

## UNIT - 1

### Evolving role of software

- Software engineering is the systematic approach to the development, operation, maintenance & retirement of software.
- We can say it is application of science & mathematics by which capabilities of computer equipment are made useful to man via computer program, procedure.

- ① It manage business information to enhance competitiveness
- ② It provide a gateway to worldwide information network & provide means for acquiring information in all.
- ③ Software transform person data so that data can be more useful in a local context.
- ④ Software act as the basis for the control of computer communication of information, creation of other programs.

**OBJECTIVE** :- (1) Scale up large system

(2) At low cost

(3) with small cycle of time

(4) can be used consistently with high quality soft.

IMP

Software myths ⇒ It propagate false belief & confusion in the mind of management, user and developers.

**(1) Manager myths** ⇒ (1) If the project is behind schedule, increasing no of programmers can reduce the gap.

(2) Standards are often incomplete, inadaptable.

(3) Developers are often unaware of all the established std.

(4) New worker take longer time to learn about project

(5) Outsourcing software to a third party does not help the organization which is incompetent in managing and controlling software project internally.

(6) Adding man power to the project

User myths → User tend to believe myths about the software because software manager & developer don't try to correct the false belief.

① Software is flexible ( requirement change can be added during any phase)

② Starting development with incomplete requirement often lead to software failure.

③ Adding requirement at a later stage after requires repeating entire development process.

④ Brief requirement states in the initial process is enough to start development.

⑤ Detailed requirement can be added at later stage.

Incorporating change requests earlier in the development process costs lesser than those occur at later stage.

Developer myths → ① Software development is considered completed when code is delivered.

② The success of software project depends on quality of product produced.

Software engineering require unnecessary documentation.

③ The only product that is delivered after completion of project is working program.

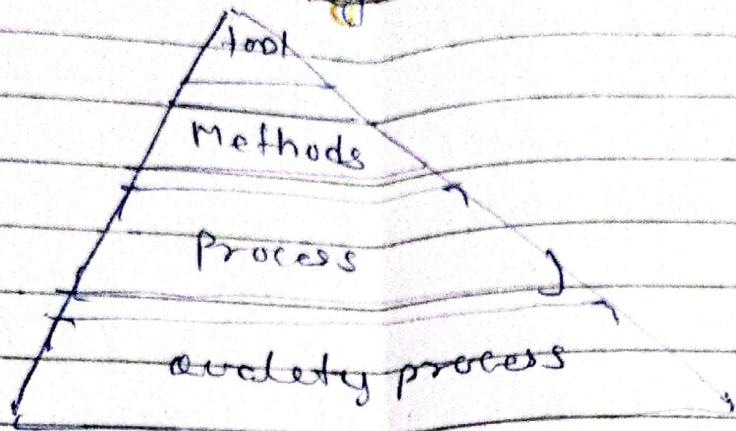
Software quality can be accessed only after program is executed.

50-70% of all efforts are expended after the software is delivered to the user.

④ Software engineering is about creating quality at every level of software project.

The quality of software can be measured during phase of development process by applying quality assurance.

Software engineering  
is a layered technology



- 1) Process layer - It allow the development on time, it defines an outline for a set of key process area that must be acclaimed for effective delivery.
- 2) Method layer - It provide the technical knowledge for developing software. This layer cover a broad array of task that include requirement analysis, design, coding, testing & maintenance phases.
- 3) Tools layer - It provide computerized or semicomputerized support for the process & method layer. Sometime tools are integrated in such a way that other tool can use information.
- 4) Quality focus - It is fundamental layer It include specification, efficiency, usability, maintenance and reusability -

## # Software process

To produce a software product the set of activities is used.., this set is called software process.

Objectives :- (1) optimality means the process should be able to produce high quality software at low cost.

(ii) Scalability = It means that It should be applicable for large software project.

(iii) Predictability = it determines how accurately of following a process in a project can be predicted before project is completed.

### Process steps

(1) Specification = Set out the requirement & constraint of the system,

(2) Design = produce a model of system,

(3) Manufacture = Build the system.

(4) Test = Check the system meets the requirement specification

(5) Install = Deliver the system to the customer and ensure it is operational.

(6) Maintain = Repair faults in the system .

## Software process model

It can be defined as a simplified representation of a software process.

It is an abstraction of the actual process.

~~IMP~~

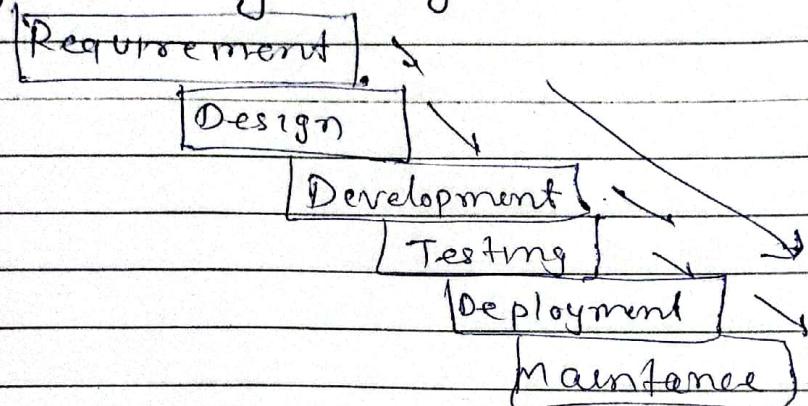
## # Waterfall model →

linear sequential model

for software engineering is known as waterfall model. It is also known as classical life cycle model.

It is a breakdown of project activities into linear sequential phases, where each phase depends on the deliverables of the previous one.

The approach is typical for certain areas of engineering design.



~~IMP~~

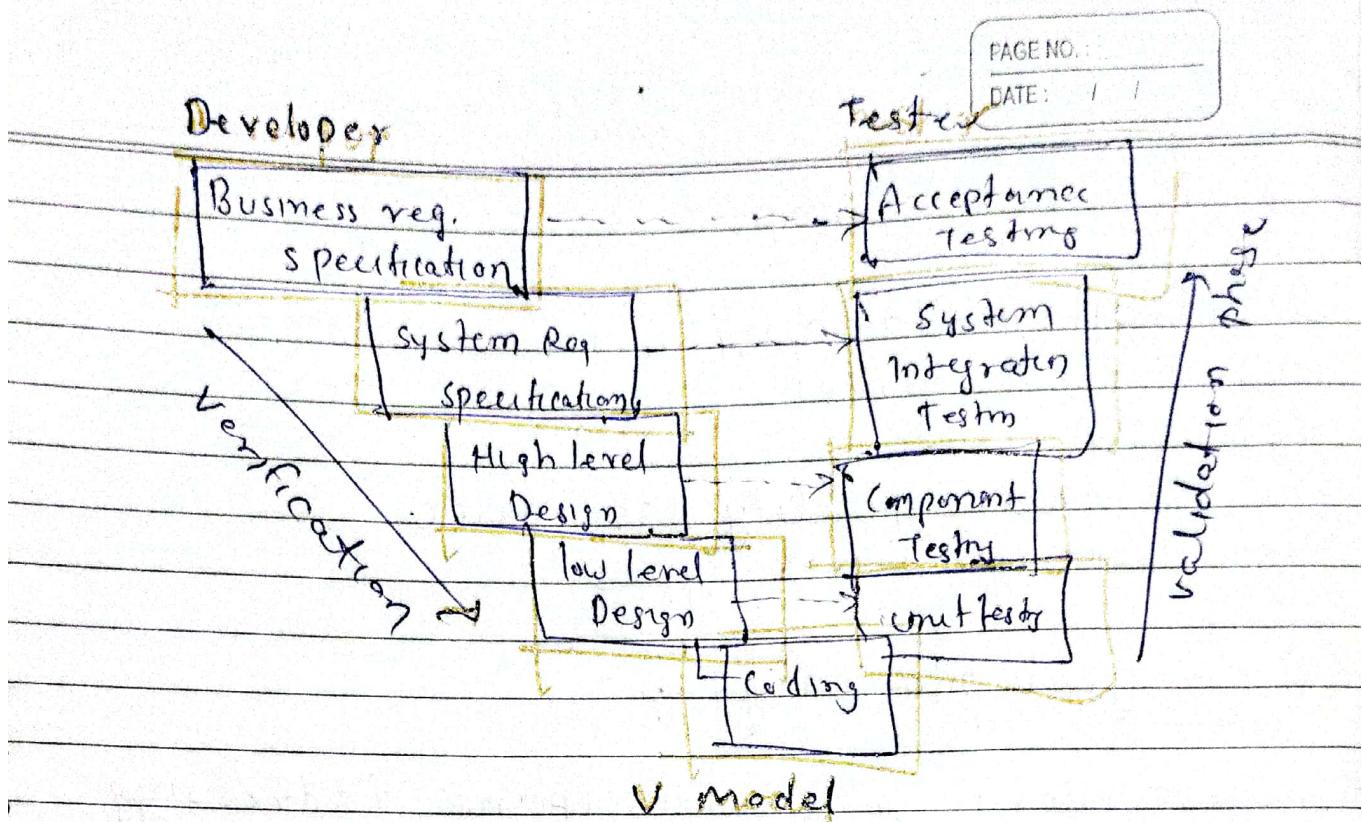
## # V model model → It is a extension of waterfall model

The process steps are bent upward after coding phase to form typical V shape.

It demonstrate the relationship between each phase of development cycle.

The horizontal axes represent time or project completeness.

The vertical axes represent level of abstraction is called V model.



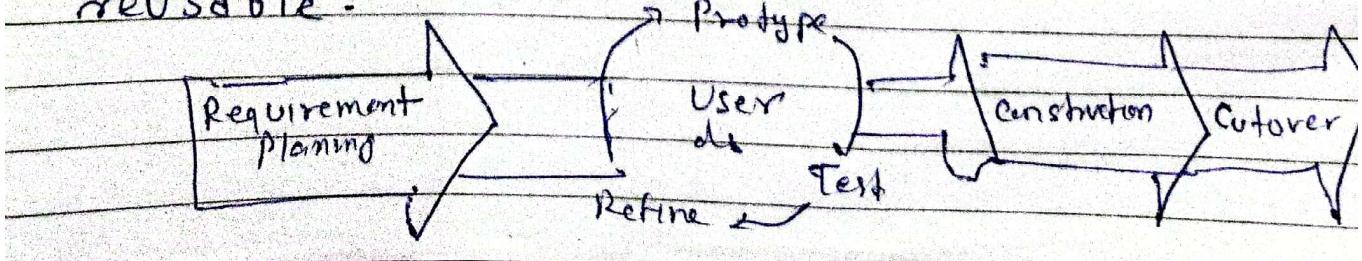
## # RAD Model

\* (Rapid Application Development) is based on prototyping and iterative development with no specific planning involved.

In RAD, it focus on gathering customer requirement through workshop or focus group early testing by customers using iterative concept, reuse of the existing prototype.

\* functional modules are developed in parallel as prototype.

RAD project follows iterative & incremental model. The important aspect <sup>for this model</sup> to be successful is to make sure that prototype developed are reusable.



**1) Evolutionary model** • It is known as successive version model (increment model). The software requirement is first broken down into several modules or function unit that can be incrementally constructed & delivered.

Each evolutionary model may be developed using an iterative waterfall model.

When customer prefer to receive product in the increment so that he can start using different feature as they are developed rather than waiting

**2) SDLC (Spiral Model)** • It combine the idea of iterative development with the systematic, controlled aspect of waterfall model. It is a combination of iterative development process model & sequential linear development model.

**Model design** → There are four phases.

① **Identification** → The phase start with gathering the business requirement in the baseline and subsystem requirements and unit requirement are all done.

At the end, of the spiral product is deployed in identified market.

② **Design** → It start with conceptual design in baseline & involve architecture design, logical design of modules finally design in subsequent spirals.

③ **Construct or Build** → It refers to the production of the actual software product at every spiral in the baseline. High clarity & design details are send customer to feedback.

④ **Evaluation and risk analysis** → It include, identifying estimating and monitoring technical feasibility and management risks. At the end of first iteration customer evaluates the software & provide feedback.

# Prototype model  $\Rightarrow$  It is applied when detailed information related to input and output requirement of system is not available.

It allows the users to interact and experiment with a working model of system known as prototype.

At any stage, if the user is not satisfied with the prototype, it can be discarded & an entirely new system can be developed.

It can be prepared by the approaches

(1) By using the system components to illustrate the function that will be included in the system to be developed.

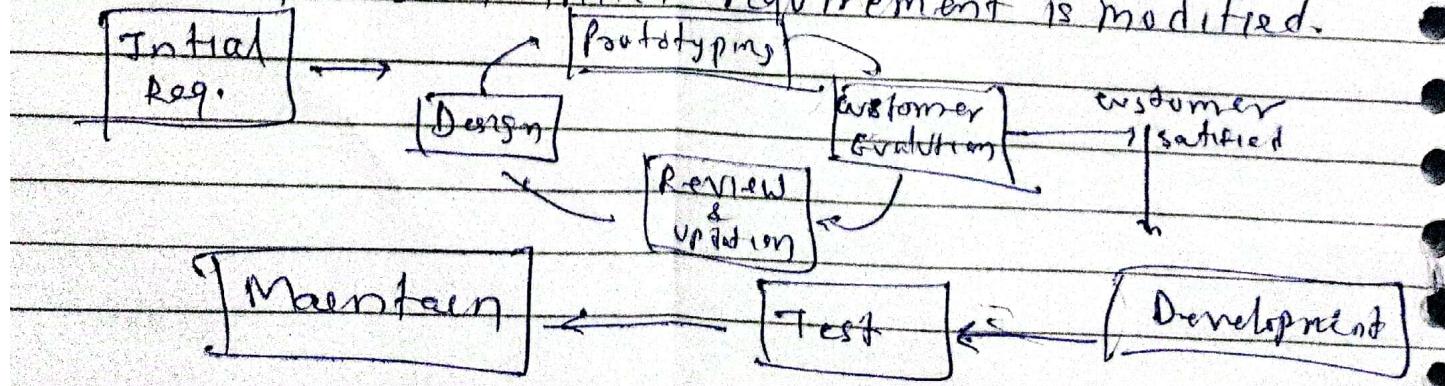
(2) By appreciating a version of the system that will perform limited subset of function.

They provide feedback to the developer regarding what is correct, what needs to be modified.

Based on feedback the prototype is modified to incorporate some of the suggested change that can be done easily then the user & clients are again allowed to use system.

The cycle repeat until judgement of the prototype and analyst.

Based on feedback initial requirement is modified.



# 0

## ② Project model ↳

### ① formal method ↳

It is branch of software engineering, in which we analyse software system.

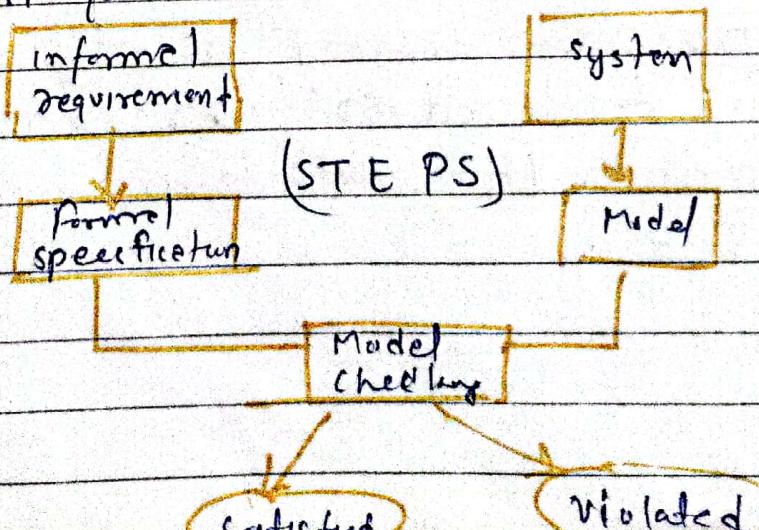
formal method used in developing computer system are mathematically based techniques for describing system properties.

formal method provide framework within which people can specify, develop & verify system in a systematic.

A method is formal if it has a sound mathematical basis, typically given by a formal specification language.

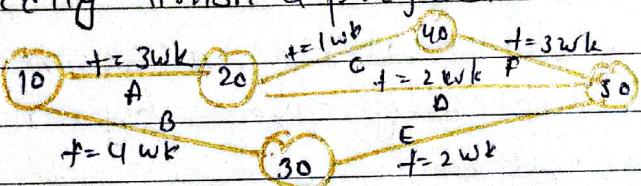
### ## Important of formal method ↳ in software

- ① It ensure that software is correct & reliable & correctness, consistency of a software.
- ② It is used to achieve software quality by applying testing, verification and validation process.



## PERT (Program Evaluation & Review Technique)

- # This technique creates a graph that shows the dependencies among the tasks.
- # The graph may not always have only one starting subtask or only one stopping subtask. The whole task is only completed when all the subtasks are completed.
- # It provides a visual representation of a project timeline & breaks down individual tasks.  
→ It is a project management planning tool used to calculate amount of time it will take to realistically finish a project.



## # fourth generation model

- # It encompasses a broad array of software tools that have one thing in common.
- # Each enables the software engineer to specify some characteristic of software at high level.
- # Implementation using a GUI enables the software developer to represent desired result in a manner  
→ In addition, 4GT developed software must be built in a manner that enable maintenance to be performed.

## UNIT - 2

### # Software project management concept.

In software project management, An effective software project manager can diagnose the technical & organizational issue that are most relevant in our software project.

It is an art & discipline of planning & supervising software project.

It is procedure of managing, allocating & timing resources to develop computer screen that full fill requirement.

There are three need for this (1) Time

(2) cost

(3) quality

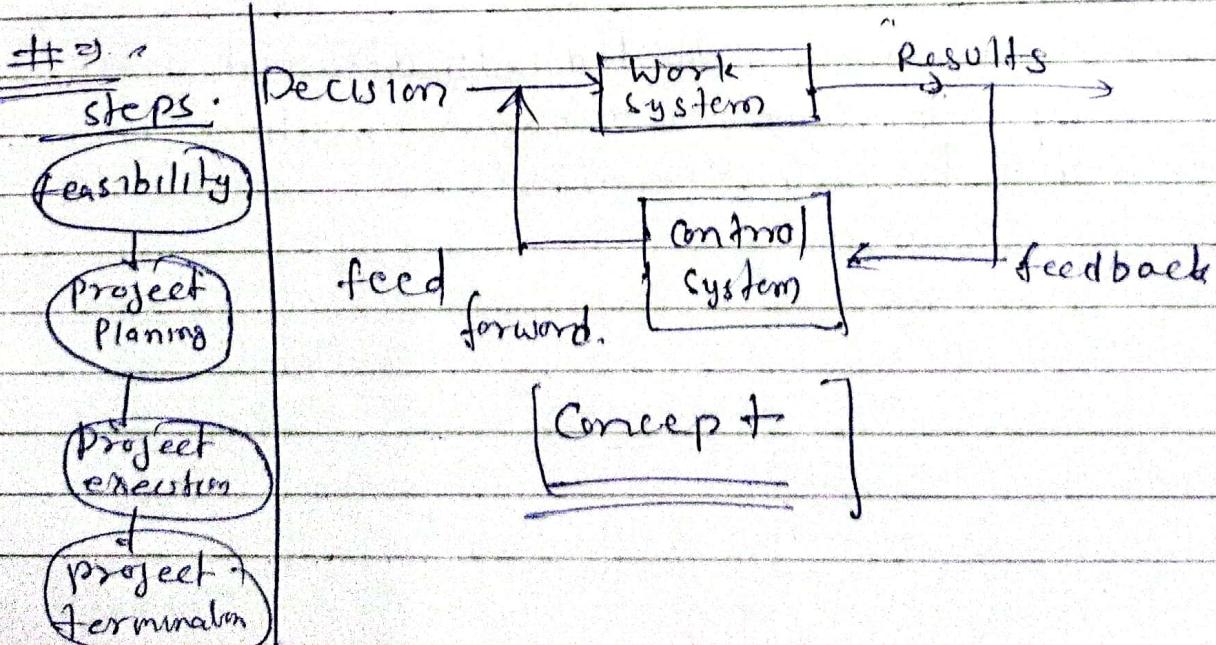
It is essential part of software organization to deliver quality product, keeping the cost within client.

It has four type (1) conflict management

(2) Risk management

(3) Requirement management

(4) Change management.



## # Software matrix -

It is a measure of software characteristic which are measurable or countable.

Software matrix are valuable for measuring

- (1) Software performance, planning work item
- (2) Measuring productivity & other uses.

## # Need of software matrix -

- (1) Create the quality of current product or process.
- (2) Anticipate future qualities of product or process
- (3) Enhance the quality of product.
- (4) Regulate the state of the project.

## # Step related to software matrix -

A matrix is a measurement of the level that any impute belongs to a system product or process.

There are 4 functions in Planning

- (1) Organizing
- (2) Controlling
- (3) Improving

IMP

## # Classification of software matrices

size

Design

control flow

information

Data structure

software scence

Test ability

ability

static

dynamics

dynamic

ability

## ① Product matrices $\Rightarrow$

These type of matrices are used to measure the characteristics of software product. The two important of software characteristics  
(1) size and complexity of software.  
(2) Quality & reliability of software.

## ② Process matrices $\Rightarrow$

These type of matrices are used to measure characteristic of software process. It is also used to measure method, techniques & tool that are used for developing of software.  
(1) efficiency of fault detection  
(2) effectiveness of defect removal during development  
(3) Pattern of testing defect arrival  
(4) The response time of the fix process.

## ③ Project matrices $\Rightarrow$ It is used to check project progress by manager time to time. It describe the project characteristic and execution as:

- ① number of software developer
- ② staffing pattern over the life cycle of software

## Scope of software matrices $\Rightarrow$ (1) cost & effort

(2) Data collection

(3) Productivity measure and model

(4) Quantity measure and model

(5) Performance & evaluation model

(6) Capability - maturity assessment

(7) Management by matrices

(8) Reliability model

## # Characteristics of software metrics

### (1) Quantitative :-

(It mean metrics can be expressed in values)

(2) Understandable  $\Rightarrow$  Metrics computation should be easily understood, the method of computing metric should be clearly defined.

(3) Applicability  $\Rightarrow$  It should be applicable in the initial phase of development of software.

(4) Repeatable  $\Rightarrow$  It should be same when measured repeatedly & consistent in nature.

(5) Economical  $\Rightarrow$  (It should be economical)

(6) Language independent  $\Rightarrow$  (It should not depend on any programming language)

## # Type of matrices :-

(1) Internal matrices  $\Rightarrow$  It is used for measuring properties that are viewed to be of greater importance to a software developer for example line of code (LOC)

(2) External matrices  $\Rightarrow$  It is used for measuring properties that are viewed to be of greater importance to the user e.g. reliability, functionality, usability etc.

(3) Hybrid matrices  $\Rightarrow$  It is matrices that combine product, process and resource metrics. For example cost per FP (Function point metric).

~~JMP~~

## COCOMO Model (constructive cost model) $\Rightarrow$

- It is a regressive model based on LOC (line of code)
- It is a procedural cost estimate model for software project.
- It often used as process of reliably predicting the various parameter such as cost, size, time & effort, quality.

- (1) Primarily effort  $\Rightarrow$  Amount of labor that will be required to complete a task.  
It is measured in person-month units.
- (2) Schedule  $\Rightarrow$  It means the amount of time required for completion of the job.  
It is measured in the unit of time such as week.

## Different system types

- (1) Organic  $\Rightarrow$  It is said to be an organic type if the team size required is adequately small, the problem is well understood & has been solved in the past & team member have a nominal experience regarding the problem.
- (2) Semi-detached  $\Rightarrow$  It is said to be a semi-detached type if the vital characteristic such as team-size, experience, knowledge of the various programming environment lie-in between that of organic & Embedded. It is comparatively less familiar & difficult to develop: of compilers.
- (3) Embedded  $\Rightarrow$  A software project with requiring the highest level of complexity, creativity & experience requirement fall under this category.

# Type of Models → There are three type of model

- (1) Basic cocomo Model
- (2) Intermediate cocomo Model
- (3) Detailed cocomo Model

(1) Basic cocomo model → It can be used for quickly & slightly rough calculations of software costs.

(2) Intermediate cocomo → The basic cocomo Model assumes that effort is only a function of a number of code & some constant evaluated according to different software system.

In reality no system's effort & schedule can be solely calculated on the basis of line of code. Reliability, experience & capability factors are known as cost drivers.

(3) Detailed model → It incorporates all the characteristic of the intermediate version with an assessment of cost driver impact on each step of software process. It uses different effort multipliers for each cost driver attribute.

In it, whole software is divided into different modules & then we apply cocomo in different modules to estimate effort then sum the effort.

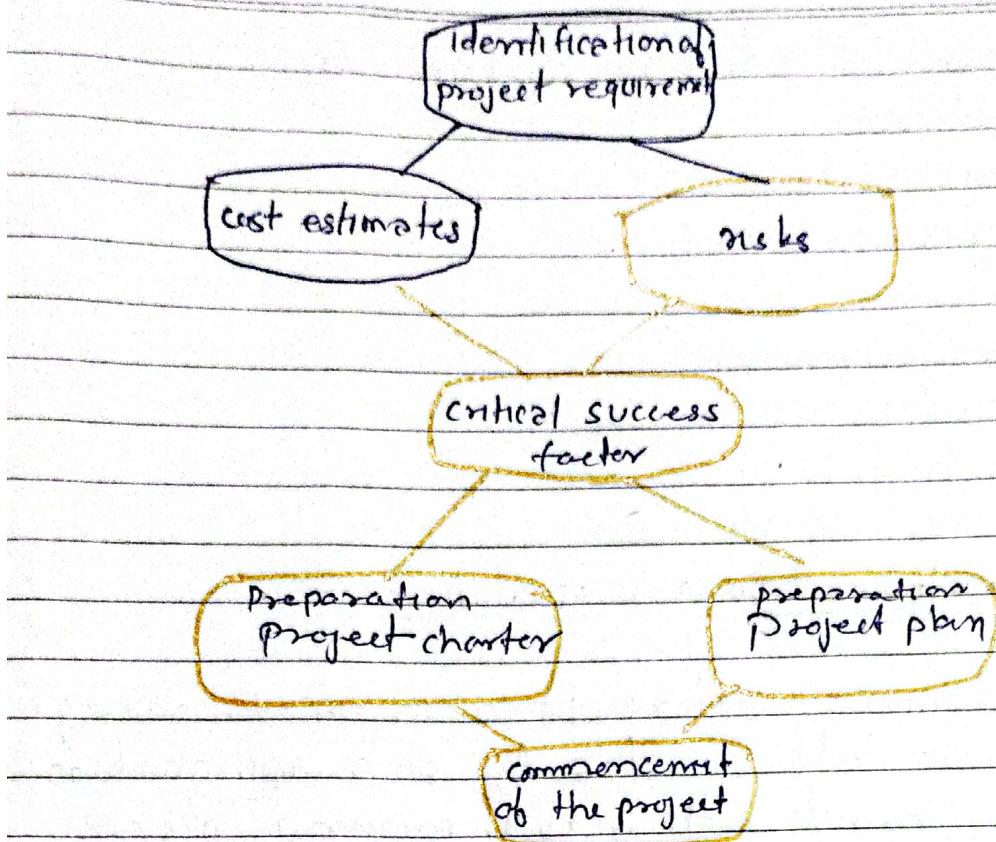
## ① Project planning →

- # Project planning is a discipline addressing how to complete a project in a certain timeframe, usually with defined stages and designated resource. It is essential.
- # Achieving success in software development require planning.
- # It involves deciding what task need to be done, in what order to do the tasks & what resource are needed to accomplish task.
- # Project planning process comprises several activities, which are essential for carrying out a project systematically.
- # These activities include estimation of time, effort & resource required & it refers to the series of task performed over a period of time for developing software.

# Project planning

PAGE NO.

DATE: / /



## Need of project planning

- ① objective and scope of project
- ② Techniques used to perform planning
- ③ Effort (in time)
- ④ Resources required for project.
- ⑤ Risks associated with project.

## Benefits of project planning

- ① It ensures that software is developed according to user requirements.
- ② It determines the available resource & activities.
- ③ It monitors the progress of project.
- ④ It identifies role of each project management team member.
- ⑤ It provides overview to management about cost of project.

# A typical project plan is divided into the following sections.

- ① Introduction
- ② Project organization
- ③ Risk analysis
- ④ Resource requirement
- ⑤ Work breakdown
- ⑥ Project schedule.

### # WBS - Work breakdown structure -

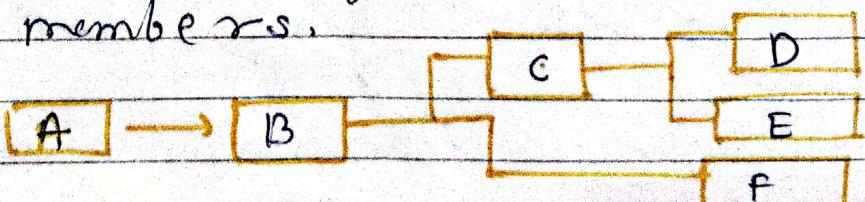
A work breakdown structure is a project management tool that take a step by step approach to complete large project with several moving pieces.

The work breakdown structure in project management & system engineering is a deliverable oriented breakdown of a project into small components

The top-level breakdown usually matches the life cycle model (LCM).

The next level breakdown usually matches organization process model (PM).

- IMPORTANT:-
- ① It must be a tree structure. There should be no loops or cycles in WBS.
  - ② Every task & deliverable description must be understandable & unambiguous.
  - ③ Every task must have a completion criterion.
  - ④ The purpose of WBS is communication with team members.



## # Software project estimation :-

Project estimation is a complex process that revolved around predicting the time, cost and scope that a project requires to be deemed finished.

It has six critical element of a project cost, Time, size or scope, Risk, quality, Resources.

### # Project Estimation Techniques.

(1) Top down estimation ⇒ It assign an overall time for a project & breaks it in to individual phase, work & task.

Top down approach allow you to take that overall timeline & estimate how much time you can commit to each project activity.

(2) Bottom-up estimation = A bottom-up estimate is the polar opposite of a top-down estimation.

You begin by estimating each individual task or aspect of project using this method. It is more accurate than top-down.

(3) Three point estimation ⇒ It is technique for generating bottom-up estimate that are sometime used. PERT method employs three point. Three points are (1) optimistic (2) pessimistic (3) most likely.

(4) Parametric model estimation = This method estimates the current project by pro-rating the details of previous project that you have completed ex. average project cost per square foot of house

#-step of estimation ⇒ ① Understanding scope of work

② Prioritizing tasks.

③ choosing estimate technique

④ Revising your estimation.

⑤ Special request from client.

~~IMP~~

~~TF~~

## RMM M

# Risk mitigation Monitoring and Management (RMM M)

# It aim to outline all risk analysis activities

+ It goal is to assist project team in developing a strategy for dealing with risks.

There are three issue of RMM M

① Risk Avoidance / Mitigation

② Risk Monitoring

③ Risk Management

① Risk Mitigation ⇒ (1) Proactive planning for risk avoidance  
(2) Risk mitigation is problem avoidance activity.

② Risk monitoring ⇒ (1) Assessing whether predicted risk occur or not.

(2) Collection of information for future risk analysis.

(3) Determine which risk cause which problem.

(4) It is used to find causes of risks.

(5) Ensuring risk aversion step are being properly applied.

③ Risk Management ⇒ Action to be taken in the event that mitigation step have failed & the risk has become a live problem.

## RMMM Plan

- # It is used by project manager as part of overall project plan.
  - # RMMM plan documents all work performed as part of risk analysis.
  - # Some software teams don't develop formal RMMM documents, each risk is documented individually using a risk information sheet (RIS)
  - # RIS is maintained using a database system so that creation, information entry and searches may be accomplished easily.
- RIS contain some information:
- (1) Risk ID, Date
  - (2) Description
  - (3) Context
  - (4) Mitigating / Monitoring
  - (5) Management
  - (6) Current status
  - (7) Assigned information

## Component based Development (CBD)

- It is defined as a set of reuse enabling technology tool, techniques that allow application development organization to go through entire Application process:
- # It is an approach to software development that focus on the design & development of reusable components.
- (1) Managing component
  - (2) Publishing components
  - (3) Referencing components
  - (4) Process pattern

## Project scheduling & tracking

# Project scheduling in a project refers to roadmap of all activities to be done with specific order & within time slot allocated to each activity.

for scheduling a process, it is necessary to

(1) Breakdown the project task into smaller

(2) findout various tasks & complete them.

(3) Estimate time frame required for each task

(4) Divide time into work-units

(5) Calculate total time required for project to finish.

Principles ➤ (1) product & process must be decomposed into manageable no. of activities & tasks

(2) task that can be completed in parallel must be separated from those that must completed serially.

(3) each task has start & completion date that take the task interdependencies into account.

(4) each scheduled task needs to be assigned to a specific team member.

(5) each task in the schedule needs to have a defined outcome.

# Project tracking is a method to follow, monitor & manage the progress of a series of tasks required for a project.

It is the best way to keep track of project because it monitors all the tasks associated with your project & help you stay on schedule.

Benefit ➤ (1) It gives Real time information

(2) Problem identifiers

(3) Team motivation

(4) Easy & precise reporting

## # Software quality Assurance →

It is ongoing process that ensure the software product meets & complies with the organization established & standaralized quality specification.

SQA is a set of activities that verifies that everyone involved with the project has correctly implemented all procedure & processes.

# It works parallel to software development.

# It focuses on improving the software development process.

II It ultimate goal is to catch a products shortcoming & deficiencies before the general public sees it.

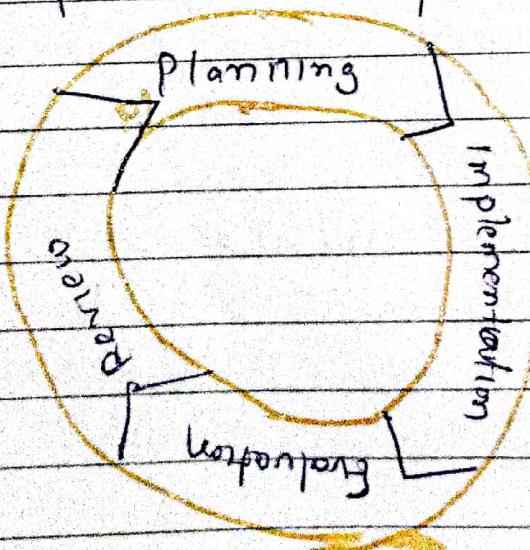
## # Quality vs Assurance →

Quality is simply stated "fit for use or purpose".  
It is all about meeting the needs & expectation of customers with respect to functionality, design, reliability, durability & price of project.

Assurance is nothing but a positive declaration on a product or service which gives confidence.

It provide a guarantee that product will work without any problem as per expectation.

Quality Assurance cycle



1. Plan      }  
2. Do          } steps.  
3. Check  
4. Act

Plan  $\Rightarrow$  organization should plan & establish the process related objectives.

Do  $\Rightarrow$  Development & testing of process & also 'do' changes in process.

Check  $\Rightarrow$  Monitoring of process, modify the process & check whether it meet the predetermined objective

Act  $\Rightarrow$  A tester should implement action that are necessary to achieve improvement in the process.

### Quality control (QC)

$\Rightarrow$  It is software engineering process used to ensure quality in a product or a service.

$\Rightarrow$  Its main aim of quality control is to check

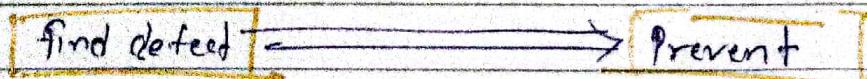
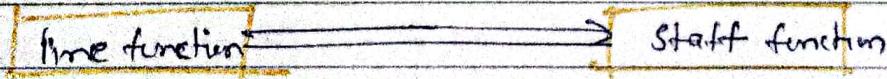
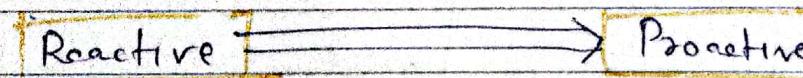
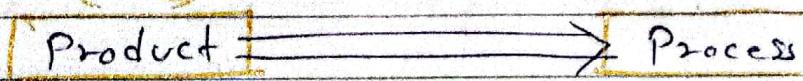
$\Rightarrow$  whether the product meet specification & requirement of the customers.

$\Rightarrow$  QC also evaluates people on their quality level level skill set & imparts training & certification.

$\Rightarrow$  This evaluation is required for the service based organization and help provide "perfect" service to the customer.

### QC v/s Q.A

### (Q.A)



## 4 Software configuration management (SCM)

The element that comprise all information produced as a part of the software process are collectively called a software configuration.

# SCM primarily deals with version selection tracking the changes and version control of software project with high productivity & minimize the error or risk factor.

# SCM refer to a process for maintaining computer system, servers and software in desired, consistent state etc.

Benefit of SCM ① SCM enhance the productivity of software as it ensure minimal error

② Communication between team member is easy with SCM

③ It ensure that software run effectively across multiple platform & operating system

④ SCM makes it easier to accomodate change in schedule

⑤ SCM help to control cost & increased efficiency

### Step of SCM

① Planning & Identification

② Version control & Baseline

③ Change control

(any change that made consistent)

④ Configuration status Accounting

⑤ Reviews & Audit

(make sure change from baseline match at post)

## UNIT-3

PAGE NO.:  
DATE: / /

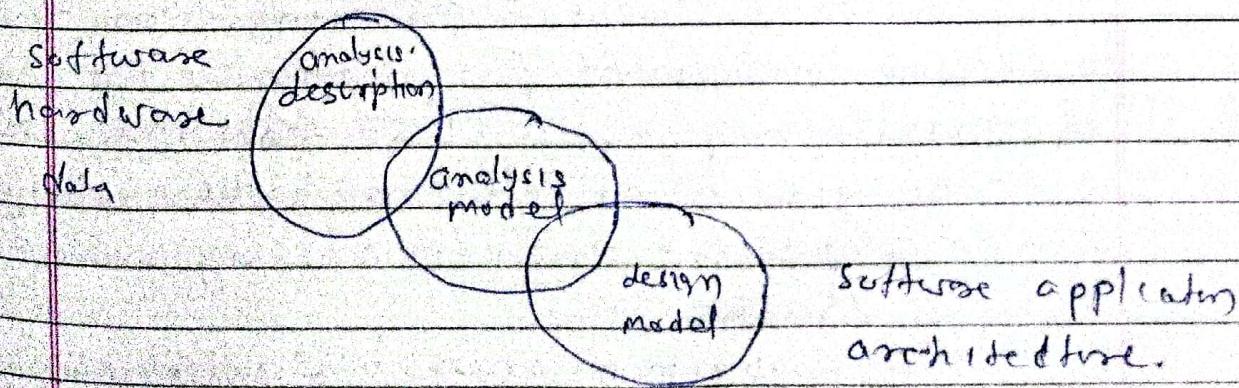
### # Analysis modeling:

- ① It is technical representation of the system
- ② It act as link b/w system description & design model
- ③ In it, Information, behaviour & function of the system are defined & translated into architecture, component, interface level design.
- ④ Development process starts with analysis phase
- ⑤ The phase result in a specification document
- ⑥ It can use two separate approaches, depending on whether the implementation phase is done using a procedural programming language.

#### Principle :- ①

The information domain must be understood & represented

- ② Models should be concrete.
- ③ function & problem statement must be defined.
- ④ The various analysis model are flow oriented modeling.
- ⑤ Model should be developed to give emphasis on system information.

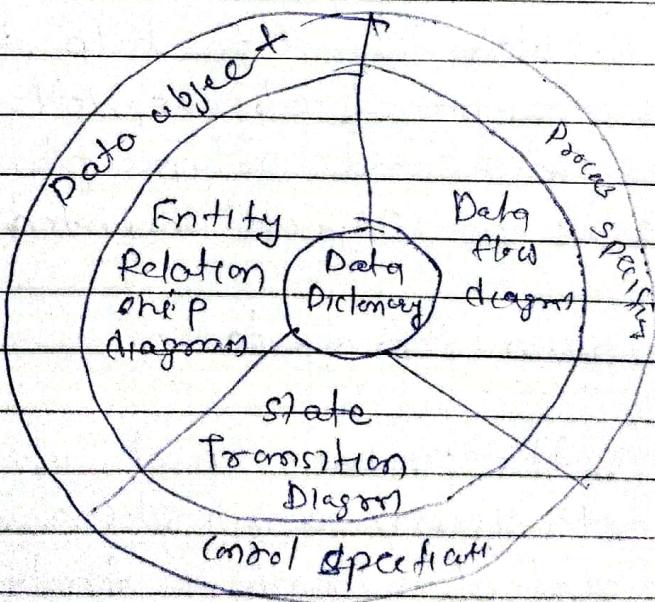


- # Activities :**
- ① It describe operational & behavioral characteristic
  - ② show the relationship b/w software inter.
  - ③ the representation of information, function & behavior
  - ④ Convert the design into more descriptive model like use case fp diagram
  - ⑤ Provide the customers & developer the means to maintain quality.

**+ # Objectives :**

- ① Describe what the customer require
- ② Establish a basis for the creation of a software design
- ③ Define a set of requirements.
- ④ Analysis model bridge the gap b/w system.

**+ # Element :**



## # Analysis modeling approaches

### (1) structured analysis =>

- ① consider data & processes that transform the data as a separate entities
- ② It is top-down approach
- ③ It focus on defining the problem with the help of function performed.

### (2) Object oriented analysis =>

- ① It focus on definition of class .
- ② Defines the system as set of object which interact with each other.
- ③ Analyse the problem domains & partition the problem with the help of object
- ④ The concept of object, attributes, class operation inheritance and polymorphism should be known to work on Object oriented analysis.

### (3) Domain Analysis =>

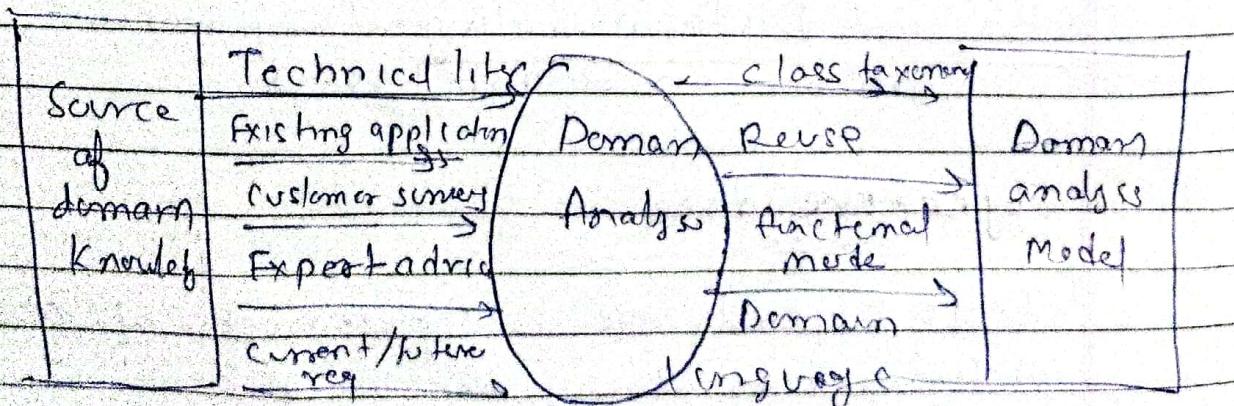
- ① It is the process that identifies the relevant object of an application domain
- ② The goal of Domain Analysis is software reuse
- ③ The higher the level of the life-cycle object to reuse, the larger are the benefit coming from its reuse .

## # Concept & Technical, application domain of software

- (1) Framework are excellent candidates for domain analysis. they are higher level but average programmer can understand.
- (2) Umbrella activity involving the identification analysis & specification of common requirement.
- (3) Object oriented domain analysis involves identification, analysis & specification of reusable capabilities.

## # Input & output structure of domain Analysis

- (1) The main goal is to create the analysis classes & common function.
- (2) The input consist the knowledge domain.
- (3) The input is based on technical survey, customer survey & expert advice.
- (4) The output domain consists of using input as reference & developing the function model.



## # Analysis model ↗

### ① Data modeling ↗

- ① It is a analysis of data object that are used in a business & identifies the relationship among these data object.
- ② It is first step in doing object-oriented programming.
- ③ Data modeler often use multiple models to view the same data & ensure that all processes data flow have been identified.
- ④ Data model are modeled to define their attributes & relationship.

### Approach ↗

- ① conceptual data modeling → (Identifies higher level of relationship)
- ② Enterprise data modeling → (It is similar to conceptual but address the unique requirement of a specific business)
- ③ logical data modeling → It illustrate the specific entities, attributes & relationship involved in a business function
- ④ Physical Data modeling → (represent on application & database)

## # Object of data ↗

- ① Data object are representation of most composite information of system.
- ② Data object are all related to each other.
- ③ Data attributes help to make reference to other instance in another table.

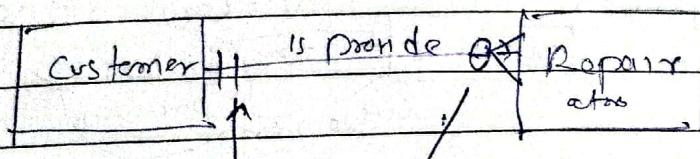
## # Cardinality (number of occurrences):

- ① It is referred to as "one" or "many".
- One to one" & "one to many", many to many.
- ② When one instance of object A relates to one instance of object B its one to one cardinality.

## # Modality:

- (i) Modality is 1 if an occurrence of the relation is mandatory.
- (ii) Modality is 0 if there is no explicit need for the relationship to occur & the relationship is optional.

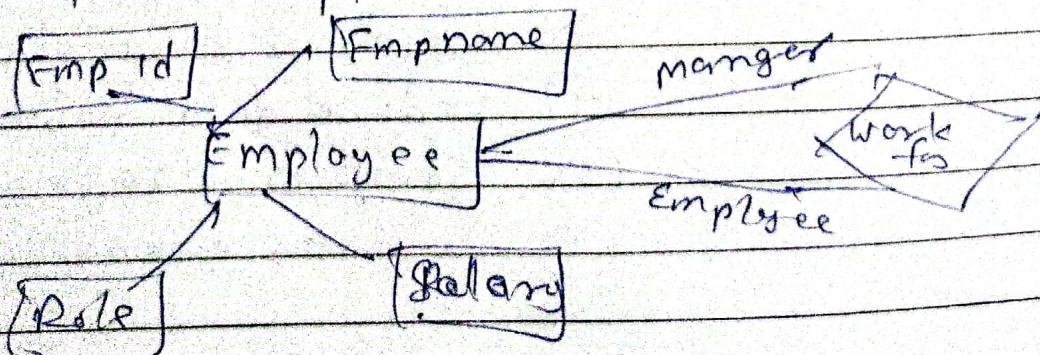
like Each faculty member advise many student



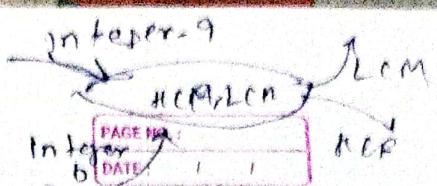
Modality implies that in order to have a repair action there may be a situation in which a repair action is not necessary.

## # Flow oriented modeling:

This represent how the data objects are transformed as they move through the system. If provide the view of the system in the graphical approach

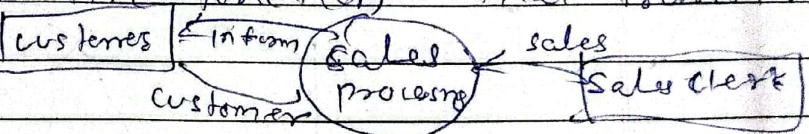


- It has (1) Processes (2) Constraints  
 (3) Data flow (4) Control flow  
 (5) Actors (6) Data stores



# **DFD** → It is a graphical technique that depicts the information flow & transform that are applied as data moves from input to output.

- (1) A level 0 DFD, also called a fundamental system model & context diagram represent entire system as a single bubble with input & output data.
- (2) A level 1 DFD, might contain five or six bubbles with interconnecting arrows.
- (3) It depicts how input is transformed into output as data object move through a system.  
 - It also depicts the function that transforms data flow.



## Data Dictionary

- A tool that used for recording & processing information about the data.
- It can be integrated with the DBMS or be separate.
- It can be used as a repository for common code.

It may be referenced during system design.

**Benefit of DDS** - (1) improved the documentation

- (2) consistency in data use
- (3) easier data analysis
- (4) reduce data redundancy
- (5) simpler programming
- (6) the enforcement of standard.

# facilities :- There are two type of facility

- (1) To record & analyse the data requirement independently of how they are going to be met conceptual data model (entities, attributes)
- (2) To record and design in term of database & file structured implemented & the program which access them internal schema.

It show the relationship between conceptual & implementation view.

# DD information = (1) name associated with element

- (2) Details of ownership
- (3) Details of user that refer to element
- (4) Details of the system & program
- (5) Details about data element in data processing unit
- (6) total storage requirement
- (7) Details of relationship of data item

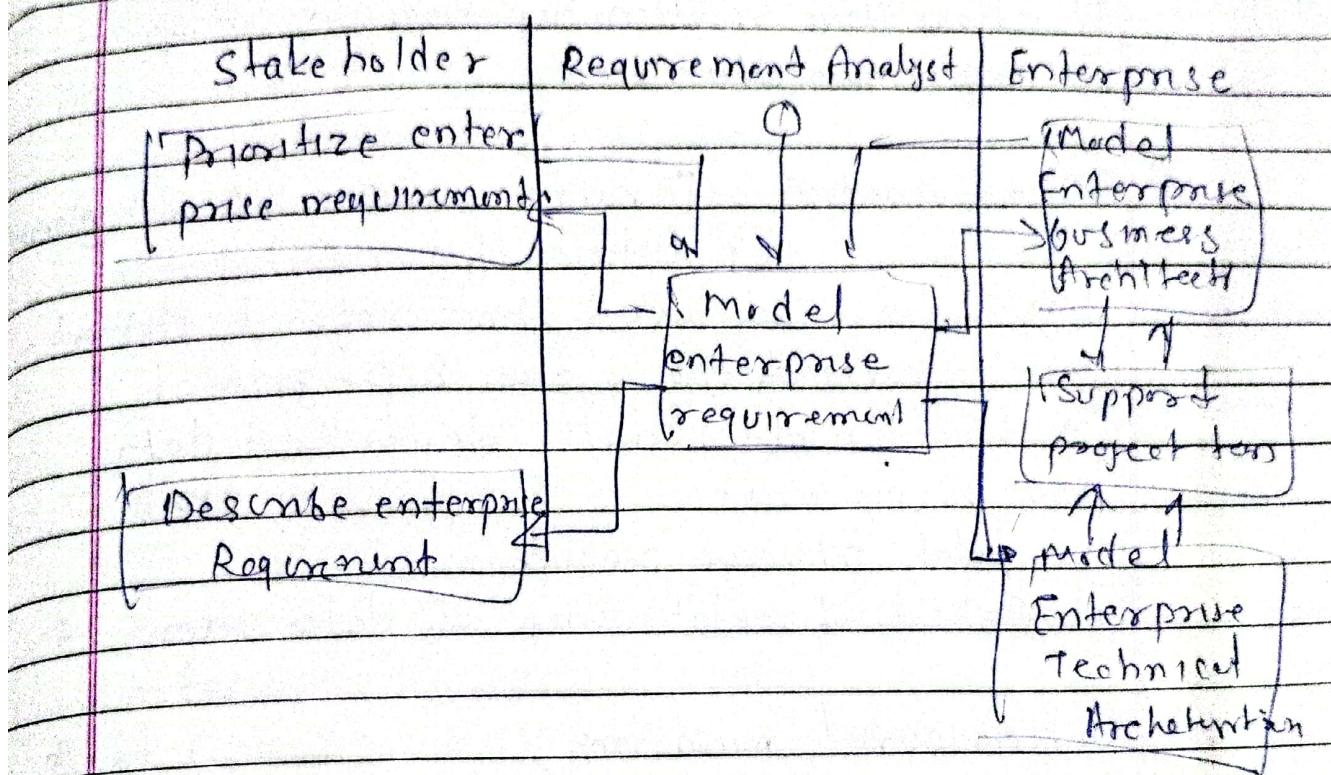
# Behavioural modeling =

# It visualize, specify, construct & document the dynamic aspect of a system

# It can be categorised into use case diagram, interaction diagram, state chart diagram & activity diagram,

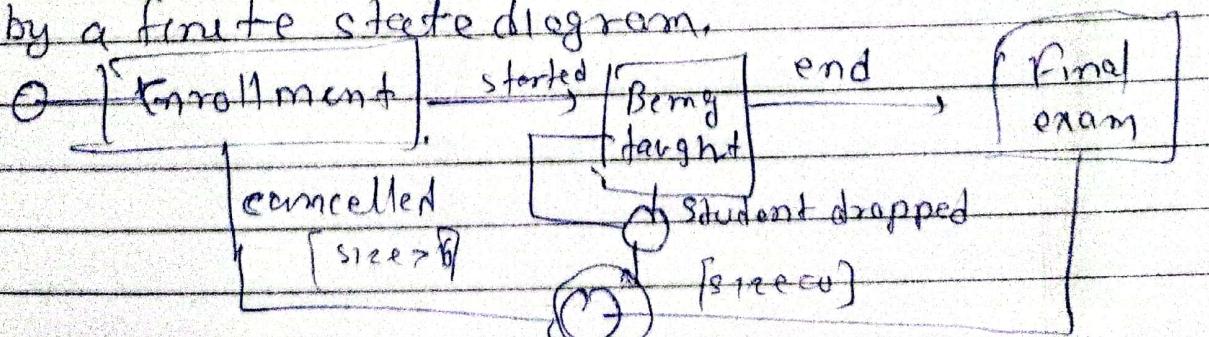
## ① Activity diagram

- It helps to describe the flow of control of the target system.
- It describes use case and business process.
- It is object-oriented equivalent of flow chart & data-flow diagram (DFDs).



## ② State machine diagram

- It can show the different state of an entity how an entity responds to various events by changing form one state to another.
- The history of an entity can best be modeled by a finite state diagram.

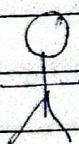


## ⑦ Use case diagram.

- # It describe the behaviour of target system from external point of view
- # It describe "the meat" of actual requirement.
- # Use case →



It describe a sequence of action that provide something of measurable value or a factor & is drawn in horizontal ellipse.



① Actor = An actor is a person, organization or external system that play a role in one & more interactions with your system.

It is drawn as stick figure.

### → Association.

Association between actor & use case are indicated by solid line. An association exist whenever an actor is involved with an interaction describe by user case.

④ These diagrams are used to model the context of a system by enclosing all the activities of a system within a rectangular

② It is also used to model requirement of a system from outside point of view

## # Sequence diagram ↗

It collaborate of object that is based on a time sequence.

+ It show how the object interact with other in a particular scenario of a use case.

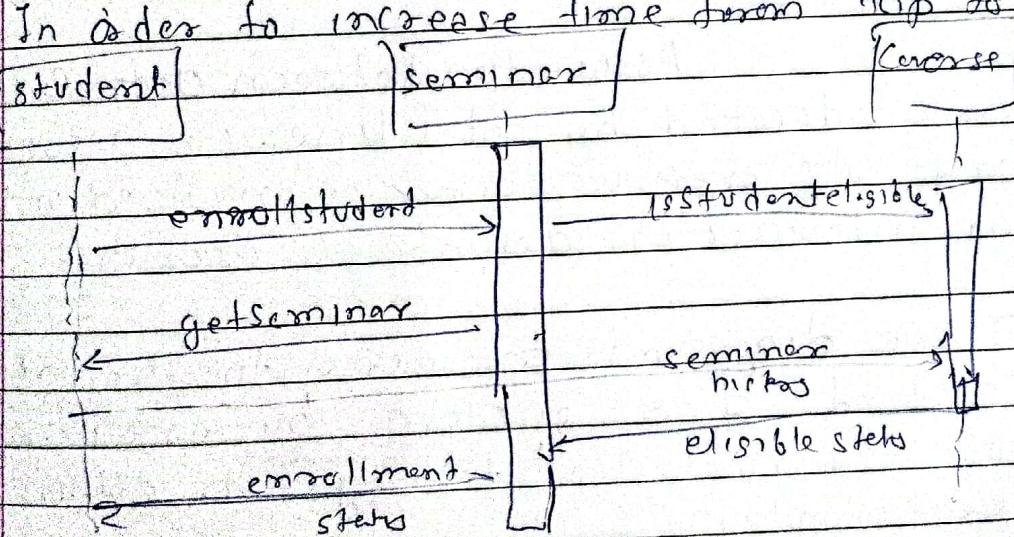
A These diagrams are interaction diagram that illustrate the ordering of message according to time

## Notations ↗

These are in two-dimensional form.  
the object that initiate the interaction are placed on X-axis.

the message that these object send & receive are placed along Y-axis.

In order to increase time from top to bottom



## # Collaboration diagram ↗

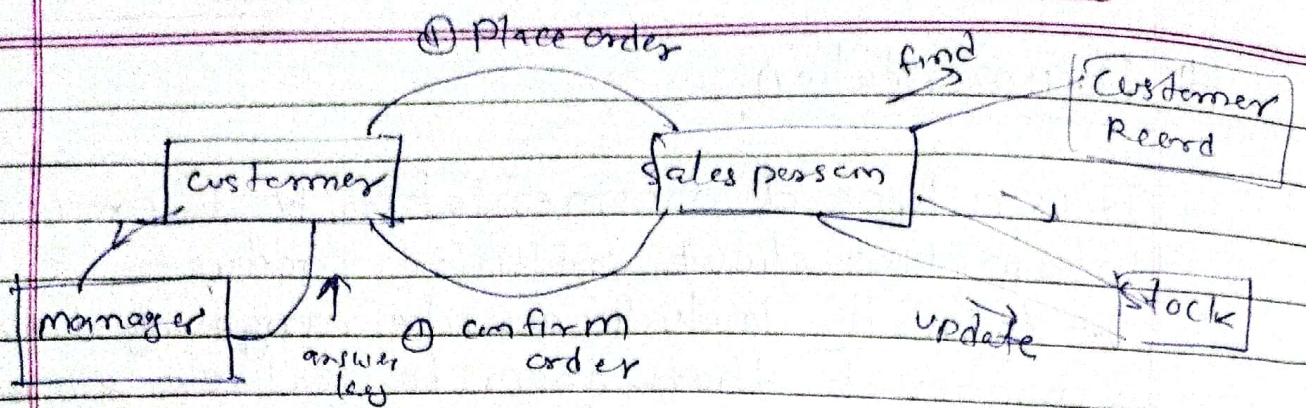
It is interaction diagram

that illustrate the structure of the object that send and receive message.

Notation : The object that participate in the interaction are shown using vertices.

The message is shown labeled arrow.

Ex Automated house Trading system.



## # Functional modeling,

- 1 It gives the process perspective of the object-oriented analysis model.
- 2 An overview of what the system is supposed to do.
- 3 It defines the function of the internal processes in the system with the aid of data flow diagrams.
- 4 It depicts the function derivation of data values without indicating,

## # Design heuristics → The main goal of heuristic evaluation is to identify any problem associated with design of user interfaces

- A heuristic evaluation is a usability inspection method for computer software that help to identify usability problem in the user interface.

Analysis → ① Establish an appropriate list.

- ② Select your evaluators
- ③ First evaluation phase
- ④ Second evaluation phase
- ⑤ Record problem
- ⑥ Debriefing session

## # Documentation.

It is a type of documentation that provides information about software product & system. It typically include a wide range of document & materials that describe features & use.

Types ① Requirement documentation

It is description

how the software shall perform and which environment setup would be appropriate to have the best out.

② Architectural → It is special type of document that concern the design. It contains very little code.

③ Technical Documentation = It contains technical aspect of software like API.

④ End-use Documentation

These are made for end users. It contains support resource for end use.

Advantage - ① Help development teams during environment

② help end-user in using product

③ It cuts down duplicative work

④ make easier to understand code

⑤ help in establishing internal coordination.

Disadvantage - ① code is time-consuming

② It has no influence on

the performance of an application

③ Documentation is not so fun. It sometimes boring to a certain extend.

## SRA (Software requirement analysis)

- # It is a systematic approach for documenting the workflow processes in each functional area.
- # It is significant & essential activity after elicitation.
- # We analyze, refine and scrutinize the gathered requirement to make consistent requirement.
- # After completion of this analysis, it is expected that the understandability of the project may improve significantly.
- # Use the interaction with customer to clarify point of confusion & to understand which requirement are more important than other

1. Draw the context diagram

2. Develop prototype

STEPS

3. Model the requirement

- # Draw the context → It defines that the boundaries & interface of the proposed system with external world.
- # Develop prototype → prototype helps the client to visualize the proposed system & increase understanding of the requirement.
- # Model requirement This consist of various graphical representation of the functions like data flow diagram, entity-relationship diagram, data dictionaries.
- # Finalise requirement After modeling requirement, we will have a better understanding of the system behaviour.

# Software design → The design phase of software development deals with transforming customers requirement as described in the SRS document into a form implementable using programming language.

- (1) Interface design
- (2) Architectural design
- (3) Detailed design

① Interface design → It is the specification of the interaction between a system & its environment. This phase proceeds at a high level of abstraction with respect to the inner working of the system. Internal system are completely ignored & system is treated as a black box. It include

- (1) Precise description of events
- (2) precise description of message
- (3) Specification of data.

② Architectural design : It is the specification of major component of a system their responsibilities, properties & interactions b/w them. It adds important details ignored during interface design. It include some levels :-

- (1) Communication & interaction between component.
- (2) Component interface
- (3) Allocation of function responsibilities.
- (4) Component scaling & performance.
- (5) Gross decomposition of the system into major component

# Detailed design ⇒ It is specification of internal element of all major system components their properties, relationship & algorithms.

It include

- (1) User interface
- (2) Unit states & state change
- (3) Data & control interaction b/w unit.
- (4) Decomposition of major system component
- (5) Algorithm & data structure

# Software Architecture ⇒ It provide an explanation of how your system behave on a structural level. It provides the foundation to which all the software a company has can be changed, built or retired.

It impact the quality, performance, maintenance & success of a system based on design.

Example ⇒ The best software architecture examples are readily available online. This is because the easiest way to create software architecture is to use existing templates. This can save team hours & hours of works.

Software architecture serves as the blueprint for the system by giving an overview of how the system works & managing the complexity of it.

### Tools → (1) Refactoring

A process of restructuring existing code without changing its external functionality.

(2) Cloud → Lack of legacy knowledge & software architecture documentation elongates this process.

(3) half - It reverse-engineers all database structure, code components & interdependencies in complex software system.

Principle → It dictate the process of converting & building software.

① An end-user application to graphically display & explore the structure of an application.

② Impact analysis capabilities to simulate the impact.

③ HTML report feature technical documentation

④ View source code & all details about any component.

⑤ Search any component & position it on a current graphical view, on a transaction.

⑥ High database model.

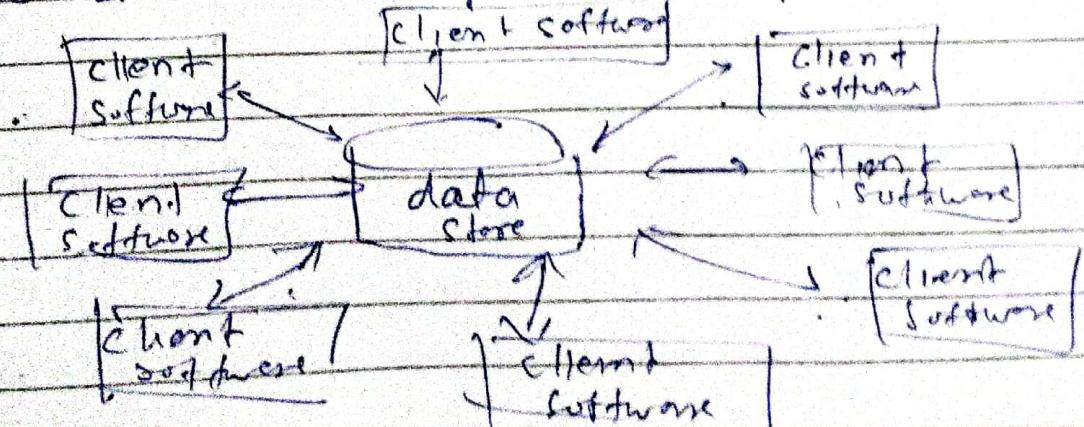
Data design: It is the first design activity which result in less complex, modular & efficient program structure.

The data, object, attributes & relationship depicted in entity relationship diagram & the information stored in data dictionary provide a base for data design activity.

- Principle →
- ① The data structure needed for implementing the software as well as operations.
  - ② A data dictionary should be developed to depict how different data object interact with each other.
  - ③ Stepwise refinement should be used in data design.
  - ④ A library containing the set of useful data structure along with operations
  - ⑤ Language used for developing the system.

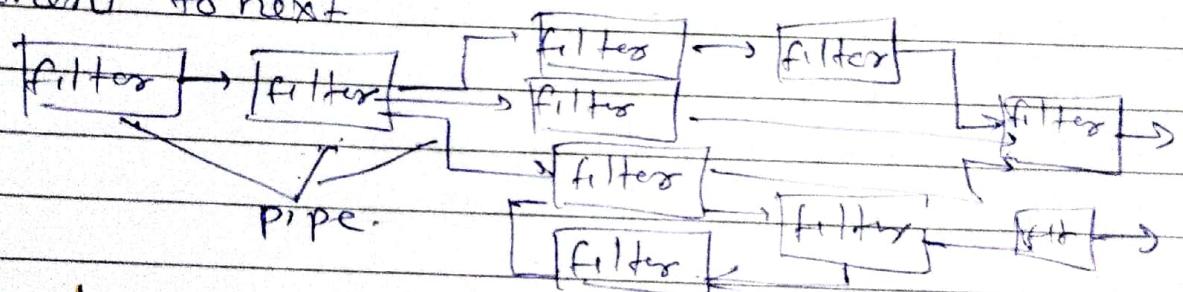
## Software architecture styles

(1) Data centred architecture → A data store reside at the centre of architecture & is accessed frequently by other component that update, add, delete or otherwise modify data within the store. Existing components can be changed & client component independently execute.



② Data flow architecture → The architecture is applied when input data are to be transformed through a series of computational & manipulative component into output data.

A pipe & filter pattern has a set of component called filters connected by pipe that transmit data from one component to next.

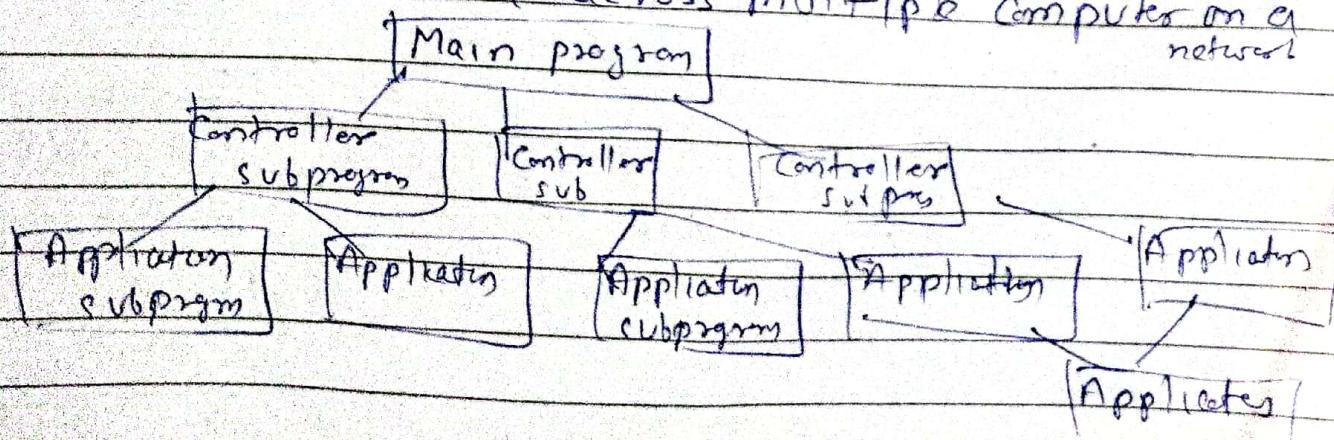


③ Call & return architecture

It enable you to achieve a program structure that is relatively easy to modify & scale. two architecture style are relative to this call & return architecture style.

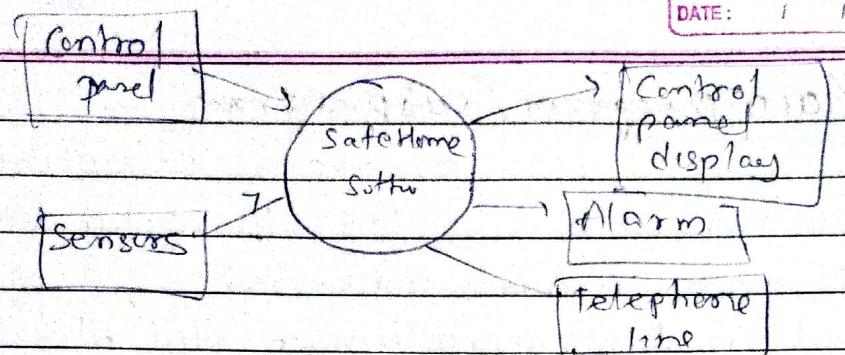
④ Remote procedure call architecture → The component of main program

architecture are distributed across multiple computer on a network



- # Main program/subprogram → This classic program structure decomposes function into a control hierarchy where a "main" program invokes a number of program component that in turn may invoke still other component.
- # Object oriented architecture → The component of a system encapsulate data & the operation that must be applied to manipulate data. Communication & coordination b/w components are accomplished via message passing.
- # Layered architecture → A number of different layer are defined, each accomplishing operations that progressively become closer to the machine instruction set.
  - At outer layers, components service user interface operation
  - At inner layers, components perform operating system interfacing

- # Transform mapping → It is a set of design step that allow a Data flow diagram with transform flow characteristic to be mapped into specific architectural style.
- In this section transform mapping is learned by applying design step to an example system a portion of the safeHome security software.
- Example → The safeHome security system is representative of many computer based product & systems in use today. The product monitor real world & react to changes that it encounter.



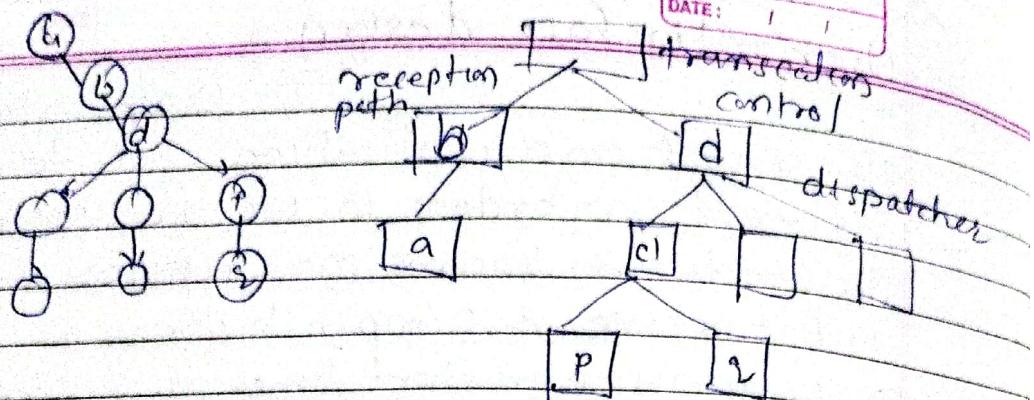
- # Steps ⇒
- ① Review the fundamental system model
  - ② Review & refine data diagram of software

- ③ Determine whether DFD has transform or transaction flow characteristic.
- ④ Isolate the transform centre by specifying incoming & outgoing flow boundaries
- ⑤ Perform "first level factoring". program structure represent a top-down distribution of control.
- ⑥ Perform "second level factoring" second level is accomplished by mapping individual transform of a DFD into appropriate modules within archt.
- ⑦ Refine the first iteration architecture using design heuristic for improved quality.

## # Transaction mapping ⇒

A single data item trigger one or a number of information design flow that effect a function implied by trigger data item. These data item is called a transaction & its corresponding flows characteristics.

In this section, we consider design steps used to treat transaction flow.



- steps →
- ① Review the fundamental system
  - ② Review & refine data flow diagram
  - ③ Determine whether DFD has branch or transaction.
  - ④ Identify the transaction centre & flow characteristic along each of action paths
  - ⑤ Map & DFD in a program structure to transaction processing
  - ⑥ factor & refine the transaction structure & structure of each action path
  - ⑦ Refine the first iteration architecture

#

## Refining architectural design

Successful application of transform & transaction mapping is supplemented by additional documentation that is required as part of architecture design. After the program structure has been developed & refined

- ① A processing narrative must be developed.
- ② local & global data structure are defined
- ③ A set of design review are conducted
- ④ All design restriction & limitations are noted
- ⑤ Refinement is considered.
- ⑥ An interface description is provide for each.

## # User interface design →

User interface is the front end application view to which user interact in order to use the software.

The software become more popular.

(1) Attractive & simple to use

(2) Responsive in short time

(3) Clear to understand

(4) Consistent on all interface screen.

### Types ↗

(1) Command line interface → It provide a command prompt, where the user type the command & feed to the system. The user need to remember the syntax of command.

(2) Graphical user interface → It provide simple interactive interface to interact with the system. GUI can be a combination of both hardware & software.

Using GUI, user interpret the software

### # Process ↗

① User task, environment analysis

analysis

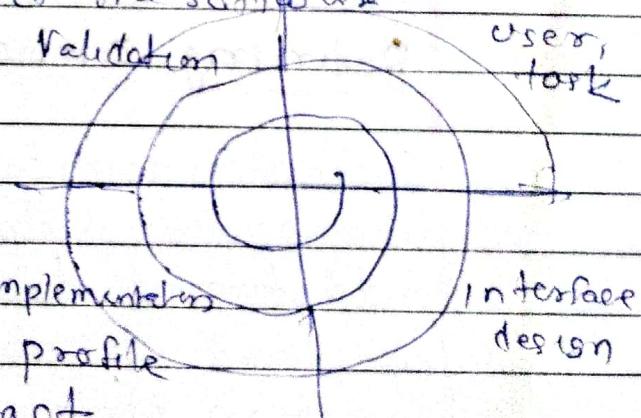
It focus is based on the profile of user who will interact

with the system, i.e. understanding, skill & knowledge.

The analysis of user environment focus on the physical work environment.

Question arise :

Where will the interface be located physically?



## ② Interface design

The goal of this phase is to define the set of interface object & control & control mechanism that enable user to perform desired task.

Design issue such as response time, command & action structure, error handling, help facilities

## ③ Interface implementation ⇒

The implementation activity begins with the creation of prototype that enable usage scenarios to be evaluated. Many other elements of an iterative environment can be used for completing the construction of an interface.

## ④ Interface validation ⇒ This phase focus on testing the interface. The interface should be in a way that it should be able to perform task correctly & it should be able to handle a variety of tasks.

### # Component level design ⇒ It is also component based software engineering. Is an approach to software development that emphasizes the concept of reusable components. CBSE can means higher software quality as well, through the quality of a single component does not necessarily reflect the quality of the system.

Steps ⇒ ① Requirement analysis & specification

② System design, implementation and testing

③ System integration, validation & verification.

## UNIT-5

~~Very important~~

### Testing & its types

PAGE NO.:

DATE: / /

Testing is the process of evaluating a system or its components.

According to ANSI/IEEE 1059 standard, Testing can be defined as "A process of analyzing a software item to detect the difference between existing & required conditions and to evaluate the feature of the software item". Following professionals are involved in testing of a system within their respective capacities.

- ① Software Tester
- ② Software developer
- ③ Project Lead/ Manager
- ④ End User

It is not possible to test the software at any time during its cycle. The next two section state when testing should be started and when to end it during SDLC.

- (1) When to start testing
- (2) When to stop testing.

### Difference between verification & validation

#### Verification

- ① Ensure that software system meet all functionality
- ② Done by developer
- ③ It takes place first & includes the checking for documentation.
- ④ It have static activities
- ⑤ It is an objective process

#### Validation

- ① Ensure that functionality meet the intended behavior
- ② Done by tester
- ③ It occur after verification mainly involve checking of product
- ④ It have dynamic activities
- ⑤ It is an subjective process

## # Difference between Testing, Assurance & Control

Assurance	Control	Testing
① It ensure the implementation of process	① It ensure the verification of developed software.	① It ensure the identification of bugs/error.
② focuses on process & procedures	③ focuses on actual testing by executing software to identify errors.	Focus on actual testing
④ Process oriented activities	④ Product oriented corrective activities	④ Product oriented preventive process
⑤ It's subset of software Test life cycle	⑤ It's subset of quality assurance	⑤ It is subset of QA

## # Difference between Testing & Debugging

Testing ⇒ It involves the identification of bug/error/defect in software without correcting it. Testing is performed in the testing phase. Quality Assurance background are involved.

Debugging ⇒ It involves identifying, isolating & fixing the problem/bug.

Debugging is the part of white box & unit testing. It can be performed in the development phase while conducting unit testing.

## # Testing myths

- (1) Testing is too expensive
- (2) Testing is time consuming
- (3) Testing cannot be started if the product is not fully developed
- (4) Complete Testing is possible.
- (5) If the software is tested then it must be bug free
- (6) missed defect are due to tester.
- (7) Tester should be responsible for quality of product
- (8) Any one can test a software application
- (9) A tester task is only to find bugs.

## Types of Testing ↳

### ① Manual Testing ↳

This type includes the testing of the software manually i.e. without using any automated tool or any script.

In this type the tester takes over the role of an end user & test the software to identify any un-expected behaviour or bug.

It also includes exploratory testing as tester explore the software to identify error in it.

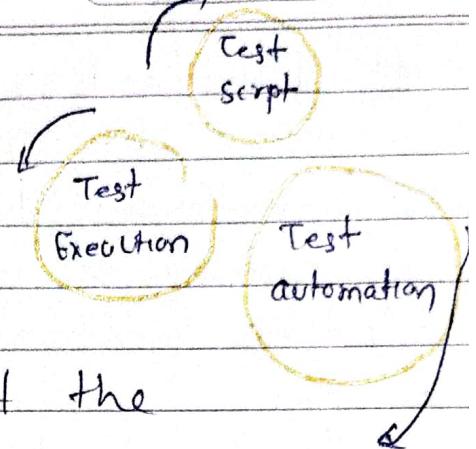
### ② Unit testing

### ③ Integration testing

### ④ System testing & user

### ⑤ Acceptance testing

## ② Automation testing ↗



- Automation testing which is also known as 'Test Automation' when tester writes scripts and uses another software to test the software.
- This process involves automation of a manual process.
- It is used to re-run the test scenarios that performed manually, quickly & repeatedly.
- It is also used to test the application from load, performance & stress point of view.

### ~~Testing method~~ ↗

#### ① Black Box Testing .



internal not known

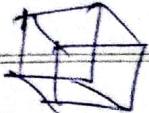
- It is that testing in which the tester does not have knowledge about coding. It does not have knowledge of language in which coding is done.
- We only check the functionality of the software that is input we pass & the output the software produce.

The technique of testing without having any knowledge of the interior working of the application is Black Box testing.

### # Disadvantage ↗

- ① Test cases are difficult to design
- ② Inefficient testing
- ③ Limited coverage. & Blind Coverage.

## # White Box Testing -



fully

In white box testing tester has knowledge of coding & internal structure of the existing software - whether structure is correct or not.

White box testing is also called glass testing or open box testing.

The tester need to possess knowledge of the internal working of code.

# If help in optimizing code.

### Disadvantage :-

- ① Due to the fact that a skilled tester is needed to perform white box testing.
- ② It is difficult to maintain white box testing.
- ③ It is impossible to look into every nook & corner to find out hidden error that may create problem as many paths will go untested.

## # Grey Box testing -



(Partially)

It is a technique to test the application with limited knowledge of internal workings of an application.

In software testing the term, the more you know the better carries a lot of weight when testing an application.

- ① Disadvantage of source code is not available.
- ② Test coverage is limited.
- ③ The test can be redundant.

## ~~3<sup>rd</sup>~~ Level of testing

levels of testing include the different methodologies that can be used while conducting software testing.

### ① Functional testing

### ② Non-functional testing.

#### ① Functional testing →

This is a type of black box testing that is based on the specification of the software that is to be tested.

It is conducted on a complete, integrated system to evaluate the system compliance with its specified requirement.

##### ① UNIT Testing → It is performed by developer

It is used to check independent & individual modules.

The goal of unit testing is to isolate each part of program & show the individual parts or correct in term of requirement & functionality.

##### ② Integration testing →

This testing will integrate all the modules one by one and check the comparatively issue whether one module is compatible with other module.

The testing of combined part of an application to determine if they function correctly together is integrated testing.

##### ① Bottom up integration

##### ② TOP - UP integration.

## # System testing →

In this testing, we put the software in the different environment & check whether the software is compatible with new environment or not mean run software in different operating system.

This is next level in the testing & test the system as whole.

This type of testing is performed by a specialized testing team.

It is first step in software development life cycle, where the application is tested as whole.

## Regression Testing -

Whenever a change in a software application is made it is quite possible that other areas within application have been affected by this change.

To verify that a fixed bug hasn't resulted in another functionality or business rule violation is Regression testing.

- Advantage -
- ① Minimize gap
  - ② Testing new change
  - ③ Mitigate Risks
  - ④ Test coverage
  - ⑤ Increased speed

# Portability testing - It include the testing of software with intend that it should be re-useable and can be moved from another software as well. It can be considered as one of the sub part of system testing.

## # Acceptance Testing

This testing is done by the customers. It is conducted by the Quality Assurance Team who will gauge whether the application meets intended specification & satisfies the client requirement.

By performing acceptance test on application testing team will deduce how application will perform in production.

### (1) Alpha Testing

This test is the first stage of testing & will be performed amongst the teams. This is the last testing done by the developer site just before the delivery of the product in the testing the development feed input measure output for spelling mistake, broken links.

### (2) Beta Testing

In this testing, customer can check the functionality of the software provide suggestion to the developer team.

This testing is performed after Alpha testing. In beta testing a sample of intended audience test the application.

It is also called pre-release testing.

## # Non-functional testing

Non-functional testing of software involves testing the software from the requirement which are non-functional in nature related but important as well such as performance, security, user interface.

## # Performance Testing -

It check the speed & effectiveness of the software to make sure that

- I. It generate the result within the specified time.
- + It is mostly used to identify any bottleneck or performance issue rather than finding bugs.
- I. It has two type load & stress.

I. ① load testing - A process of testing the behaviour of software by applying maximum load in term of software accessing & manipulated large data.

This type of testing identifies the maximum capacity of software & its behavior at peak time.

② stress testing -

In this testing, we check how the system behave under heavy load mean comparatively of software how much load software can handle. It include the testing of software behaviour under abnormal conditions.

## # Usability testing - It ensure that a good & user friendly GUI is designed & is easy to use for end user.

It is a black box technique & is used to identify any error & improvement in software by observing the user through their usage & operation. UI testing can be considered as a sub part of Usability testing.

## # Security testing - It involves the testing of software in order any flaws ad gap from security & vulnerability point of view. It includes -

- ① confidentiality, Integrity, Authentication, Availability.

THANK YOU