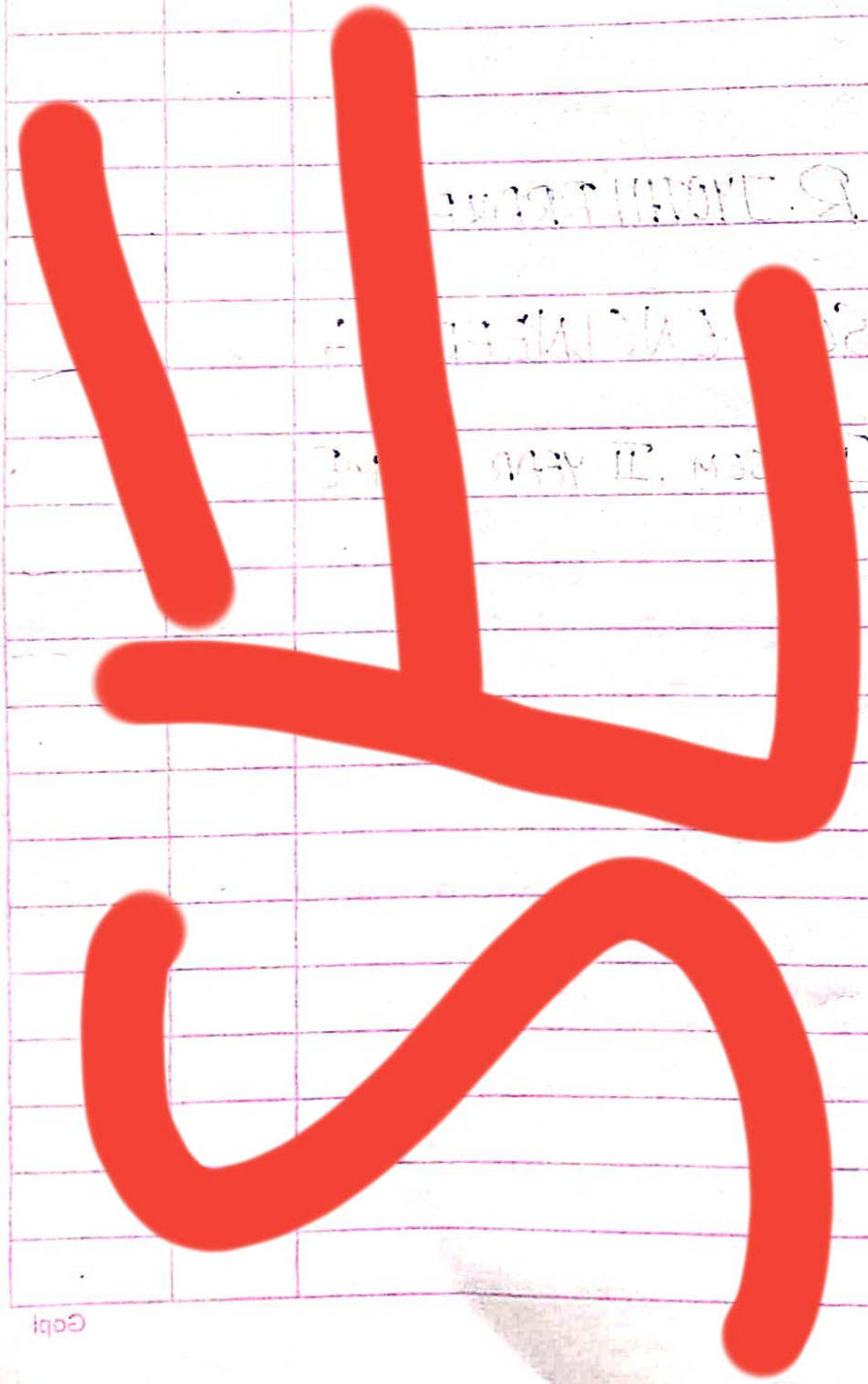
INDEX

Name: _____ STD: _____ SEC: _____ ROLL NO: _____

| S.No. | Date | Title | Page No. | Teacher's Sign / Remarks |
|---|---|---|---|---|
| | | | | |

# SOFTWARE ENGINEERING



# CHAPTER-1

## Basics Of Software Engineering And Life Cycle Models

**Software Engineering :-** It is the application of systematic, disciplined, quantifiable approach to the development of operation and Maintainance.

**Software crisis :-** A software crisis is a situation which made people thing seriously about the software development process to ensure a successful post effective system implementation which could be delivered on time to the customer.

**Software Engineering Myths :-**

There are two types of Myths.

1. Management myths.
2. Developer myths.

Management is always under pressure to maintain budgets, keeping schedule and improving the quality of a products.

**Myth :-** State of the art, tools are solutions.

**Reality :-** ease tools are used to achieve the good quality of the product.

**Myth :-** If you get behind schedule we can add more programs and catch up.

**Reality :-** Software development is not a mechanistic

process like manufacturing. Adding new people must be through up to speed to speed and communication overhead increases.

Myth:- project requirement continuously changed out changes can be easily accomdate because software is flexible.

Reality:- Requirement changes can be easily accomdate in the early stages. But when changes are requested later in software development life cycle the time and cost impact broast.

Myth:- A general statement of objectives is sufficient to begin the programs that we can fill the details later.

Reality:- Poor requirements are the major causes for software failure. A detailed description of domain information is essential. To gather the requirements there should be a good communication between customer and developer.

Developer Myths

Myth: Once we write the program and get it to work our job is over.

Reality:- Once the program is get it to work only 70% of work is over. Then remaining work will be completed. Once a project is delivered to the customer.

Myth:- Until I get the program running. I really have no way of assigning it quality.

Reality:- One of the most effective software quality assurance can be applied from the inspection of a project. One of the project is developed it is difficult to assign the quality.

Myth:- The working program is only a delievarable successful project.

Reality:- The working program is only one part of a software product. If software projects includes programs, Documentation and other data. document is more important to the user to provide greater for software maintainance.

## High Level Language Programming :-

A program consist of algorithms and flowcharts that operate on data structure.

* There are two types of data flow designs.
  * Object oriented design
  * component based design.

**Object oriented design:-** It is entirely different from structure oriented design like c, pascal., In object oriented design objects are used to send the messages to other objects. Every objects can perform set of activities, each activity perforims a specific action.

Ex:-


Component based design:-

## Component Based Design:-

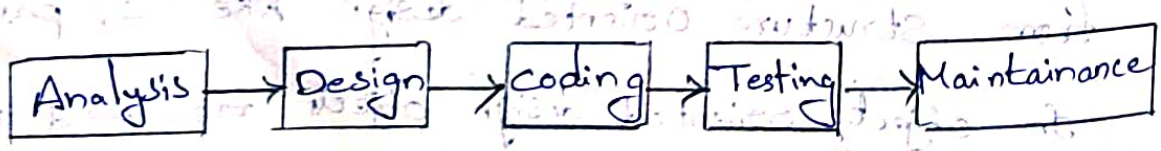
Component based programming is entirely different from object oriented design, Component based design is a data flow design developed in a structural format.

## Software Development Life Cycle Models:-

* There are different life cycle models in Software development. There are:
  * Water fall model
  * Iterative model
  * Prototype Model
  * Evalutionary software process model.
  * Spiral model.

**Water fall Model:-** It is also called as classic life cycle model (or) linear sequential model.

* It is a systematic sequential approach to develop the software.




* In waterfall model all the process are done one after another. If any problems arises at any level during the process then it returns back at any

to the previous level.

* In software development life cycle analysis is a first phase. Here we are going to gather the requirement we need.

* Design phase is the second phase of SDLC and it converts the requirements in to software representation.

* Coding is the third phase of SDLC. Here software code is developed by the developer and also unit testing is done.

* Unit testing is a processor conducting tests to uncover the errors for good results.

* The fourth stage of waterfall model is testing. Here test engineer is responsible for integration and system testing.

* The last stage of waterfall model is maintainence takes place after completion of installation put it in to use by preparing documentations user manuals.

## Advantages:-

* It is simple and easy to develop.

* Testing is performed in every stage for error free production.

* Maintainence is easy.

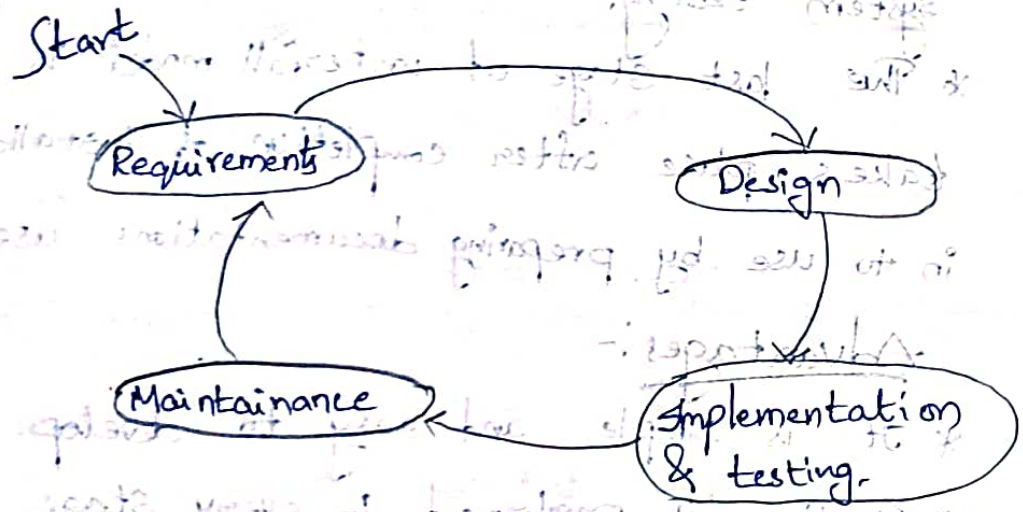
* It is very flexible to develop the proposed system.

## Disadvantages:-

* Difficult to gather the requirements.
* If there is a delay in one stage but it effects the other stages.

## * Iterative Model (or) Iterative Waterfall Model:-

* An iterative life cycle model doesn't attempt to start with a full requirements.
* Iterative model development begins by specifying and implementing a part of a software.
* Iterative model helps to produce a new version of software.
* Iterative model uses system software development.
* An iterative model consisty of following four phases

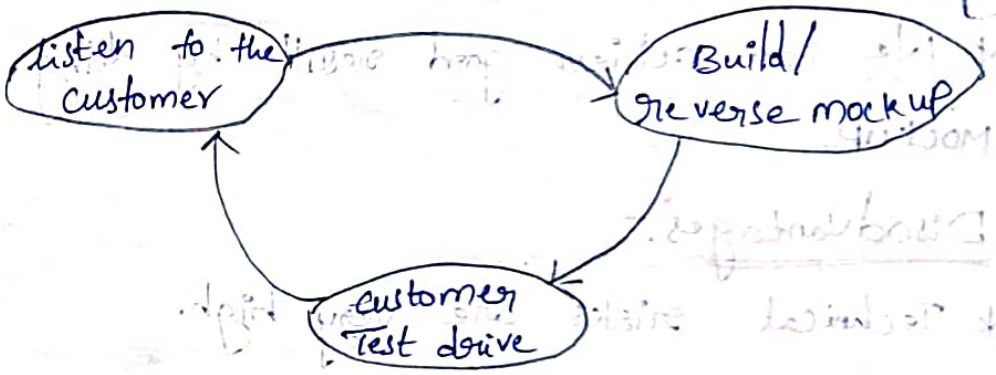



* An iterative model the requirement analysis and gather the requirements.
* The design phase means all the requirements to design (or) plan for an existing software (or) Newly created software.

* Implementation & testing for developing programming, unit testing, integration, performance and acceptance testing.
* In review phase the current requirements are reviewed, and make changes for proposed.

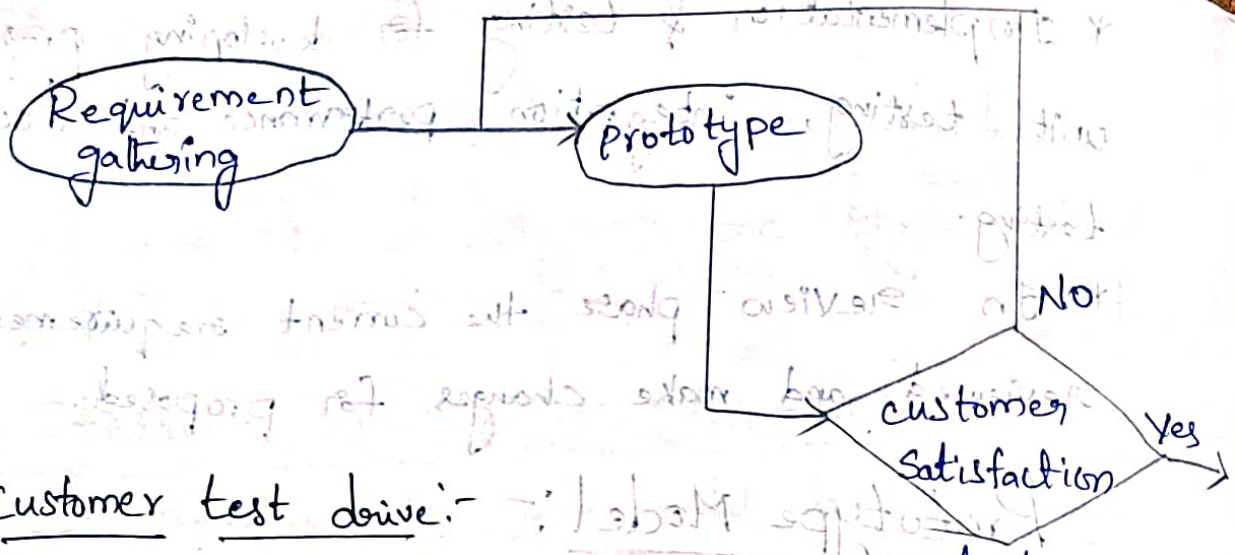
## Prototype Model :-

* It is designed to eleminates the drawbacks and disadvantages of classic life cycle models.
* It consists of three phases.
  1. Listen to the customer
  2. Build (or) reverse Mockup.
  3. Customer test drive.




Listen to the Customer :- In this phase the requirement are gathered from the customers.
* Requirment gathering is achieved through a communicatio between customer and developer.

Build/Reverse Mock UP :- In this phase the design focus on software that will be visible to the Customers.
* The below fig. shows build/Reverse Mockup.




Customer test drive:-

In this phase the customer evaluates the prototypes. If the prototype is accepted by the customer and satisfies. Otherwise the iteration occurs until it satisfied the customer.

Advantages:-

* Requirement gathering is very easy by maintaining good communication with customer.
* We can achieve good result by doing reverse mockup.

Disadvantages:-

* Technical risks are very high.
* If customer unaware/unimagine.
* Iteration of process takes places many times.

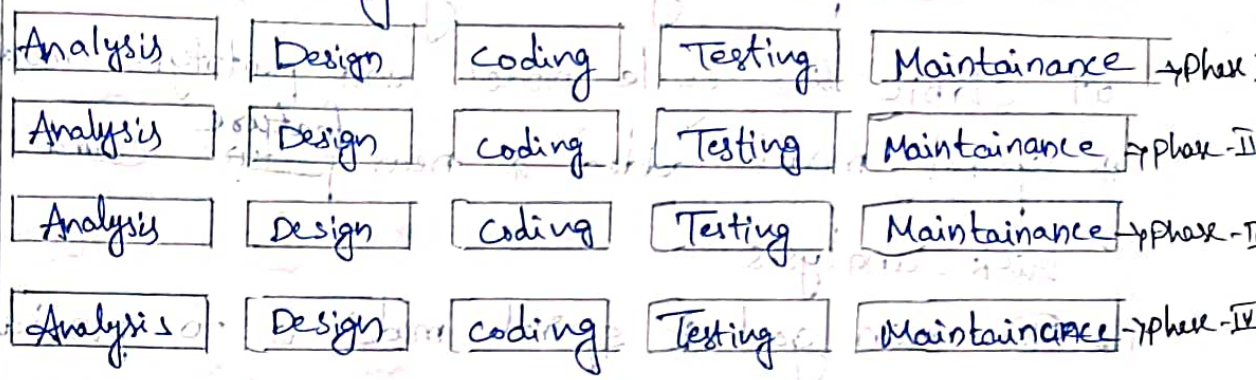
Evalutionary Model:-

It is also called as incremental model.

* The incremental model deliveres software with high quality but usuable pieces called increments.

* Incremental model is a combination of linear Sequential model and prototype model.

* The below fig. shows the sequence of increments.




* This model is suitable for object oriented Software development.

* It is suitable for large projects where requirements can be divided into modules.

* There are four crises in evalutionary model. Each phase will be released with a new added functionality.

* Phase-I delivers a core product that means basic feature features.

* Phase-II delivers with manipulation on core product

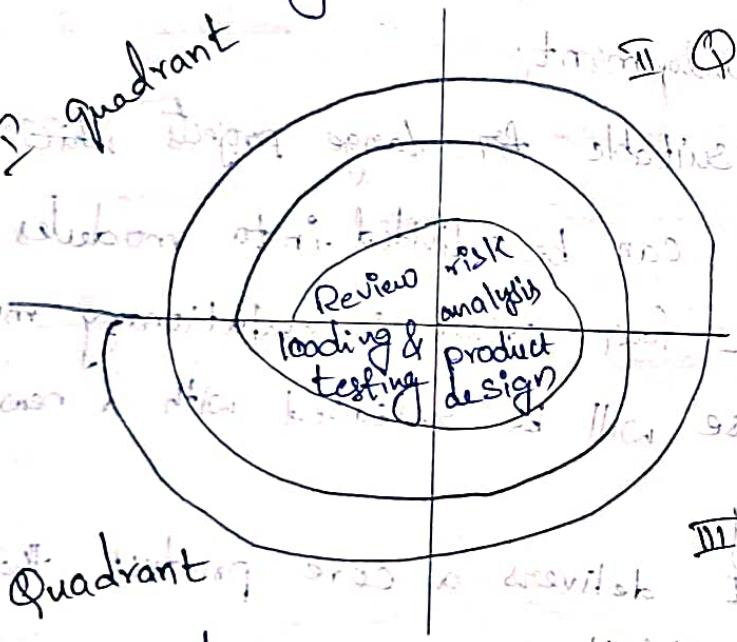
* phase-III delivers with updated features of phase-II

* Phase-IV delivers with proposed system.

## Spiral Model :-

Spiral model for software engineering has been developed by combining the best features of classic life cycle model and prototype model.

* Spiral model added a new feature called risk analysis.
* The main goal of spiral model is to provide a good framework for software development.
* The below fig shows spiral model.




* Risk analysis is an objective to identify, address and eliminate software risks. Before become to successfull software operation.
* Spiral model focus an identifying and eliminating the major risks by careful design.
* The spiral model characterstic is cyclic not like linear structure.
* Each cycle of the spiral consists of four

Quadrant, each Quadrant represented by cartesion diagram.

Advantages:-

* Spiral model can be applied through off life of software.
* In spiral model risk analysis helps to avoid the risks inception.

Disadvantages:-

* It is a more time consuming process.
* It is very expensive.

Solution for Software crisis:-

* There are various problem that exist in software industries like expenses, delay, increase in maintainance lack of staffing etc.
* Due to the above ~~produ~~ problems Software crisis may occur automatically.
* The only way to determine the solution for Software crisis is to adopt a systematic software engineering practises with more knowledgable persons.

Difference between program and Software?

| Program | Software. |
|---|---|
| 1. This are developed by individual users. | 1. Software contains large no. of programs that are developed by users. |
| 2. Program are smaller in size. | 2. Software contain large no. of phase like Analysis, coding, testing, maintainance design. |
| 3. program contains limited functionality | 3. Software contains many functions and objects. |
| 4. Easy to prepare documentation. | 4. Difficult to prepare documentation. |
| 5. Lack proper documentation | 5. Good documentation support. |

CHAPTER - 2

# Software Project Management

**Project Management:-** It is the application of knowledge, skills, tools and techniques to describe organise and control and control varias project processes.

**Types of Project Management:-**

1. Integration Management.
2. Scope Management
3. Time and cost Management.
4. Human Resource Management
5. Communication Management
6. Risk Management
7. Quality Management.

**Responsibilities of a Software project Manger:-**

* Project Manager is a person having objectives responsibilities and Authourities.
* The Main objective of every Project Manager is to deliver the project on time within a budget with required quality.
* The responsibility of a project Manager may vary from company to company and from project to project, but should always include planning

and forecasting.

* Additional responsibilities of project manager are Inter personal responsibilities leading the project.
* Informational responsibilities are monitoring the performance of staff and implementation of project plan.

## SPMP [Software Project Management plan] Document:-

The primary output of project planning is SPMP document that contains.

* Definitions of deliverable products.
* Process models, like cycle approach, methods and tools are to be used.
* List of resources.
* Total cost estimation.
* Project schedule.

## METRICS for Project size estimation:-

* for any kind of project size is a primary factor.
* There are two common ways to measure a software size.
* Loc [Lines of code]
* fp [function point]

## LOC:-

* LOC stands for lines of code and it is a most commonly used measure of software program length.
* There are two kinds of LOC.
* NLOC [Non-commented lines of code].
* CLOC [commented lines of code].
* KLOC is used to define thousands of lines of code [KILO LOC]

## Function point [FP]:-

Function point is a metric is used to measure the size of source code in terms of no. of functions in a software.

* No. of functions are calculated based on the following categories:

1. External Inputs :- Like file names and menu selections.
2. External outputs :- Such as reports and Messages.
3. External Inquiries :- Such as interactive inputs.
4. External files :- Require a response.
5. Internal files :-

## Project Estimation Techniques:-

* There are "3" project Estimation Techniques.

1. Empirical size estimation.

2. Heuristic Estimation.

3. Analytical Estimation.

## Empirical Estimation technique :-

=> It is also called as expert judgement technique.

=> This type of estimations defends on various activities like formula based and on earlier workdone on the similar type of projects and experience.

=> Experts estimate the cost of the project based on no. of functions and modules and integrate the cost of all the components to get the final estimation.

=> Project estimation is done by group of experts without considering influences and recommendations.

=> Delhi cost estimation is a type of emphrical estimation which has been developed to overcome the limitation of expets judgement techniques.

=> It is operated by a team of expert along with a co-ordinates.

=> Co-ordinates responsibilities is to furnish the software requirements specifications document expert fill their forms and hand it over to the co-ordinate.

=> The job of the co-ordinator is collecting all the forms and select the best one is the final outcome of estimation.

# HEVRISTLC ESTIMATION:

It works based on some mathematical formulas and expressions.

⇒ It can be categorized in to two classes.

1. Single Variable Model.
2. Multi Variable Model.

⇒ Single Variable Model is used for estimation the desired characteristics of a program such as effort project duration, no. of staff and project size.

Effort = no. of staff x duration

$$e = c_1 \times d_1$$

⇒ Multi Variable Models are used to act the accurate estimate of the project by using the below formula.

$$e = c_1 \times d_1 + c_2 \times d_2 + \ldots$$

## Analytical Estimation:

It derives the output by considering basics about the projects.

→ It attains the results by considering basic assumptions

⇒ It is mainly used for estimating software maintainance efforts.

# Staff Level Estimation:

⇒ Staffing is the activity of grouping the people in to team and assigning them different software engineering task.

⇒ Effort estimation is performed before staffing which gives the Measures of efforts we required to complete the project.

⇒ There are two studies proposed for staff level estimation.

1. Nordon's
2. Putum's.

⇒ Nordon's worked on various research and development projects of staffing pattern.

⇒ Nordon's determined that staffing patterns similar to ray height distribution curve.

$$ebb = \frac{area}{mt^2} \times \text{no. of engineer requires.}$$

maximum height

⇒ Putum's worked on the problem of staffing and determining that the characteristic of staff are relatively similar to research/development projects

⇒ Putum's recommended that the ray height distribution staffs to build his projects.

## Project Scheduling:-

It is a project planning activity where decisions are made recording the sequence to be followed performing the task.

⇒ During project scheduling the following activities are performed by the project manager step by step

### Step 1 :- Identify the task:

* It is the initial step of project scheduling all the tasks are required to complete the project. This can be done by the project manager. have good knowledge.

### Step 2 :- Fragmentation larger tasks.

* If a task is consuming more times and effort is fragmented into set off smaller subtasks which are performed different indivisible by different engineers

Ask task fragmentation is carried out by using work break down structure.

### Step 3:- Identification of Dependencies.

* An engineers analysis some of the activities are interrelated to each other. The Main idea of finding the dependency is to perform Sequence of generations.

Step 4 :- Estimation of time duration.

* This stage helps the manager to track the time required for completing the activity.

Step 5 :- Allocation of Resources

* It is done with respect to the activity

Step 6 :- Identification of critical Part/Path

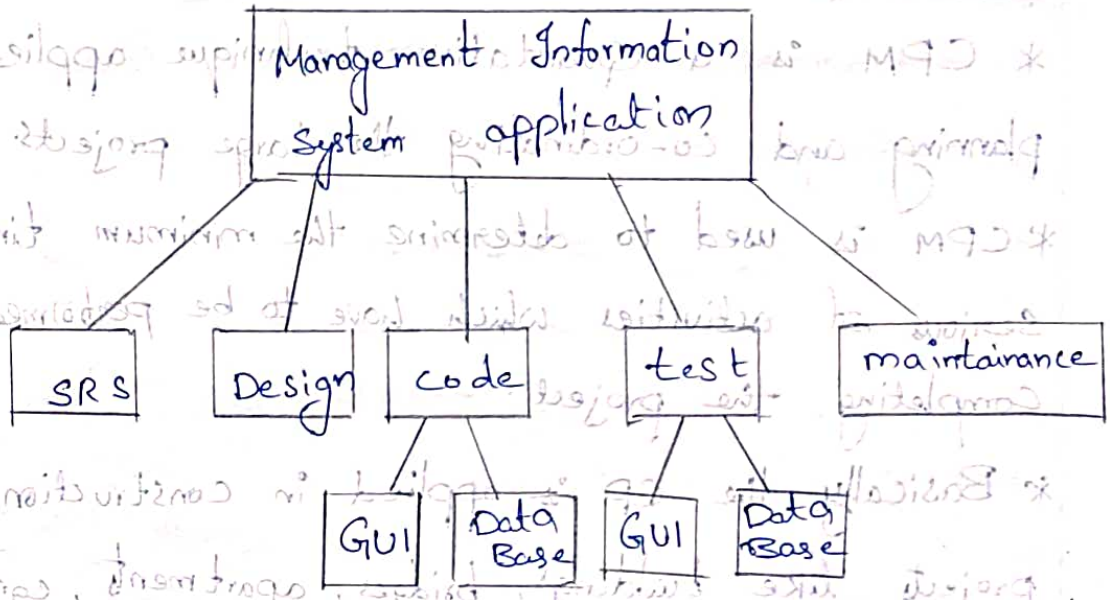
Critical path determines the duration of time and cost is identified.

Work Break Down Structure :-

It is a hireachical representation of a project plane which is divided into sub tasks.

* Hierachical represents a tree structure with Problem name / Project name as a root.
* A node is a leap of root and root is divided in to several nodes
* Nodes are decomposed into activities, sub activities till it reaches the leaf node.
* Project manager is responsible for breaking down the large task into a small ones.
* After dividing the task assigned to several engineers to start the work.

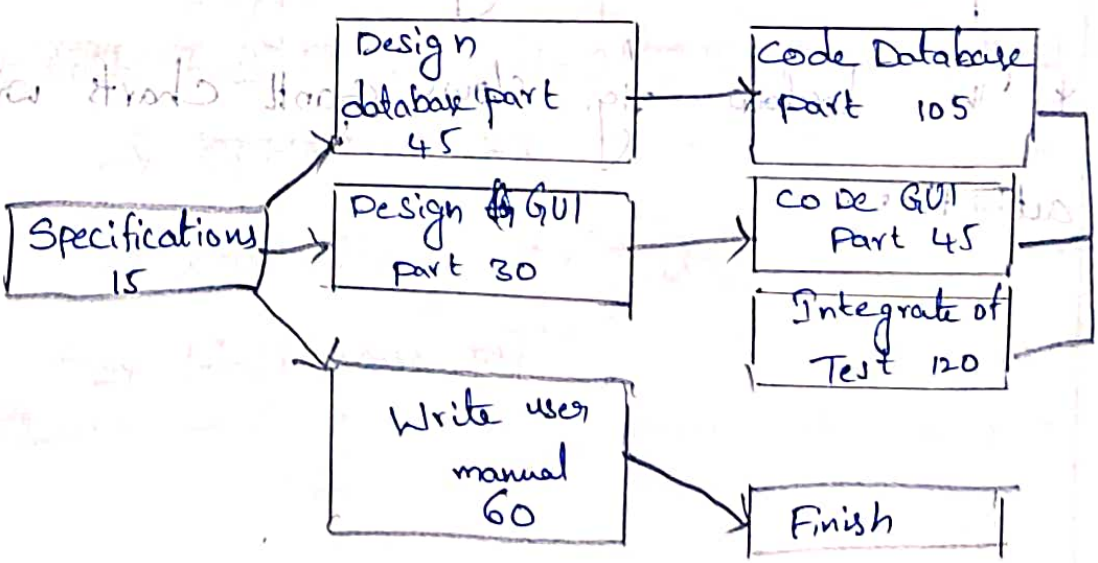
Ex:-




## Critical Path Method [CPM] And Activity:-

Network:- It describes various activities that buildup a project.

* Activity specifies the estimated time for completion of project and its inter independencies.
* The estimation time depends on the manager preferences.
* The below fig. shows the activity diagram and each activity is denoted by rectangular nodes with its estimated duration.



## CPM: [Critical Path Method]:-

* CPM is a quantative technique applied for planning and co-ordinating the large projects.
* CPM is used to determine the minimum time and serious of activities which have to be performed for completing the project.
* Basically the CP is applied in construction projects like buildings, bridges, apartments, canals etc..
* CPM give a periodic reports to the management (or) project performance.

## Gantt Chart:-

* It is also called as time line charge and it is process of creating a schedule for software project development.
* It begins with a set of activities by taking the work break down structure as inputs.
* The effort starting date and duration are the inputs for each activity.
* The below fig. shows Gantt charts with activities.

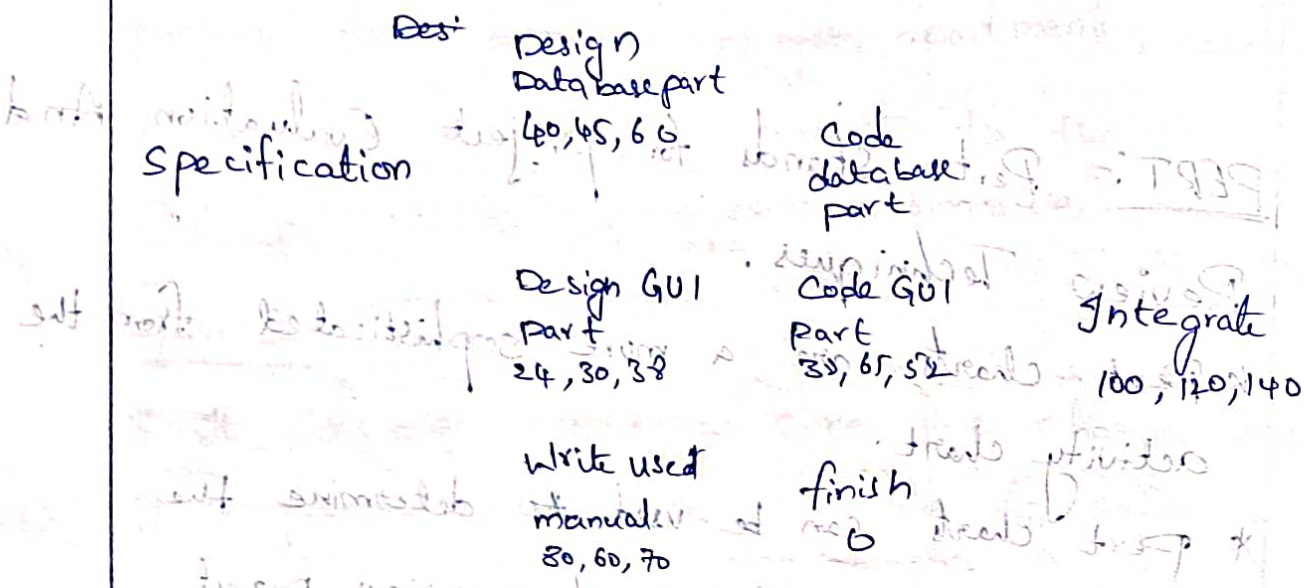
PERT:- Pert stands for project Evaluation And Review Techniques.

* Pert charts are a more sophisticated from the activity chart.
* Pert charts can be used to determine the probability time for reaching various project milestones including the final mile stone.
* Pert Chart like activity network consists of a network of boxes and as rows.
* The rectangular boxes represent activities.
* Each task in annotated with three estimates.
  - => Optimistic [O]
  - => Most likely estimate [M]
  - => Worst case [W]

Optimistic:- And work case estimate represent the extremities of all possible scnervinus of task completion.

$$te = \frac{to + 4tM + tP}{6}$$

* The below part chart representation of the [MIS] problem is shown below.




* Pert chart is used for maint monitoring the above fig.

## Risk Analysis / Management:-

Risk is debined as unfortable event which occurs during the development of project.

* The unidebined risk may result in unsuccessful completion of the project to overcome this situation necessary to identify the risk that can occur with in the project development.

* Risk management consists of three essential activities.

=> Risk identification

=> Risk accessment

=> Risk Containment.

## Risk identification:-

It is an important activity. It identifies in early stage helps the project manager to reduce the effect of risk.

* The different types of risks are

1. Project risks
2. Technical risks
3. Business risks
4. Market risks.

* If project risks becomes difficult then automatically the project development work will extend the schedule time.
* Project risks causes increases of project cost.
* Technical risks occurs due to inefficient domain knowledge.
* It causes incompleteness of project.
* Business risks arises unsuccessful Software.

## Risk Assessment:-

The main aim of Risk assessment is to rank the risises by determining the effect and damage occured on the project.

## Risk Containment:-

It must be determined and project manager should be intelligent and knowledgable in order to handle the risk efficiently / sufficiently.

* There are three important techniques for risk containment.

1. Risk avoidance
2. Risk Tranfer
3. Risk Reduction.

1. Risk Avoidance:- The risks are avoided by talking to the customer and convience him to extend the project duration otherwise giving incantives and benefits to the project engineering in order to minimize the risk of manpower.

2. Risk Transfer:- It is a technique to obtain the risk component built by third party (or) buying insurance cover.

3. Risk Reduction:- It is a technique includes the planning approaches to reduce the effect.

Ex:- When a team member / leader leaves the job then immediately recruitement of a new member should be planned project manager is responsible for handling the risks.

CHAPTER - 3

# Requirement Analysis & Specifications

Requirements gathering is a process of collecting relavant information from customer about the project to be developed.

⇒ Requirement gathering is done to get the clear understanding of the customer requirements to avoid incompleteness and inconsistence.

⇒ The system analyst is a person visits the customer size and know the problems, identify the possible solutions to the problems etc....

⇒ The following are major problems that must be eliminated by the system administrator / and list.

1. Anomoly :- refers to the ambiguiti that might occur in requirements.

2. Inconsistent :- The requirements are said to be inconsistent when thy clash each other

3. In Completeness :- If all operations to be performed by a software but those are not mentioned in SRS Document if leads to incompleteness.

# SRS Document [Software Requirement]

SRS is a finally outcome of the well defined requirements.

* SRS document is designed by the project manager in a well documented format.
* The SRS document contains functional requirements and as well as non functional requirements.
* Writing SRS document is a difficult job then development.

## Contents of SRS Document:-

The SRS document contains the following contents.

* Functional requirements / requirement
* Non-functional requirements
* Goal of implementations
* Functional Requirements:- Describes which are specified to functioning a system or project.
* Functional requirements depends up on the peoples choice.

Non functional requirements:- Dont related / which are not related to a system (or) project but which are associated with it.

* Non functionals requirements of a system are availability, reliability, Performance, security etc...,

Hence we can easily avoid the errors, unauthourised manipulations.

## Goal of Implementation:-

It is a part of SRS document and specifies general ideas and suggestion internet of development this ideas and suggestion helps in design the SRS document.

* It focus on proposed system, development of non features, readsability etc.....

## Characteristics of good SRS Document:-

A good SRS document must have the following characteristics.

* SRS document must be consistancy.
* SRS document must be well organised.
* SRS document must be complete.
* SRS document must be verifiable.
* The SRS document must be Modifiable.

## Characteristics of Bad SRS Document:-

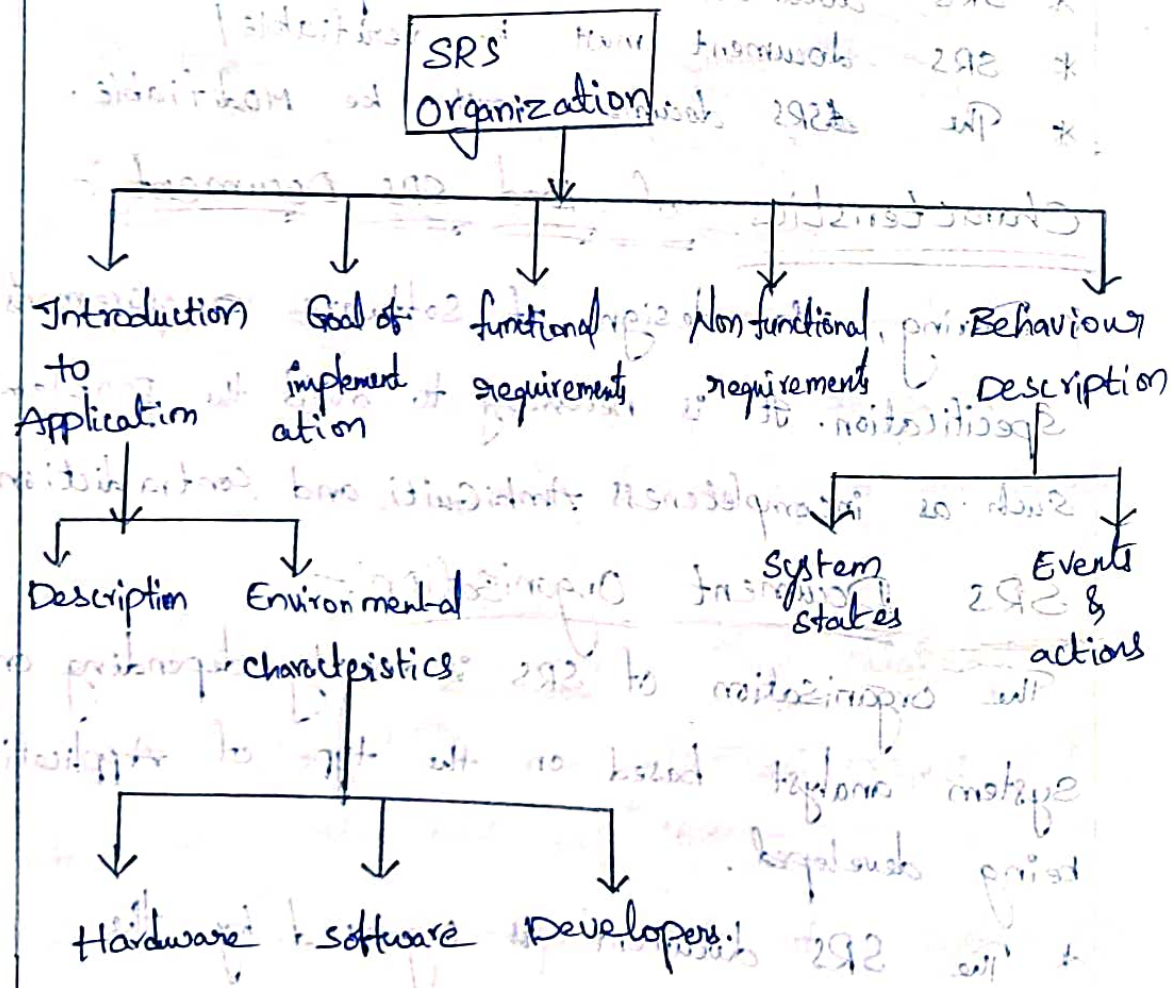
* During the design of software requirements specification. It is necessary to avoid the problems such as incompleteness Ambiguiti and contradiction.

## SRS Document Organisation:-

The organisation of SRS is mostly depending on the system analyst based on the type of Application being developed.

* The SRS document is organised by the following five sections

1. Introduction of the application
   a) Description
   b) Environmental characteristics.
      i) Hardware
      ii) Software
      iii) Developers.
2. Goal of implementation.
3. Functional Requirements
4. Non functional Requirements
5. Behavioural Description.
   a) System states.
   b) Events & Actions.



## Introduction of the Application:-

This phase mainly deals with the overall descriptions of the / a system with its Environmental characteristics that specifies hardware and software requirements and knowledgable developers.

Refer to previous page :- Goal of implementation, Functional requirements, Non functional requirements.

## Behaviour Description:-

It describes the behaviour of a system that depends upon the system that depends which varies according to the set of events and actions.

* The finite state machine specifies the behaviour of possible states dividing the occurance of the event.

CHAPTER-4

# Software Design, Coding & Testing:

**Software Design:-** A good software Design is a Graphical User Interface, includes latest and tested technology.

* A good software design should have a scalable structure which support easy association with other systems.
* A good software design should be flexible to allow changes as per our need.

## Charcteristics of a Good Software Design:

1. correctness
2. Efficiency
3. Understandable / understandability.
4. completeness
5. Portability.
6. Simplicity
7. Maintainability
8. Reuseability.

## Cohesion & coupling:-

* A cohesion is defined as mesure of functional strength with respect to a module form/model.
* There are seven classes of cohesion.

1. **Functional cohensive:-** Two elements are said to be functionally cohensive. if that performs same type of functions.

2. Sequential cohensive:- The elements in a modules are sequential cohensive when the output of one module with act as the input of another module.

3. Procedural cohensive:- The elements belongs to the same procedural unit are said to be procedural cohensive. It occurs due to the divisions of module into sub modules.

4. Temporal Cohensive:- If two elements are execute at a time, then the elements are said to be temporarily cohensive.

5. Logical Cohensive:- The elements that performs the same logical functions are said to be logical cohensive.

6. Coincident cohensive:- When decomposing of a program into several modules, it is possible that the elements one module part in to another module that are not related to each other. Then that elements are said to be coincident cohensive.

7. Communicational cohensive:- If two elements are said to communicational cohensive when they need same Input/output data.

## Coupling:-

Coupling is defined as the degree of relationship between different modules.

* There are different types of coupling.

1. Control coupling:- It is said that the data of one module is utilize and executed by other modules.

2. Stamp Coupling:- It occurs when two modules pass the data through parameters.

3. Common Coupling:- It occurs several modules occurs access the global data.

4. Content Coping:- It occurs when one module uses the code of another module. It is also called as code Reusability.

5. Data Coupling:- It occurs when several modules are communicate with each other by using some elementary data items.

User interface Design:-

It is defined as a communication medium between a user and a computer.

* User interface is a process where user interacts with a computer and helps to create internal system.
* The main objective of interface is to identify the user and analyse the task in its requirements.

Characteristics of good user Interface:-

* Speed of use:- Can be competeted by time and effort required by the user to initialize and implement different types of commands.
* ERROR PRVENTION:- A good user interface must avoid errors at the time of initating commands.
* Speed of Recall:- It is a feature for infragment users.

* Consistance / consistency:

Object oriented design is a mature methodology that improves software development as for the SRS document.

* Object oriented design emphasizes on how software models are connected to each other.
* Object oriented design focus on object.
* The object consists of parents, chains and leaf objects.
* Object oriented design emphasizes on what the system consists of.
* Object oriented design follows incremental model of SDLC.
* There is very less risk in object oriented design. because that is more code realability / reusability.
* OOD suitable for projects with changing user requirements.
* Object oriented design is a bottom approaches contain entities known as objects.
* A class is a group of objects with attributes.
* Object communicate with other objects by message passing.

Ex:- Student Admission process.

* It involves many objects like college, department, course, examinations etc.
* The objects are binded with relationship and that exist between them.

* An object oriented design, encapsulation refers to binding the data and information that contained in an object.

## Function Oriented Design:-

It is a methodology which includes software development by defining all the functions in SRS Document

* In function oriented design the system consist of many smaller subsystems called functions.
* Functions are capable of performing significant tasks.
* It is a top down approach it inherit some properties of structured design. Where divide and conquire methodology is used.
* In this design when a program calls a function then automatically.

## Differences between ODD and FOD.

| FOD | ODD |
|---|---|
| * In FOD, the system is centralized among different functions. | * In ODD, the system is decentralized among object. |
| * In FOD, functions performs operations. | * In ODD, objects performs operations. |
| * In FOD, the data is not secure | * In ODD, the data is highly secured. |
| * Push and pop, push(), pop() two operations (or) functions to get the data and to set the data. | * In ODD, stack operation is used. |
| * In FOD, the function is divided in to sub function. | * In ODD, the objects (or) entities are designed based on their relationship |
| * FOD is a top down approach | * OOD is bottom up approach. |

Differences between Graphical user interface [GUI] and Text Based Interface [TBI].

| GUI | TBI |
|---|---|
| * In GUI, the user can access different information simultaneously display in different windows on user screen. | * User can access a single window on the user screen at a time. |
| * GUI support symbolic information. | * Text based interface doesn't support for symbolic information and requires commands. |
| * GUI support pointing devices like mouse, light pen etc. | * TBI have text command for performing each operation therefore users have to memoraize the commands. |

Coding :- This is the third phase of SDLC.

* The coding is developed by the developers by considering all the requirements and specifications related to a particular module.
* A module specification consists of data structures and Algorithms.
* The significances of coding is to translate the design of the system in to a high level programming language.
* The developers computers the coding and performs unit testing.
* To develop a good software the organisation follows their own completely and rely account upon them.

* The coding standards / Guidelines are software that are prepared based on the requirements of the software that is to be developed.

## Coding Standard :-

Coding standard provides rules that are to be followed during the development of the code.

1. Restrict global data types.
2. Follow the naming conventions.
3. Specify proper format for the header.
4. Adopt a standard exception handling policy.

## Coding Guidelines :-

Coding guidelines provides ideas that are to be followed during the implementation of code.

1. Develop a single code :- ⇒ Simple code must be easily understandable and easy to maintain.
2. Avoid uncertain side effects :- Side effects creates difficulty in understandable and it increases the size of a code and also it changes program parameters and I/O operations.
3. Provid proper Documentation :-
4. Don't use same identifiers Multiple times :-

It creates confusion and makes the program difficult to understand.

5. Create Small function :- A function must be "small" should not exceed "10" Lines.

* Component Based Development :-

* A component based GUI [Graphical user Interface] is a development style based on widgets.

* Widgets are used as building blocks, because the user can easily operated.

* Widgets are used to create user interface for various applications.

Types of Widgets :-

1. Label :- Label widgets are simple. This are used to display the label.

2. Container Widget :- It contains no. of child widgets. If any variations made in a container automatically it effect child widgets.

3. Pop-up menu :- It occurs when a mouse is pressed.

4. Pulldown menu :- It occurs when a mouse is pressed.

5. Dialog box :- It is useful to idea the given information and select the reflected option. Some Dialog boxes are simply allow the user;

* A Dialog box includes various commands like ok, cancel, next and so, on.

Push button:- It consists of keywords or images.
* It indicates that next dialog box will appear.

Radio button:- It is used while selecting one option.
It is representation by allow circle.

Combo box:- It displays the list of items to be chosen when it is ~~checked.~~ clicked.

Testing:-
* Testing is a process used to estimate the productivity and quality of a software.
* Testing is done by providing the necessary detail of the software such as efficiency, portability, usability, capability etc.
* Testing refers to the ~~pro~~ process of determining the errors and debugging ~~de~~effects to generate error-free software.
* Testing process involves three phases.
  1. Test planning
  2. Testcase Design.
  3. Test case Execution.
* For any kind of software before going to test the developer has to plan for it.
* To develop test plan several inputs are required such as
  1. project plan
  2. Requirements Document
  3. Design Document.

* Test case design is done before the actual testing commences due the following reasons.

⇒ The effectiveness of testing depends on the nature of the process.

⇒ Test case are evaluated throw testcase Reviews.

⇒ Test case execution is a last step of executing process.

⇒ Testcase execution require testplan and testcase specification.

## Verification & Validation:-

* Verification is the process of ensuring the given software satisfies all the credentials which were included prior to its development.

* Validation is a process of ensuring that the given software can satisfies the customer requirements.

## Unit Testing:-

* Unit testing is a process of executing a single unit of module without effecting the other modules present in the software.

* Unit testing concentrates on data structure and internal processing logic of a module.

* Unit testing is done by the developers as per requirements.

* Unit testing can be done simultaneously for all the units (or) modules which are independent to each other.

* Unit testing is carried out atleast once during software development and it may be repeated depending on any changes made on software.

* There are five tests are carried out in unit testing.

1. * Interface :- Testing the module information confirms that the information coming into and going out of the module is correct. This is also called as a data flow testing.

2. * Data Structure :- Local data to the module are tested inorder to confirm their integrity during the program execution and also the effect of global data on any other local data must be tested.

3 * Boundary Conditions :- For each module some boundaries are set such that the module operates within this limits.

4. Independent Paths :- It is presented in the control structure (or) tested in order to make sure that no. of even a single function (or) statement is unexecuted.

5. Error handling :- It is tested after completion of all the above tests. This is also called as "anti-bugging".

## White Box testing (or) Clean room testing:-

Internal logic and structure of a program.

* It is also called as glass box testing, Structured testing (or) open box testing.
* In white box testing, test cases are derived from test suit.
* It is performed depending on the knowledge of flow the system is implemented.
* It concentrates on dataflow, control flow, exception and error handling techniques.
* White box testing also checks the implementation is accordance with the planned designed (or) not.
* It is done to check the implementation of secured functions and to explore the risks.

### Advantages:-

* The application is effective because of the internal knowledge of a program.
* The code is optimized.
* Hidden errors are removed.

### Disadvantages:-

* It is very expensive to perform because the tester should have a knowledge about the internal structure.

Black Box Testing:- It concentrates on all the functional requirements of a program through a set of input conditions.

* It is also called as a behaviour testing.
* It finds the errors that arises due to important (or) missing functions / functions.
* Due to interfaces.
* It find the errors in external data base.
* It finds the errors due to invalid initialization (or) Termination.

Advantages:-

* It is more effective than white box testing.
* It is performed in the later stages of a program

Disadvantages:-

* It takes long time to test each and every input as well as internal database.
* It is difficult to design the test cases without knowledge.

Debugging:- It is a process of fixing the errors that were identified during the process of testing.

* The following are some of the approaches adopted by the developer for debugging.

1. Back tracking Method:- It identifies from which point an error symptom is observed till the discover of errors.

2. Brute force Method :- In this method the program is loaded with various print statements to print the intermediate values.

* The result of print statement are used to identify the no. of errors (or) Buggs in a program.

3. Cause elimination method :-

In this method a list of causes are responsible for error generation. Then, tests are conducted against to each cause to eliminate it.

4. Program slicing method :- In this method to reduce the length of a search in a program is minimized by defining various programs slices. A program is a subset of a program containing lines of code.

## Guidelines for effective Debugging :-

* As program become large finding buggs are very difficult and debugging is a time consuming process.
* Understanding the program design and structured top down approach to writing a programs well.
* Document your program using Comments.
* Trace your program Normally.

* Fixing of one error may lead to generation of other errors. So, Hence it is necessary to perform.

Regrestation testing:- each time an error is fixed.

Program Analysis tools:-

* It produces a detailed report of a given program
* It compares program characteristics like program size, complexity
* These tools are classified into two types.
    1. Static Analysis tools
    2. Dynamic Analysis tools.

1. Static Analysis tool:-

* It analysis the structure of a program and produces a report on it.
* Static Analysis tool doesn't even execute the program for computation of software characteristics.
* It checks whether all coding standards are followed (or) not
* It checks for the errors like uninitialized variable, parameters and used variables.

2. Dynamic Analysis tool:-

* It executes the program and determines the behaviour of a program
* It produces a report in the form of charts, like bar charts, pie charts etc.....
* It eliminate the Rendunation of code

Renduance.

## Integration Testing :-

* It assembles all individual modules of a System. Then it performs a test on the module interfaces.
* ~~The follow~~
* It is performed after a successful integration of all the modules.
* ~~The following after a successful~~
* The following are/ of some methods of developing integration testing.

1. BigBang Method :- It integrates all the modules of the system together and performs a test on it.
* It is simple method and gives good results for small systems.

2. Top down Methods :-

* It performs testing from beginning and combines the middle level and low level modules one after another. This process continuous till the entire system has to be tested.

3. Bottom Up method :-

* This method intialize testing from beginning low level modules and combines the intermediate systems then perform test on high level modules.

4. Mixed Testing:- It is a combination of top down and bottom up method.

* To down method starts its testing after coding and unit testing all/of top level modules.
* Bottom up method starts its testing after coding of bottom level modules.

Performance Testing:-

* It checks whether all the non-functional requirements documented in the SRS have been satisfied or not.
* There are different types of performance testing. That depends upon the non-functional requirement of a system.
* Various types of performancing testing functions.

Performance testing:-

* It check whether all the non-functional requirements documented in the SRS have been satisfied or not.
* There are different types of performance testing. That depends upon the non-functional requirements of a system.
* Various types of performancing testing follows.

1. Stress Testing:-

It computers the performance of a system by enforcing unusual and illegal input conditions into the system. It also checks the capability of a

Software like input data rate, input data volume, memory utilization, processing time or tested.

2. Configuration Testing:-

This type of testing analysis the behaviour of the system under various software and hardware requirements.

3. Volume testing:-

It verifies whether the data structure have been correctly defined / declared.

4. Documentation Testing:-

It checks the consistency of a documentation like user manuals, technical manuals and maintainance manuals.

5 Comparability testing:-

It verifies system interface compared / comparable to other system.

6. Recovery Testing:-

It determines how the system responds in faculty conditions like power loss, loss of device, loss of service etc...

## System Testing:-

System testing is validates the systems and ensure that is satifies all the requirements.

* It will be classified types of three.
  1. Alpha testing
  2. Beta testing
  3. Acceptance testing.

### * Alpha Testing:-

It is conducted by the team members in their organisation.

### * Beta Testing:-

It is conducted by team leader of project.

### * Acceptance testing:-

It is conducted by the developer who is frequently communicating with the customers.

CHAPTER - 5

# Realiability, Quality Management & Maintainance.

## Realiability :-

It measures of the degree of consistancy and Quality of a product and performance of a product.

=> Software realibility refers to the probability of software running correctly without causing any failure.

=> If any defects arises, while running a Software then it may be called as unreliable software.

=> Generally, only 10% of the programs are frequently executed. They are called as core programs. The rest of 90% of the programs are rarely used.

=> The reliability of software can be include when only if the no. of defects are decreased.

=> The software reliability depends upon two pins.

1. Exact location of the error.
2. Execution profile means the way product is being used.

## Hardware Reliability:-

It refers to the ability of the hardware to perform its task without any failure.

=> Hardware failures are mostly physical failures where as software failures are mostly Design failure.

=> It is very to easy to rectify the hardware failures.

=> It is very difficult to verify the software failures.

=> The quantity of a software can't be changed once it is Explaced in to the storage

=> The Quality of a hardware can be changed even if it is fixed (or) Assembled.

## Software reliability Metrics:-

* It is a measurement of typical problem as a single person doesn't have any idea of the nature of the software.

* A single person can't find a suitable way of measuring the software ~~realibity~~ reliability.

* Software reliability metrics are four types.

1. project Metrics
2. product Metrics
3. process Metrics
4. fault & Failure metrices.

## 1. Project Metrics:-

In research and development most of the researches are been moved that good management can give good results.

* High reliability can activated by using better development process risk management process.

## 2. Product Metrics:-

It refers the size of the software in LOC (or) KLOC.

* A source code is processed and processed (comments) and other non executable code is not measured.

* Function point [FP] is an function oriented metric to measure the functionality of a program based on inputs, outputs and interfaces.

* Function point metric can be used to estimate the size of a software system in the form of the number of functions are to be used.

## 3. Process Metrics:-

It can be used to estimate and monitor to improve the reliability and quality of a Software.

* To measure the quality of the software ISO [International Standard Organisation] goes Certification (or) quantity management software.

## 4. Fault & Failure Metrics:-

The main goal of fault & failure metrics to determine the software design is approaching failure free ~~education (or)~~ execution (or) Not.

* Failure metrics are represented by ~~collect~~ collecting all the faults found during testing.

## Stastical Testing:-

It helps in determining the product reliability. It maintain focus on the reliability of a product rather than determining the errors in the problem.

* The below steps for performing stastical testing as follows.

=> Initially, determine the software operational profile.

=> Generate set of test cases that are to be tested.

=> Apply the test cases on the software product.

=> Finally, observe the failure rate.
=> Calculate the reliability depending upon the failure rate.

## Advantages:-

* Stastical testing generate accurate results when compared to other methods.
* Stastical testing improves the reliability of developed product (or) project.

## Disadvantages:-

* It is a cost consuming process.
* It is very difficult to define the operational profile.

## Software Quantity:-

It defines a developed product is said to be good quantity, if it fulfills all the user requirements and factors.

* It can be described using the following factors.
  1. Portability
  2. Useability
  3. Reliability
  4. Correctness
  5. Reusability
  6. Maintainability.

Portability:- Refers to the ability of the program being adaptable to any platform. Then it is said to be portable.

Usability:- Refers to the ability to which destrict users can invoke the functions very easily.

Reliability:- Refers to the ability of a software running correctly without any failures.

Reusability:- Refers to the ability of a program to be used in other application.

Correctness:- Refers to the ability of a software satisfying all the customer requirements.

Maintainability:- Refers to the effort applied in finding and removing the errors in a given software.

## Software Quantity Management:-

* A software Quantity management systems in a standard methodology adopted by various Organisation for assuming the Quantity.
* Quantity management system ensures that the developed product meet all the requirements specified by the users and it is a finished product without any defects.

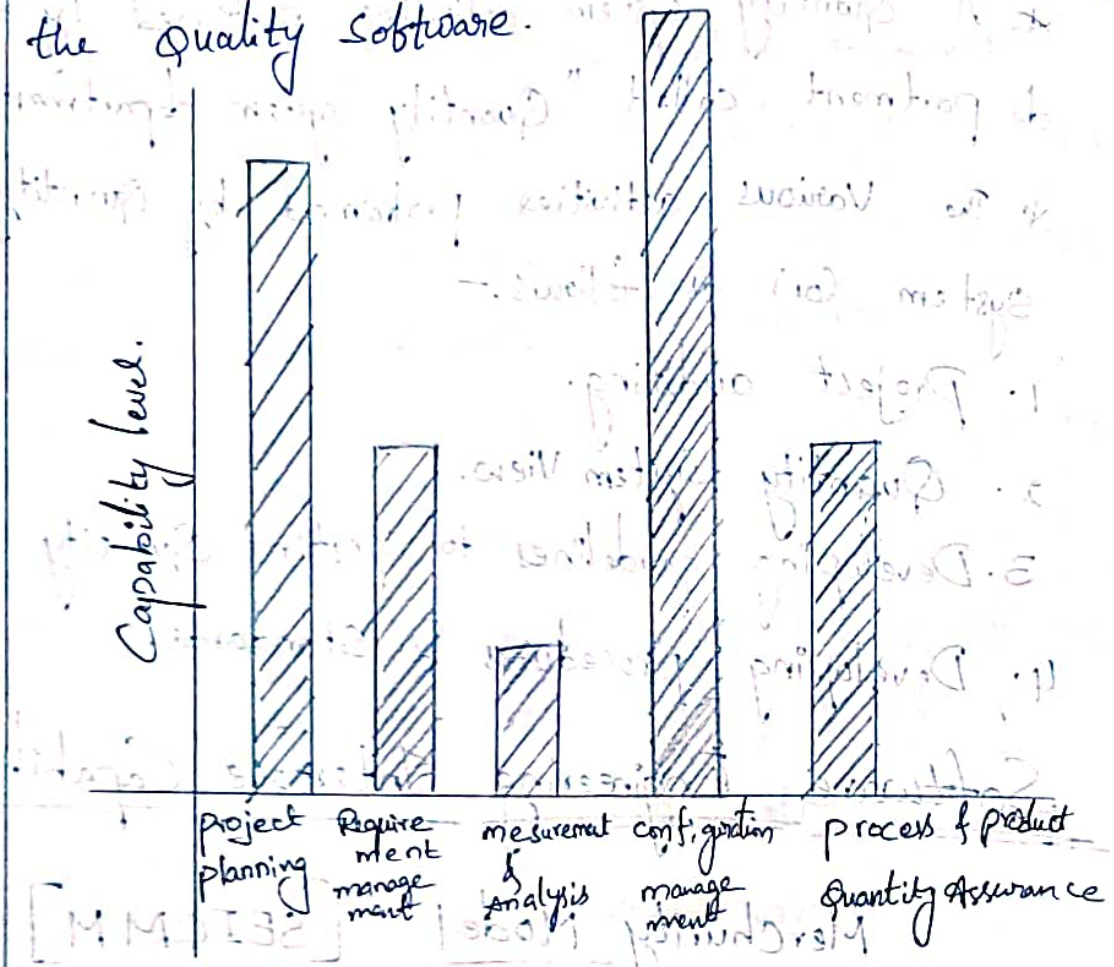
* A Software Quantity management system comprises of ~~mag~~ managerial structure as well as individual responsibility.
* A Quantity system activities performed by separate department called "Quantity system department".
* The various activities performed by Quantity system (or) as follows:-

1. Project auditing.
2. Quantity System View.
3. Developing guidelines to perform Quantity works.
4. Developing procedures & Standards.

## Software Engineering Interface Capability Merchurity Model [SEICMM]

* ∅ SEI assisted the organisation in impairing their software Quantities and business benefits.
* SEI focus on continuous improvement of an organisational work force.
* CMM provides the basic requirements for building the software process.
* CMM helps to ~~pre~~ improve the knowledge, skills and development of software process.

* CMM involves developed by [SEI]. It is a model of 5 levels of organisational merchurity that determines the effectiveness in delievring the quality software.




Level 1 :- It describes the periodic planning and effort required by individual to complete the project successfully.

Level 2 :- It describes softwares project requirements management, planning and configuration management.

Level 3 :- It describes ~~software~~ the size of a software and requirements analysis.

Level 4 :- It describes configuration management of a software.

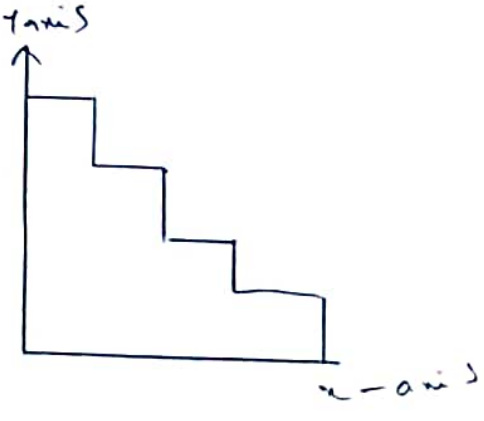
Level 5 :- It focus on continuous process impairing and product quality assurance can be predicted and effectively implemented when required.

## Reliability Growth Model

* Reliability growth modeling in a mathematical model that shows how the software reliability grows by detecting and connecting the errors.
* It helps in predicting the reliability in initial stage and attaining a given reliability level on bases of tests performed.
* The two important reliability growth model are as follows.

1. Step function Modeling
2. Little Wood model.

1. Step function Modeling :- It is a simple model that improves reliability by detecting and repairing errors with constant increment on each step.



## *2. Little Wood Model:

States that increasing in error repair may leads to decreases in software reliability. It is also called as negative reliability