

REQUIREMENTS ENGINEERING.

Topics covered

1. Functional and non-functional requirements
→ functional requirements, non-functional requirements, case study.
2. The software requirements document.
3. Requirement specification.
4. Requirement engineering process:
→ Requirement elicitation and analysis.

Requirement?

The requirements for a system are the descriptions of the services provided by the system and its operational constraints. It may range from high-level abstract sysb statement of a service of a system constraint to a detailed mathematical functional specification.

Requirement Engineering.

The process of finding out, analysing, documenting and checking these services and constraints is called requirement engineering (RE).

Types of requirements.

User requirements are statements, in a natural language plus diagrams, of

what services the system is expected to provide and the constraints under which it must operate.

2. System requirements are more detailed description of the software system's functions, services and operational constraints in detail. The system requirement document (sometimes called a functional specification) should be precise. It should define exactly what is to be implemented. It may be part of the contract between the system buyer and the software developers.

case study below figure shows distinction between user and system requirements for a mental health care Patient management system (MHC-PMS) show how user requirement can be expanded in detail into several system requirement.

User Requirement definition

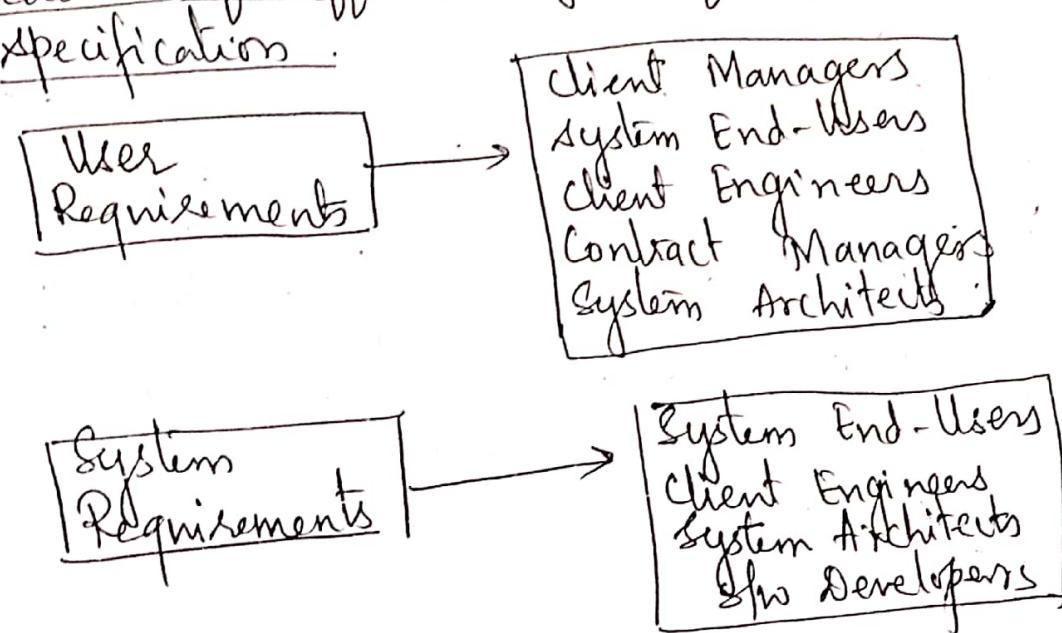
1. The MHC-PMS shall generate monthly management reports showing the cost of drugs prescribed by each clinic during that month.

System Requirement Specification

- 1.1 On the last working day of each month, a summary of drugs prescribed their cost, and the prescribing clinics shall be generated

- 15
- 1.2 The system shall automatically generate the report for printing after 17.30 pm on the last working day of a month.
 - 1.3 A report shall be created for each clinic and shall list the individual drug names, the total number of drugs prescribed, the number of doses prescribed and total cost of prescribed drugs.
 - 1.4 If drugs are available in different dose units (eg 10 mg, 20 mg) separate reports shall be created for each dose unit.
 - 1.5 Access to all cost reports shall be restricted to authorized users listed on a management access control list.

Readers of different types of requirements specification



1. The readers of user requirements are not usually concerned with how the system will be implemented and may be managers who are not interested in the detailed facilities of the system.
 2. The readers of system requirements need to know more precisely what the system will do because they are concerned with how it will support the business processes or because they are involved in the system implementation.
- 2 Functional and Non-functional requirement
- Software system requirements are further classified as functional requirements, non-functional requirements
- Functional requirements: These are statements of services the system should provide, how the system should react to particular inputs, and how the system should behave in particular situations. In some cases, the functional requirements may also explicitly state what the system should not do.

(3)

Non functional requirements: These are constraint on the services or functions offered by the system. They include time constraints, constraints on the development process and standards.

Non functional requirement apply for a whole system, they do not apply to individual system features or services.

Functional requirement

1. It describe what the system should do.
2. These requirement depend on the type of software being developed, the expected users of the software and the general approach taken by the organization when writing requirements.
3. Functional user requirements may be high-level statements of what the system should do.
4. Functional system requirements should describe the system service in detail. (its I/P, O/P, exceptions & so on.)

Case Study: Functional requirement for MHC-PMS

1. A user shall be able to search the appointments lists for all clinics.
2. The system shall generate each day, for each clinic, a list of patients who are expected to attend appointments that day.
3. Each staff member using the system shall be uniquely identified by his or her 8-digit

employee number.

Requirement Imprecision.

Imprecision in the requirement specification is the cause of many software engineering problems. i.e. problems arise when the requirements are not precisely stated. Ambiguous requirements may be interpreted in different ways by developers and users.

Consider In the above case study example consider the term 'Search' in requirement.

- User intention - search for a patient name across all appointments in all clinics.
- Developer intention interpretation - search for a patient name in an individual clinic, user chooses clinic & then search.

The functional requirement specification of a system should be both complete and consistent.

Completeness means that all services required by the user should be defined.

Consistency means that requirements should not have contradictory definitions.

study: functional requirements Management system for library

1. Add new users ie User request for registration.
2. Issue book
3. Return book
4. Add new book category

Detail description of each functionality

1. Add new user.
 - User request for registration.
 - Librarian add new user to system
 - Enter all details in registration form
 - Generates a unique ID for that user.
 - It is high priority requirement.
2. Issue book.
 - Request for issuing a specific book which is available.
 - Student book limit is available than book will be issued.
 - Limit exceeds than book will not be issued.
 - Only registered member can issue a book.
 - This feature(functionality) is high priority requirement.
3. Return book
 - Request for returning book
 - Get member ID and check return date.
 - Date exceed than charge fine against member ID.

- Returned book must be available in database
4. Add new Book category.
- If new book arrives then identify its category
 - If it is of existing category then Librarian adds to it else new category of book will be created and added.
 - This feature is medium priority.

Non-functional requirements.

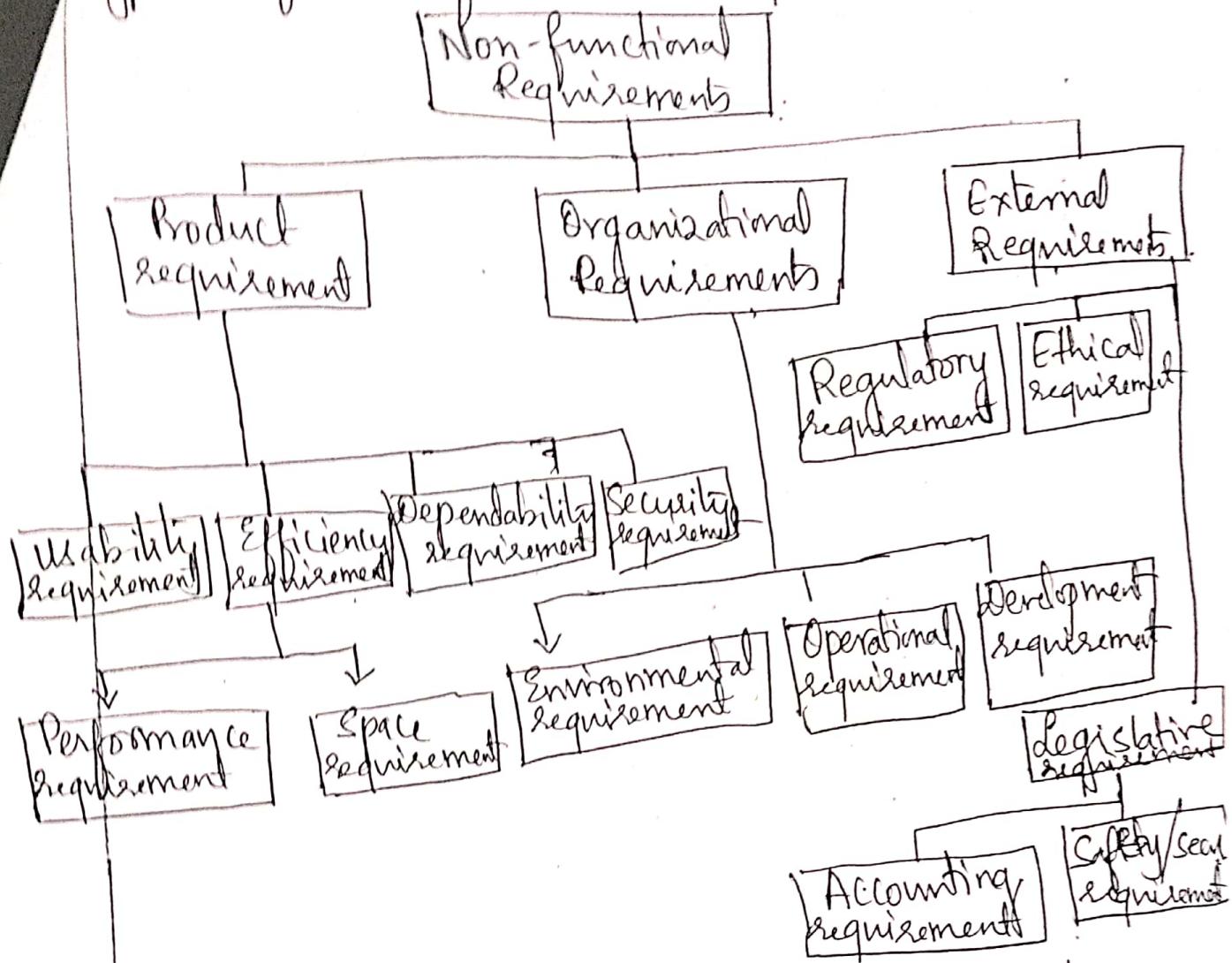
Non functional requirements are constraints on the services or functions offered by the system. They include timing constraints, constraints on development process and standards.

Non-functional requirement often apply to the system as a whole rather than individual features or services.

For example ① if an aircraft system does not meet its reliability requirements, then it will not be certified as safe for operation.
② If a real-time control system fails to meet its performance requirements, the control functions will not operate correctly.

Types of non-functional requirements.

(5)



Types of non functional requirements are

- a. Product requirements: These requirements specify (constraints) product behaviour.
Example: performance requirements on how fast the system must execute and how much memory it requires, reliability requirements that set out the acceptable failure rate, security requirements and usability requirements.

- b. Organizational requirements: These requirements are derived from policies and procedures of the customer's and developer's organization. Example process standards used (development), implementation (operational) requirements, environment requirements etc.
- c. External requirements: Requirements which arise from factors which are external to the system and its development process. e.g. Regulatory requirements. Example Regulatory, which sets out what must be done for the system to be approved for use by a regulator, such as central bank, system operates within laws (legislative requirements), ethical requirements etc.
- Product requirement: The MHC-PMS shall be available to all clinics during normal working hours (Mon-Fri 8:30-5:30).
- Organizational req: Users of the MHC-PMS system shall authenticate themselves using their health identity card.
- External requirement: The system shall implement Patient privacy provisions as set out in H Stan - 03-2006 - priv.

Non-functional requirements for library system⁽⁶⁾

Product requirement:

- Performance of the system should be fast & accurate
- The system shall be available during the working hours from 8 to 5.
- Should be able to handle large amount of data. (Space req).

Organisational requirement:

- The system development process and deliverable documents shall conform to the process and deliverables defined in XY260-SP-STAN-95.

External requirements:

- The system shall not disclose any personal information about the system users apart from their name and library reference number to the library staff who use the system (security req).
- Must be two servers one main and one backup server (Safety req).

Goals and verifiable requirements:-

A common problem with non-functional requirements is that they can be difficult to verify. User or customers intention is ease of use, ~~and~~ the ability of the system to recover from failure.

System goal.

A system should be easy to use by experienced controllers and should be organised in such a way that user errors are minimized.

A verifiable non-functional req.

Experienced controllers shall be able to use all the system functions after a total of two hour's training. After this training the average number of errors made by experienced users shall not exceed two per day.

Metrics for specifying non-functional requ

Property

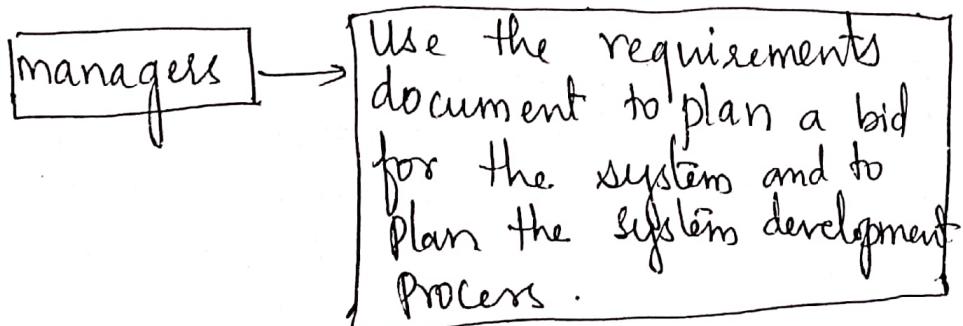
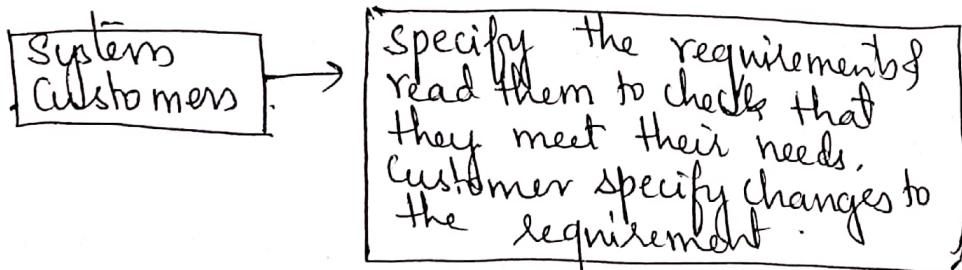
Measure

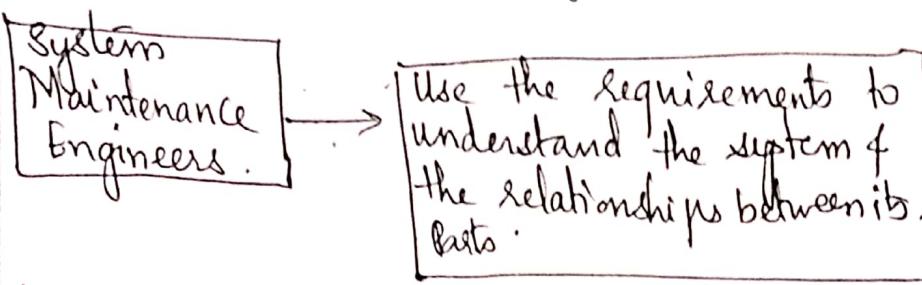
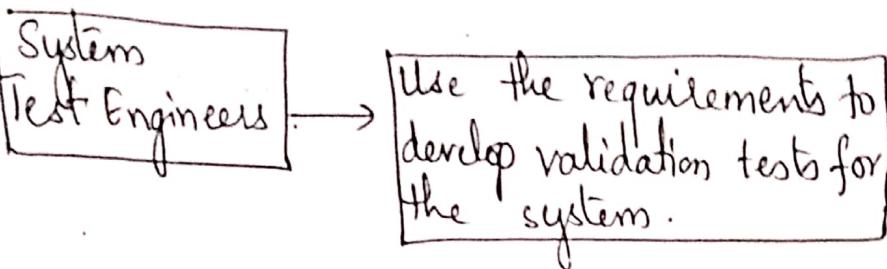
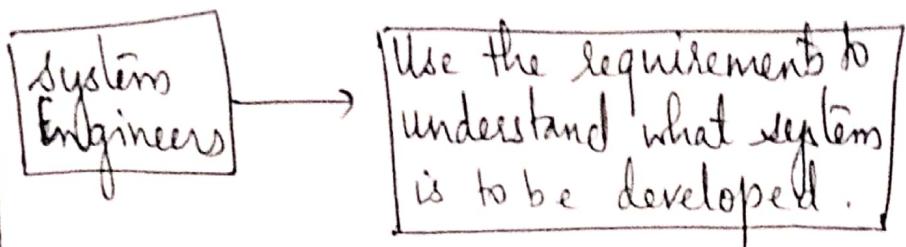
- 1) Speed → Processed transactions/second
User/event response time
Screen refresh time.
- 2) Size → K.bytes
Number of ROM chips.
- 3) Ease of use → Training Time
No of Help frames.
- 4) Reliability → Mean time to failure
Probability of unavailability
Rate of failure occurrence
Availability.
- 5) Robustness → Time to start after failure
Percentage of events causing failure
Probability of data corruption on failure.
- 6) Portability → Percentage of target dependent statements
Number of target statements.

The software requirements document or (SRS) (7)

- The software requirements document is the official statement of what is required of the system developers.
- It should include both definition of user requirements and specification of the system requirements.
 - Many agile development methods argue that requirements change so rapidly that a requirement document is out of date as soon as it is written so the effort is largely wasted, so it uses Extreme programming where user requirements are collected incrementally and write this on cards as user stories. The user then prioritizes requirements for implementation in next increment.

The Users of requirements document.





IEEE standard for requirement document

The structure of a requirements document.

- | <u>Chapter</u> | <u>Description</u> |
|------------------------|--|
| 1. <u>Preface</u> | : This should define the expected relationship of the document and describe its version history, including a rationale for the creation of a new version and a summary of the changes made in each version. |
| 2. <u>Introduction</u> | : It should briefly describe the system functions and explain how it will work with other systems. It should also explain how the system fits into overall business or strategic objectives of the organization commissioning the S/w. |

- 3. Glossary: This should define the technical terms used in the document. ⑧
- 4. User requirement definition: Describe the services provided for the user. The non-functional reqs. should also be described in this section. This description may include natural language, structured diagrams or mathematical notations that are understandable to customers.
- 5. System architecture: It should present a high level overview of the anticipated system architecture, showing the distribution of function across system module. Reused component should be highlighted.
- 6. System requirement specification: This should describe functional and non-functional requirement in detail. Interfaces to other systems may be defined.
- 7. System models: This may include graphical system models showing the relationships between the system components and its environment. Ex are object models, data-flow models or semantic data models.
- 8. System evolution: It should describe the fundamental assumptions on which the system is based and anticipated changes due to hardware evolution, or user needs so on.

9. Appendices : These should provide detailed, specific information that is related to application being developed for ex h/w & database descriptions etc.
10. Index : Several indexes to the document may be included. As well as a normal alphabetic index, diagram index, an index of functions, tables so on.

Requirement Specification.

Requirement specification is a process of writing down the user and system requirement in a requirements document.

- User and system requirements should be clear, unambiguous, easy to understand, complete and consistent,
- User requirements have to be understandable by end users and customers who do not have a technical background.
- System requirements are more detailed requirement and may include more technical information.

Ways of writing a system requirements specification.

(9)

Notation

1. Natural language sentences.

Description:

: The requirements are written using numbered sentences in natural language. Each sentence should represent (express) one requirement.

2. Structured natural language

: This requirement are written in a natural language on a standard form or template.

Each field provides information about an aspect of the requirement.

3. Design description languages

: This approach uses a language like a programming language, but with more abstract features to specify the requirements by defining operational model of the system. This approach is rarely used but it can be useful for interface specifications.

4. Graphical notations:

Graphical models, supplemented by text annotations are used to define the functional requirements for the system. UML use case & sequence diagram are commonly used.

5. Mathematical specification

: It uses mathematical concepts such as finite-state machine or sets etc. Ambiguity is reduced but it is difficult to understand to a customer.

Requirement and design

- In practice, requirements and design are inseparable.
- In principle requirements ^{should} state what the system should do and design should describe how it does.
- A system architecture may be designed to structure the requirements.
- A system may inter-operate with other systems that generate design requirements.

Natural language specification

User requirements are almost always written in natural language supplemented by appropriate diagrams and tables in the requirements document.

Guideline for writing natural language requirements

1. Invent a standard format and ensure that all requirement definitions adhere to that format.
2. Use language consistently to distinguish between mandatory and desirable requirements.
Mandatory requirements are requirements that the system must support and are usually not essential and are written using 'shall'. Desirable requirements are written using 'should'.
3. Use text highlighting (bold, italic or color) to pick out key parts of the requirements.

4. Avoid the use of computer jargon, abbreviations, and acronyms. (10)
5. Whenever possible, try to include an explanation (rationale) of why a requirement is necessary.

Problems

case study Example of ~~non~~ requirements for the insulin pump software system.

1. The system shall measure blood sugar and deliver insulin, if required every 10 minutes.
2. A system shall run a self-test routine every minute with the conditions to be tested and associated actions defined in a table.(further)

Problems with natural language:

1. Lack of clarity: Precision is difficult without making the document difficult to read.
2. Requirement confusion: functional and non-functional requirements tend to be mixed-up.
3. Requirement amalgamation: Several different requirements may be expressed together.

Structured specifications [written in standard form or template]

Structured natural language is a way of writing system requirements where the freedom of the requirements writer is limited and all requirements are written in a standard way.

- This works well for some types of requirements e.g. requirements for embedded control systems but is sometimes too rigid for writing business system requirements.
- The advantage of this approach is that it maintains most of the expressiveness and understandability of natural language but ensures that some degree of uniformity is imposed on the specification.

case study: A structured specification of a requirement for an insulin pump.

Insulin Pump/Control software /SRS /3.3.2.

function : → compute insulin dose: safe sugar level

description : → computes the dose of insulin to be delivered when the current measured sugar

Inputs : → current sugar level is in the safe zone between 34.7 units

Inputs : → current sugar reading (s_0), & previous reading (s_0 & s_1).

Source : → current sugar reading from sensor. Other readings from memory.

Outputs : → CompDose - the dose in insulin to be delivered

Destination : → Main control loop.

Action : → CompDose is zero if sugar level is stable.
If the level is increasing and the rate of increase is increasing, then

Compose dose is calculated by dividing between the current sugarlevel reading & previous sugar level & rounding the result. If the result is rounded to zero the Compose dose is set to the minimum dose that can be delivered.

Requirements \rightarrow To previous readings so that the rate of change of sugar level can be computed.

Pre-condition \rightarrow The insulin reservoir contains at least the maximum allowed single dose of insulin.

Post-condition \rightarrow r_0 is replaced by r_1 then r_1 is replaced by r_2 .

Side-effects \rightarrow None.

An example of a form-based specification shows in above example shows how to calculate dose of insulin to be-delivered when the blood sugar is within a safe band.

P.T.O.

Ex → for a Bus reservation system list the system requirements using structured specification form.

Bus reservation system / management s/w / SRS / 3.32 .

function: Compute Total amount of tickets.

Description: Compute the total amount to be paid after buying the ticket when the number of tickets are less than or equal to total number of available tickets

Inputs : Number of tickets.

Source : Current available bus chart.

Outputs : Reservation details

Destination: Bus reservation chart.

Action :- When a seat is selected and booked, the total number of seats minus the number of seats currently booked is calculated and new chart is generated by blocking the seats booked by the user.

Requirements: Reservation chart.

Pre-condition :- The bus-reservation chart should have some vacant seats.

Post-condition :- Updated bus reservation chart .

Side-effects :- None.

When a standard form is used for specifying functional requirements, the following information should be included: ⁽¹²⁾

1. A description of the function or entity being specified.
2. A description of its input and where these come from
3. A description of its outputs and where these go to.
4. Information of what other entities are used.
5. A description of the action to be taken.
6. If a functional approach is used, a pre-condition setting out what must be true before the function is called and a post-condition specifying what is true after the function is called.
7. Description of the side effects (if any) of the operation.

Advantages

1. Using structured specifications removes some of the problems of natural language specification.
2. Variability in the specification is reduced and requirements are organized more effectively.

Disadvantages

1. It is difficult to write requirements in an unambiguous way, particularly when complex

- computations are required.
- Things cannot be specified clearly.

Tabular specification

- Tables are particularly useful when there are a number of possible alternative situations and need to describe the actions to be taken for each of these.
- Graphical models are most useful when you need to show how state changes or where you need to describe a sequence of actions.

Example Below, figure shows a tabular description for specification of computation for an insulin pump.

Condition	Action
Sugar level falling ($r_2 < r_1$)	CompDose = 0 .
Sugar level stable ($r_2 = r_1$)	CompDose = 0
Sugar level increasing & rate of increase decreasing ($(r_2 - r_1) < (r_1 - r_0)$)	CompDose = 0 .
Sugar level increasing & rate of increase stable or increasing ($(r_2 - r_1) \geq (r_1 - r_0)$)	CompDose = round($(r_2 - r_1)/4$) If rounded result = 0 then CompDose = MinimumDose.

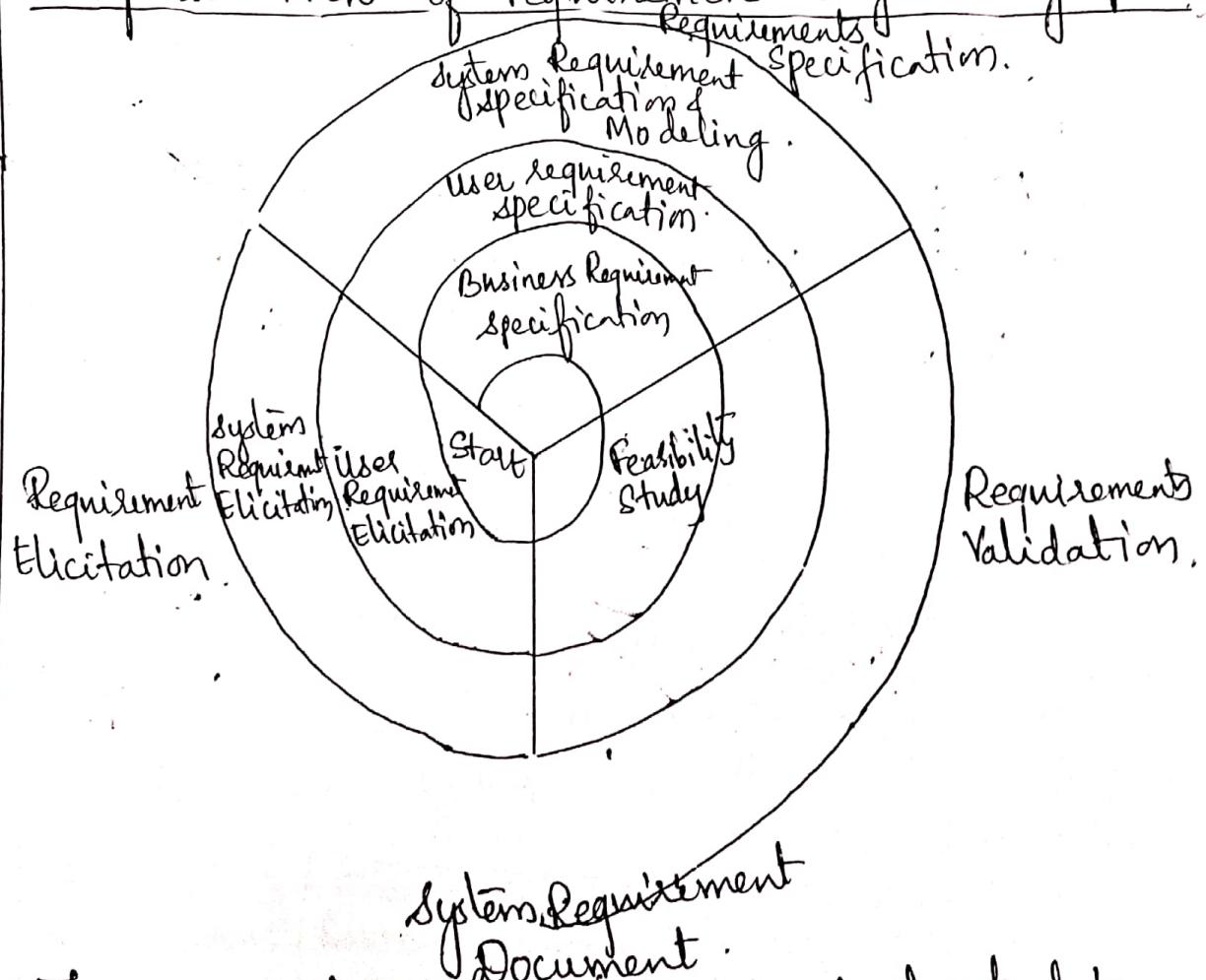
Requirements engineering Processes

Requirements engineering processes may include 4 high-level activities (this includes 1. These focus on assessing if the system is useful to the business (feasibility study).

2. Discovering requirements (elicitation & analysis)
3. Converting these requirements into some standard form (specification) user req, system req.
4. Checking that these requirements actually define the system that the customer wants (validation).

The above 4 activities can be shown interleaved which is an iterative process. These activity are organized as an iterative process around a spiral, with output as system requirement document.

A spiral view of requirement engineering process



→ The amount of time and effort devoted to each activity in each iteration depends on the stage of the overall process and the type

of system being developed.

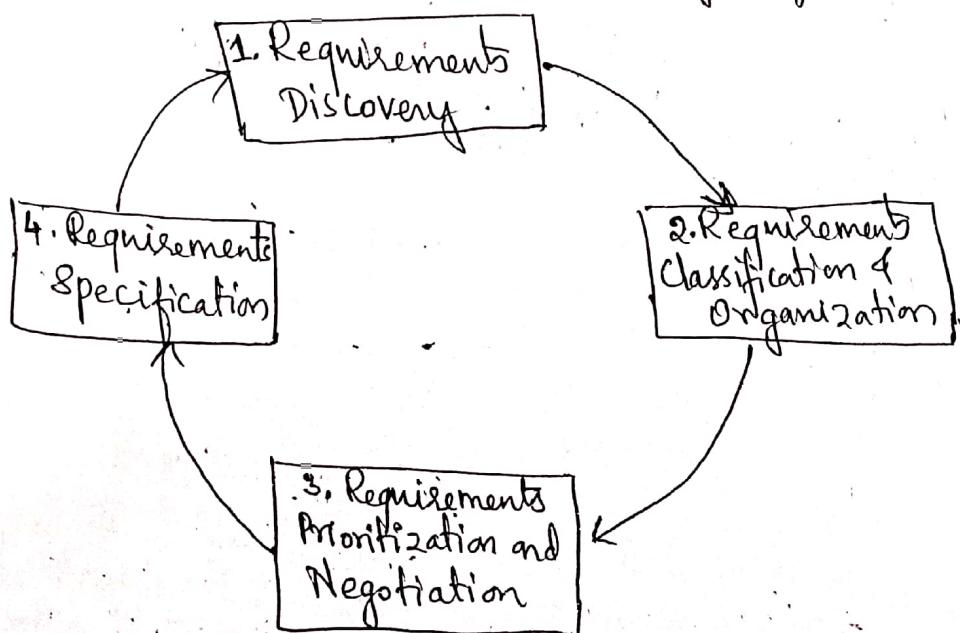
- Initial stage most effort is spent on understanding high-level business and non-functional requirements & user requirements for a system.
- Later stage in the process, ie the outer ring of spiral - more effort is spent on eliciting & understanding the detailed system requirements

Requirement elicitation and analysis

After an initial feasibility study the next stage of the requirements engineering process is requirement elicitation and analysis. i.e. Researching i.e. Researching and discovering requirements of the system from user and customers.

A process model of the elicitation and analysis process is shown in below fig.

Requirement elicitation and analysis process.



The process activities are

1. Requirements discovery: This is a process of interacting with stakeholders of the system to discover their requirements, domain requirements are also discovered at this stage.
2. Requirements classification and organization: This activity takes unstructured collection of requirement groups related requirements, and organizes them into coherent clusters.
3. Requirements prioritization and negotiation:- When multiple stakeholders are involved, requirements will conflict. This activity is concerned with prioritizing requirements resolving requirements conflicts.
4. Requirements specification (document): The requirements are documented and input into the next round of the spiral.

Problems of requirement elicitation:

Eliciting and understanding of the requirements from system stakeholders is a difficult process for several reasons.

1. Stakeholders don't know what they really want from computer system, except in most general
2. Stakeholders express requirements in their own terms and with implicit knowledge of their own work.
3. Different stakeholders have different requirements and they may express these in different ways.

requirement engineers must discover the commonalities & conflicts.

4. Political factors may influence the requirement of a system.
5. The requirements change during the analysis process hence the economic & business environment in which the analysis takes place is dynamic.

Question Bank

Case Study

Identify Functional and Non-functional requirements for ATM.

Functional Req.

1. The card reader determines the account number from the entered card.
2. User is prompted to enter a PIN after a card is entered.
3. Menu is displayed to user with following options withdraw, Deposit, Pay Bill, Account Update & Exit.
4. A transaction record can be printed upon demand.
5. Transaction can be canceled at any prompt by the user pressing the CANCEL button.

Non function requirement

- 1) ATM unit consists of a display monitor, card reader, cash dispenser, keypad and a printer.
- 2) ATM server should be available to the user 24x7 to users.
- 3) PIN must be entered within 3 times. 20 seconds & maximum
- 4) The system should be secure.

Question Bank (Previous year questions)

- Q1. Distinguish 'User level' requirements with the 'System level' requirement using an example show the change in requirement description at different levels of process.
- Q2. Compare functional & non-functional requirement, with a neat block diagram explain non-functional requirements types?
- Q3. With a neat diagram explain the various steps in Requirements Elicitation and Analysis.
- Q4. Explain any five metrics for specifying non-functional requirements?
- Q5. Explain the users of requirement documents?
- Q6. Explain the different ways of writing systems requirement specification (SRS).
- Q7. Define user requirement & system requirement. Who are the readers of these requirements?
- Q8. Describe the structure of requirement document in detail.
- Q9. List the problems of requirement analysis & explain the requirement elicitation & analysis process with a neat diagram.
- Q10. Discover a set of non-functional requirements of the ticket booking system for airlines, setting out its expected reliability and response time.
- Q11. Discuss the problems of using natural language for defining user and systems requirements with examples.

Q12 Reliance is planning for an online shopping mart. List and briefly explain the non-functional requirement for the application?

Security

Reliability

Maintainability

Portability

Extensibility

Reusability

Resource Utilization

Secure access of confidential data, user's details

24x7 availability of the server

Better component design to get better performance at peak time.

Advertisement space

Flexible service based architecture will be highly desirable for future extension. NF req. defines system properties & constraints

Q13 Write ~~a~~ ^{systems} requirements

Q13 Write a system requirements using structured specification for issuing permanent licence for an applicant.

Function: Issue Permanent Licence

Description: Issue Permanent licence to an applicant if he is above 18 years & holding valid learner licence

Inputs : Age of an applicant and LLR No,

Source : RTO

Output : Permanent licence details

Destination: RTO rules & regulations

Action : When an application is received, it is checked for age validity and learner licence validity, if both are satisfied then permanent licence can be issued.

Requires: RTO rules

Pre condition: Applicant should be above 18 years

Post-condition: - & holding LLR

Side effects : None