

Course name

Linus Falk

September 4, 2023

Contents

1	Requirement engineering	2
1.1	Customers	2
1.2	User vs System requirements	2
1.3	User requirements	2
1.4	System requirements	3

1 Requirement engineering

Requirements are about **What** not **How**. The **definition**: Requirements are a specification of what should be implemented. They are descriptions of how a system should behave, or of a system property or attribute. They may be a constraint on the development process of the system.

1.1 Customers

Stakeholder is a person, group or organization that:

- is actively working in a project
- is affected by the process or outcome
- or can influence the outcome
- Stakeholders can be internal or external to the project team and to developing organization.

A **customer** is an individual or organization that derives either direct or indirect benefit from the product that is being developed.

The **user** are a subset of the customer who will actually use the product. We can distinguish two types of these:

- Direct users that will operate the product hands-on
- Indirect users might receive outputs from the system without touching it/come in contact themselves.

1.2 User vs System requirements

User requirements are goals or tasks that **specific classes of users must** be able to perform with a system, or a **desired** product attribute.

System requirements are Top-level requirements for a product and can be software or software + hardware.

The user requirements are usually *abstract, in natural language and what the customer wants* while the system requirements are more *concrete and detailed, Natural+formal language, what system provides and is used as a product description*

A software example

User requirements:

- I need to print a mailing label for a package
- As the lead machine operator, I need to calibrate the pump controller first thing every morning

System requirements:

- If the pressure exceeds 40.0 psi the high pressure warning light should come on.
- The user must be able to sort the project list in forward and reverse alphabetical order.

1.3 User requirements

Written as the use cases/user stories (agile). Informed by 0. Domain requirements: "everyone knows that...".

- **Use case**
 - Sequence of interactions between a system and external actor to achieve an outcome of value.
- **User Story**
 - Short, simple description of a feature told from the perspective of who wants it, e.g., user or customer from the system.
 - As a **USER ROLE**, I want **DO SOMETHING** so that **REASON**
- **User classes**
 - Subset of product's customer
 - An individual can belong to several classes

- Each user class must have a set of requirements
- Classes need to be human beings
- **What process to use**
 - User personas - A description of a representative of a user class with similar characteristics and needs
 - User representatives - A suitable representative to provide the voice of the user.
 - Product Champion - An intermediary gathering requirements from the users for us.

In *agile* methods all user classes are represented by the **product owner**

1.4 System requirements

We can divide it into two categories: functional and non-functional:

Functional Requirements

- Behavior that a system will exhibit under specific conditions
- Describes **what** developers must implement to satisfy user requirements
- Written as "shall" statements

Example: Functional system requirements

- The passenger **shall be able to** print boarding passes for all flight segments for which he has checked in.
- If the passenger's profile does not indicate a seating preference, the reservation system **shall** assign a set.

Non-functional Requirements

- Describe **how well** a system does what it must do
- Quality standard of the system
- Must be measurable

Here are some **metrics** for **Non-functional System Requirements**

Property	Measure
Speed	Processed transactions/second User/event response time Screen refresh time
Size	Mbytes Number of ROM chips
Ease of use	Training time Number of help frames
Reliability	Mean time to failure Probability of unavailability Rate of failure occurrence Availability
Robustness	Time to restart after failure Percentage of events causing failure Probability of data corruption on failure
Portability	Percentage of target dependent statements Number of target systems

Table 1: Example