Software Requirement analysis & Specification

Introduction : Requirements of a customer plays a key role in designing a software Product. It is very imp to define the Precise requirements of the customer ; bfr we start to design a software Product. Understanding requirements of any Project is most crucial to that Project and forms foundation of that Project.

The key component is "Requirement Specification Document (RSD) or Software Requirement Specification" (SRS). and Proces of developing this document is called "Requirement Engineering".

Requirement defines "what" of a system not "how".

So Requirement engineering Produces one large document, written in a natural language, contains a description of what the system will do without describing how it will do.

REQUIREMENT ENGINEERING DEFINITIONS :- A stk requirement is a capability that somebody needs or wants.

IEEE defines a requirement as : (a) a condition or capability needed by a user to solve a Problem or achieve an objective;

(b) a condition or capability that must be met or
system or system component to satisfy a contract, standard
or other formally imposed document
(c) a document representation of a condition or capability

NEED FOR SRS ! The origin of most s/w systems is in need of
client, who either wants to automate an existing manual system
or desire a new s/w system. The s/w system itself is created
by the developer. Finally, completed system will be used by end
user.

So three major parties interested in new system: the client,
the user and the developer.

The main problem is here that client usually doesn't under-
stand s/w or s/w development process, and developer often
doesn't understand the client's problem and application area.
This cause a communication gap b/w parties involved in develop-
ment project.

(1) So a basic purpose of SRS is to bridge this communication
gap. SRS is a medium through which client and user
needs are accurately specified.

(2) SRS is helping the clients understand their own needs.

(3) In order to satisfy the client, which is basic quality
objective of s/w development, the client has to be made aware
of these potentials and aided in visualizing and conceptualizing

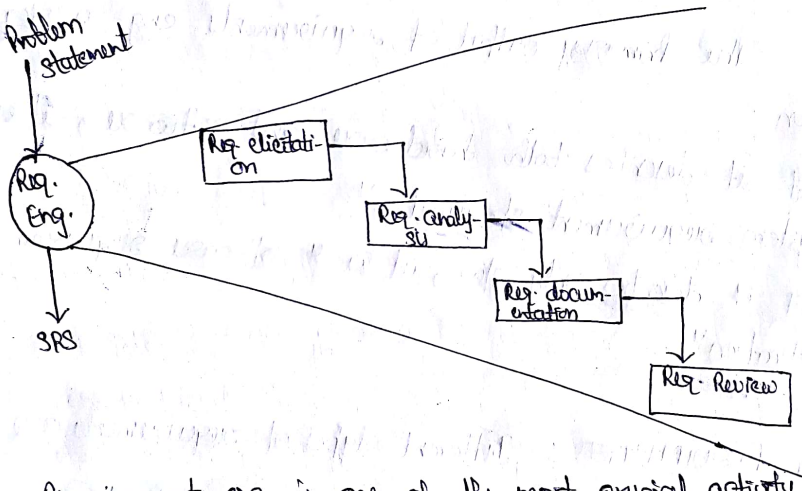
item eg: ... and requirements of his organization. (8)

- Ranges SRS establishes the basis for agreement b/w the client and the supplier on what the s/w product will do.
- (5) An SRS provides a reference for validation of final product.

or (6) A high quality SRS is a prerequisite to high quality s/w.

(7) A high quality SRS reduces the development cost.

REQUIREMENT ENGINEERING : Crucial Process Step



Requirement eng. is one of the most crucial activity in the creation process. Without well-written req. specification, developers don't know what to build, customers don't know what to expect, and there is no easy to validate that build system satisfies the requirements.

(i.) REQUIREMENT ELICITATION: Also known as gathering of requirement. Here requirements are identified with help of customer and existing system process, if available.

(2.) REQUIREMENT ANALYSIS: Analysis of requirements starts with req. elicitation. The requirements are analyzed in order to identify the inconsistencies, defects etc.

(3.) REQUIREMENT DOCUMENTATION: This is the end product of req. elicitation and analysis. It is very imp as it will be the foundation of design of s/w. Document is known as SRS.

(4.) REQUIREMENTS REVIEW: Review Process is carried out to improve the quality of SRS. Also be called as "requirement verification".

The primary output of requirements eng. is req. specification.

(a.) If it describes both hardware and software, it is a "system requirement specification".

(b.) If it describes only s/w, it is "software requirement specification".

TYPES OF REQUIREMENTS: Different types of requirements are:

(1.) Known Requirements: Something a stakeholder believes to be implemented.

(2.) Unknown Requirements: Forgotten by stakeholder bic they are not needed right now or needed by another stakeholder

(3.) Undreamt Requirements: Stakeholders may not be able to think of new requirements due to limited domain knowledge.

Stakeholder is used to refer to anyone who may have some direct or indirect influence on system requirements. Stakeholders include end-users who will interact with the system and every one else in an organization who will be affected by it.

REQUIREMENTS ELICITATION

The real requirements actually reside in users mind. Hence most important goal of requirement engineering is to find out what user really want. Users need can be identified only if we understand the expectations of users from desired S/w.

* Requirements are gathered by asking questions, writing down the answers etc.

Developers and users have different mind sets, expertise and vocabularies. Due to communication gap, there are chances of conflict that may lead to inconsistencies, misunderstanding etc.

General techniques that are used in requirement elicitation Process are as follows:

(1) Interviews
(2) Brainstorming
(3) Facilitated Application Specification Technique (FAST)
(4) Quality function deployment
(5) Use Case Approach.

... at ease. After making atmosp...
...ay be asked to

INTERVIEWS

After receiving the problem statement from Customer, the first step is to arrange a meeting with the customer. During the meeting, both the parties will try to understand each other.

Specialised developers, called "requirement engineers" interact with each other.

The objective of Conducting the interviews is to understand the customer's expectations from the software.

Interviews may be OPEN-ENDED or STRUCTURED.

OPEN-ENDED INTERVIEW: In this, there's no pre-set agenda. Content free questions may be asked to understand the problem.

for eg: for "result mgt. system", req. engineer may ask:

(a) who is the Controller of examination?
(b) who will use the S/w
(c) Is their any opposition for this project?

STRUCTURED INTERVIEW: agenda of fairly open questions is prepared. Proper questionnaire is designed for the interview. Interviews may be started with simple

questions to set people at ease. After making atmosphere (4)
comfortable and calm, specific questions may be asked to
understand the requirements.

SELECTION OF STAKEHOLDER :- It will be impossible to interview every stakeholder. Thus, representative from groups must be selected on their technical expertise, domain knowledge, accessibility.

Various groups to be considered for conducting interviews :

(1.) ENTRY LEVEL PERSONNEL : may not have sufficient domain knowledge and experience, but may be very useful for fresh ideas and different views.

(2.) Mid-level Stakeholders- have better domain knowledge and experience of the Project. They know the sensitive, complex and critical area of Project. Project leader should always be interviewed.

(3) Managers or other Stakeholders- Higher level management offices like vice-Pres, Presidents, General managers, managing directors should also be interviewed. Their expectations may provide different but rich inf. for SRS development.

(4) Users of S/w : most imp. group bc they will spend more time interacting with S/w than any one else.

Types of Questions : Questions should be simple and short.

for "result mgmt. system" we may ask :

(1.) Are there any Problem with the existing system?
(2.) How many students are enrolled Presently
(3.) Are you satisfied with current Processes and Policies.

BRAINSTORMING SESSION

It is a group technique that may be used during req. elicitation to understand the requirements. The group discussion may lead to new ideas quickly and help Promote creative thinking.

Brainstorming has become very Popular and is being used by most of companies.

It promotes "creative thinking", "generates new ideas", "Provide Platform to share view", difficulties of implementation.

This group techniques may be carried out with specialised groups like actual users, middle level managers etc.

Every idea will be documented in such a way that everyone can see it. white board, overhead transparencies or Projector system can be used to make it visible.

After a session, a detailed report will be prepared and ⑤ then facilitator will review the report. Finally, a document will be prepared which will have list of requirements and their Priority, if possible.

Every idea will be written in simple english so that it convey same meaning to every stakeholder. Incomplete ideas may be listed separately and should be discussed at length to make them complete ideas. write here.

Various modern authors have characterized the brainstorming process as;

* a part of problem solving that involves the creation of new ideas by suspending judgement.
* a technique that maximize the ability to generate new ideas.
* designed to obtain the max. no. of ideas relating to a specific area of interest.
* the creation of an optimal state of mind for generating new ideas.

Facilitated Application Specification Technique (FAST)

This approach is similar to brainstorming session and obj. is to bridge the expectation gap - a difference between what developers think they are supposed to build and what customers think they are going to get.

In order to reduce expectations gap, a team oriented approach is developed for requirement gathering and is called "FAST".

This approach focus on the creation of a joint team of customers and developers who work together to understand the expectation and propose a set of requirements.

Basic Guidelines :- (1) Arrange a meeting at a neutral site for developers and customers.

(2) Establishment of rules for Preparation and Participation.

(3) Appoint a facilitator to control the meeting. A facilitator may be a developer, a customer or an outside expert.

* Share a goal :

- Identify the Problem
- Propose element of solution
- Negotiate different approaches
- Specify a Preliminary set of solution requirements

PREPARATION FOR A FAST SESSION :

- making a list of objects that are :
  (i) Part of environment that surrounds the system
  (ii) Produced by system
  (iii) used by the system
- Making a list of services (Processes or functions) that manipulate or interact
- Making a list of constraints and performance criteria.

ACTIVITIES OF FAST SESSION :- The activities during FAST session may have foll. steps :

- Each Participant Presents his or her lists of objects, services, constraints and performance for discussion. List may be displayed in the meeting by using board, large sheet of Paper or any other mechanism.
- The combined lists for each topic are Prepared by eliminating redundant entries and adding new ideas.

translated into

- The combined lists are again discussed and consensus lists are finalized by facilitator.
- Once the consensus lists have been completed, the team is divided into smaller subteams, each works to develop mini-specifications for one or more entries of the lists.
- Each subteam then presents mini-specifications to all FAST attendees. After discussion addition or deletions are made to the list.
- During all discussions, team may raise an issue that can't be resolved during the meeting. An issue list is prepared so that these ideas will be considered later.
- Each attendee prepares a list of validation criteria for Product/System and presents the list to team. A consensus list of validation criteria is then created.
- A subteam may be asked to write the complete draft specification using all inputs from FAST meetings.

QUALITY FUNCTION DEPLOYMENT

It is a quality management technique that helps to incorporate the voice of the customer. The voice is then translated into "technical Requirements". This technical requirements are documented and result is software requirements and specification documents.

- These requirements are further translated into design requirements.
- Customer satisfaction is of prime concern and thus it emphasizes understanding of what is valuable to the customer.
- Then deploys these values throughout S/w eng. Proces.

In this, 3 types of requirements are identified:

(1.) NORMAL REQUIREMENT: (i.) The objective of Proposed S/w are discussed with customer.

(ii.) If this category of requirements is present, then Customer is satisfied.

for eg:- for "student result" requirement may include:

- entry of marks
- calculation of marks into %
- grade

(2.) EXPECTED REQUIREMENTS: (i) These are those req. that are implicit to S/w Product and may be so obvious that customer don't feel the need to state them explicitly.

(ii) In the absence of such requirements customer will be dissatisfied with S/w Product.

(iii) eqs are : Protection from unauthorized access, some warning system for wrong entry of data.

(3) EXCITING REQUIREMENTS :- Some features go beyond customer's expectations and prove to be very satisfying when present in S/w Product.

Eg:- An additional copy of important files is maintained and may be accessed by system administrator.

QFD methods has foll. steps;

(i.) Identify all the stakeholders i.e customers, users and developers. Also identify initial constraints identify by customer.

(ii.) List out requirements from customer; inputs, considering different viewpoints.

(iii.) A value indicating a degree of importance, is assigned to each requirement. Thus, customer determines the importance of each requirement on scale of 1 to 5 as given below!

5 Point : Very Important
4 Point : imp.
3 " : not imp., but nice to have
2 " : not imp
1 " : unrealistic, requires further exploration.

Requirement engineers may review the final list of req. and categorise like:

(i.) it is possible to achieve
(ii.) it should be deferred and reason thereof
(iii) it is impossible and should be dropped from consideration.

USE CASE APPROACH

Use Case:- This is a technique for capturing requirements from user's point of view.

Use Case are structured descriptions of the user requirements. This approach uses a combination of text and pictures in order to improve the understanding of requirements.

Use Case only give functional view of the system. It describes the seq. of events.

USE CASE SCENARIOS:- are unstructured descriptions of users requirements.

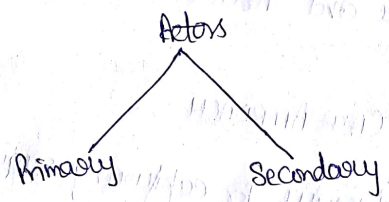
USE CASE DIAGRAMS:- are graphical representation to show the system at different levels of abstraction. Use Cases are identified and this inf. is used to draw use case diagrams.

Following Components are used for the design of use case approach-

(1) ACTORS:- An actor or external agent, lies outside the system model, but interact with in some way.

actors appears outside of

- An actor may be a person, machine or inf. system that is external to the system model.
- Devices are typically mechanisms that actor used to communicate with the system.



- A "Primary Actor" is one having a goal requiring the assistance of the system.
- A "Secondary Actor" is one from which system needs assistance.

USE CASES:- A use case is initiated by a user with a particular goal in mind, complete successfully when goal is satisfied.

- It describes the sequence of interactions b/w actors and system necessary to deliver the services that satisfies the goal.
- It may also include possible variants of this sequence.

Use cases are written in an easy to understand structured narrative - the vocabulary of the domain. The users may validate the use cases and may involve in the process of gathering and defining the requirements.

USE CASE GUIDELINES :- The foll. provides an outline of a ④ process for creating use cases:

- Identify all the different users of the system
- Create a user profile for each category of users, includes all the roles the users play that are relevant to system.
- Create a use case for each goal, foll. the use case template.
- Structure the use cases. Avoid over structuring, as it can make the use cases hard to follow.
- Review and validate with users.

USE CASE TEMPLATE

1) Brief description: Describe a quick background of the use case.

2) Actors :- List the actors that interact and participate in this use case.

3) Flow of Events :-

(a) Basic Flow : List the primary events that will occur when this use case is executed.

(b) Alternative Flows :- Any subsidiary event that can occur in use case should be separately listed. List each such event as an alternative flow.

4.) Special Requirements:- Business rules for basic and alternate flows should be listed as special requirements. Both success and failure scenarios should be described.

5.) Pre-Conditions:- Pre-Conditions that need to be satisfied for use case to perform.

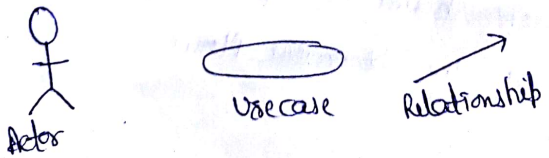
6) Post-conditions:- Defines the different states in which you expect the system to be in, after use case executes.

7.) Extension Point.

USE CASE DIAGRAM

Visually represents what happens when an actor interacts with a system. Hence, it captures the functional aspects of a system. The system is shown as a rectangle with name of the system inside.

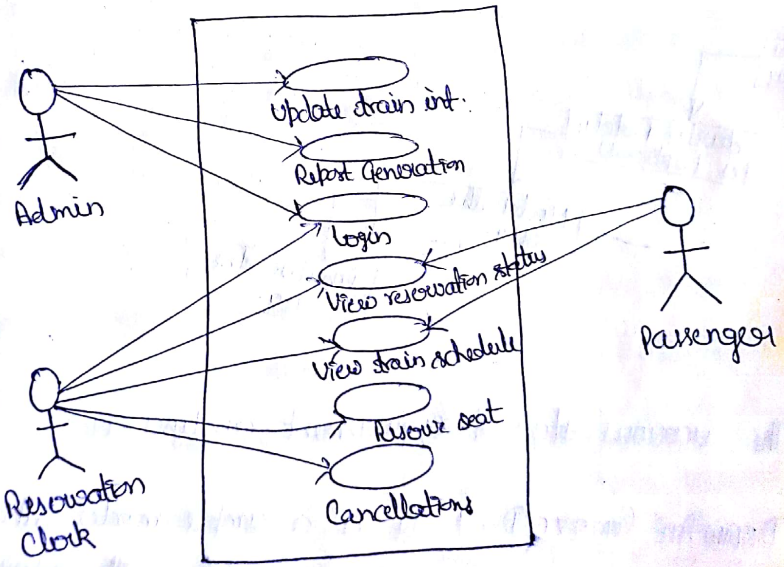
- The actors are shown as stick figures
- Use cases are shown as solid bordered ovals labeled with name of use case,
- Relationship are lines or arrows b/w actors and use cases and/or b/w use cases themselves.



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

USE CASE DIAGRAM FOR RAILWAY RESERVATION SYSTEM

Reserve seat, Cancel Reservation, Update train Int., Report Generation, login, Views reservation status, Views train schedule.

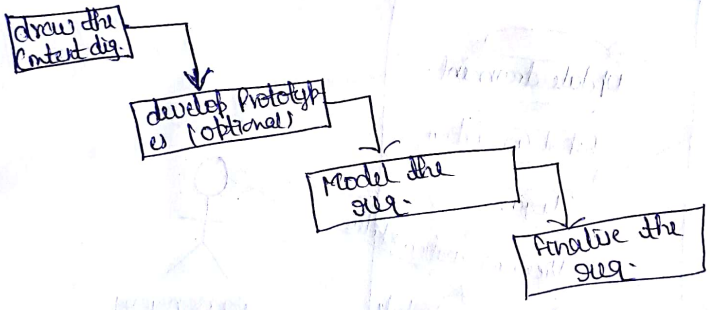


REQUIREMENT ANALYSIS

Requirement analysis is very important and essential activity after elicitation.

We analyze, refine and scrutinize the gathered req. in order to make consistent and unambiguous requirements.

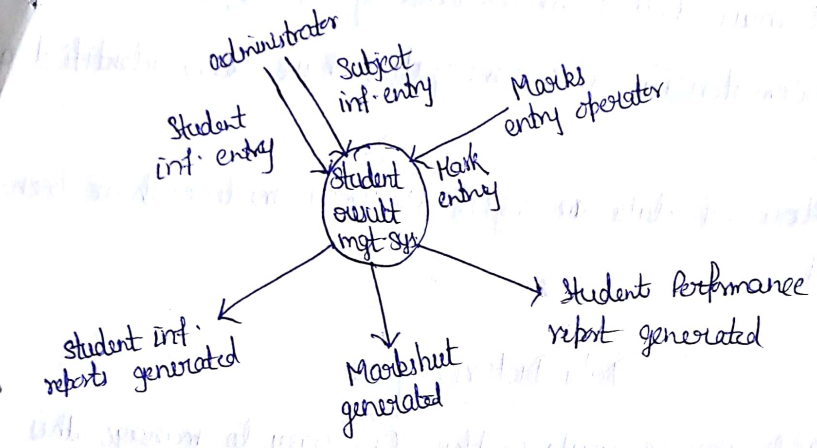
This activity reviews all requirements and may provide a graphical view of the entire system.



The various steps of requirement analysis are:

(1.) DRAW THE CONTEXT DIG.:- It is a simple model that defines boundaries and interfaces of proposed system with external world. It defines the entities outside the proposed system that interact with the system.

Context dig. of Result mgt System



(ii) **Development of Prototype (Optional)** :- One effective way to find out what the customer really wants is to Construct a Prototype.

We can use their feedback to Continuously modify the Prototype until the customer is satisfied. Hence Prototype helps the client to visualise the Proposed system and increase the understanding of requirements.

(iii) **Model the requirement** :- This Process usually Consists of various graphical representation of functions, data entities, external entities and relationship btw them.

The graphical view may help to find incorrect, inconsistent, missing, superfluous requirements.
Such models include DFD, E-R dig, data dictionary, state transition dig. etc.

(iv.) Finalise the requirements: After modeling the requirements we will have better understanding of system behaviour. The inconsistencies and ambiguities have been identified and corrected.

Flow of data amongst various modules have been analysed.

Data Dictionary

DFDs can be quite complex. One way to manage this complexity is to augment DFDs with data dictionaries (DD).

Data dictionaries are simply repositories to store info. about all data items defined in DFDs.

Typical info. store include :-

1.) Name of data item
2.) Aliases (other name for item)
3.) description / Purpose
4.) Related data items
5.) Range of values
6.) Data structure definition / form

- Name of data item is self-explanatory.
- Aliases include other names by which this data item is called. eg. DEO for Data entry operator.
- Description / Purpose is textual description of what data item is used for or why it exists.

• Related data items capture relationship b/w data item eg. total marks = internal + external marks.

• Range of values records all possible values eg. total marks must be +ve and b/w 0 and 100

• Data flow captures the name of processes that generate or receive the data item.

DD notation and mathematical operators

Notation	Meaning
$x = a+b$	x consist of data elements a and b
$x = [a\|b]$	x is either a or b
$x = a$	x consist of an optional data element a
$x = y\{a\}$	x consist of y or more occurences of data element a
$x = \{a\}z$	x consist of z or fewer occurences of data element a
$x = y\{a\}z$	x consist of some occurences of a which are b/w y and z.

Data dictionary can be used to:-

- Create an ordered listing of all data items
- Create an ordered listing of a subset of data items.
- Find a data item name from a description.
- Design the s/w and test cases.

Data Flow Diagram

also known as bubble chart, is a simple graphical notation that can be used to represent a system in terms of input data to system, various processing is carried out on these data and output data is generated by system.

DFDs shows Flow Of Data Through A System.

System may be Company or Organization.

Imp. Points:- (1.) All names should be unique. This makes it easier to refer to items in the DFD.

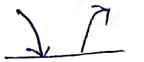
(2.) DFD is not a flowchart. Arrow in flowchart represent the order of events; arrow in DFD represent flowing of data.

Basic Elements :-

1.)  Process :- Performs some transformation of input data to yield output data.

2.)  Source or Sink :- A source of system input or sink of system output.

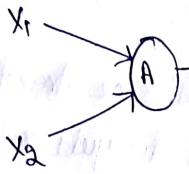
3.)  Data flow :- used to connect process to each other, to source or sinks.

4.)  Data Store :- A repository of data; the arrowheads indicate net inputs and net output to store.

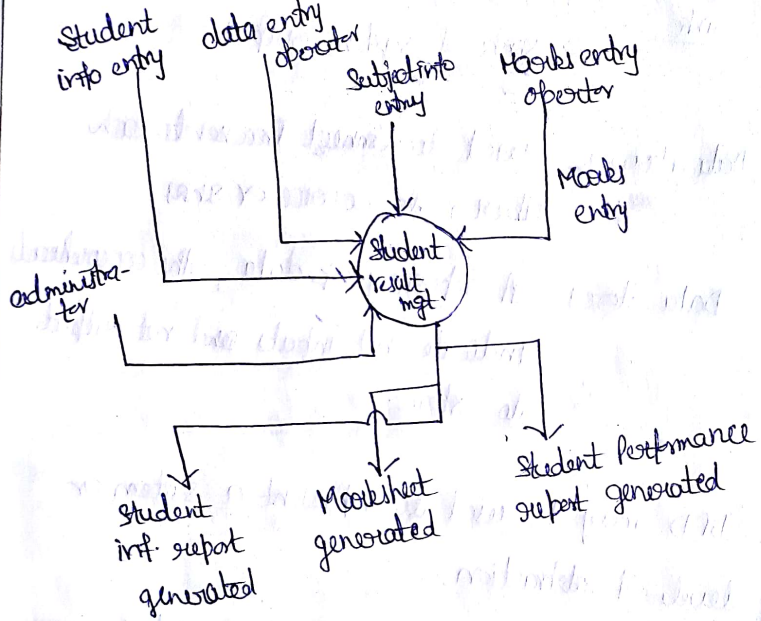
LEVELING :- DFD may be used to represent a system or s/w at any level of abstraction.

DFD may be partitioned into levels that represent increasing information flow and functional detail.

0-Level DFD :- also called a functional system model or content diagram represent the entire s/w element as a single bubble with input and output data indicated by incoming and outgoing arrows.

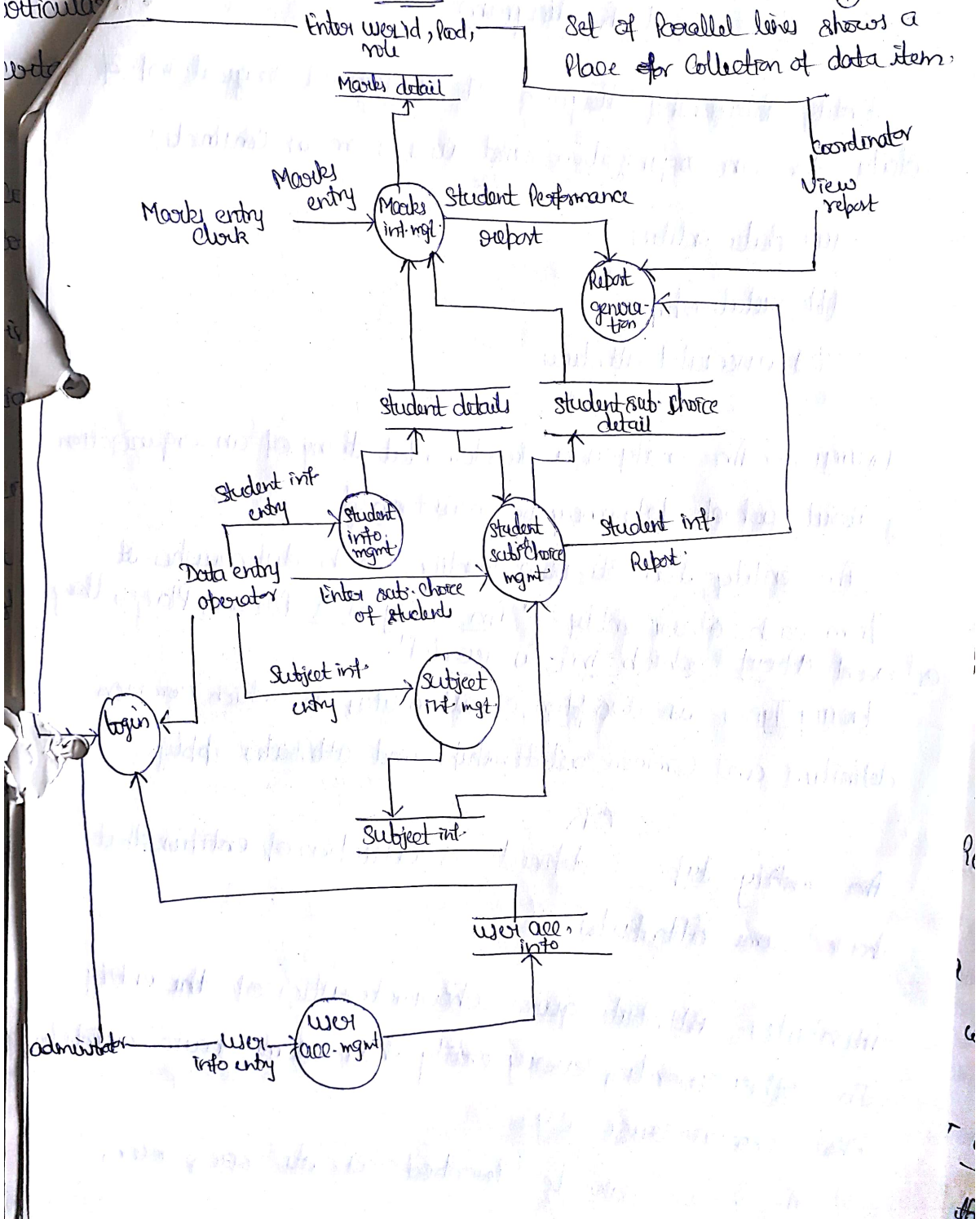


if bubble "A" has two inputs, X_1 and X_2 and one output y.



Level-1 DFD.

Set of Parallel lines shows a Place for Collection of data item.



E-R diagram

Entity-Relationship diagram. It is detailed logical rep. of data for an organization and uses 3 main constructs:

(a.) data entities
(b) relationships
(c.) associated attributes.

ENTITIES:- An entity is a fundamental thing of an organization about which data may be maintained.

An entity has its own identity, which distinguishes it from each other entity. "An entity is a Person, Place, thing or event about which info. is recorded".

ENTITY TYPE:- is description of all entities to which common definition and common relationships and attributes apply.

OR

An entity type is defined as collection of entities that have same attributes.

Attribute:- Attribute gives characteristics of the entity. In other words, every entity has some basic attributes that characterize eg.

1.) A house can be described as its size, color, surroundings etc.

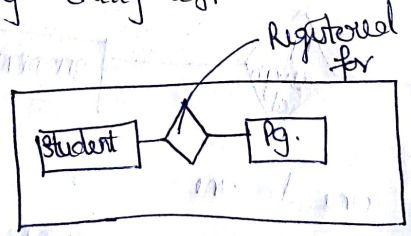
Entity type — EMPloyee

Attributes

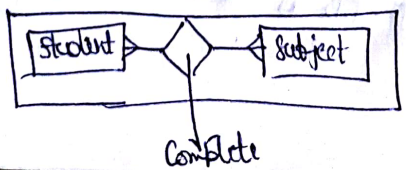
name, age, Pay
e_1
(Ram, 40, 25000)
e_2
(Rahi, 55, 8500)

Relationship :- which connect entities and represent meaning ful dependencies b/w them. The way in which two or more entity types are related is called "relation type".

Degree of Relationship :- degree of relationship is no-of participating entity types.



Relationship :

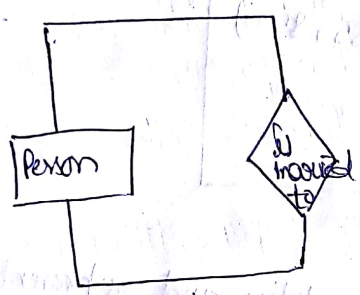


→ Complete relationship.

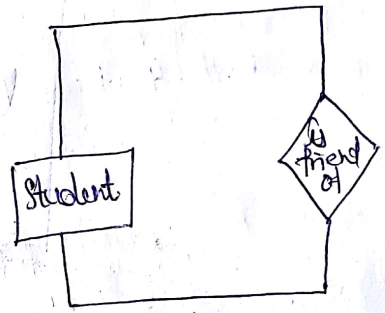
called SRS. SRS is a specification

Unary Relationship :- also called recursive relation

It is a relationship b/w the instances of one entity type.



one to one



one to many.

Binary Relationship :- It is a relationship b/w instances of two entity types and is most type relationship encountered in E-R dig.

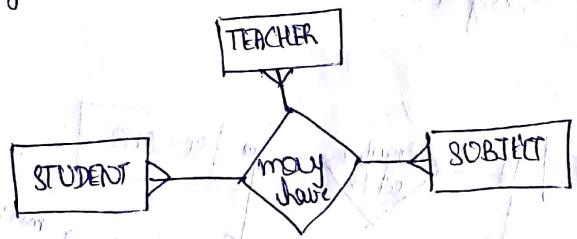
one to one

STUDENT is assigned a STUDENT-ID, and each STUDENT-ID is assigned to a STUDENT.

(one to many)

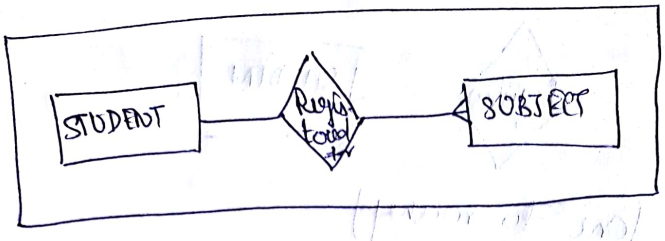
PROGRAMME may have many students and each Students belongs to only one Pg.

Ternary Relationship: Relationship among instances of 3 entity types.



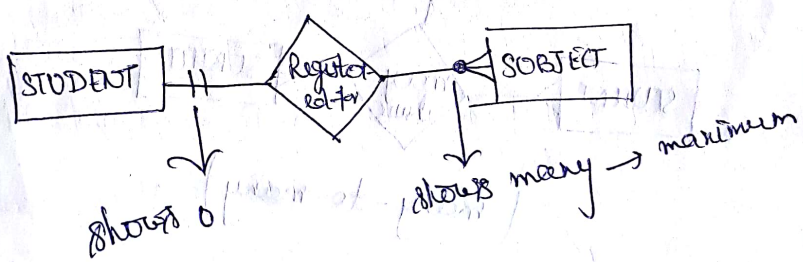
(many-to-many)

Cardinality: Suppose there are two entity type, A & B, that are connected by a relationship. The cardinality of a relationship is number of instances of entity B that can be associated with each instance of entity A.



The minimum cardinality of a relationship is minimum no. of instances of entity B that may be associated with each instance of entity A.

If minimum no. of students available for subject is zero, we say that subject is an optional participants in register for relationship.



Candidate key :- e.g. Stu-id, Reg-no

~~one act as primary key~~

Both act as candidate key.

A candidate key is an attribute (or combination of attributes) that uniquely identifies each instance of an entity type.

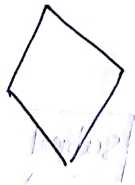
time
whether

If there is more than one candidate, the designer must choose one of the candidate keys as identifier. ③

An Identifier is a candidate key that has been selected to be used as the unique char. for an entity type.

ER dig Notations :-

entity type



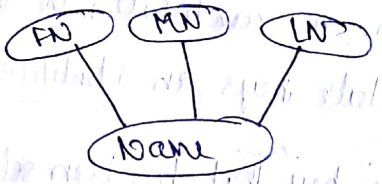
Relationship type

attribute

identifier

Multivalued attribute

Color of a car entity

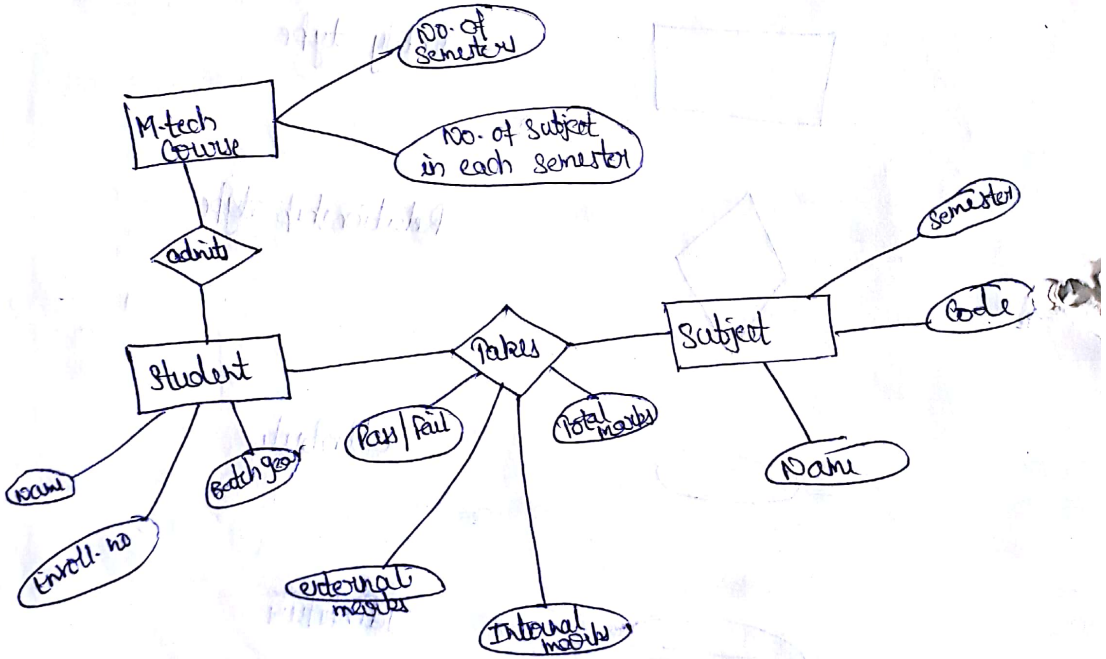


Composite attributes

3.5 1. STUDENT RESULT MGT SYSTEM.

BOOK.

E-R dig 1.



SMRITI M&M

REQUIREMENT DOCUMENTATION

Requirement document is called SRS. SRS is a specification for a particular S/W Product, Pg. or set of Pgs. that Perform certain functions in a specific environment.
It serves a no. of purposes depending on who is writing it.

- SRS could be written by customer of a system
- SRS could " " by developer " "

In first case, SRS is used to define the needs and expectations of the user.

In second case, SRS is used to define or written for different purpose and serve as a contract document b/w customer and developer.

NATURE OF SRS :- Basic issues that SRS writer shall address are following :

1) Functionality : what the S/W is supposed to do?
2) External Interfaces : How does S/W interact with people, the system's hardware, other hardware and S/W.

3) Performance : what is speed, response time, recovery time, availability and various S/w functions.

4.) Attributes :- what are considerations for correctness, maintainability, security, reliability etc.

5.) Design constraints imposed on an implementation :- Are there any required standards in effect, implementation language, Policies for resource limits etc.

ROLE OF SRS :- (1.) SRS should correctly define all S/w requirements. A S/w req. may exist b/c nature of task to be solved or special characteristic of Project.

(2.) should not describe any design or implementation details. These should be described in design stage of Project.

(3.) should not imposed additional constraints on S/w.

Characteristics of Good SRS :- The SRS should be:

(1.) Correct : • SRS is correct, if and only if ; every req. stated therein is one that S/w shall meet.

• There is no tool or Procedure that assures correctness.

2) Unambigous :- SRS is unambigous if and only if every requirement stated therein has only one interpretation. Each sentence in SRS should have unique interpretation.

3) Completeness :- SRS should be complete. An SRS document said to be complete if only if, it includes foll. details:

(1) All significant requirements, whether relating to functionality, Performance, design constraints.

(2) SRS document must include response to all valid and invalid class of data.

(3) All figure, table and Pages are numbered. They are also Properly labelled and referenced and definitions of all terms and units of measure are specified.

(4) CONSISTENT :- SRS is said to be consistent if there is no conflict btw requirements described in SRS document.

Foll. types of conflicts in requirement are:-

(1) Two or more requirements describe the same real world objects in different terms.

2) The specified characteristics of real world object might conflict.

3) There may be logical or temporal conflict b/w two specified actions.

4) Verifiable :- SRS is verifiable, if and only if, every requirements stated therein is verifiable.

An SRS document is said to be verifiable if for each requirement written in SRS, there exist some cost effective process using which a person or tool can ensure that S/w meets the requirements.

Nonverifiable requirements include statement, such as "work well", "good human interface", and "shall usually happen". These requirements can't be verifiable b/c it is impossible to define the terms "good", "well" or "usually".

5) Modifiable :- SRS is modifiable only if and only if its structure and style are such that any changes to the requirements are made easily, completely and consistently while retaining the structure and style.

The requirements should not be redundant.

6) Traceable :- The SRS is traceable if origin of each of requirements is clear and if it facilitates the referencing of each requirement in future development or enhancement documentation.

Two types of traceability are:

(1) Backward traceability :- This depends upon each requirement explicitly referencing its source in earlier documents.

(2) Forward traceability :- This depends upon each requirement in SRS having a unique name or reference number.

The forward traceability is imp when the SRS product enters the operation and maintenance phase.

ORGANIZATION OF SRS :- IEEE (Institute of electrical and electronics engineers) has published guidelines an standards to organize an SRS document.

1.) INTRODUCTION :- (1.1) PURPOSE :- Identify the purpose of SRS and its intended audience.

1.2 Scope: (i.) Identify the S/w Product to be name and a paragraph description of its function.

(ii.) Explain what the S/w Product will and will not do.

(iii.) Describe the app. of S/w being specified, including relevant goal, objective etc.

(iv.) Be consistent with similar st. in higher level specification if they exist.

1.3 Definitions, Acronyms and abbreviations: Provide the definitions of all terms, acronyms and abbreviations required to properly interpret the SRS.

1.4 References: In this:

(a.) Provide a complete list of all documents referenced elsewhere in SRS.

(b) Identify each document by title, report number, date and publishing organization.

(c) Specify the sources from which references can be made.

(d.) Use as may web sources as possible. (4)

1.5 Overview: Describe the rest of SRS and how it is organized.

2) Overall description: Describe the general factors that affect the Product and its requirements.

2.1) Product Perspective: This subsection of SRS relates the Product to other Products or Projects.

1) If Product is independent and totaly self-contained, it should be stated here.

2) If SRS defines a Product that is component of a larger system or Project:

(a.) Describe the functions of each component of larger system or Project, and identify the interfaces.

(b.) Identify the Principal external interfaces of S/w Product.

(c.) describe the computer hardware and Peripheral equipment to be used.

development time of ?, the ans.

In m-

2.1.1 :- System Interfaces :- List each system interface and identify the functionality of s/w to accomplish system requirement and interface description to match system.

2.1.2 :- Interfaces :- Specify :-

(i.) Logical char. of each interface b/w s/w Product and it users.

(ii.) All aspects of optimizing the interface with Person who must use the system.

2.1.3 :- Hardware Interface :- Specify logical char. of each interface b/w s/w Product and hardware components of system. This include configuration characteristics.

2.1.4 :- S/w Interfaces :- Specify the use of other required s/w Products and interface with other app. system.

for each required s/w Product, include

1.) Name
2.) Mnemonic
3.) Specification no.
4.) Version no.
5.) Source.

2.1.5 :- Communication Interfaces :- Specify various interfaces to communications such as local network protocols.

2.1.6 :- Memory Constraints :- Specify any applicable char. and limits on primary and secondary memory.

2.1.7 :- Operations :- Specify normal and special operations required by user such as:

i) Various modes of operations in user organization
ii) Periods of interactive operations
iii) Data processing support functions
iv) Backup and recovery operations.

2.1.8 :- Site adaptation requirements :- Define requirements for any data or initialization sequences that are specific to a given site, mission or operational mode.

2.2 :- Product functions. Provide a summary of major fns. that s/w will perform. Sometimes fn. summary can be taken direcly from system specification section that allocates particular fns. to s/w product. The fn. should be organized in a way that makes the list of fns. understandble to customer or to anyone else reading document for first time.

2-3 :- USER CHARACTERISTICS :- describe the general char. of intended users of Product including education level, experience, technical expertise.

2-4 :- Constraints :- Provide a general char. to any other items that will limit the developers option.
These can include:
1.) Regulatory Policies
2.) Hardware limitations
3.) Parallel operation
4.) Control fns.
5.) Higher-order lang. requirement

2-5 :- Further from Book.