

Module - 2

Software Requirements:

The requirements for a system are the descriptions of the services provided by the system & its operational constraints. These requirements reflect the needs of customer's for a system that helps solve some problems.

The process of finding out, analysing, documentation & checking these services and constraints called Requirement Engineering (RE). There are different types of requirements. They are:

- ① Known requirements
- ② Unknown requirements
- ③ Undreamt requirements.

① Known requirements :-

Something a stakeholder believes to be implemented.

② Unknown requirements :-

Forgotten by the stakeholder because they are not needed right now.

③ Undreamt requirements :-

Stakeholder may not be able to think of new requirements due to limited domain knowledge.

A known, unknown and undreamt requirements may be functional & non-functional.

* Functional Requirements :-

These are related to the expectations from the intended Software. They describe what the Software has to do. They are also called product features. Sometimes functional requirements may also specify what the software should not do..

* Non functional Requirements :-

These are quality requirements, that are especially important to users, include specifications of desired performance, availability, reliability, usability and flexibility. The functional requirements are directly related to customer's expectations. Non-functional requirements may make the customer happy and satisfied. These requirements are important for the success of any product.

Technically Software requirements are classified into 2. They are:

- * User requirements
- * System requirements
- * User requirements :-

These are written for the users and include functional & non-functional

requirements. Users may not be the experts of the software field, hence simple language should be used. The software terminologies & notations etc.. should be avoided. user requirement should specify the external behaviour of the system with some constraints & quality parameters.

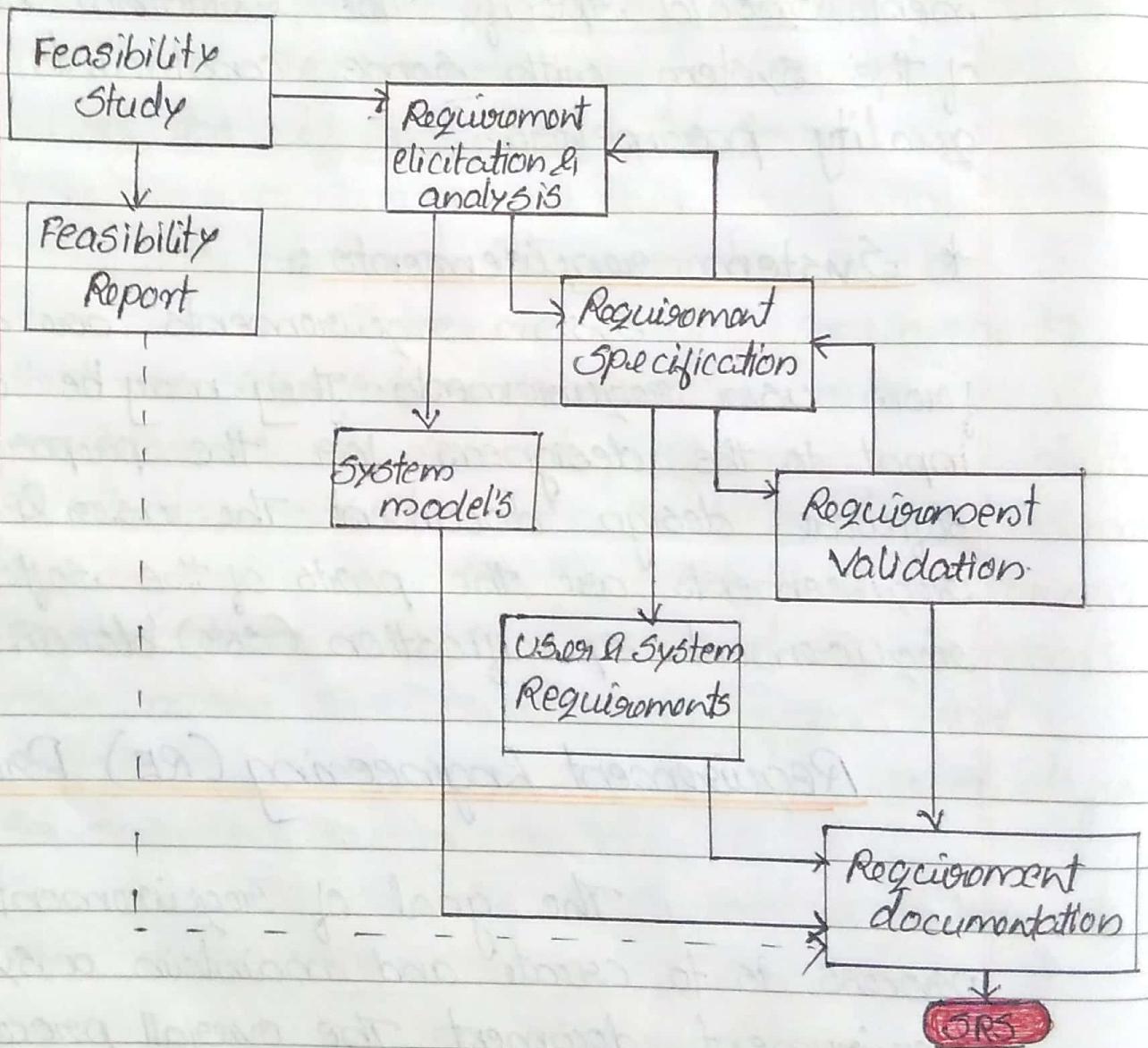
* System requirements :-

System requirements are derived from user requirements. They may be used as input to the designers for the preparation of software design document. The user & system requirements are the parts of the software requirement specification (SRS) document.

Requirement Engineering (RE) Process:-

The goal of requirement engineering process is to create and maintain a system requirement document. The overall process include 4 high level sub process. These are concerned with assessing whether the system is useful to the business (Feasibility Study), discovering requirements (elicitation & analysis), converting

These requirements into some standard form (specification) and checking that requirements actually define the system that the customer wants (validation). RE process is given below:



① Feasibility Study :-

The requirement engineering process should start with a feasibility study. The input to the feasibility study is an outline

description of the system and how the system intended to support business process. The feasibility study is a short focussed study that aims to answer no1 of questions. Questions are:

- ① Does the system contribute to the overall objective of the organization?
- ② Can the system be implemented using current technology and within given cost and schedule constraints?
- ③ Can the system be integrated with the other systems which are already in place?

Carrying out a feasibility study involves information collection & report writing. These information identifies to answer the 3 questions set out above. In a feasibility study you may consult information sources such as managers of departments where the system will be used, software engineers who are familiar with the type of system that is proposed, technology experts and end user's of the system. Feasibility study will complete in 2 or 3 weeks. After this we can writing feasibility study report.

Purpose of feasibility Study:-

The decision to implement any new project or programs must be based on thorough analysis of the proposed project or program. Feasibility Study is defined as an evaluation or analysis of the potential impact of a proposed project or programs. It should also contain extensive data analysis related to financial operational impact & should include advantages & disadvantages of the current situation & proposed project.

② Requirement Elicitation and Analysis:-

The next stage of requirement engineering process is requirement elicitation & analysis. In this activity software engineers work with customers & system end users to find out what services the system should be provided, the required performance of the system & so on.

Requirement elicitation & analysis may involve variety of people in an organization. The term 'stake holders' is used to refer any person or group

who will be effected by the system directly or indirectly.

Stakeholders include end users who interact with the system & everyone else in an organization that may be effected by its installation. Other systems, stakeholders may be engineers who are developing or maintaining related systems & business managers etc.

A very general process model of elicitation & analysis process is shown below:

i) Requirement classification & organization

ii) Requirement prioritization & negotiation.

iii) Requirement discovery

iv) Requirement documentation.

① Requirement discovery :- process of interacting

with stakeholders in the System to collect their requirements.

② Requirement classification & organization :-

This activity takes the unstructured collection of requirements, group related requirements & organize them to clusters.

③ Requirement prioritisation & negotiation :-

where multiple stakeholders are involved requirements with conflicts. This activity is concerned with prioritizing requirements & finding resolving requirements conflicts through negotiation.

④ Requirement documentation :-

The requirements are documented and are put into the next round of spiral.

Requirement elicitation is most difficult & most communication intensive aspect of software development. This activity helps to understand the problem to be solved. Requirements are gathered by asking questions, writing down answers, asking other questions etc.. There are no of requirement elicitation methods. They are:

① Interview

② Brain Storming Session.

- ③ Facilitated application specification technique (FAST)
- ④ Quality function deployment (QFD)
- ⑤ Use case approach.

① Interview :-

After receiving the problem statement from the customer, the first step is to arrange a meeting with the customer. During the meeting or interview both the parties would like to understand each other. The objective of conducting an interview is to understand the customers expectations from the software. Both parties have different goals, opinions, vocabularies & understandings, but one thing is common - both want the project to be a success. Interview may be an open ended, there is no preset agenda.

Eg:- For a "Result management system" requirement engineer may ask some questions:

- ① who will use the Software?
 - ② who will explain the manual systems?
 - ③ How many stakeholders are computer friendly?
- Such questions help to identify all stakeholders who will have interest in the software to be developed.

* Selection of Stakeholders (Using interview):

① Entry level personnel :-

They may not have sufficient domain knowledge & experience, but may be very useful for fresh ideas & different views.

② Mid level Stakeholders :-

They have better domain knowledge & experience of the project. They know the sensitive complex and critical areas of the project.

③ Managers or other stakeholders:-

Higher level management officers like president, general manager, MD etc.. should also be interviewed.

④ Users of the Software :-

This group is perhaps the most important because they will spend more time interacting with the software than anyone else.

⑤ Brainstorming Session :-

It is a group technique, that may be used during requirement elicitation to understand the requirements. The group

discussion may lead to new ideas quickly & help to promote creative thinking. Brainstorming has become very popular & is being used by most of the companies. It promotes creative thinking, generate new ideas & provides platform to share views. All participants are encouraged to say whatever ideas come to mind, no one will be criticized for any idea. This group technique may be carried out with specialised groups like actual group users, middle level managers etc or with total stakeholders. In order to handle situations, a highly trained facilitator may be required. The facilitator may handle groups & group conflicts carefully. The facilitator must also be cautious about individual ego's and will be responsible for smooth conduct of brain storming sessions.

Every idea will be documented in such a way that everyone can see it. white boards, computer projection systems can be used to make it visible to every participants. After the session, a detailed report will be prepared and facilitator will review the report. Every idea will be written in simple english ~~so that it conveys~~

③ Facilitated Application Specification Technique (FAST):

This approach is similar to brain storming session, a team oriented approach is developed for requirements gathering & is called FAST. This approach encourages a creation of a joined team of customers & developers who work together to understand the expectations & propose a set of requirements. The basic guidelines are:

- * arrange a meeting for developers & customers.
- * prepare an informal agenda that encourages free flow of ideas.
- * Appoint a facilitator to control the meeting (a facilitator may be a developer, customer or an outside expert)
- * Prepare a definition mechanism board, worksheets, wall sticks etc.
- * Participants should not criticize or debate.

FAST Session Preparations :-

Each ~~goes~~ FAST attendee is asked to make a list of objects that are:

- * parts of the environment that ~~so~~ surround's the system
- * produced by the system.
- * used by the system.

Activities of FAST Session :-

- ① Each participant present his or her list of Objects, Services, constraints & performance for discussion. List may be displayed in the meeting by using both, large sheets of paper or any other mechanisms, so that they are visible to all participants.
- ② The combined list for each topic are prepared by eliminating redundant entries and adding new ideas.
- ③ The combined list are again discussed and list are finalised by the facilitator.
- ④ Once the list have been completed, the teams is divided into smaller subteams, each works to develop mini specification.
- ⑤ Each subteam then presents mini specification to all FAST attendee. After discussion, addition or deletion are made to the list.
- ⑥ During all discussion, the teams may raise an issue that can't be resolved. An issue list is prepared so that these ideas will be considered later.

① Quality Function Deployment (QFD)

QFD is a quality management technique, that helps to incorporate voice of customer. The customer voice is then translated into technical requirements. These technical requirements are documented & result is the software requirement specification document (SRS document). In this method, customer satisfaction is very important & also understanding of what is valuable to the customer. 3 types of requirements are identified.

- ① Normal requirements
- ② Expected requirements
- ③ Exciting requirements.

① Normal requirements :-

The objectives & goals of the proposed software are discussed with the customer. If this category of requirements are present, the customer is very satisfied.

Eg:- In case of "result management system"
entry of marks, calculation of results etc...

② Expected Requirements :-

These are implicit to the software

product and may be that customer does not explicitly state that. If such requirements are not present, the customer will be dissatisfied.

Eg:- expected requirements of "gascult management system" is protection from unauthorised access, some warning systems for wrong entry of data.

③ Exciting Requirements :-

Some features go beyond the customers expectations and prove to be very satisfying when present.

Eg:- exciting requirements of "gascult management system" is, if an unauthorized access is noticed by the Software, it should immediately all the processes and an email is generated to the System administrator.

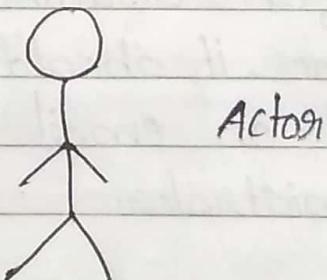
④ Use case Approach :-

This approach uses a combination of text & pictures in order to improve the understanding of requirements. The use cases describe 'what of a system'. They only give functional view of the system. The term 'use case' & use case diagram are different, use cases are structured outline or templates for the description of user requirements.

modelled in a structured language like english. Use case diagram are graphical representations. The following components are used for the design purpose.

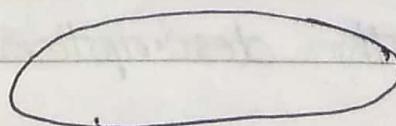
D Actor :-

An actor or external agent lies outside the system model but interacts with it. An actor may be a person, machine, or an information system that is external to the system model. An actor is represented by stick figure (*) & is not a part of the system itself.



② Use Case :-

A usecase is initiated by a user with a particular goal in mind and completes successfully when that goal is satisfied. It describes the sequence of interactions between actors and the system. The usecase representation is:

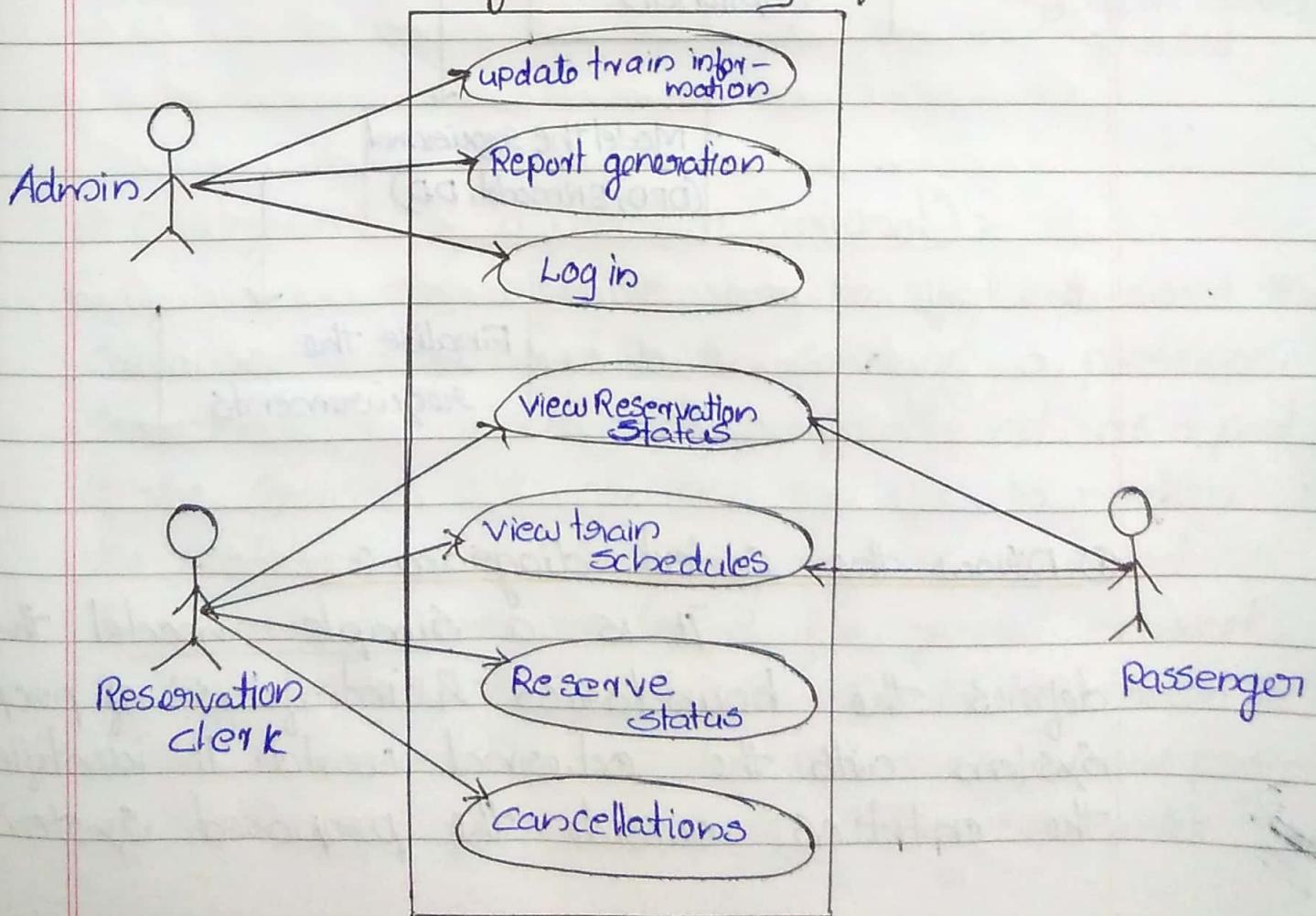


use case.

* Use case Diagrams :-

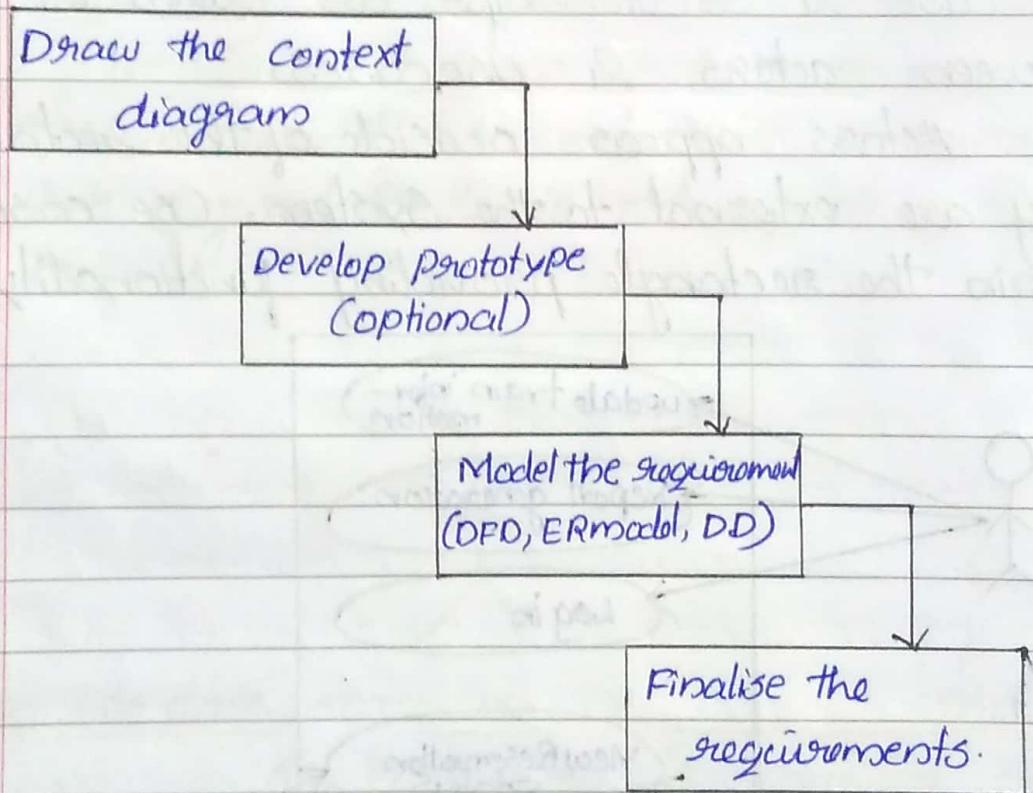
It visually represents what happens when an actor interacts with a system. The system is shown as a rectangle with the name of the system inside, the actors are shown as stick figures, the use cases are shown ovals, labelled with the name of the use case & relationships are lines or arrows (\rightarrow) between actors & usecases.

Actors appear outside of the rectangle, since they are external to the system. Use cases appear within the rectangle providing functionality.



③ Requirement Analysis :-

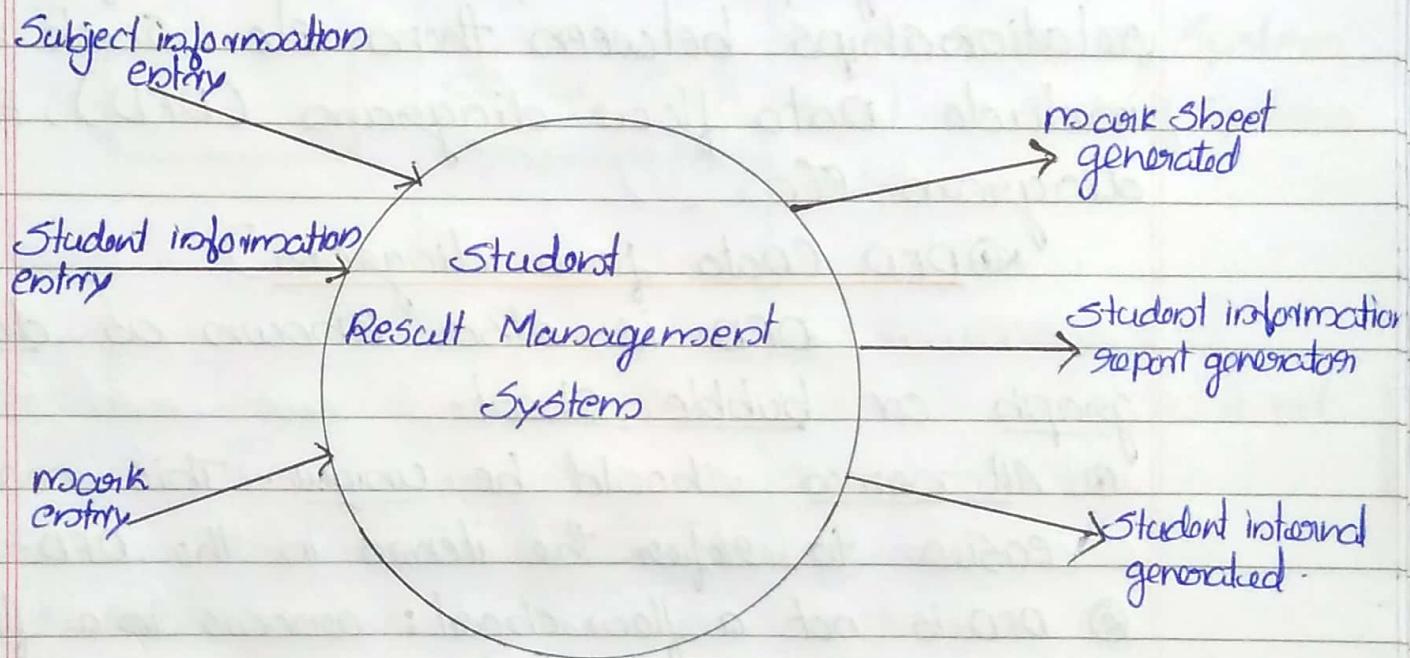
Requirement analysis is very important & essential activity after evaluation. We analyse, refine & Scrutinize the gathered requirements to make an unambiguous Requirements. The activities of requirement analysis are :



① Draw the Context diagram :-

It is a simple model that defines the boundaries & interfaces of proposed System with the external world. It identifies the entities outside the proposed system.

that interact with the system. The context diagram of "Student result management system" is given below.



② Development of a prototype (optional) :-

One effective way to find out what the customer really want is to construct a prototype. Something that look and preferably act as a part of the system. we can use that file to modify the prototype until the customer is satisfied. Some projects are developed for general market, in such cases, the prototype should be shown to some representative of the purchaser. Some projects are developed for a specific customer under

Contract. On such projects, only that customer's opinion are considered.

③ Model the requirements :-

This process usually consist of various graphical representation of the functions, relationships between them etc. Such models include Data flow diagrams (DFD), E-R diagram etc..

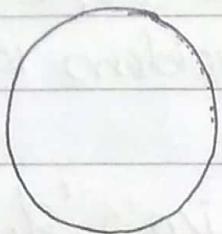
*① DFD (Data flow diagram)

DFD is also known as data flow graph or bubble chart.

- ① All names should be unique. This makes it easier to refer the items in the DFD.
 - ② DFD is not a flow chart: arrow in a flow chart represents the order of events, arrow in a DFD represent flowing data.
 - ③ In flowchart, logical decisions, drawing diamond shaped box, in DFD is not allowed.
- Standard Symbols of DFD are:-

Symbol	Name	Function
① 	Data flow	Used to connect processes to each other. Arrow head indicate direction of data flow.

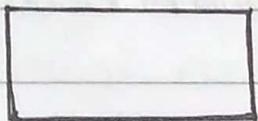
②



Process

performs some transformation of input data to yield output.

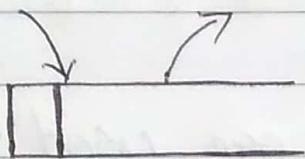
③



Source / Sink
(external entity)

A source of System input or sink of System output.

④



Data Store

The arrow head

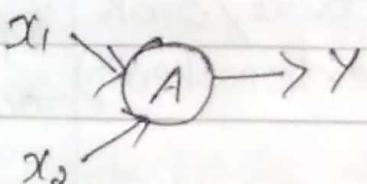
indicates net input & net output to store.

* Levelling

The DFD may be used to represent a system or software at any level of abstraction. DFD may be partitioned into levels that represent increasing information flow and functional details. A Level-0 DFD is called Context diagram (fundamental system model) or represents the entire software element as a single bubble with ip and op data. Then the system is ~~decom~~ divided into multiple bubbles. Then these bubbles are ~~decom~~ divided and documented as more and more detailed

DFD. This process may be repeated at many levels until the problem is well understood.

Eg:- If bubble 'A' has 2 ip's 'x₁' and 'x₂' and one o/p 'y'. Then the expanded DFD that represents 'A' is shown below (Level-0)



* Data Dictionary (DD)

Data dictionaries are used to store information about all data items defined in DFD. The typical information stored includes:

① Name of the data items.

② Aliases (other names for item)

③ Description

④ Related data items

⑤ Range of values.

① Name of data items :- ^{used for} identification ~~name~~ of data.

② Aliases :- include other names by which this data items are called.

Eg:- DEO represents Data Entry Operator.

③ Description :-

Textual description of what the data items is used for or, why it exist.

④ Related data items :- Shows the relationship between data items.

Ex:- Total mark must always equal to internal mark + external mark.

⑤ Range of values :- records all possible values.

Ex:- Total mark must be tve and between 0 to 100. The mathematical operators used within data dictionary are :

$x = a+b \Rightarrow x$ consist of data elements a & b.

⑥ Requirement Documentation :-

It is very important activity after requirement documentation elicitation & analysis. This is the way to represents in a consistent format. Requirement document is called software requirement specification. The SRS is a specification for a particular software product, program that perform certain functions in a specific environment. SRS is written for different purpose and serve as a contract document between customer & developer.

Nature of SRS :-

The different steps for creating SRS is :

- ① Functionality
- ② External interfaces
- ③ Performance
- ④ Attributes.
- ⑤ Design constraints

① Functionality :-

what the software product is supposed to do ..

② External interfaces :-

How does the software interact with people, systems, hardware or other software.

③ Performance :-

what is the speed, availability, response time etc of various software functions

④ Attributes :-

what are the considerations for portability, maintainability, correctness, security etc.

⑤ Design constraints imposed on an implementation

⑥ implementation language, database...

Characteristics of Good SRS

SRS Should be

- ①) correct
- ②) complete
- ③) Verifiable
- ④) Modifiable
- ⑤) Traceable

① Correct :-

The SRS is correct iff every requirement stated therein is one that the software shall meet. There is no tool or procedure that assures correctness.

② Complete :-

The SRS is complete iff it includes,

- * All requirements relating to functionality, performance, design etc..
- * Definition of their responses of the software to all i/p data in all situations.
- * Full labels & references to all figures, tables & diagrams in the SRS.

③ Verifiable :-

SRS is Verifiable iff every requirement stated therein is Verifiable.

In general, any ambiguous requirement is not verifiable.

④ Modifiable :-

The SRS is modifiable if its structure and style are such that any changes to the requirement can be made easily & completely. The requirement should not be redundant. Redundancy is not an error but it can easily lead to errors.

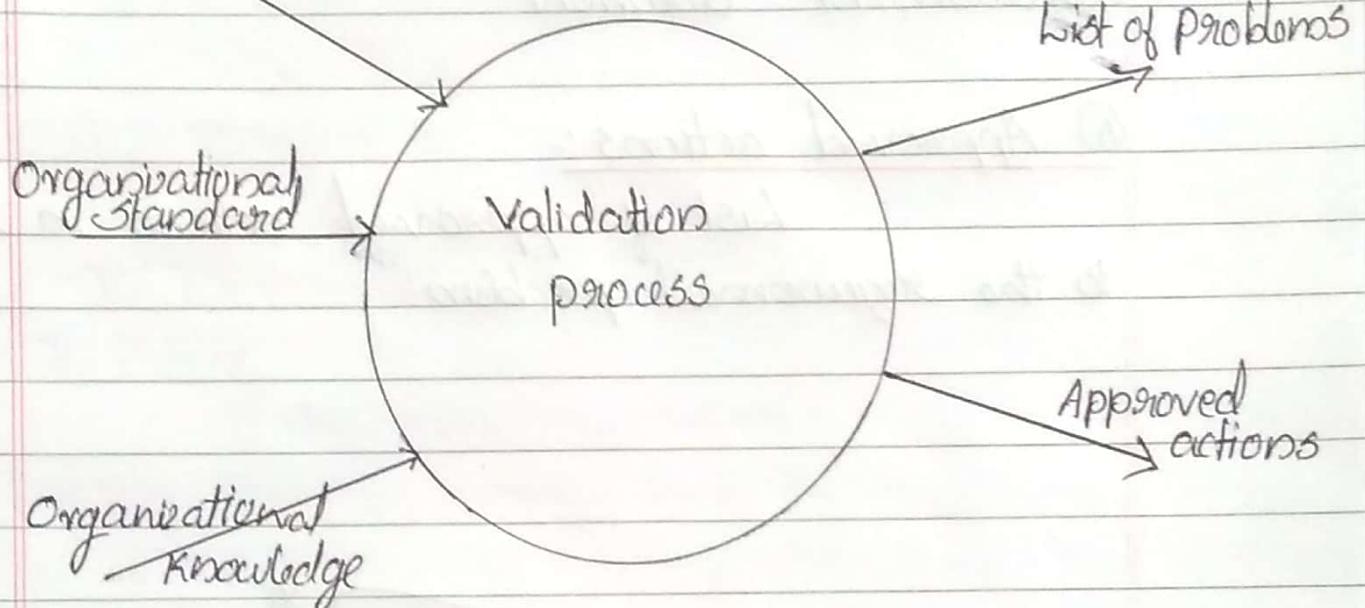
⑤ Traceable :-

The SRS is traceable if the origin of each of the requirement is clear.

⑥ Requirement Validation :-

After the completion of SRS document, we need to check the document for completeness and consistency. The objective of requirement validation is to verify that SRS document is an acceptable document of the System. The input and output of validation process are:

SRS document



① SRS document :-

It should be a final draft and not an unfinished draft and should be formatted or organized.

② organisational standards :-

Every organisation should have some quality standards for SRS document. These standards will be used for reviewing.

③ organisational knowledge :-

Knowledge of the organisation is used to judge the realisms of requirements.

④ Problems List :-

List of discovered problems in the requirement document.

⑤ Approved actions :-

List of approved actions in response to the requirement problem.

