Chapter 7: Building the Requirements Model

1. 7.1. Preparations to Begin the Analysis.
2. 7.2. Developing the Requirements Model: The Project Scope.
3. 7.3. Developing the Requirements Model: The Context Diagram.
4. 7.4. Developing the Requirements Model: The Use Case Model.
5. 7.5. Developing the Requirements Model: The Interface Descriptions.
Chapter 7: Building the Requirements Model

7.1. Preparations to Begin the Analysis.

- Our goal is to learn about the users and
  - *Their* World
  - *Their* Business
  - *Their* Technology
  - *Their* Jargon
  
  while not alienating them with ours!

- This task is complicated by:
  - Communication problems
  - Users’ fears
  - Our ignorance of what they do.

Systems Analysts require:

- Technical knowledge and understanding
- Business knowledge, and

  what sets a **great analyst** apart
  
  from a **merely good one,**

**People Skills**
7.1. Preparations to Begin the Analysis.

- Theoretically, users are responsible for telling us what we need to know,
- To build the software they need to use,
- But the world just doesn’t work that way!
- Generally, users have very little idea of what we need to know about their business.

Users are experts in their own field, but have never looked at their business from the point of view that we need,

The viewpoint of Information and Data.
7.1. Preparations to Begin the Analysis.

We will do whatever is necessary to carry out a Successful Analysis!!
Our project depends on it.

There are different ways to approach the task of Analysis:

- One-on-one
- Facilitated Team Sessions (FTS), also called Joint Application Design (JAD) Sessions
- Meetings (with busy executives):
  - Meet with each user in turn,
  - Combine results,
  - Circulate for feedback,
  - Meet each again to discuss,
  - Iterate Until Done.
7.1. Preparations to Begin the Analysis.

- The method that follows may be adapted for any of these approaches.
- It may be used one-on-one,
- But these things are best done in a FTS / JAD setting.
- With continual major input from the users.
Chapter 7: Building the Requirements Model

7.1. Preparations to Begin the Analysis.

Attendees

We need a range of users, from all levels and all segments (departments) of the Business Area we are modeling.

● Senior Management
  - as high-level as you can get
  - to lend the weight of influence and authority
  - to give the broad overview (the “Big Picture”).
  - Most important at first meeting (or first few)

  to firmly set the scope and boundaries of the project.

Attendees

● Workers -
  - From the “Shop Floor”
  - Often know details or secrets the Boss doesn’t!
  - We need to know what actually happens,
  - Not what the boss or the manual says ought to happen.
  - Select them carefully - workers who are especially
    ● Aware
    ● Intelligent
    ● Able to verbalize and describe things
7.1. Preparations to Begin the Analysis.

Attendees

- Junior and Middle Management.
- Foremen,
- Supervisors,
- Managers.

- Close to the operational level
- Have a clear idea of what goes on
- But have a more strategic (long-term) view than their staff do.

Recording Analyst
Chapter 7: Building the Requirements Model

7.1. Preparations to Begin the Analysis.

Recording Analyst

- Designated person to document sessions,
- Publish “minutes”
  - List of attendees,
  - Project Scope,
  - List of Candidate Classes (See Chapter 9),
  - The Model as it exists so far at the end of each session.
- This person must understand the process
  - Must be an analyst/modeler
  - Or an experienced and knowledgeable user.
  - Do not assign a clerk or steno; use someone who is a “Knowledgeable Filter.”

User Majority
7.1. Preparations to Begin the Analysis.

**User Majority**

- Systems people, Modelers, Programmers, Analysts should be a *Minority*.
- **Users** should be a *Majority*, so they are not overwhelmed by our numbers and do not feel “Blinded by Science.”
- With *more of them* than us, they are more likely to risk standing up and being heard, more willing to make their point.
- Generally we need at least 2 modelers: a *Leader* and a *Recording Analyst*.

**Distractions**
Chapter 7: Building the Requirements Model

7.1. Preparations to Begin the Analysis.

Distractions

- Meet at **Your Premises**
- Or **neutral premises**
- Or even better a retreat of some kind.
- All are better than the users’ office or even their conference room, because of:
  - Phones
  - Staff interrupting
  - People not returning after breaks because of “emergencies” that their staff would have handled in their absence anyway!

- You need their **solid concentration** and focus for whatever time has been set aside.

Background Research
Chapter 7: Building the Requirements Model

7.1. Preparations to Begin the Analysis.

Background Research

Research the user group beforehand:
- Reporting relationships (who works for whom).
- Jargon, terminology, abbreviations, acronyms.
- Get these from:
  - Glossaries,
  - Training Documents,
  - Introductory brochures,
  - Sales literature,
  - Annual reports,
  - Other reports,
  - etc.

Environment
7.1. Preparations to Begin the Analysis.

Environment

- Whiteboard or chalkboard (or flipchart)
- Lower the temperature for wakefulness (20°C or 68°F)
  better that some wear sweaters than any should sleep!
- Encourage casual dress
  - Especially user management (less intimidating to staff)
  - Especially if away from corporate eyes
  - Modelers dress one level above what users will wear
    - Look just a little bit professional
    - Aim for a “dressier level of casual”

- Comfortable seating, informal arrangement
- Coffee, tea, juice, water in the room
- Have refreshments served in the room at breaks
  - Minimizes disruption from the break
  - Keeps conversation on topic
- Beware smoking!
  - Best if not allowed
  - Unless users usually smoke in their meetings
  - Be guided by user management
  - If it is allowed, give frequent short smoke breaks.
- **Be sure to confirm** the first meeting with everybody the day before.
7.1. Preparations to Begin the Analysis.

Scheduling

- Users must be aware in advance of the time commitment they will need to make.
- 2 to 4 hours per session is optimal.
  - But if you must do it in all-day sessions, then so be it.
- ½-day sessions in the mornings if possible
  - Alertness drops after lunch!
  - When eyelids droop, call a 3-minute break to fetch coffee.

- If you are forced to do the model in straight days, then so be it, but be conscientious about all the things on the previous slide.
Chapter 7: Building the Requirements Model

1. Preparations to Begin the Analysis.

2. 7.2. Developing the Requirements Model: The Project Scope.

3. 7.3. Developing the Requirements Model: The Context Diagram.

4. 7.4. Developing the Requirements Model: The Use Case Model.

5. 7.5. Developing the Requirements Model: The Interface Descriptions.

---

Chapter 7: Building the Requirements Model

2. 7.2. Developing the Requirements Model: The Project Scope.

- There are two sets of objectives to consider:
  - Objectives of the company, and
  - Objectives of the system.

The objectives of the system must match with those of the company!
- Resolve any discrepancies **NOW, Early!**
- **The system must be built to support the objectives of the business area.**
The Project Scope.

- The Project Scope is a document stating what the project is to produce.
- A brief statement (2-4 paragraphs to 2-4 pages)
- Describing the software system we are about to produce.

The Scope says in general terms:
- What the eventual system will do,
- What functions will be part of the system,
- Which users it will service,
- And any other things you may think are important.

It must also state:
- What will \textit{NOT} be part of the system.
7.2. Developing the Requirements Model: The Project Scope.

- This last is because users have a tendency to “Push the Scope,”
- Which can give rise to “Scope Creep,”
- Where more and more gets added to the project.
- This can be deadly for fixed-price consultants.
The Context Diagram

- The **Context Diagram** models the data flows into and out of the system.
- It shows the system as a “Black Box”
- It is also known as:
  - A **Context Data Flow diagram**, or
  - A **Context Model**, or
  - An **Environment Model**.
- The Context Diagram is an excellent starting point for analysis.

Definition:

**External Entities**

are people, organizations, other systems, and a variety of other things, that are **external to our system**, and that either **provide data** to it, or **draw data** and information from it.
Chapter 7: Building the Requirements Model

7.2. Developing the Requirements Model:

The Context Diagram

External Entities may be:

- **People** - Customers, Employees, Members, Patients.
- **Organizations** - Government Depts, Regulating Bodies, Subsidiaries, or your Parent Corporation.
- **Other systems within your company** - Accounts Payable, Payroll, Human Resources, Inventory, Production Control, Fleet Management, etc.
- **Systems belonging to Vendors, Customers or Banks, etc.**
- **Software embedded in a product**, - cell phone, GPS, vending machine, machine tool, etc.
- **Software in a physical system** - telephone switch, or an oil refinery or other industrial process.

---

A Sample Context Diagram

- **Time Reporting and Billing System**
- **Unemployment Insurance**
- **Revenue**
- **Unions**
- **Life Insurance**
- **Medical / Dental Plan**
- **Board of Directors**
- **Employees**
- **Human Resources Database**
Chapter 7: Building the Requirements Model

7.2. Developing the Requirements Model:

The Context Diagram

- The arrows represent data flows
  - It is in fact a high-level Data Flow Diagram.

- Name the arrows for the kind of data
  - Not for the report name, etc.

- Show external databases or other computer files with a DFD Data Store symbol.

Chapter 6: The Object-Oriented Development Life Cycle (OODLC)

6.2. The Object-Oriented Analysis Phase

Requirements Model

Context Diagram

Radio CHQT Advertisers Database System

- Advertisers
- Billing
- Regulatory Authorities
- Statistics & Reports
- Listeners
- Better Business Bureau

- Requests
- Financial Reports
- Revenue Canada
- Quarterly Reports
- Credit Ratings
- Program Info
- Listeners
- Shareholders
This model documents what functions the system should offer to the users. It helps to:

- Construct our view (developers’ view) of what the users want,
- Provide a starting point for discovering object classes,
- Provide a starting point for discovering some of the operations or behaviors for the classes.
Chapter 7: Building the Requirements Model

7.3. Developing the Requirements Model:

The Use Case Model

- The users’ view of the eventual system will consist of:
  - The **commands** they give it, and
  - The **responses** they get back from it.
- The users see the system as a “**black box.**”
- We write down in **step-by-step form** the steps a user takes to do a task with the help of our software system.
- We document the **user’s concept** of the steps a user would follow or does follow interacting with the system to get a job done.

Definition:

A Use Case is the **steps** a user could follow to make use of the system.

A Use Case represents a **scenario** of what might happen when a user (i.e., an Actor) **makes use of** one or more features of the system.

So a Use Case is in a way a “**case of the usage**” of the system.
7.3. Developing the Requirements Model:  

**The Use Case Model**

- An *Actor* is a person, an organization or another system that can *initiate an instance* of a use case, thus making use of one or more features of the system.
- If we were to discover *all* the ways that all the Actors could interact with the system, we would have a *behavioral model* of our system.
- Our aim is to find the *major* ones.
- From these we can identify classes and behaviors for our Class Diagram.
- Use the person’s title, rather than their name.

Actors are diagrammed as stick figures. Use cases are initially drawn as an ellipse with the name of the Use Case inside.
Chapter 7: Building the Requirements Model

7.3. Developing the Requirements Model:

The Use Case Model

Finding Actors
- From the Context Diagram.
  - Some of the External Entities will turn out to be Actors
- Users are also Actors - every time they touch the mouse or keyboard.
- An Actor may represent several people
  - e.g., a Sales Clerk actor includes several people, including the Manager.
- A person may sometimes act as different Actors
  - e.g., the Manager person is sometimes a Sales Clerk Actor
- Thus an Actor is a **role** a person assumes when they interact with the system.
Chapter 7: Building the Requirements Model

7.3. Developing the Requirements Model:

The Use Case Model

- **Primary Actors:**
  The people (Actors) whom the system is primarily intended to benefit. This usually includes the users.

- **Secondary Actors:**
  People (roles) that exist only to ensure that the Primary Actors can use the system. This would include:
  - Service Technicians
  - Operators and on-site attendants
  - Technical support staff and maintenance programmers.
  - System and Network administrators.

- Use Cases are an *informal* description of the system;
  - They do not give information about *how* the system does things
  - Or any other details *internal* to the system.

- They just tell us *what the system will do* for the users.
  - Concentrating on *what* rather than *how* makes them more a tool for analysis than design, but . . .

- They do give us a good starting point for both
  - Testing the system, and
  - Prototyping the user interface.
Chapter 7: Building the Requirements Model

7.3. Developing the Requirements Model:

The Use Case Model

Finding the Use Cases

- Ask the users to list everything they need the system to do for them. Put this list up on the board.
  - This list is a first pass only. Do not expect it to be anywhere near complete.
- Show the system as a large box,
- Draw around it all the Actors you have discovered so far.
- Draw the ellipses and write in the names of the Use Cases.
- Join with arrows. The arrows show the net flow of data between the Actor and the system.

Once you have names for the Use Cases, it’s time to spell them out in detail. There are 3 steps to this process:

- Validate the name.
- Write a narrative description.
- Optionally write a detailed step-by-step.
  - There is considerable flexibility in how much detail you use with the last two. It is a judgement call. Do what you and your users are comfortable with; do whatever you find gives good results for you and them.
7.3. Developing the Requirements Model:

**The Use Case Model**

**Validate the Name:**

- Certain words have different meaning for different countries, cultures, or corporations. Be sure you and the users are all attaching the *same meaning* to each name.
- When the name is correct, it will prompt us to come up with the step-by-steps.
- Refining the name often shows that there are really two or more Use Cases where before we saw only one.
  - e.g., “Register Student” may resolve into different steps for Full-time, Part-time and Special students.

**Validate the Name:**

- Go down the list and ask the users:
  - Does the name tell all it should?
  - Does it tell the whole story?
  - Are there any exceptions?
  - Special cases?
  - Possible errors?
  - Occasional variations?
  - Does the name cover several related or similar processes?
  - Can you find a more enlightening or informative name?
7.3. Developing the Requirements Model:

The Use Case Model

Write a Narrative Description

- Get the users to describe a “course of events” that happens as they do the task.
- Include every step. Don’t miss any.
- Speak as if you were training an assistant to do a job, and you want the job done right.
- Tape or write the steps exactly as described.

Direct Orders Use Case

“When a customer calls in to order something, I call it up on the screen. I sometimes have to type in a part of the name or description, and then it helps me find the Product Code. One way or another, I get the details of the stock on the screen.

“Then I tell the customer whether I can satisfy his order, and if I can’t I call up stocks in each of the other branches, until I can. If not, I can place an order on the supplier, and reserve it for this customer.”
7.3. Developing the Requirements Model: The Use Case Model

The Outhouse Sales Reporting System

Use Case Scenario 7: Book a Room

0. System is at Main Menu.
1. Salesperson or Customer (title) phone to book a room (initiation). Scheduling Clerk (Actor, title) selects “Book a Room” from the Operations menu, and sees screen #11: the “Book a Room” screen.
2. Scheduling Clerk keys in Date and Time Requested and Store ID and sees a pick-list of rooms available at that date and time at that store.
3. Scheduling Clerk picks a room and sees screen #13: the “Customer/Consultant dialog box.”
4. Scheduling Clerk Customer ID and/or Consultant Employee # if known and clicks on the dialog box OK button. Dialog box disappears.
5. Scheduling Clerk checks screen and makes any changes by clicking on a field.
6. Scheduling Clerk clicks on Main Screen OK button. System prints booking slip.
7. When printing done, screen 11 is replaced by Main Menu (final state).

Use Case Scenario 4: Not Available

0. System is at Main Menu.
1. Salesclerk (Actor) keys in stock number or scans it from tag. System returns stock, backorder and on-order quantities for this store, plus a menu of actions, plus a menu of all the stores with a box marked “All Stores.”
2. Clerk goes to step 3, or chooses a store (or “All Stores”). System returns stock figures and the menu of actions.
3. Clerk chooses a store and an action.
4. Action: Dial. System dials that store on the voice phone for a verbal query or verification of stock. return to step 2.
5. Action: Reserve Stock. Clerk may enter a quantity; default is one. System places that quantity on hold at that store for customer pick-up.
6. Action: Reserve from On-Order. Clerk may enter a quantity; default is one. System places that quantity on hold for when the order arrives at that store, for customer pick-up.
7.3. Developing the Requirements Model:

**The Use Case Model**

- Remember, users do not request a Use Case by name!
- The Use Case is *our* invention for us to do our job.
- Typically, a user begins to use the system,
  - this starts a **course of events**,  
  - that results in the user acting out a Use Case.
- The resulting historical record of **what this user actually did** is an **instance** of the Use Case.

---

7.3. Developing the Requirements Model:

**The Interface Descriptions**
7.3. Developing the Requirements Model:

The Interface Descriptions

There are two main categories:

- **Human Interfaces**

- **Interfaces to other systems**
  These fall into subcategories:
    - Other systems or databases *within* your company
    - Other systems or databases *outside* your company
    - Communication systems and protocols
    - Real-time and process-control systems

**Human Interfaces**

- As we work at finding Actors and Use Cases, we can rough out screen designs
  - Maybe on paper
  - Or a screen tool, CASE tool or Development Environment
    - *e.g.*, PowerBuilder, one of “The Visuals,” or the IDE that comes with an OOPL.

- As you develop the detail of a Use Case, place these screens in front of the user
  - This is actually a first pass at prototyping.
  - Data Fields only. Leave scroll bars etc until later.
7.3. Developing the Requirements Model: The Interface Descriptions

Human Interfaces - Screens

- Each screen must have a title and a unique screen number.
- List as notes any details that the user mentions, but you think are for later design and prototyping stages.
- The **Sequence** of the screens is critically important.
  - You may add the screen numbers to the steps in the Use Case.

Interfaces to other systems

- Other systems or databases *within* your company
- Other systems or databases *outside* your company
  - Databases
  - Communication systems and protocols
  - Real-time and process-control systems
Chapter 7: Building the Requirements Model

7.4. Developing the Requirements Model:

The Interface Descriptions

Other systems or databases

**within** your company:

- If these are already documented, then either
  - Refer to the existing documentation. This is generally better than copying it and having the copy go out of date.
  - Copy it only if your documentation is going out to people who will need the details but would not normally have them.
- Otherwise, you will need to research the interface and write it up.
  - Don’t go into too much detail at this point. It can be fleshed out in the Design phase.

Other systems or databases

**outside** your company:

- Databases
- Communication systems and protocols
- Real-time and process-control systems
Chapter 7: Building the Requirements Model

7.4. Developing the Requirements Model:

The Interface Descriptions

- External Databases
  - Document the interface the same as databases within your company.

- Communications systems and links.
  - WANs, LANs Intranets and the Internet.
  - Refer to the published protocols.
  - Be sure you have a copy on hand!

- Real-time and Process-control Systems
  - For existing real-world systems, refer to documentation from the hardware engineers
  - If you are doing the entire project, at this time you need to at least rough out the requirements for the hardware interface.

End of Chapter 7