Software Requirements Analysis and Specification

Target group: 4th year Computer engineering students

ECE, GIT,DTU

By Misganaw Aguate

MSc. In Computer Engineering

ethiomisgie@gmail.com

YouTube:ethioptech

https://project.ethioptec.com

https://academics.ethioptec.com

Topics includes…

- REQUIREMENTS
- REQUIREMENT CATEGORIES
- GATHERING REQUIREMENTS
- REFINING REQUIREMENTS
- RECORDING REQUIREMENTS
- VALIDATION AND VERIFICATION
- CHANGING REQUIREMENTS

Requirements

Requirements

- Are the features that your **application** must provide.
- At the beginning of the project, you gather requirements from the **customers** to figure out **what you need to build**.
- Throughout development, you use the **requirements to guide development** and ensure that you're **heading in the right direction**.

Requirements

Requirements

- At the **end** of the project, you use the **requirements to verify** that the finished application actually does **what it's supposed to do**.
- Depending on the project's scope and complexity, you might need only a **few requirements**, or you might need **hundreds of pages of requirements**.
- The number and type of requirements can also depend on **the level of formality the customers want**.

Requirements cont....

Properties of Requirement

- Clear
- Unambiguous
- Consistent
- Prioritized
- Verifiable

Words to Avoid

- **Comparatives**: Words like faster, better, more, and shinier. How much faster? Define "better." How much more?
- **Imprecise adjectives**: Words like fast, robust, user-friendly, efficient, flexible, and glorious.
- **Vague commands**: Words like minimize, maximize, improve, and optimize.

Requirement Categories

In general, requirements tell **what an application is supposed to do**.

Good requirements share certain characteristics. They are

- clear
- Unambiguous
- Consistent
- prioritized, and verifiable.
- but there are several kinds of requirements that are aimed at different audiences or that focus on different aspects of the application.
- For example, business requirements focus on a project's high-level objectives and functional requirements give the developers more detailed lists of goals to accomplish.

Requirement Categories cont....

Audience-Oriented Requirements

- These **categories** focus on **different audiences** and the different points of view that each audience has.
- They use a somewhat **business-oriented** perspective to classify requirements according to the people who **care** the most about them.

Requirement Categories cont....

Audience-Oriented Requirements

Types:

- **Business Requirements.** Business requirements lay out the project's **high-level** goals. They explain **what** the **customer hopes to achieve with the project**.
- **User Requirements**. User requirements (which are also called stakeholder requirements), describe **how** the project will be used by the **eventual end users**.
- **Functional Requirements**. Functional requirements are **detailed statements** of the project's **desired** capabilities. They're similar to the user requirements but they may also include **things that the users won't see directly**.

Requirement Categories cont....

Example of functional requirement

No	Requirements	Functional
1	The system scan student ID, staff ID and equipment serial	Yes
2	The system store, modify, process, display and delete required information	Yes
3	The system crosscheck the input ID or serial to the database data	Yes
4	The system show alert message, image and sound if ambiguity is occurred	Yes
5	The system proceed pre checking system	Yes
6	The system scan student ID and check for double feeding with time range	Yes
7	Generating weekly report for cafe manager	Yes
8	The system scan equipment serial and show owners photo with full information of equipment and owner	Yes
9	The system show available books in the library to be borrow	Yes
10	The system enables any user to remained deadline to return books	Yes
11	Show book issuing date and deadline	Yes
12	Show balanced punishment per late day while returning book	Yes
13	Generate weekly report for the library manager	Yes
14	The system can register and manipulate non café students information	Yes
15	The system can crosscheck that incoming person are either allowed or not	Yes
16	Sending SMS while equipment exit from the university for the owner	No

Requirement Categories cont....

Types cont…

Nonfunctional Requirements

- Nonfunctional requirements are statements about the **quality** of the **application's behavior** or constraints on how it **produces a desired result**.
- They specify things such as the application's performance, reliability, and security characteristics.

Requirement Categories cont....

Example of non functional requirement

No	Requirements	Priority
1	The resulting implementation should be flexible (extendible).	Yes
2	The user interface should be friendly interactive, easy to understand and operate.	Yes
3	The implementation should combine existing technologies like integration of database system and barcode reader machine to read student and staff ID number and also PC serial	Yes
4	The implementation should be feasible in technically, operational and economically	Yes
5	Generally the project implementation should be necessary for the society of DTU	Yes

Requirement Categories cont....

Types cont…

Implementation Requirements

- Implementation requirements are **temporary features** that are needed to transition to using the new system but that will be later discarded.
- For example, suppose you're designing an **invoice-tracking** system to replace an existing system. After you finish testing the system and are ready to use it full time, you need a **method** to copy any pending invoices from the old database into the new one.
- That method is an implementation requirement.

Requirement Categories cont....

FURPS

FURPS is an acronym for this system's requirement categories: functionality, usability, reliability, performance, and scalability. It was developed by Hewlett-Packard.

Categories:

- *Functionality:* What the application should do.
- *Usability:* What the program should look like.
- *Reliability:* How reliable the system should be.
- *Performance:* How efficient the system should be.
- *Supportability:* How easy it is to support the application.

Requirement Categories cont....

FURPS+. FURPS was extended into FURPS+ to add a few requirements categories that software engineers thought were missing.

Categories:

- *Design constraints:* These are constraints on the design that are driven by other factors such as the hardware platform, software platform, network characteristics, or database.
- *Implementation requirements:* These are constraints on the way the software is built.
- *Interface requirements:* These are constraints on the system's interfaces with other systems.
- *Physical requirements:* These are constraints on the hardware and physical devices

Gathering requirements

Listen to Customers (and Users).

- Learn as much as you can about the problem they are trying to **address** and any ideas they may have about **how the application might solve that problem**.
- Initially, focus as much as possible on the problem, not on the customers' suggested solutions, so you can keep the requirements flexible.

Gathering requirements

Use the Five Ws (and One H).

- Sometimes customers have trouble articulating their needs.
- You can help by using the five Ws (who, what, when, where, and why) and one H (how).

Study Users

- Interviewing customers (and users) can get you a lot of information, but often customers (and users) won't tell you everything they do or need to do.
- They often take for granted details that they consider trivial but that maybe important to the development team

Gathering requirements

- **Who?** Ask who will be using the software and get to know as much as you can about those people. Find out if the users and the customers are the same and learn as much about the users as you can.
- **What?** Figure out what the customers need the application to do. Focus on the goals as much as possible rather than the customers' ideas about how the solution should work.
- **When?** Find out when the application is needed. If the application will be rolled out in phases, find out which features are needed when.
- **Where?** Find out where the application will be used.
- **Why?** Ask why the customers need the application

Refining Requirement cont....

Copy Existing Systems

- If you're building a system to replace an existing system or a manual process, you can often use many of the behaviors of the **existing system as requirements** for the new one.
- If the old system sends customers e-mails on their birthdays, you can **require** that the **new system does that too**.

Refining Requirement cont. ....

Clairvoyance.

- This technique is particularly effective if the **project lead has previously built a similar** system.
- In that case, the lead already knows **more or less**

✓ what the application needs to do

✓ which things will be easy and which will be hard

✓ how much time everything requires and

✓ which kinds of donuts motivate the programmers the best.

Refining Requirement cont....

Brainstorm

- Copying an **existing application** and **clairvoyance** are good techniques for generating requirements.
- but they share a common disadvantage: They are unlikely to lead you to new innovative solutions that might be better than the old ones.
- To find truly revolutionary solutions, you need to be more creative.
- One way to look for creative solutions is the group creativity exercise known as brainstorming.

Refining Requirement cont....

- The basic approach that most people think of as brainstorming is called the **Osborn** method because it was developed by Alex Faickney Osborn
- He was an advertising executive who tried to develop new, creative problem-solving methods starting in 1939.
- Basically, the gist of the method is to gather as many ideas as possible, not worrying about their quality or practicality.
- After you assemble a large list of possible ideas, you examine them more closely to see which deserve further work.

Refining Requirement cont....

Brainstorming Techniques

Popcorn:

People just speak out as ideas occur to them. This works fairly well with small groups of people who are comfortable with each other.

Subgroups

Break the group into smaller subgroups and have each group brainstorm.

Refining Requirement cont....

Brainstorming Techniques

Sticky notes

- Participants write down their ideas on sticky notes, index cards, papyrus, or whatever.
- The ideas are collected, read to the group, and the group votes on each idea.
- The best ideas are developed further, possibly with other rounds of brainstorming.

Idea passing

- Participants sit in a circle.
- Each person writes down an idea and passes it to the next person.
- The participants add thoughts to the ideas they receive and pass them on to the next person.

Refining Requirement cont....

Brainstorming Techniques cont...

Circulation list

- This is similar to idea **passing** except the ideas are passed **via e-mail**, **envelope**, or some other method outside of a single meeting.
- This can take a lot longer than idea passing but may be more convenient for busy participants.

Rule breaking:

- List the **rules** that govern the **way you achieve a task or goal**.
- Then everyone tries to **think of ways** to break or circumvent those rules while still achieving the goal.

Refining Requirement cont....

Brainstorming Techniques cont...

Individual

- Participants perform their own solitary brainstorming sessions.
- They can write (or speak) their trains of thought, use word association, draw mind maps (diagrams relating thoughts and ideas—search online for details), and any other technique they find useful.
- Some studies have shown that individual brainstorming maybe more effective than group brainstorming.

Recording Requirements

UML

- The Unified Modeling Language (UML) lets you specify how **parts of the system** should work.
- Despite its name, UML isn't a single unified language.
- Instead it uses several kinds of diagrams to represent different pieces of the system.
- Some of those represent program items such as classes.
- Others represent behaviors, such as the way objects interact with each other and the way data flows through the system.

Recording Requirements

User Stories

A user story is exactly what you might think: a short story explaining how the system will let the user do something.

Use Cases

A use case is a description of a series of interactions between actors. The actors can be users or parts of the application.

Recording Requirements

Prototypes

- A prototype is a mockup of some or all of the application.
- The idea is to give the customers a more intuitive hands-on feel for what **the finished application will look like** and how it will behave than you can get from text descriptions such **as user stories and use cases**.

Requirements Specification

These typically list major categories of requirements such as user documentation, user interface design, and interfaces with other systems.

Validation and Verification

Requirement validation is the process of making **sure that the requirements say the right things**. Someone, often the customers or users, need to work through all the requirements and make sure that they:

(1) Describe things the application should do.

(2) Describe everything the application should do**.**

Requirement verification is the process of checking that the finished application actually satisfies the requirements.

Validation : Are we doing the right things?

Verification : Are we doing the things right?

Review questions

1. Why we need software project requirement?
2. What happen if we did awesome software with out clint requirement?
3. How functional requirements are help us to develop a good software?
4. Why we change requirement after we settled?
5. why we used brainstorming in requirement refinement.
6. Suppose you gathered the requirement from the owner of super market to develop a software that can control the entering and exit of customer. While doing so, the customer inform you to add a module that used to display the name of customers on the display screen. At what type of requirement does the newly module addition is categorized?

End of Chapter 3