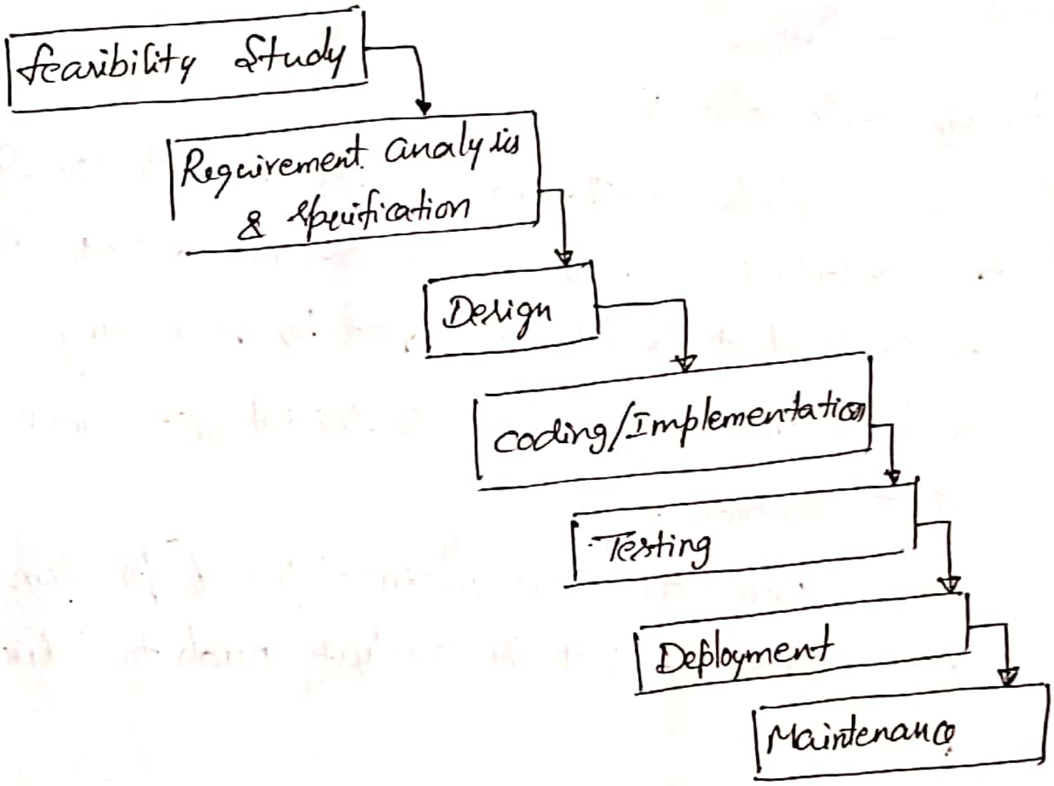
UNIT-2

Software Requirements

Software Deployment life cycle (SDLC)



Requirement Gathering

It is also known as Requirement Elicitation.

Various ways to gathers requirements :

* Studying The existing documentation

The Analyst studies all existing documents regarding the System to be developed before visiting the customer site.

* Interview : Interviews are strong medium to colect requirements. organization may conduct several types of interviews such as

- oral interview
- written interview
- one-to-one interviews are held between two
- Group interviews which are held b/w groups of participants.

* Task analysis :- A detailed task analysis can be conducted to understand The current System

* Scenario analysis : A task can have many Scenarios of operations

Requirements Analysis

After Gathering all requirements The System analyst The collected information to obtain a clear understanding of The product to be developed by removing all ambiguities and inconsistencies from The initial customer perception of The problem.

There are Three main types of problems in The requirements That the analyst needs to identify and resolve are

* Ambiguity

An anomaly is an ambiguity in the requirements when a requirement is anomalous, Several interpretations of That requirement are possible

Example: In a process Control application, a requirement by one user is That when the temperature becomes high, The heater should be Switched off.

* Inconsistency: Two requirements are said to be inconsistent if one of the requirements Contradicts The other, or two-end user of the System give inconsistent description of The requirements.

* Incompleteness :- An Incomplete Set of requirements is one in which some requirements have been overlooked

Ex: The Analyst has not recorded, When temp falls below 90

-Heater should be turned ON

- water shower turned OFF.

* The Main aim of the Requirement specification is to Systematically organize the requirements gathered during requirements analysis and document them properly in document called Software Requirement Specification

* The SRS states the functions and capabilities that a software system needs to provide, as well as the constraints That it must respect.

* The SRS provides The basis for all subsequent project planning, design, coding, testing and maintenance.

Characterstics of Good SRS Documents

* Concise : The SRS document should be concise and at the same time unambiguous, consistent, and complete.

* Structured : It should be well-structured. A well-structured document is easy to understand and modify. often the customer requirements evolve over a period of time.

* Black box view : This means That The SRS document should specify The external behaviour of The System and not discuss The implementation issues.

* Response to undesired events : It should characterize acceptable responses to undesired events. These are called system response to exceptional conditions

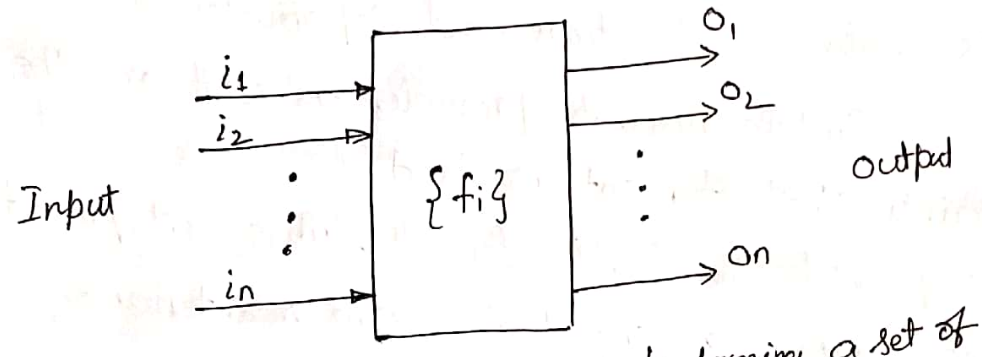
* Verifiable : All requirements of the system as documented in The SRS document should be Verifiable.

Categories of Customer Requirements

1. Functional requirements
2. Non-functional requirements
3. Goals for Implementation

Functional Requirements

A function is a set of inputs, The behaviour, and outputs ~~transforms set of input data the output of set~~ The System is considerd to perfom a set of high level functions {fi}. Each function tranforms set of input data to a set of output data.



View of a System performing a set of functions.

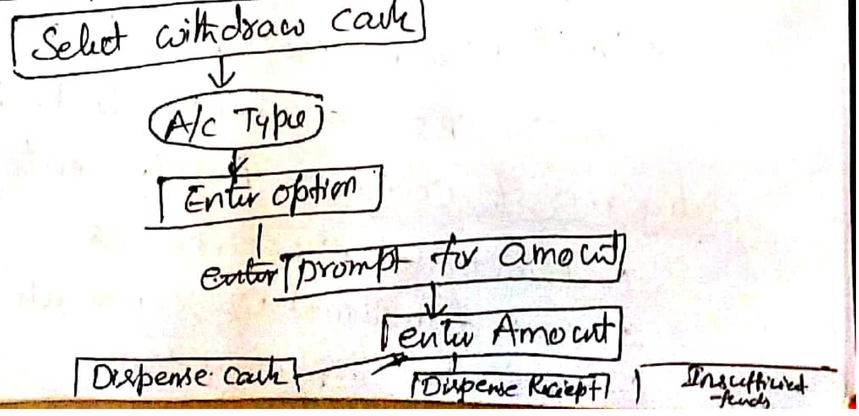
Example : Search Book in library automation Software

Input : Author's name
output : Details of The author's books and the locations of These books in The library

Functional requirements describe The features, functioning, and usage of a product / System / Software

- * A set of high level requirements
  Each high level requirements
  - * Takes in some data from The user.
  - * outputs some data to The user

Interaction b/w user and The System in the withdraw - Cash high level functional requirement



How to document functional Requirements

Documentation of The withdraw-cash function of an ATM (Automated Teller machine) System

Example :- withdraw cash from ATM

R1 :- withdraw cash

Description :- The withdraw cash function first determines The type of account That the user has and the account number from which the user wishes to withdraw cash it checks The balance to determine whether the requested amount is available in the ~~account~~ account. If Enough balance is available, it outputs the required cash, otherwise it generates an error message.

Non functional Requirements

Non-functional requirements deal with the characteristics of the System which can't be expressed as functions Such as ~~a~~ maintainability of the System etc.

Non-functional requirements may include:

* Reliability issues
* Accuracy of results
* Human-computer interface issues
* Constraints on the System implementation etc
* Maintainability, portability, usability etc
* performance - for Example; response time, Throughput, utilization, static Volumetric.
* Scalability
* Capacity
* Availability
* Recoverability
* Maintainability

User Requirements

* The user Requirements Specification (or) user requirements document usually used in Software Engineering, That specifies what the user expects The software to be able to do.

* It Describes the business needs for what user require from the system.

* user Requirements Specification are not intended to be a technical document

* The user Requirement document can be used as a guide to planning cost, timetables, milestones, testing etc.

The User Requirements Specification include

✓ Introduction

Including the scope of the System, Key objectives of the project

✓ program Requirements :

The functions and workflow that the System

✓ Data Requirements

The type of information that a System must be able to process

✓ Life cycle requirements

Include How the System will be maintain & user trained.

Example for User Requirements

⇒ Screen A accepts production Information

⇒ Screen B produces the lab Summary Report

⇒ Twenty users can use System C concurrently without noticeable

⇒ Sys' Screen D can print on-screen data to the printer.

System Requirements :

⇒ These requirements written for developers

⇒ Detailed functional and non-functional requirements

⇒ These are cleary and more rigorously specified

⇒ Requirements related to Software and Hardware.

Example

User - Requirements

The Software must provide a means of representing and accessing external files created by other tools.

System - Requirements

* The user should be provided with facilities to defines the type of external files
* Each external file type may have an associated tool.
* facilities should be provided for the icon representing etc.

Interface Specification

A user interface specification (UI specification) is a document That captures ~~of~~ the details of the Software user interface into a written document.

⇒ The Specification covers all possible actions That an end user may perform and all visual, auditory and other interaction elements.

⇒ The UI specification is the main source of implementation information for how The software should work beyond implementation

⇒ A UI specification may also be encorporated by Those within The organization responsible for marketing, graphic design, and software testing.

The Software Requirements Document

An SRS is a document that describes what the Software will do and how it will be expected to perform.

=> An SRS describes the functionality the product needs to fulfill all stakeholders (business, user) needs.

=> A typical SRS includes

- * A purpose
- * An overall description
- * Specific requirements

=> SRS is used to provide critical information to multiple teams - development, Quality assurance, operations, and maintenance.

SRS Template

1. Introduction

1.1 purpose

1.2 Intended use

1.3 Scope

1.4 Definitions and Acronyms

2. Overall Description

2.1 user needs

2.2 Assumptions

3. System Features and Requirements

3.1 Functional requirements

3.2 External Interface requirements

3.3 System Features

3.4 Nonfunctional requirements

Requirements Engineering process

The process to gather the Software requirements from client, analyze and document Them is known as requirement engineering.

The Goal of requirement engineering is to develop and maintain Sophisticated and discriptive system requirements specification document.

Requirement Engineering process → Requirement Managmt

which includes Requiremnt Development

- * Feasibility studies
- * Requirements elicitation
- * Software Requirement Specification
- * Software Requirement Validation
- * Requirements management.

✓ Feasibility Study

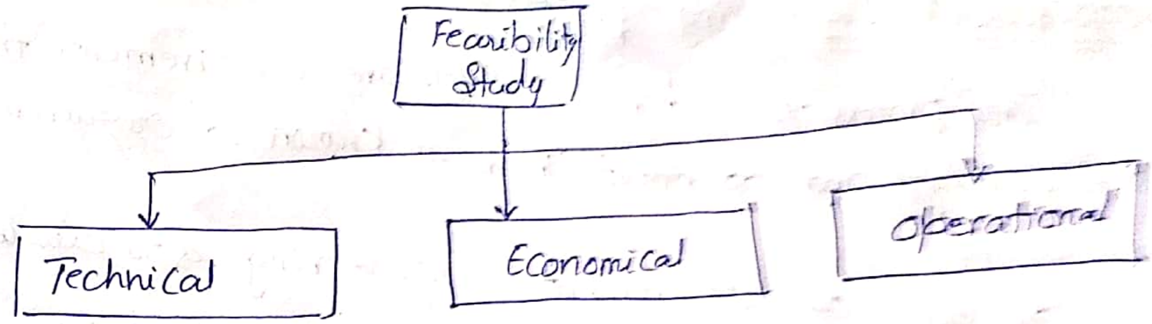
Feasibity Study is focused towards goal of organi-zation.

⇒ The Study analyzes whether the software product can be practically materialized in terms of implementation, contribution of project to organization, cost Constraints and as per values and objectives of the organization.

⇒ It Explores technical aspects of the project and product such as usability, maintainability, productivity and integration ability

Types of Feasibility

Various types of feasibility That are Commonly considered include technical feasibility, operational feasibility, and economic feasibility.



Technical Feasibility :-

-> Analyzes the technical skills and capabilities of The Software development team members

-> Determines whether the relevant technology is stable and established.

Operational Feasibility :-

-> Determines whether the problems anticipated in user requirements are high priority

-> Software development team is acceptable.

-> Analyzes whether users will adapt to a new software

Economic Feasibility :-

-> It determines whether the required software is Capable of generating financial gains for an organization

-> cost of hardware, software, development team, and training

-> cost required to conduct full software Investigation

Requirement Elicitation :-

=> It is related to The Various ways used to gain Knowledge about the project domains and requirements.

=> The Various sources of domain knowledge include customers, business manuals, The existing software of same type, standards and other stakeholders of the project.

⇒ The techniques used for requirements elicitation include interwiews, brainstorming, task analysis, prototyping etc.

Software Requirement Specification :

⇒ SRS is document created by System Analyst after The requirements are collected from various stakeholders.

⇒ SRS defines how The intended Software will interact with hardware, external interfaces, speed of operation, response time of System, portability of Software across Various platforms, maintainability, speed of recovery after crashing, security, Quality, limitations etc.

⇒ The requirements received from client are written in natural languages. It is the responsibility of System analyst to document The requirement in technical language.

SRS should come up with following features

* user Requirements are expressed in natural language.

* Technical requirements are expressed in structured language.

* Design description should be written in Pseudo code.

* Format of forms and GUI Screen prints

* Conditional and mathematical notations for DFDs etc.

Sofware Requirement Validation

After Requirement specifications are developed, The requirements mentioned in This document are validated.

⇒ User might ask for illegal, impractical solution or experts may interpret the requirements incorrectly

Requirements can be checked against following Conditions

* If They can be practically implemented

* If They are valid and as per functionality and domain of Software

* If They are Complete
* If They can be demonstrated
* The requirements should be practically achievable

✓ **Requirement Management**

Requirement management is the process of analyzing, documenting, tracking, prioritizing requirement and controlling and communication to relevant stakeholders

⇒ This stage takes care of the changing nature of requirements.

⇒ These tasks start with identification and assign a unique identifier to each of the requirement.

⇒ After finalizing the requirement tracability table is developed

⇒ The examples of tracability table are the features, sources, dependencies, subsystems and interface of the requirement.