Come, Harry, let us look for the Room of...

Requirements

"The hardest single part of building a software system is deciding precisely what to build. No other part of the conceptual work is as difficult as establishing the detailed technical requirements… No other part of the work so cripples the resulting system if done wrong. No other part is more difficult to rectify later."

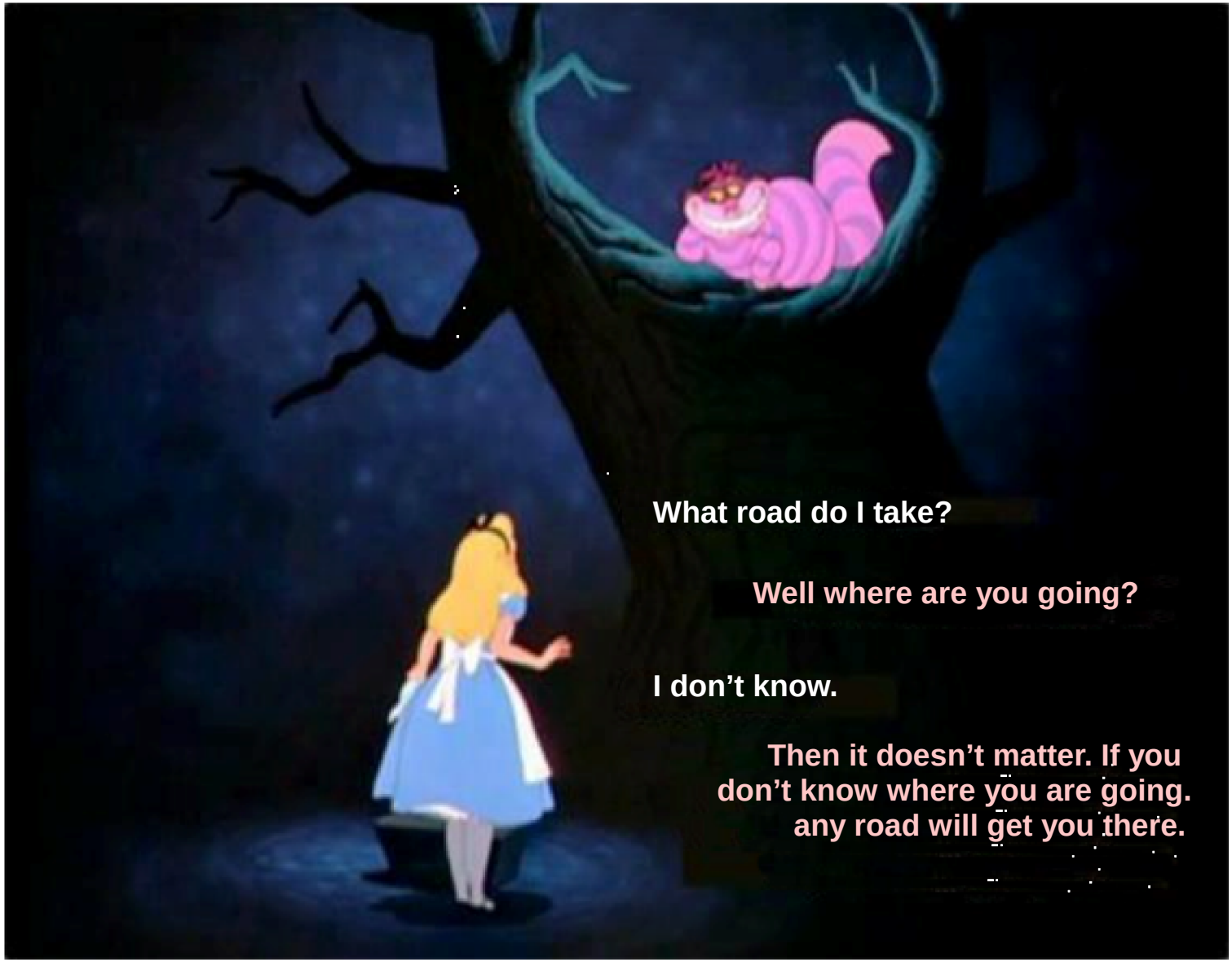
Frederick P. Brooks, Jr.

Requirements

Let’s start with a definition...

An expression of desired behavior for a software system.
It‘s a specific thing your system has to do to work correctly.

Why are Requirements so important?

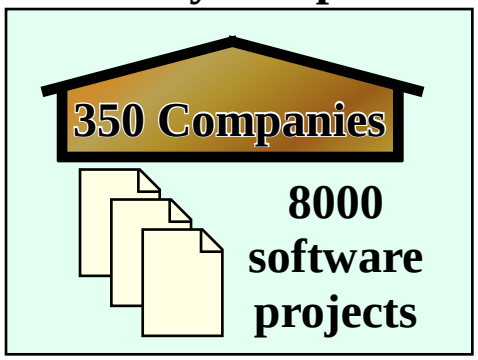


Why are Requirements so important?

First you have to know where you are going!

Consider a study conducted by the Standish Group*

Study Sample



Sample Findings

ONLY

9% of large companies
16% of small companies

Delivered on time and within budget

Why?

*** The Standish Group is an independent research organization specializing in IT (Information Technology) research.**
For more information visit: http://www.standishgroup.com/

Why?

Because of...

- Incomplete requirements
- Lack of user involvement
- Unrealistic expectations
- Changing requirements and specifications
- Lack of planning

More important terms...

Requirements Analysis Determining exactly what the software should do. It is the process of studying and analyzing what the customer wants in order to develop a stated list of requirements.

Object-oriented analysis and design A software engineering approach that models a system as a group of interacting objects.

Object-oriented Analysis (OOA) Applies object-modeling techniques to analyze the functional requirements for a system. OOA focuses on *what* the system does.

Object-oriented Design (OOD) Elaborates the analysis models to produce implementation specifications. OOD focuses on *how* the system does it.

Application Domain The specific subject matter content which the software will operate on and the environment in which the software will operate.

Stakeholders

Stakeholders Anyone involved in any way with the design, development, or use of the proposed software system.

- Anyone who operates the system
- Anyone who benefits from the system
- Anyone involved in purchasing the system
- Organizations which regulate aspects of the system
- People or organizations opposed to the system
- Organizations responsible for systems which interface with the system under design.

Remember: "You gotta keep the customer satisfied..."
Simon and Garfunkle

Steps in Determining the Requirements

1. Elicitation

Find out what the requirements are.

2. Analysis

Make sure you understand the requirements

3. Specification

Clearly state the requirements

4. Validation

Make sure they are correct

There's an easy nmemonic for this one: "EASy Value"

Steps in Determining the Requirements

Elicitation

That's a fancy word that means "find out what they are."

That's easier said than done!

- Interviews with the stakeholders.
- Questionnaires
- User observations
- Workshops
- Brainstorming
- Define use cases
- Prototyping
- Anticipate future requirements

Steps in Determining the Requirements

Elicitation

Possible Problems

Problems of Scope

Problems of Understanding

Problems of Volatility

Steps in Determining the Requirements

Analysis

Study and understand exactly what the requirements mean and what "behavior" the software system must demonstrate.

Study how the customer/user plans to use the software and try to anticipate other needs, requirements, and potential problems.

Steps in Determining the Requirements

Specification

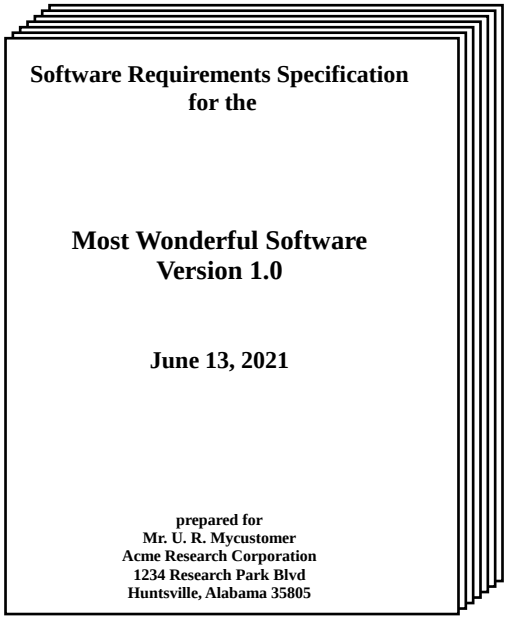
Requirement Specification — A complete description of the behavior of a system to be developed.

After you think you have all the information you need from the user write down a preliminary list of requirements clearly stating each in detail.

Steps in Determining the Requirements

Specification

Writing a Software Requirements Specification*



- Architectural Requirements
- Behavioral Requirements
- Interface requirements
  - System interfaces
  - User interfaces
- Functional requirements
- Database requirements
- Performance requirements
- Non-functional requirements

- Constraints
  - Design constraints
  - Hardware constraints
  - System constraints

****This term can refer to any of the requirements documents.***

Stating Functional Requirements

"The system shall…"

"The system shall have the capability to update an inventory database." *This is not a good example:*

What is the database going to be updated with?
What specific data elements will be updated?
What if the data elements are missing from the database?
How will this be tested?

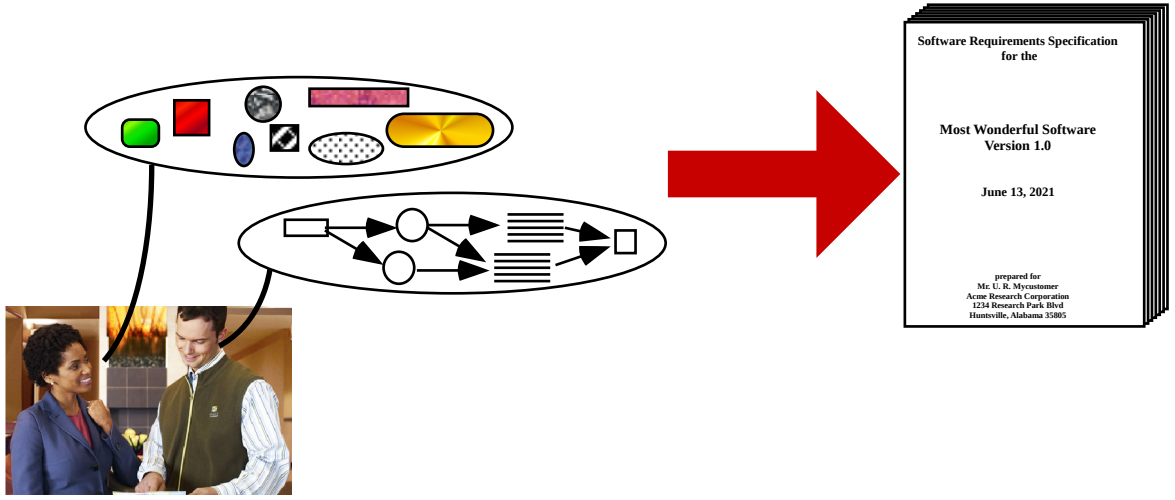
Stating Functional Requirements

- Be specific not prescriptive
- State requirements in measurable terms
- Be Realistic

From Real World Objects to Requirements

1. If the customer names parts or components in the desired system. These become requirements.

2. If the customer describes functions or actions for the system. These become requirements.



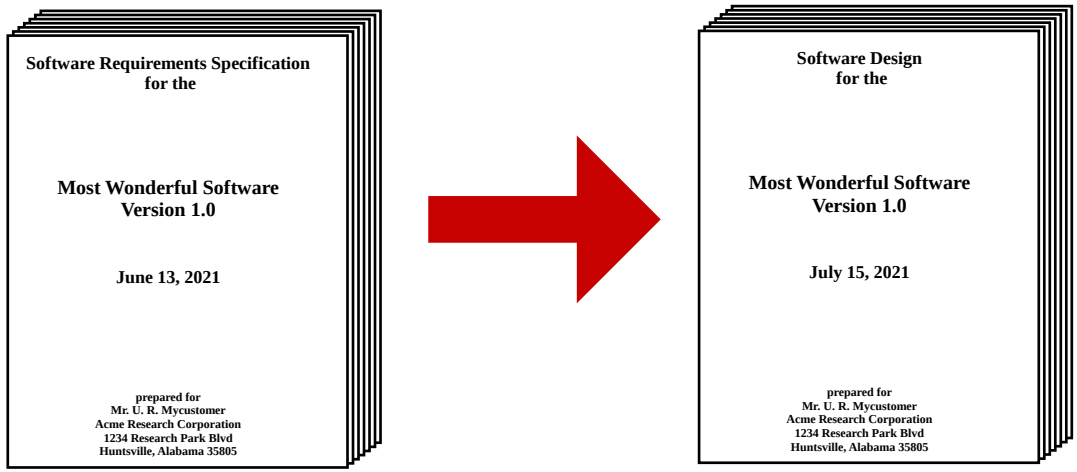
Our Policy:

Rule 1: ***The customer is always right!***
Rule 2: ***If the customer is ever wrong, re-read Rule 1!***

From Requirements to Program Objects

1. If a requirement refers to a part or component of the system. These become classes.

2. If a requirement describes an action, function, or behavior of the system. These become behaviors of a class.



Use Cases

*OK, describe for me one **case** of how you will **use** this software.*

Use Case - A technique for capturing the potential requirements of a new system or software change. Each Use Case provides one or more scenarios that convey how the system should interact with the end user or another system to achieve a specific goal.

In other words, a Use Case is just the steps that a system must follow to make something happen.

Doug has a great idea. Create software to control his dog doors.

Tired of cleaning up your dog's mistakes?
Ready for someone else to let your dog outside?
Sick of dog doors that stick when you open them?

It's time to call...

Doug's Dog Doors

- Professionally installed by our door experts.
- Patented all-steel construction.
- Choose your own custom colors and imprints.
- Custom-cut door for your dog.

Call Doug today at 1-800-Dog-Door

Here's what Harry and Blanche say they want the dog door to do.

Harry and Blanche's Requirements List

1. **We don't want Fido to hurt his back when he goes in and out so the door must be at least 12 inches high.**
2. **We want a remote control with a button we can press to open and close the door.**
3. **Once the door is open it should close automatically after a few seconds.**

Harry and Blanche, Doug's first customers

Use Cases

Harry and Blanche's requirement list.

Harry and Blanche's Requirements List

1. We don't want fido to hurt his back v
out so the
least 12 in
2. We want
with a bu
to open a
3. Once the
should clo
after a fe

Here's a new list with details on what the door actually does.

Harry and Blanche's Dog Door
What the Door Does

1. Fido barks to be let out.
2. Harry or Blanche hears Fido barking.
3. Harry or Blanche presses the button on the remote control.
4. The dog door opens.
5. Fido goes outside.
6. Fido does his business.
7. Fido goes back inside.
8. The door shuts automatically.

This is a Use Case

Did we cover everything?

Does Fido always bark to go out? What if he just scratches at the door?

What if Harry or Blanche aren't home? What if they don't hear Fido barking?

What if Fido barks because he's excited or hungry? Will it be a problem if Harry or Blanche opens the door and Fido doesn't need to go outside?

Do we need to think about what happens if the door jams? Or, maybe that's more of a hardware problem.

What if Fido stays inside?

Happy, happy! Joy joy!

What happens if the door has automatically closed by the time Fido is finished?

If Fido is stuck outside can Harry or Blanche hear him bark to press "Open" on the remote and let him back in?

Maybe a revision is in order

Harry and Blanche's Dog Door
What the Door Does

1. Fido barks to be let out.
2. Harry or Blanche hears Fido barking.
3. Harry or Blanche presses the button on the remote control.
4. The dog door opens.
5. Fido goes outside.
6. Fido does his business.
   - 6.1 The door shuts automatically.
   - 6.2 Fido barks to be let back inside.
   - 6.3 Harry or Blanche hears Fido barking (again).
   - 6.4 Harry or Blanche presses the button on the remote control
   - 6.5 The dog door opens (again).
7. Fido goes back inside.
8. The door shuts automatically.

This is an Alternate Path

Fido gets to do his thing. Harry and Blanche have it easy.

Start=Fido barks
Stop=Door shuts automatically.

Fido

A complete path (main or using alternate paths) through a Use Case from the first step to the last is called a scenario.

The one constant in software development...

Change

In the real world requirements are always changing and it‘s up to you to incorporate these changes into your software to keep your customer satisfied.

Steps in Determining the Requirements

Validation

Goes to the Customers for Approval

Requirements Definition Document for the

Most Wonderful Software Version 1.0

June 13, 2021

prepared for
Mr. U. R. Mycustomer
Acme Research Corporation
1234 Research Park Blvd
Huntsville, Alabama 35805

Requirements Definition Document

Requirements Specification Document for the

Most Wonderful Software Version 1.0

June 13, 2021

prepared for
Mr. U. R. Mycustomer
Acme Research Corporation
1234 Research Park Blvd
Huntsville, Alabama 35805

Requirements Specification Document

Goes to the Developers to Guide the Software Design