Chapter 1
GETTING STARTED

Use Cases are used to describe the outwardly visible requirements of a system. They are used in the requirements analysis phase of a project, as well as contributing to test plans and user guides. They are used to validate a proposed design and to ensure it meets all requirements. Use Cases are also used when creating a project schedule, helping to plan what goes into each release.

This book will give practical guidelines for applying Use Cases to a project. We will cover a project from it's initial inception ("Hey! How about..."), to just before we start to actually build it. We will also look at applying Use Cases to testing the system code and to creating User Manuals.

In this book we'll look at Use Cases from many viewpoints, showing how they contribute to the architecture, scheduling, requirements, testing, and documentation of a project. We'll look at the system from the user's point of view, discuss issues such as boundaries, interfaces, and scoping, and look at how to break a really large system into manageable chunks. We also look at who would be interested in the documentation you'll be writing and what to look for in a review. We need to consider such things as how to build flexibility into a system, making a build versus buy decision, and how to turn the documents into an Object Oriented design.

This book will not go into depth on software architecture, project planning, testing, process, or methodology. Instead, you will find a listing of books we like on these topics in the Resource List at the end of the book. There are a number of good books on these topics; the resource list will give you a starting point.

Chap01.fm Page 14 Friday, October 27, 2000 11:36 AM

An Iterative Software Process

Use Cases can be used in many processes. Our favorite is a process that is iterative and risk driven. It works well with Use Cases and Object Oriented methodologies. It helps identify and address risks early in the process, leading to more robust and better quality systems. We will give a very brief description of the process here, showing where Use Cases fit into the process. Subsequent chapters will go into more detail on how Use Cases are used at each phase.

This iterative, risk driven process is divided into 4 primary phases: Inception, Elaboration, Construction, and Transition.

During the Inception phase you will determine the scope of the project and create a business case for it. At the end of the Inception phase you should be able to answer the question “Does it make good business sense for us to continue with this project?”

During the Elaboration phase you will do requirements analysis and risk analysis, develop a baseline architecture, and create a plan for the construction phase.

During the Construction phase you will progress through a series of Iterations. Each Iteration will include analysis, design, implementation, and testing.

During the Transition phase you will complete the things that make what you developed into a product. This can include beta testing, performance tuning, and creating additional documentation such as training, user guides, and sales kits. You will create a plan for rolling the product out to the user community, whether internal or external.

So where do Use Cases fit into all this? In the Inception phase, high level Use Cases are developed to help scope out the project. What should be included in this project, and what belongs to another project? What can we realistically accomplish given our schedule and budget?

In the Elaboration phase, you will develop more detailed Use Cases. These will contribute to the risk analysis and the baseline architecture. The Use Cases will be used to create the plan for the Construction phase.

In the Construction phase, Use Cases will be used as a starting point for developing test plans. More detailed Use Cases may be developed as part of the analysis of each Iteration. Use Cases provide some of the requirements that have to be satisfied for each Iteration.

Chap01.fm Page 15 Friday, October 27, 2000 11:36 AM

In the Transition phase, Use Cases can be used to develop user guides and training.

An Example Project

Throughout this book we will use an example project. We will work through all the techniques using the same example. The notation we will use is the Unified Modeling Language. See Appendix??? For an overview of the UML.

The example we will be following is for an order processing system for a mail order company. So let's start at the very beginning when four friends gathered around a table after dinner, and someone got an idea.

"This is crazy!" Dennis exclaimed, sitting down next to Tara, almost spilling his coffee.

"What is?" said Lisa, sitting down with Gus, and her own cup of Mocha.

"The fact that I can't find a single supplier that will give me reasonable service and parts without all the headaches! I've got one supplier who has great service, and I like dealing with him. But he takes 3 weeks to get me even the simplest order! And the other one, oh, they're something else. Sure, I can get orders within 3 days, but half the time the orders are wrong, and when I call them back about it, they make it sound like it's my fault! It's almost as if they are TRYING to make me go elsewhere."

"Yes, I know what you mean," says Lisa. "I've had my own problems with mail order companies. You'd think they would pay more attention to their customers!"

"I really think I could do a better job myself. I sure know a lot about what not to do."

This has been a common complaint from Dennis in the last several months, and by now the group is well acquainted with it. Tara suddenly smiles, and pipes up with "Why don't you start one?" "Start one what?" Dennis mutters into in his coffee.

"Start a mail order company! What would you do to fix the problems you've seen?"

Chap01.fm Page 16 Friday, October 27, 2000 11:36 AM

"Well, it seems like automating the order processing would help a lot. It would also let me run the company myself for a while. But I don't know anything about software."

At this point, Gus joins the conversation. "You need to plan it out. I learned a method in my OO class we could use. And we could help! By putting us and all of our experiences together, we could figure out how to use our different skills in the right places, and work out the sections we don't know!"

"OO? What's that? You know I'm not a programmer. I don't know anything about programming languages."

"No, OO isn't about a programming language. It's a way of thinking about a problem, a way of modeling and breaking it down into identifiable objects so you can work with them. It really doesn't matter what the problem is, whether it's a programming problem or a problem like starting a new business. It's just an approach on how you look at it."

"Hmmm...", mused Dennis, liking the idea the more he thought about it. "And you would all be willing to help?"

"Sure!" "Why not?" "Sounds like fun!"

"Well... Okay! So, Gus... where do we start with this all-dancing-all-singing magical OO process?"

Before getting into writing Use Cases, you have to gather some information that will provide a starting point. This is part of the Inception phase of the Rational Objectory Process. The information we collect includes a project description, market factors that effect our project, risk factors for our project, and assumptions we are making. The rest of this chapter will touch on all those sources of information.

The Project Description

So you have an idea for a project. The next step is to write out a description of what you plan to do. It sounds simple, and it can be. But the larger the group you have writing this description, the longer it will take, and the more complex it will be. It is best to have just a couple of people write out a brief, but complete, description of the project.

Chap01.fm Page 17 Friday, October 27, 2000 11:36 AM

The project description should range in size from one short paragraph for a small project, up to no more than a couple of pages for a really large project. This is not a description of the requirements, but a description of the project in general.

The biggest mistake made at this point is not writing the description. Usually this is because everyone thinks they know what the project is about, so why write it down. We have spent several days in meetings while the project team argued about what a one paragraph description should say. Until you write it out, you can't be sure everyone agrees on the same project description.

"So let's get started. We need to write out a description of order processing in normal every day language. This will become the problem statement, a way to start getting the requirements for the project."

"Why do we need to write it out? We're all familiar with ordering products from mail order companies. This seems too formal for such a simple problem."

"Well, we should write it all down to make sure everyone has the same idea. Even if you were working alone, it's still a good idea to write it down so you don't leave out something important. Besides, it'll give us someplace to start, and we can add to it as we go along. Here, I'll start."

> We are developing order processing software for a mail order company called National Widgets, which is a reseller of products purchased from various suppliers.
>
> Twice a year the company publishes a catalog of products, which is mailed to customers and other interested people.

Figure 1-1: Problem Description

Chap01.fm Page 18 Friday, October 27, 2000 11:36 AM

"You think twice a year is good? What if our products change faster than that?" "Remember, this is just to start us off. We will add to it and change it as we get farther along, and understand more about what's going on. Let's keep going."

> Customers purchase products by submitting a list of products with payment to National Widgets. National Widgets fills the order and ships the products to the customer's address.
>
> The order processing software will track the order from the time it is received until product is shipped.
>
> National Widgets will provide quick service. They should be able to ship a customer's order by the fastest, most efficient means possible.

Figure 1-2: More Requirements

"Great! That looks like us! Now, what else should we do?".

What have our friends done right? They kept the description brief, talking about what they want to accomplish, not how to do it. They wrote down the elements important to them: It's a catalog company, a reseller not a manufacturer, it provides quick service, and the software is used throughout the process to track orders. These are the key characteristics of their project. They didn't worry about making their description perfect. If something were important that they NOT do, they would have written that down as well.They got a basic description which they agree on, which may need to be modified later. However, the key characteristics should not change.

Starting Risk Analysis

Now that you have a description, the next step is to write down other things you know about your project. You are looking for marketing factors that will influence your project, good or bad, and things that are required that may not have appeared in the description. We will use these with the problem statement to create Use Cases, other requirements, and risk factors. Here are some things to consider:

Chap01.fm Page 19 Friday, October 27, 2000 11:36 AM

- who or what is the competition
- what technologies are you depending on
  - web
  - object databases
  - power pc chip
- market trends that influence your project
- number of expected users
- future trends you are depending on
  - more home offices
  - more small companies
- number of transactions per time frame
- expected duration of some functionality
- legacy systems you have to interface with
  - software
  - business processes
  - data stores, databases

In most households, all adults work at least part time. They have less time available for shopping, so are usually willing to pay for conveniences such as having purchases delivered.

Web shopping and home shopping networks are popular and are competitors in this market.

Other mail order companies provide 24 hour order takers, delivery times ranging from overnight to 2 weeks, gift wrap, and volume discounts.

Figure 1-3: Mail Order Market Factors

Be creative as you make this list. Brainstorm a lot. Put down just about anything you can think of. Put down the wildly unlikely as well as things that will probably happen. The idea at this stage is to look at the project from many viewpoints. This will help solidify your ideas. Look at some books on marketing trends for ideas. What are competing companies doing right or doing wrong?

You also need to consider risk factors in your project. You need to include things that can go wrong. Writing down only the things that you'd *like* to happen is a sure recipe for disaster. Far better to think of how it can go wrong so you can plan on it, than to be wandering along and get surprised.

Chap01.fm Page 20 Friday, October 27, 2000 11:36 AM

You also need to include things like being wildly successful. Sometimes you'll find things can actually go wrong if you are too successful. For example, what would happen to National Widgets if, in the first month, they get 4000 calls? We now have to handle multiple order takers and large amounts of data. Presuming the company can handle this surge, can the software handle it? The company could lose business if the software is not up to the demands placed on it, possibly getting a bad reputation because of it. Our friends will write this down as one of their risks.

Here are some things to consider as possible risk factors:

- lack of user acceptance
- dependence on a technology that changes
- not fast enough to market
- too many users
- team not experienced enough
- too short a schedule
- too fast company growth
- too fast to market
- supplier can't/won't deliver product we depend/need

- Some of the people designing the software are inexperienced
- How can we prevent lost orders on system failure?
- The system has to be easy for non-technical people to use.
- Can we be successful if we don't support a web interface?
- What if the system is flooded with orders immediately?
- How do we handle many simultaneous users in different parts of the company?
- How do we handle the database crashing?

Figure 1-4: National Widgets Risk Factors

Take this list of risks and the list of project factors, eliminate extremes like a comet crashing into your company and putting you out of business, or the sun fails to rise, and prioritize the rest. This prioritized list you've created is your first risk analysis. All the things you've listed here put you at risk for not completing your project. The more serious items are listed first, with less severe risks at the bottom of the list. The high risk factors must be addressed if you expect your project to succeed.

Remember that this is just a starting point, and that you'll be adding to it as you continue!

Chap01.fm Page 21 Friday, October 27, 2000 11:36 AM

As part of your risk list, put together and maintain a list of assumptions. These are decisions you make with little or no hard data. You may need to pick one way of doing something based on gut feel or experience. Record that decision and why you made it. This list should be reviewed regularly. Some things will remain as assumptions. For others, you will be able to get hard data on which to base your decisions. Remove things which are no longer assumptions, and add the new assumptions you've made.

Let's go back and see how the group is doing with their lists.

"Okay, it's been a busy evening. Let's see what we've gotten out of it." Gus hands around his lists.

Problem Description

- We are developing order processing software for a mail order company called National Widgets, which is a reseller of products purchased from various suppliers.
- Twice a year the company publishes a catalog of products, which is mailed to customers and other interested people.
- Customers purchase products by submitting a list of products with payment to National Widgets. National Widgets fills the order and ships the products to the customer's address.
- The order processing software will track the order from the time it is received until product is shipped.
- National Widgets will provide quick service. They should be able to ship a customer's order by the fastest, most efficient means possible.
- Customers may return items for restocking, but will sometimes pay a fee.

Assumptions

- An electronic interface, such as the Web, would be good for some customers
- We expect to use multiple shipping companies and insured methods

Figure 1-5: Order Processing Problem Statement

Chap01.fm Page 22 Friday, October 27, 2000 11:36 AM

Risk Factors

- Some of the people designing the software are inexperienced
- How can we prevent lost orders on system failure?
- The system has to be easy for non-technical people to use.
- Can we be successful if we don't support a web interface?
- What if the system is flooded with orders immediately?
- How do we handle many simultaneous users in different parts of the company?
- How do we handle the database crashing?

Market Factors

- In most households, all adults work at least part time. They have less time available for shopping, so are usually willing to pay for conveniences such as having purchases delivered.
- Web shopping and home shopping networks are popular and are competitors in this market.
- Other mail order companies provide 24 hour order takers, delivery times ranging from overnight to 2 weeks, gift wrap, and volume discounts.

Figure 1-5: Order Processing Problem Statement

"Wow" was Dennis' comment. "We sure can fail in a lot of ways. But how has this helped us define the software? So far, all it's done is make me worry that we forgot something."

Chapter 1 Review

At this point you should have:

Complete	Deliverables
✔	Project Description
✔	Risk Analysis
	Use Case Diagram
	Description of Actors and Use Cases
	Project Proposal

Table 1: Inception Phase Deliverables

Chap01.fm Page 23 Friday, October 27, 2000 11:36 AM

Your risk analysis should include known risks, other known market factors, and assumptions you have made about the project.

These are just preliminary versions, showing what you know right now. You will modify these things as you learn more about your project. The next chapter covers using Use Cases to find the boundaries of the system and the scope of the project.