

Prof. Rania Elgohary
rания.elgohary@cis.asu.edu.eg

Dr. Yasmine Afify
yasmine.afify@cis.asu.edu.eg

WHAT IS A REQUIREMENT?

A Requirement is a **usable representation** of a **need**.

Requirements focus on **understanding what kind of value** could be **delivered** if a requirement is **fulfilled**.

A requirement defines:

- A **Feature** that a future solution has to enable (Cloud Access)
- A **Function** that a future solution has to Execute (Calculate Savings)
- A **Fact** that a future solution has to enforce (regulations)
- A **Quality** that a future solution has to exhibit (Access file in 1 second)



Smartsheet Inc. © 2022

REQUIREMENT CLASSIFICATION SCHEME

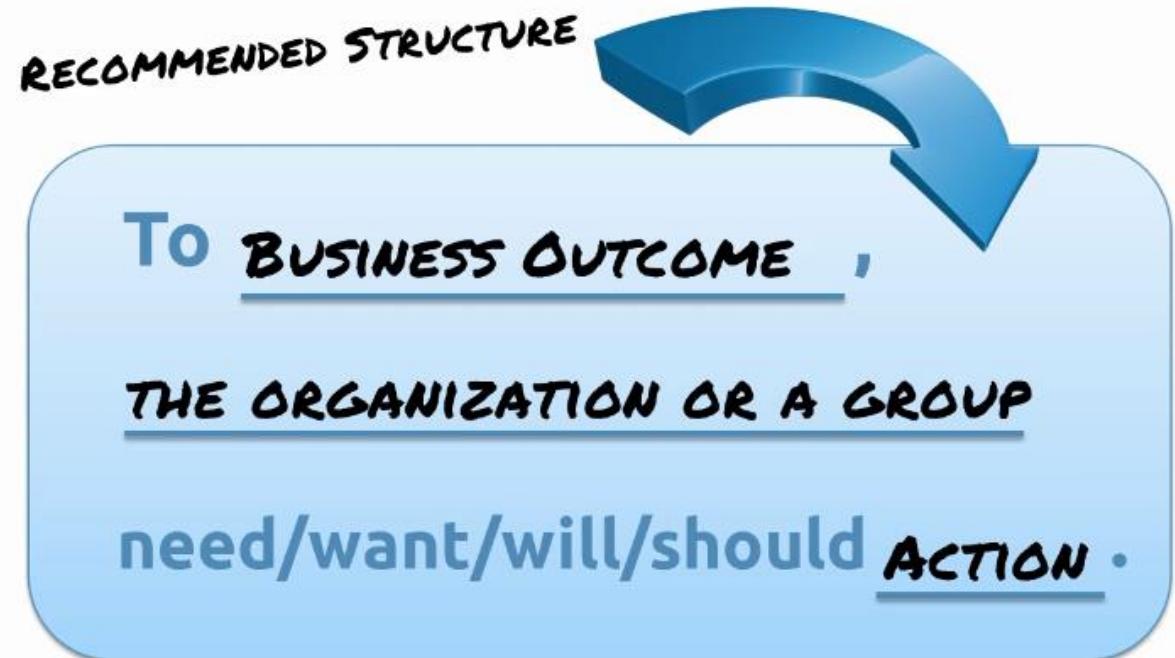
BUSINESS REQUIREMENTS

Defines the high-level *goals* and *objectives*.

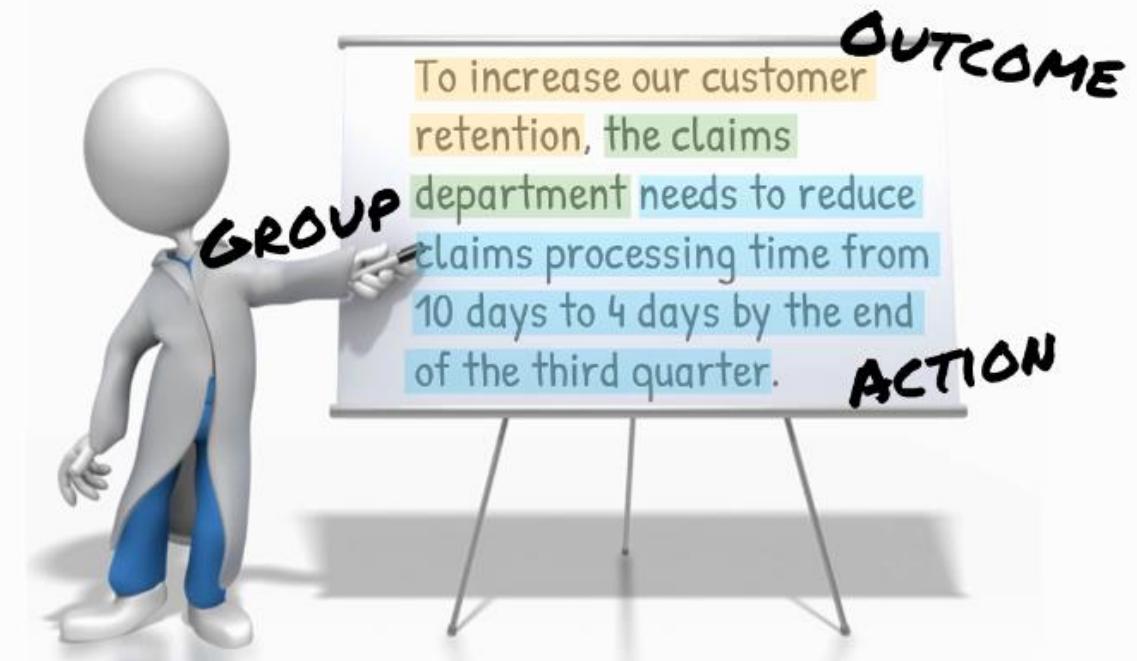
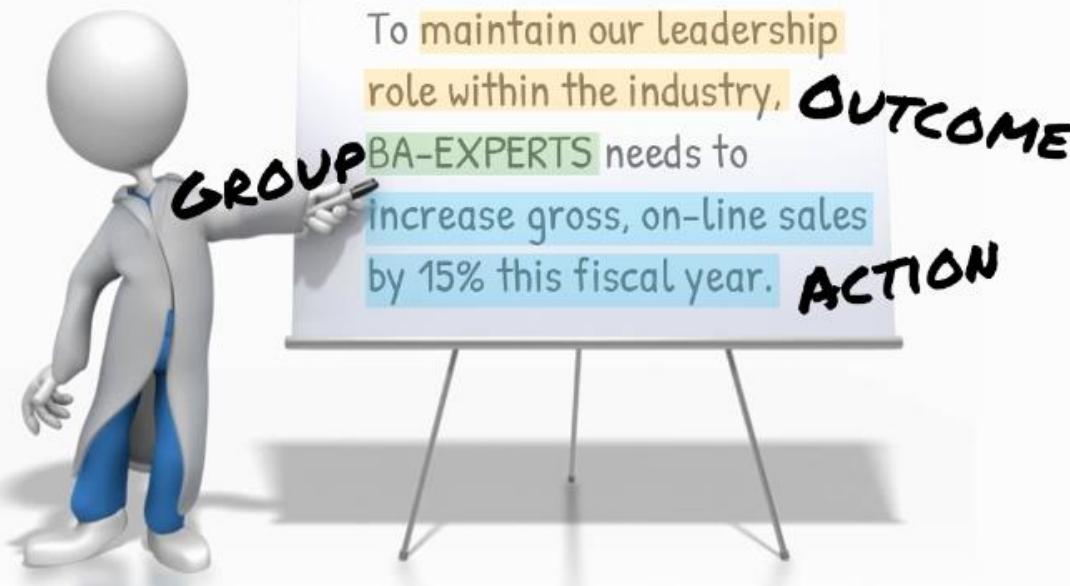
Address the question “*Why the project is needed*”

Executive levels usually define the business requirements

The team defines the requirements in a business requirements document (**BRD**). A business requirement might be a process, data necessary for the process, or a business rule influencing that process or data. These requirements align the project goals with the business objects.



BUSINESS REQUIREMENTS EXAMPLES



STAKEHOLDER REQUIREMENTS

Expresses the **needs** and **wants** of one or more **stakeholders** and **how** they will **interact** with this **solution**.

Bridge business requirements and Solution requirements.

What is needed not **How to achieve**



STAKEHOLDER REQUIREMENTS EXAMPLES



SOLUTION REQUIREMENTS

Describe specific ***characteristics*** of the ***solution*** that need business and stakeholder requirements.

Functional requirements should be ***easy*** for the ***developer*** to use. Expressed by:

- List of functions
- List of data elements
- Use cases
- Prototypes

Process diagrams
Calculate total charges including delivery costs and taxes Activity diagrams

Do not ship goods to customers with overdue accounts

TRANSITION REQUIREMENT



Describe **capabilities** need to **integrate** the **proposed solution** to the **existing solution** into the existing environment.

Describes **capabilities** the solution must have to **facilitate** getting from as-is
☒ **to-be**.

Not needed when solution is in production.

They are differentiated from other requirements types because they are of a **temporary nature**.

Transition requirements address topics such as **data conversion**, **training**, and **business continuity**.

TRANSITION REQUIREMENTS EXAMPLES

Sales personnel must attend a 2-day new customer acquisition program prior to using the new sales support system

All existing customer data will be maintained in both the old and new database format until the end of the first quarter

Data entry successful!



Is takes 5 minutes just to save a file?!



FUNCTIONAL AND NON-FUNCTIONAL REQUIREMENTS

Functional requirements

- ❑ Statements of services the system should provide and how the system should react to particular inputs.
- ❑ May state what the system should not do.

Non-functional requirements

- ❑ Constraints on the services or functions offered by the system such as timing constraints, constraints on the development process, standards, etc.
- ❑ Often apply to the system as a whole rather than individual features or services.

FUNCTIONAL REQUIREMENTS

WHAT THE SYSTEM SHOULD DO



PRODUCT FEATURES



USER REQUIREMENTS

NON-FUNCTIONAL REQUIREMENTS

HOW THE SYSTEM SHOULD DO IT



PRODUCT PROPERTIES



USER EXPECTATIONS

MENTCARE SYSTEM: FUNCTIONAL REQUIREMENTS

A user shall be able to search the appointments lists for all clinics.

The system shall generate each day, for each clinic, a list of patients who are expected to attend appointments that day.

Each staff member using the system shall be uniquely identified by his or her 8-digit employee number.

REQUIREMENTS IMPRECISION

Problems arise when functional requirements are not precisely stated.

Ambiguous requirements may be interpreted in different ways by developers and users.

Consider the term ‘search’ in requirement 1

- ❑ User intention – search for a patient name across all appointments in all clinics;
- ❑ Developer interpretation – search for a patient name in an individual clinic. User chooses clinic then search.

REQUIREMENTS 3CS



In principle, requirements should be both clear, complete and consistent.

Clear

Complete

Consistent

In practice, because of system and environmental complexity, it is challenging to produce a complete and consistent requirements document.

FUNCTIONAL REQUIREMENTS TYPES

Authentication. This group is about verifying the identity of a user before allowing access to the system, including entering usernames and passwords, biometric verification, or multifactor authentication.

Authorization levels. These requirements aim to define and control the access levels of different users within a system. For example, an admin may have complete system access, while a regular user has limited access to certain features.

Data processing. These requirements can include data entry, validation, storage, and retrieval.

FUNCTIONAL REQUIREMENTS TYPES

User interface and user experience (UI/UX). These are the requirements related to the design and interaction elements of the system. Their goal is to ensure that it's user-friendly and meets users' needs.

Reporting. These requirements define generating reports, e.g., data sources, formats, etc.

System integration. These requirements describe how the system interacts and integrates with other systems or third-party services.

FUNCTIONAL REQUIREMENTS TYPES

Transaction handling. This group contains requirements for handling transactions. They are especially important in systems that deal with financial processes or require record-keeping of transactions.

Error handling and logging. These requirements specify how the system should handle errors and log them, e.g., defining error messages, troubleshooting steps, and maintaining logs for system activities.

Backup and recovery. These are the requirements for data backup and system recovery processes, ensuring data integrity and system availability in case of failure.

FUNCTIONAL DECOMPOSITION AND WORK BREAKDOWN STRUCTURES (WBS)

A functional decomposition is a process of breaking down a complex problem, system into simpler, more understandable parts.

Functional decomposition helps create a detailed, visual representation of the system functionality – a Work Breakdown Structure.

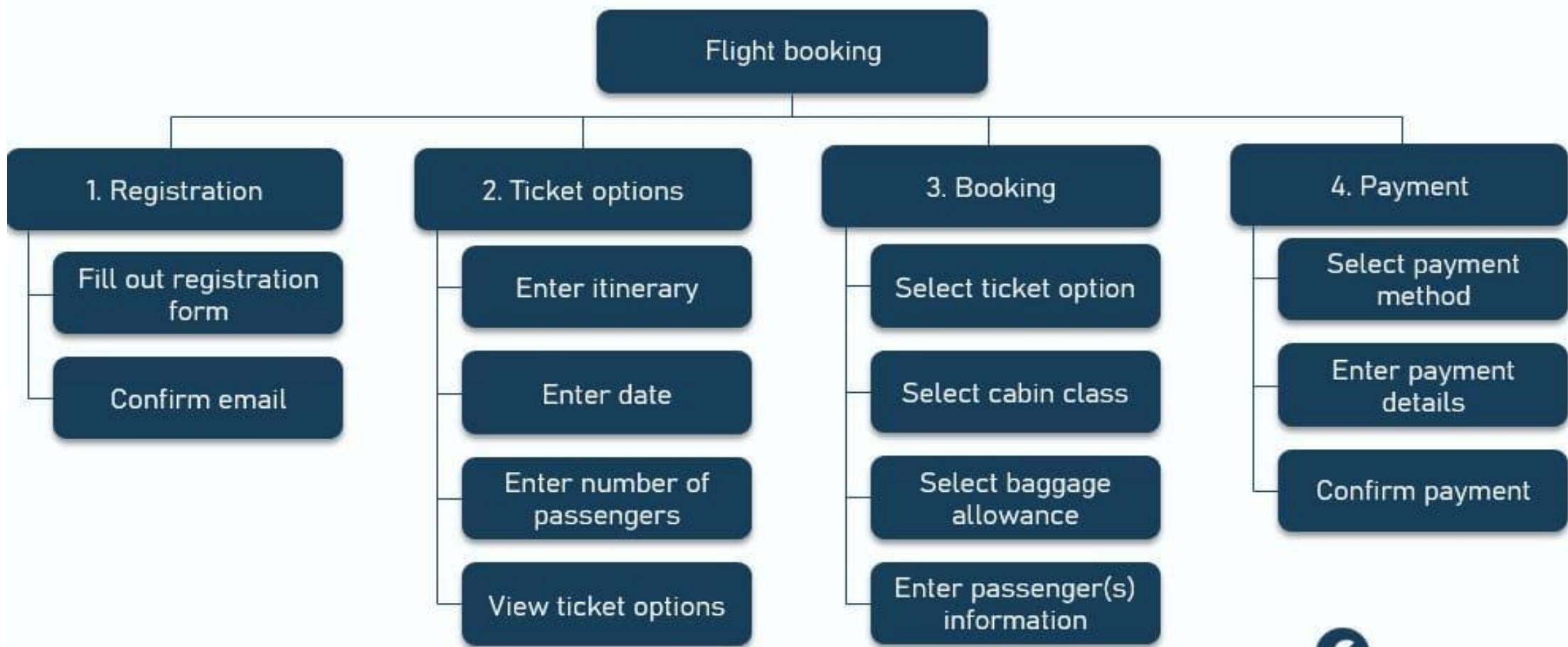
A WBS is an effective approach to allow for an independent analysis of each part. It also helps capture the full picture of the project.

The decomposition process may look like this:

High Level Function ->Subfunction -> Process -> Activity

We should decompose the features to the point where the lowest-level parts can't be broken down further.

FUNCTIONAL DECOMPOSITION: EXAMPLE



NON-FUNCTIONAL REQUIREMENTS

These define system properties and constraints e.g. reliability, response time and storage requirements, I/O device capability, system representations, etc.

Non-functional requirements are **as critical as** the functional requirements. If these are not met, the system may be useless.

Non-functional requirements may **affect overall architecture** of a system rather than the individual components.

Verifiable non-functional requirement: A statement using some measure that can be objectively tested.

GOALS AND REQUIREMENTS

Non-functional requirements may be difficult to state precisely and imprecise requirements may be difficult to verify.

Goal

❑ A general intention of the user such as ease of use.

Verifiable non-functional requirement

❑ A statement using some measure that can be objectively tested.

VERIFIABLE NON-FUNCTIONAL REQUIREMENT

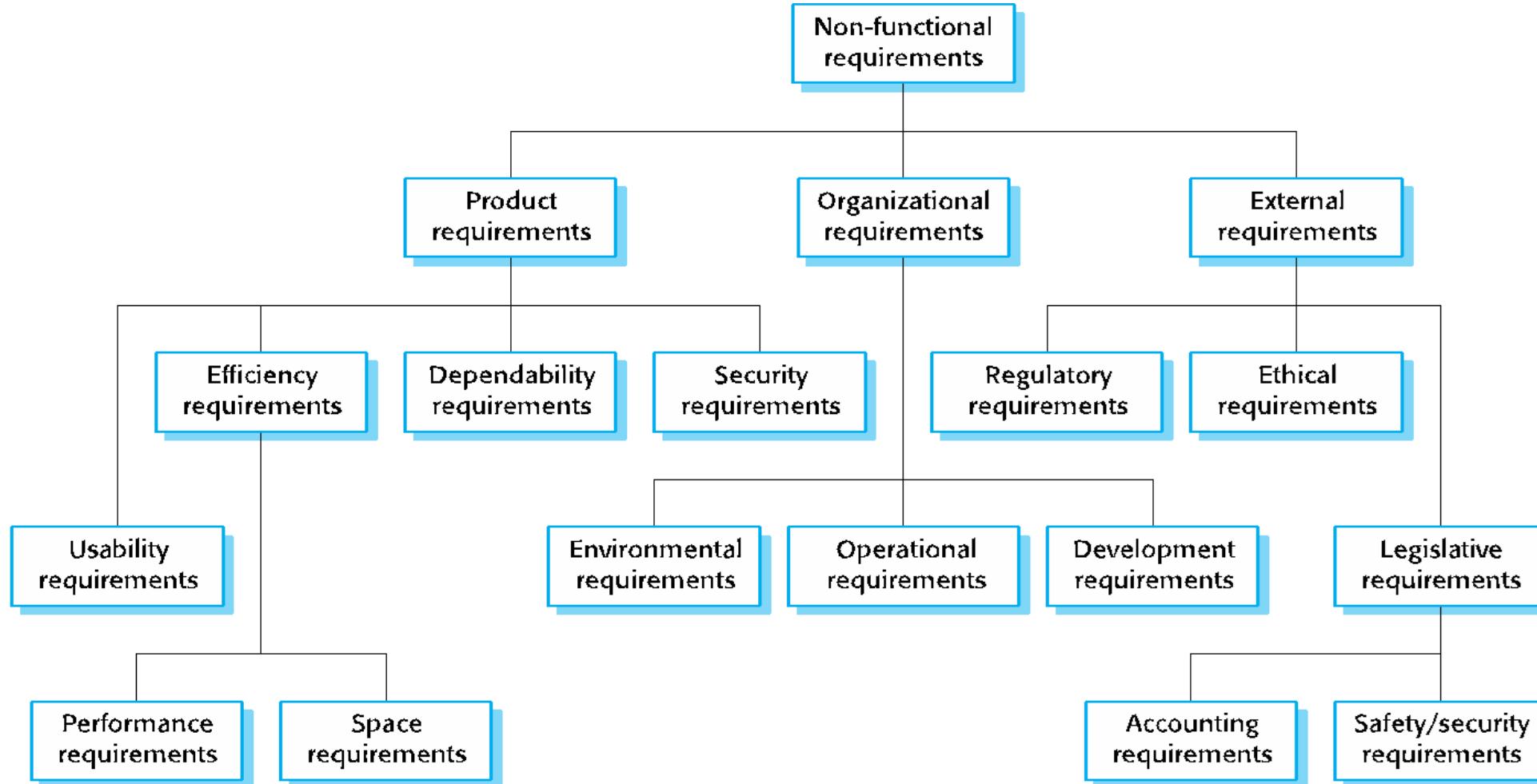
The system should be easy to use by medical staff and should be organized in such a way that user errors are minimized.

(Goal, cannot be tested)

Medical staff shall be able to use all the system functions after 4 hours of training. After this training, the average number of errors shall not exceed 2 per hour of system use.

(Verifiable non-functional requirement)

TYPES OF NONFUNCTIONAL REQUIREMENTS



NON-FUNCTIONAL REQUIREMENTS CLASSIFICATION

Product requirements

- ❑ Requirements which specify that the delivered product must behave in a particular way e.g. execution speed, reliability, etc.

Organisational requirements

- ❑ Requirements which are a consequence of organisational policies and procedures e.g. process standards used, implementation requirements, etc.

External requirements

- ❑ Requirements which arise from factors which are external to the system and its development process e.g. interoperability requirements, legislative requirements, etc.

NONFUNCTIONAL REQUIREMENTS IN THE MENTCARE SYSTEM

Product requirement

The Mentcare system shall be available to all clinics during normal working hours (Mon–Fri, 08.30–17.30). Downtime within normal working hours shall not exceed five seconds in any one day.

Organizational requirement

Users of the Mentcare system shall authenticate themselves using their health authority identity card.

External requirement

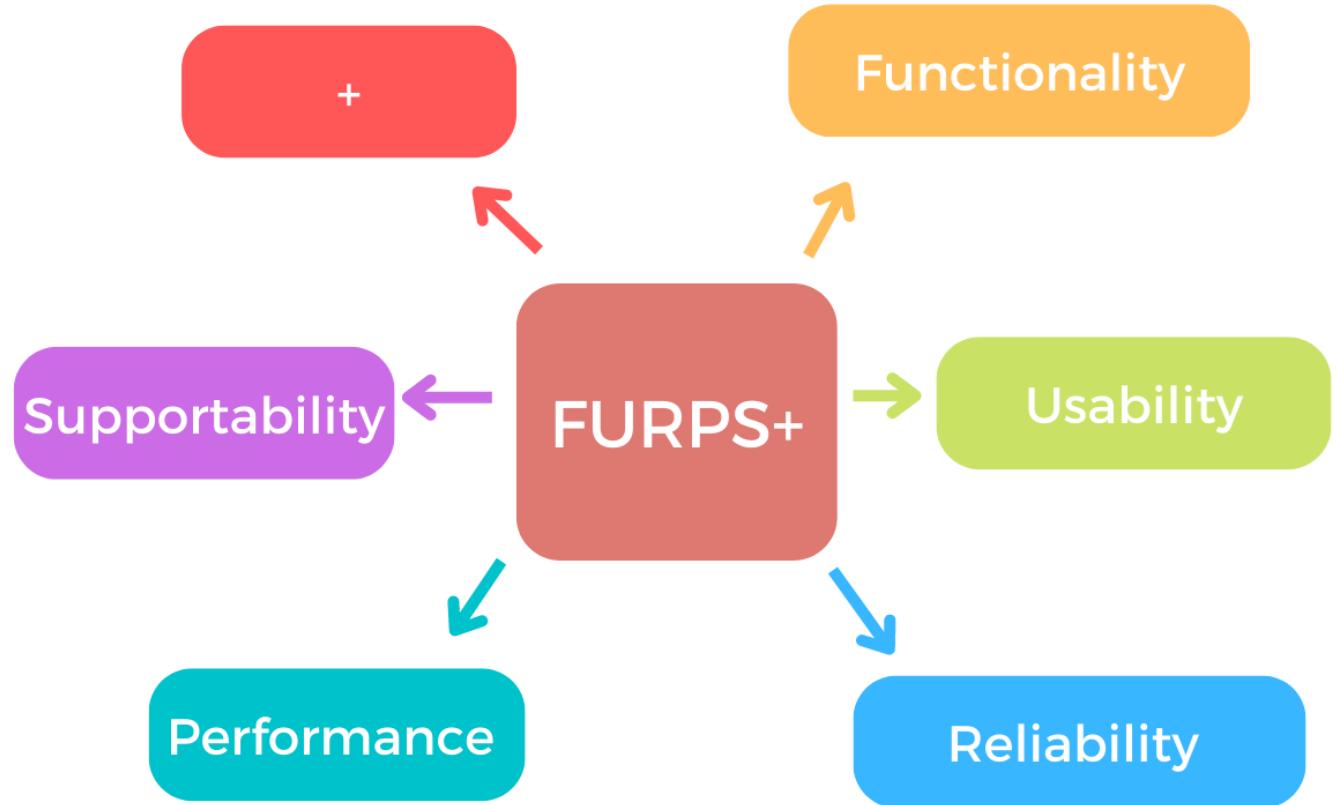
The system shall implement patient privacy provisions as set out in HStan-03-2006-priv.

METRICS FOR SPECIFYING NON-FUNCTIONAL REQUIREMENTS

Property	Measure
Speed	Processed transactions/second User/event response time Screen refresh time
Size	Mbytes
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

FURPS

FURPS provides a good way to categorize non-functional stories.



FURPS

FURPS is an acronym that stands for functionality, usability, reliability, performance, and supportability.

The “F” refers to the functional requirements.

Usability requirements refer to UI, online help, and documentation.

Reliability requirements refer to system dependability and the risk of system to be crashed.

Performance requirements describe measures workload, such as throughput and response time.

Supportability requirements describe the systems installation, configuration, monitoring, and maintenance.

FURPS+

FURPS+ extension includes design constraints as well as implementation, interface, physical requirements.

Design constraints refer to restrictions to which the hardware and software must adhere.

Implementation requirements refer to constraints related to programming languages and tools.

Interface requirements refer to interaction among systems, that might be internal or external systems.

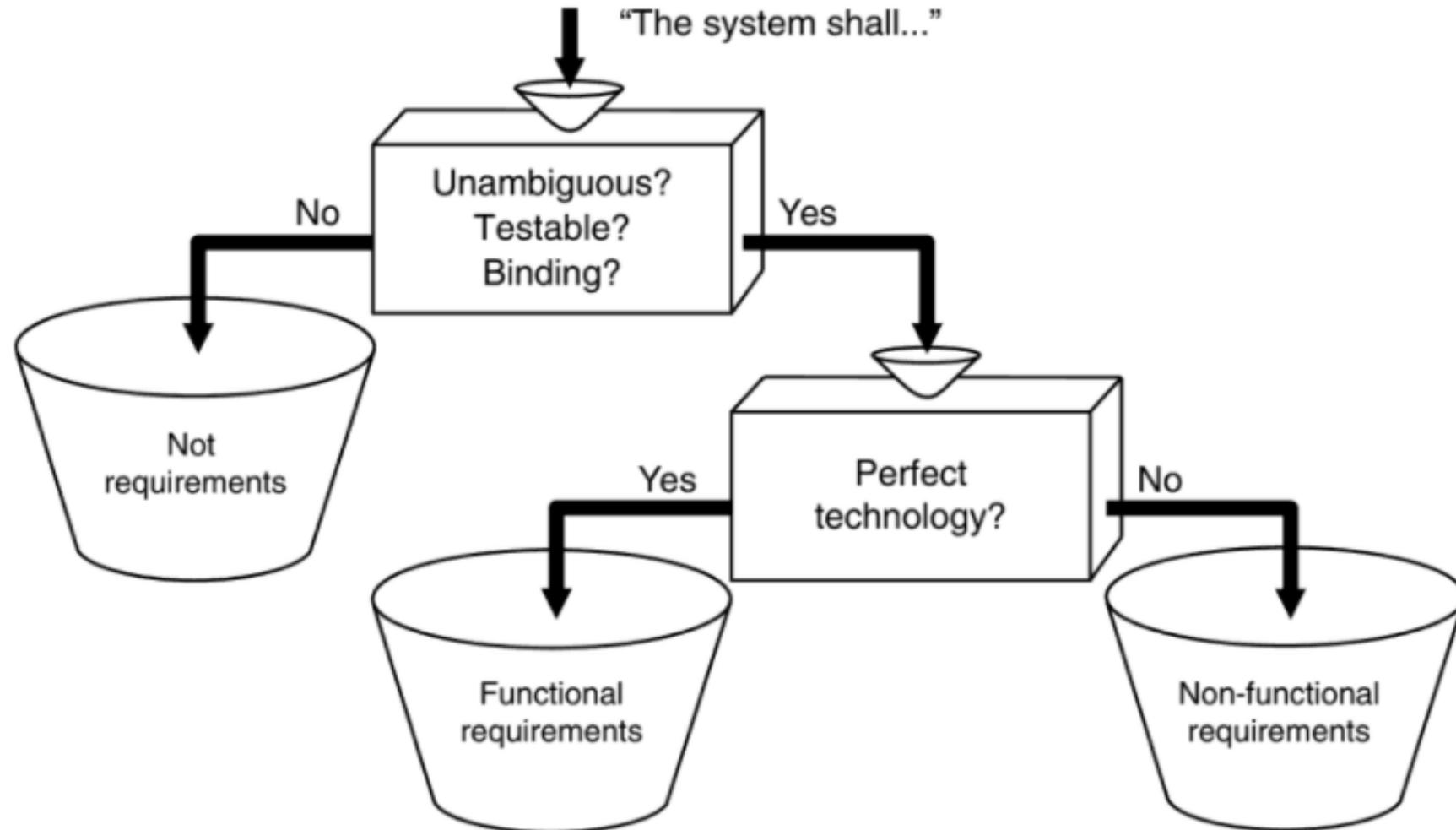
Physical requirements describe hardware characteristics in its size, weight, electrical consumption, and operating conditions.

CONCEPT OF “PERFECT TECHNOLOGY”

The book *Essential Systems Analysis* by Steve McMenamin and John Palmer⁹ introduces the concept of “perfect technology.” A perfect computer would have:

- Infinite speed
- Unlimited memory
- Transparent interface
- Consume no energy
- Generate no heat
- Take up no space
- Never fail
- Cost nothing
- ...

CONCEPT OF “PERFECT TECHNOLOGY”



CONCEPT OF “PERFECT TECHNOLOGY”

The perfect technology assumption states that events should be included during analysis only if the system would be required to respond under perfect conditions.

If requirement would still to be stated in spite of having perfect technology, then it is a functional requirement. Otherwise, it is a non-functional requirement.

MORE ON NON-FUNCTIONAL REQUIREMENTS

The nonfunctional requirements can be further separated into:

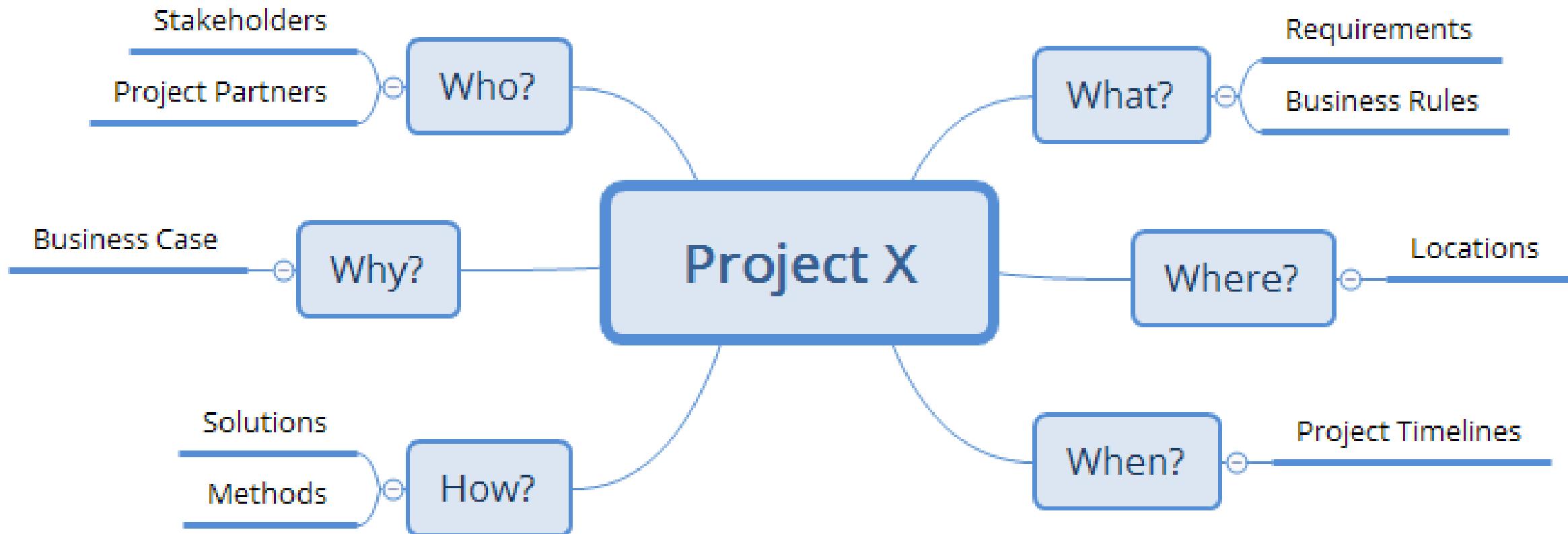
- *Nonfunctional how requirements*—Requirements that constrain use of specific automation technologies. Examples would be use of specific computing platforms (Windows™, MacOS™, Android OS™, iOS™, etc.), programming languages (Java, C++, C#, Python, etc.), compatibility with specific Web browsers (Chrome™, Safari™, Internet Explorer™, Edge™, etc.), use of given database engines (Oracle™, SQL Sever™, MySQL™, etc.), Tinker toys, and so on
- *Nonfunctional how well requirements*—Requirements that don't constrain developers to specific, named technologies. Instead, these specify “qualities of service”: levels of performance that any automated solution must exhibit. Examples would be response time, throughput, accuracy, reliability, scalability, and so on.

	Functional requirements	Nonfunctional requirements
Objective	Describe what the product does	Describe how the product works
End result	Define product features	Define product properties
Focus	Focus on user requirements	Focus on user expectations
Essentiality	They are mandatory	They are not mandatory but desirable
Origin type	Usually defined by the user	Usually defined by developers or other tech experts
Testing	Component, API, UI testing, etc. Tested before nonfunctional testing	Performance, usability, security testing, etc. Tested after functional testing
Types	Authentication, authorization levels, data processing, reporting, etc.	Usability, reliability, scalability, performance, etc.

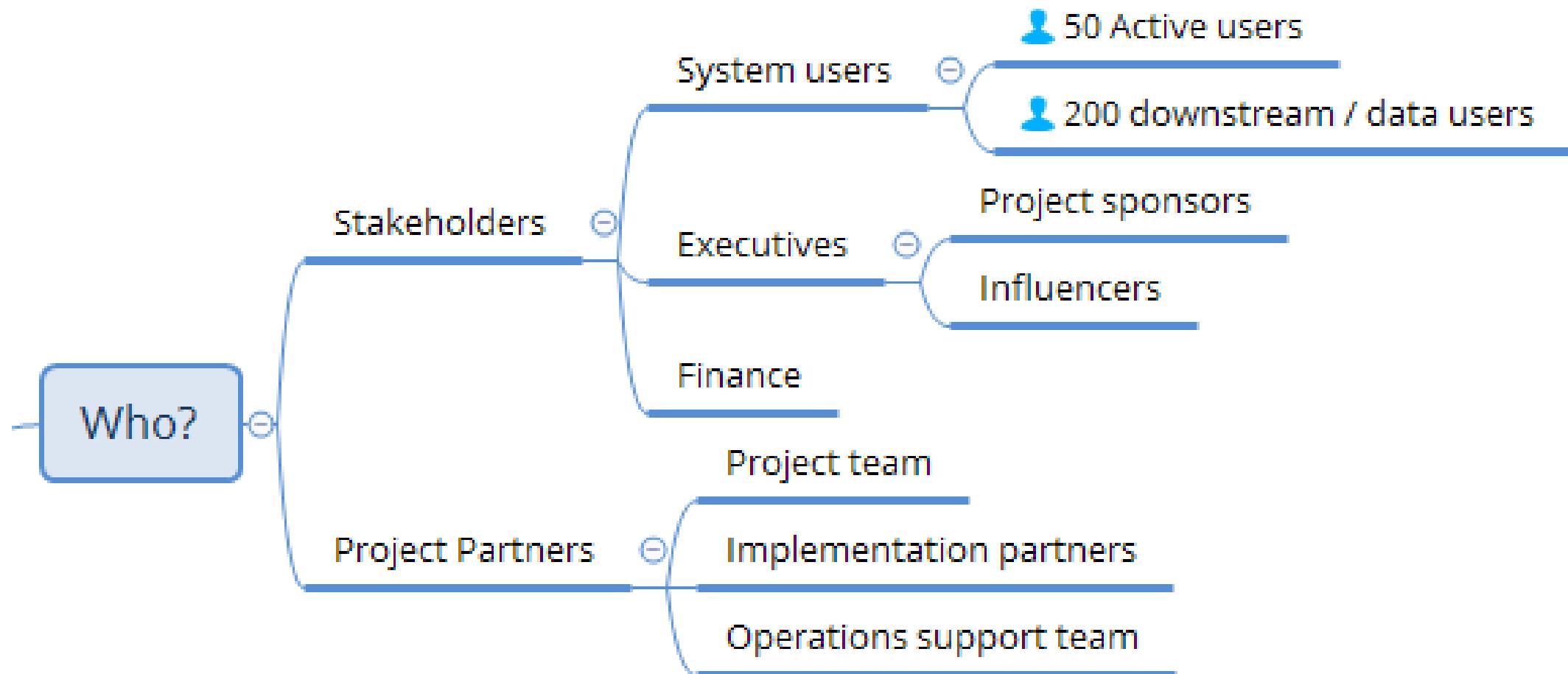
REQUIREMENT ELICITATION TASKS

- 1- Prepare for Elicitation
- 2- Conduct Elicitation
- 3- Confirm Elicitation results
- 4- Manage Stakeholder Collaboration

MIND MAP - EXAMPLE



MIND MAP - EXAMPLE



MANAGE STAKEHOLDER COLLABORATION

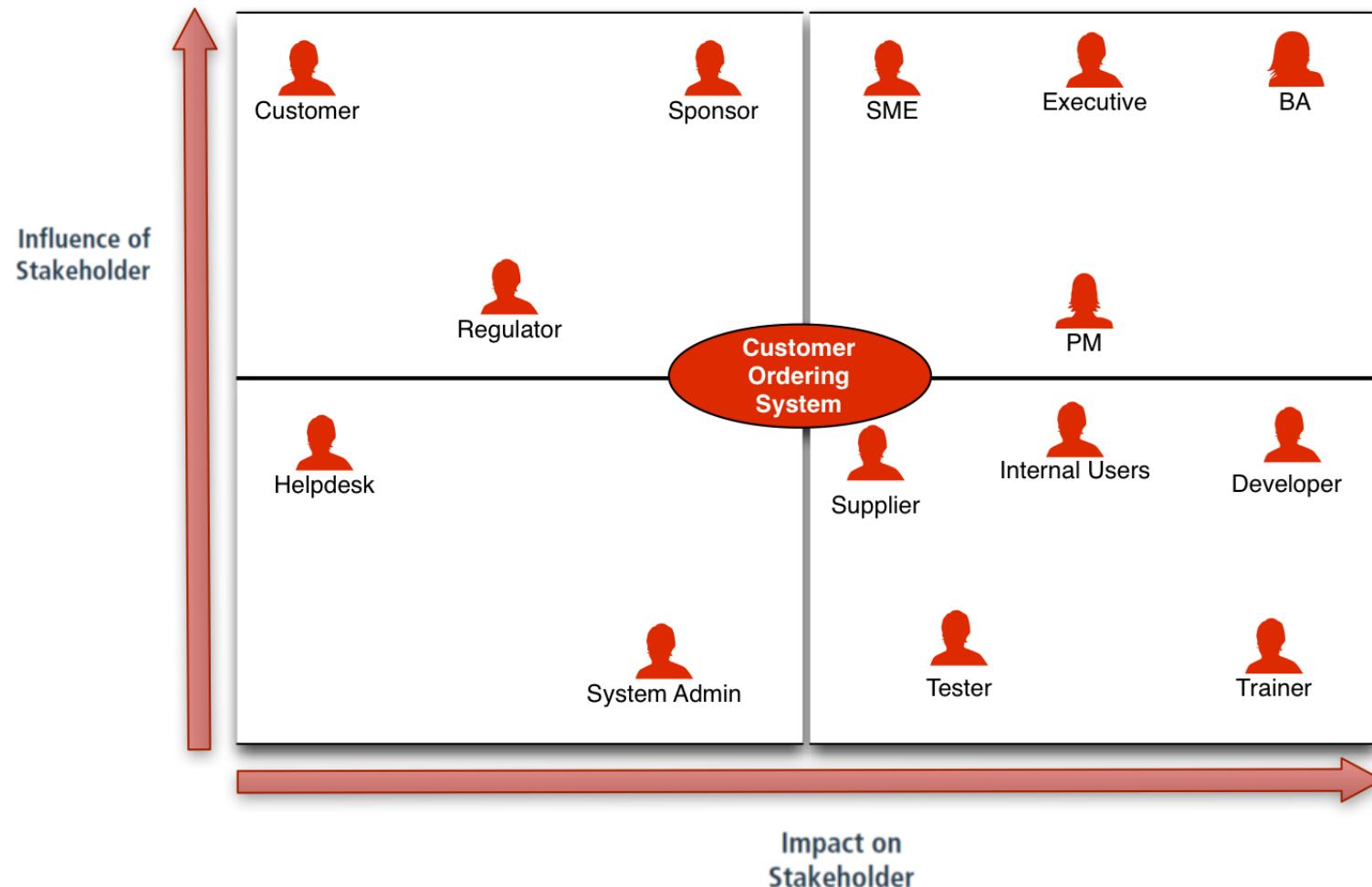
1- Stakeholder Lists

2- Stakeholder Maps

3- Persona

STAKEHOLDER MATRIX

Stakeholder Matrix: maps the level of stakeholder influence against the level of impact on stakeholder.



PERSONA

One popular technique for summarizing user classes is to give each user class a fictional representative, with typical characteristics and often a little back story.

- Bob is an IBM sysadmin in New York
- Fritz is the 50-year-old father of a German swimmer



PERSONA

Personalization

Include personal information about the individual

Job-related

Include details of the individual's job

Persona

Include details of their interest in the product

Relevance

Include details of their education and experience

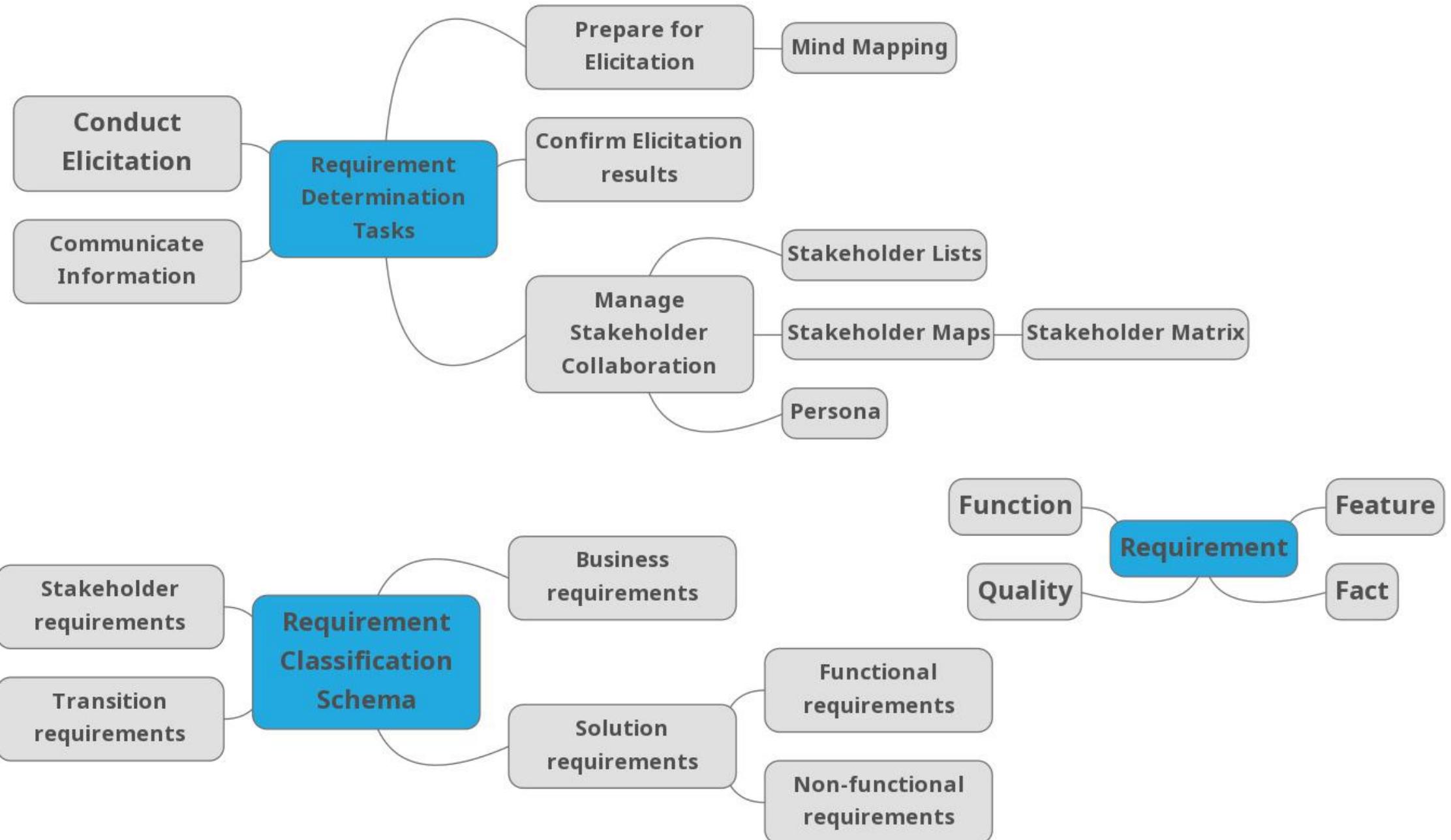
Education

PROTO-PERSONA

Personas that are developed on the basis of **limited user information** are called proto-personas.

Proto-personas may be created as a collective team exercise using whatever information is available about potential product users.

They can never be as accurate as personas developed from detailed user studies, but they are **better than nothing**.





A magnifying glass is positioned over the word "ASSESSMENT". The word is written in a bold, dark blue sans-serif font. The magnifying glass has a black handle and a silver frame. The lens is focused on the letter "S", making it appear larger and more prominent. The background is a light gray gradient.

ASSESSMENT

EXERCISE: E-COMMERCE SYSTEM

An E-commerce software system is going to be developed. This system allows customers to browse the store catalog, pick the items, and put them in an e-cart. It provides several forms of payments such as credit card, personal checks, and money orders. It shall provide to customers a set of shipment methods, which are ground, second business day, and next business day. The system has an administration backend that allows administrators to add new products, manage customers' accounts, and manage inventory. This system should provide secure payment transactions. It also has capabilities to contact customers via e-mail. The system should conform to all applicable local and international laws. It should also conform to the company standard STD0945. The requirements at the client side are no more than a web-browser and a reasonable amount of memory on a computing device. This system should have a fast response time and should tolerate common types of faults.

EXERCISE: VAGUE REQUIREMENTS/GOALS

An E-commerce software system is going to be developed. This system allows customers to browse the store catalog, pick the items, and put them in an e-cart. It provides several forms of payments such as credit card, personal checks, and money orders. It shall provide to customers a set of shipment methods, which are ground, second business day, and next business day. The system has an administration backend that allows administrators to add new products, manage customers' accounts, and manage inventory. This system should provide **secure payment transactions**. It also has capabilities to contact customers via e-mail. The system should conform to all applicable **local and international laws**. It should also conform to the company standard **STD0945**. The requirements at the client side are no more than a **web-browser** and a **reasonable amount** of memory on a computing device. This system should have a **fast response time** and should **tolerate common types of faults**.

EXERCISE: EXTRACT FUNCTIONAL REQUIREMENTS

An E-commerce software system is going to be developed. This system allows customers to **browse the store catalog, pick the items, and put them in an e-cart**. It provides several forms of **payments such as credit card, personal checks, and money orders**. It shall provide to customers a set of **shipment methods, which are ground, second business day, and next business day**. The system has an administration backend that allows administrators to **add new products, manage customers' accounts, and manage inventory**. This system should provide **secure payment transactions**. It also has capabilities to **contact customers via e-mail**. The system should conform to all applicable local and international laws. It should also conform to the company standard STD0945. The requirements at the client side are no more than a web-browser and a reasonable amount of memory on a computing device. This system should have a fast response time and should tolerate common types of faults.

EXERCISE: EXTRACT NON-FUNCTIONAL REQUIREMENTS

An E-commerce software system is going to be developed. This system allows customers to browse the store catalog, pick the items, and put them in an e-cart. It provides several forms of payments such as credit card, personal checks, and money orders. It shall provide to customers a set of shipment methods, which are ground, second business day, and next business day. The system has an administration backend that allows administrators to add new products, manage customers' accounts, and manage inventory. This system should provide secure payment transactions. It also has capabilities to contact customers via e-mail. The system should conform to all applicable local and international laws. It should also conform to the company standard STD0945. The requirements at the client side are no more than a web-browser and a reasonable amount of memory on a computing device. This system should have a fast response time and should tolerate common types of faults.

EXERCISE

Hospital management system keeps information related to the doctors, patients, visits, and prescriptions.

1. Doctor can visit the patients, examine them, add diagnoses, record symptoms, prescribe drugs, and schedule medical procedures, report the patient condition, and finally discharge the patient.
2. System keeps information regarding symptoms, diagnosis details and prescription drug details.
3. Patient can register himself, unregister, confirm the medical procedure, and fills request form for discharge.
4. Patient history can be accessed by the doctor to make better judgements.
5. Patient status includes: hospitalized, intensive care, away and discharged.

EXERCISE

- Re-write requirement 3 as a stakeholder requirement.
- Re-write requirement 4 as a non-functional requirement.
- Suggest a transition requirement for the hospital MS.
- Which of these requirements is a business requirement?
- Suggest two personas after reading requirement 3.