – customer who has three CDs and rents another one

Printer Jam – simulate a printer jam (maybe out of paper)

***End Background

“As I mentioned before, the pre-conditions convert to the setup for these tests and the post-condition are the expected results.   If there is an exception, we should see something other than the post-condition, since the use case did not completely execute. Tom will be talking about the tests more in the next chapter. (** Ref Forward**)”

“If a business rule such as CD Rental Limit is complicated, then you would have tests that exercise just the business rule. The test scenarios for the use case would exercise two conditions – when the business rule passes and when it fails. If there was a particular risky aspect to the business rule, then you might create more test cases for the scenario.”

The Procedure

“In general, use cases,” Debbie states, “are more than just our joint understanding of how things should work.   They also document the computer part of the workflow. If you create a user’s manual for the clerks, you could just put the use case into the manual. Or you could rephrase it, so that it reads better for a non-computer savvy person. Each use case captures all of the issues for a particular operation. So it is a document that it is worth making correct.”

Communication

In lean/agile development, the triad communicates more through face-to-face interactions, than through written documentation. The user stories, use cases, and acceptance tests are developed interactively. Face-to-face meetings with a whiteboard to record and display ideas are the most effective form of communication. (**Ref Alistair Cockburn**). If they are separate, then having a video meeting with a shared desktop is an alternative. (**Footnote - video conferencing sites such as oovoo.com or skype.com**). Let's take a bird's eye view of how Tom, Debbie, and Cathy interact in these face-to-face meetings.

All three perform 'Active Listening' (**Ref **). In Active Listening, they listen to understand. If they understand, they acknowledge their understanding with a "I follow you" gesture – a nod or a verbal affirmation. They focus on what the speaker is saying, not what they are going to say next. If they need clarification, they ask for it, such as "Give me an example."

When recording ideas on the whiteboard, they practice what I term Active Writing. Recording on a whiteboard, rather than on paper provides instant feedback. When a person is recording ideas, the speaker waits till each idea is recorded before proceeding to the next. That keeps the pace reasonable. If an idea is not recorded clearly, the group can immediately suggest a correction. Ideas are clarified in person and recorded with a common understanding.

When documenting ideas, they recognize that each person may have a preferred way of receiving information. Some like textual descriptions in either prose format or outline form. Others would rather view diagrams and charts than text. If necessary, information is recorded in both formats, so that both preferences can be honored.

When they are brainstorming or describing ideas, they realize that each person can have different responses. Some people get their energy from verbal discussions with other people (extroverts) (**Ref Myers-Briggs Type Indicators **) while others process their ideas internally (introverts). So the triad has mechanisms for allowing both to interact. They have times when people think individually and write down thoughts and when people discuss thoughts as a group.

They understand that some people like to see the big picture without getting into details (intuition) while others want to see the details (sensing). So they have both brief requirements such as user stories and detailed requirements, such as use cases. They recognize that usually progress can be made without first gathering all the details. But they acknowledge the times that work needs to stop if an important detail is unknown.

They realize that clarity is important. So they develop a common terminology. The developer and test units accept that the terms and definitions come from the business unit. The business unit understands that the ambiguous terms they use may have to be renamed in order to provide clarity.

Communication, even when you understand it, can be difficult. We each have our own preconceived notions as to what is clear and what is correct. I was working with a colleague on developing a powerpoint presentation for a conference. We had gone through the slides together and had a good working understanding of what we were going to present.

A little before the presentation, I got a printed copy of the slides from him. They were printed four to a page – two rows and two columns. I looked at the printout and exclaimed that he had rearranged the slides. He looked at the printout and said that he had not.

Here's a question for you. If the first slide is in the upper left portion of the page and the fourth slide is in the bottom right portion, where should the second and third slides be located?

If you said the second should be upper right and the third should be lower left, then you would have been as surprised as I was. If you said the reverse, then you would have had no issue with the printout.

Communication is about more than just words. It's about how you organize those words.

Summary

- A use case details a user story
- A use case states the pre-conditions, post-conditions, and main course.
- A use case may have exceptions which do not allow it to successfully complete.
- A use case may have alternative ways to achieve the post-condition.
- If a use case is large, exceptions and alternatives may become user stories.
- Use cases suggest acceptance tests.
- Collaboration requires an understanding of the difference in how people create and process information.