

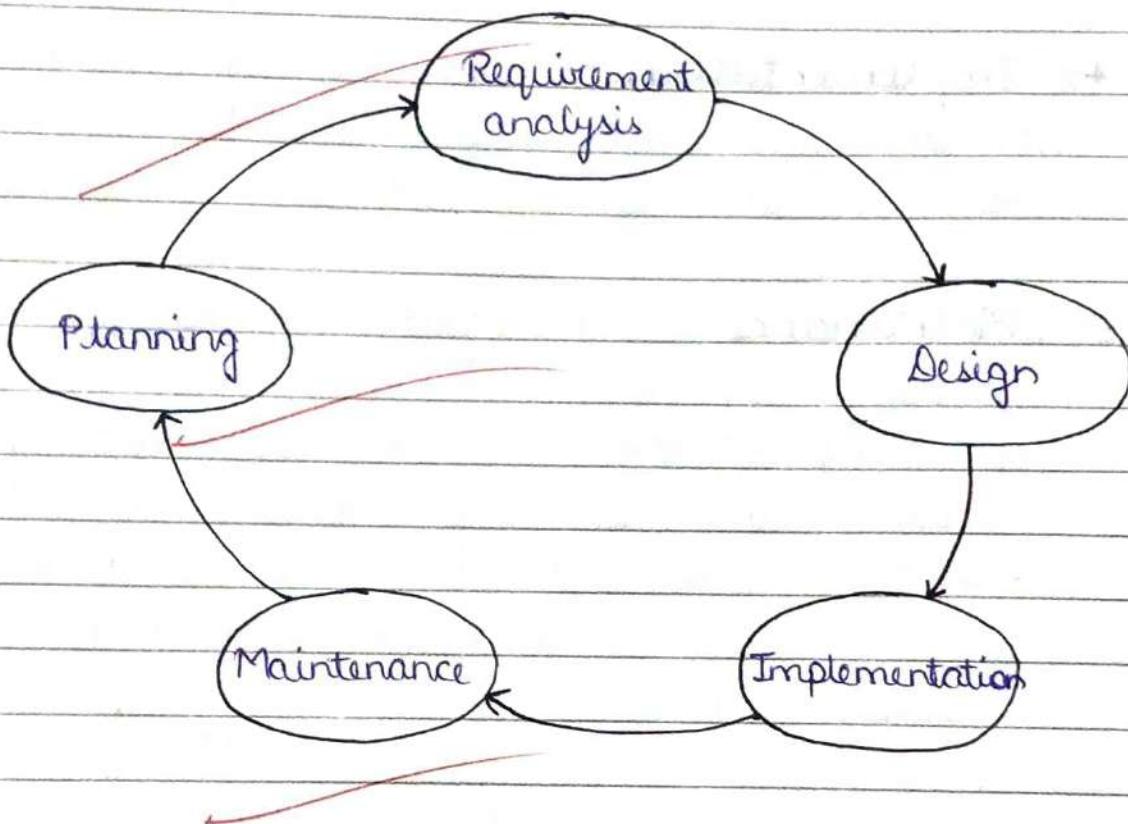
## UNIT - 1

### Overview Of System Analysis And Design

★ SDLC ( System Development Life Cycle ) :-

SDLC is a process used by the software industry to design, develop and test the high quality softwares. The aim of SDLC is to produce a high quality software that meet the expectation and reaches the completion of work within times and done the software cost estimation.

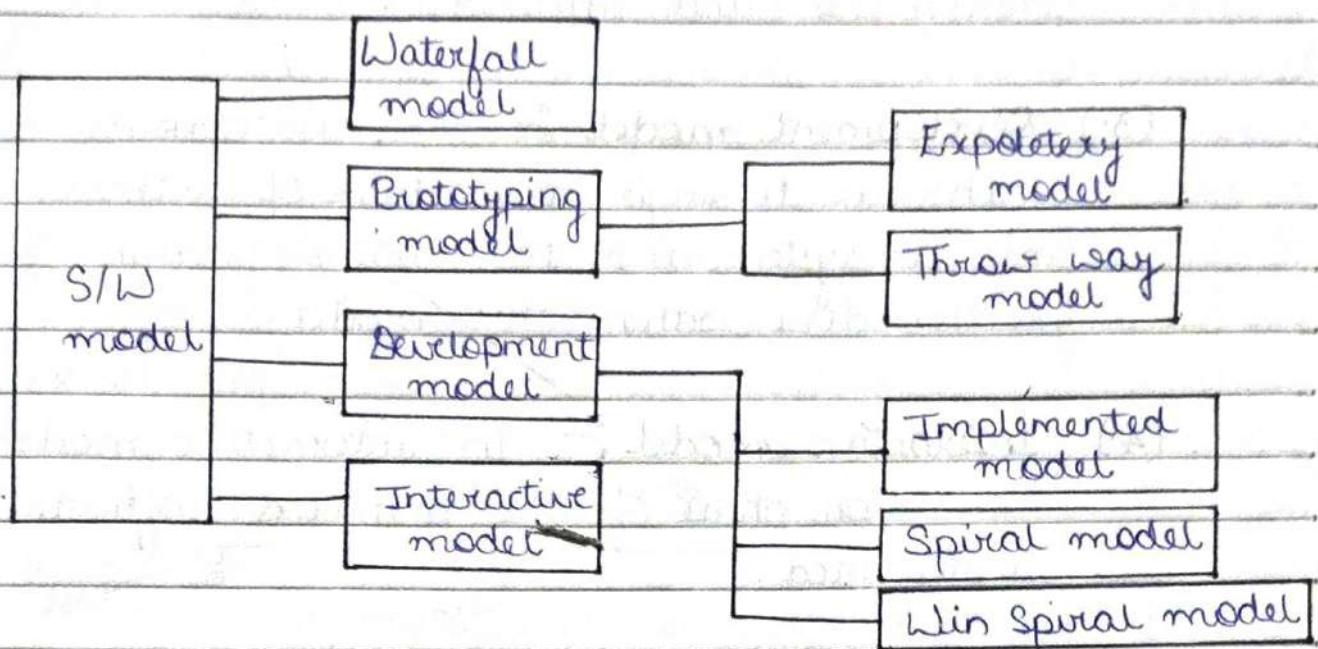
There are following phases of SDLC which is given below -



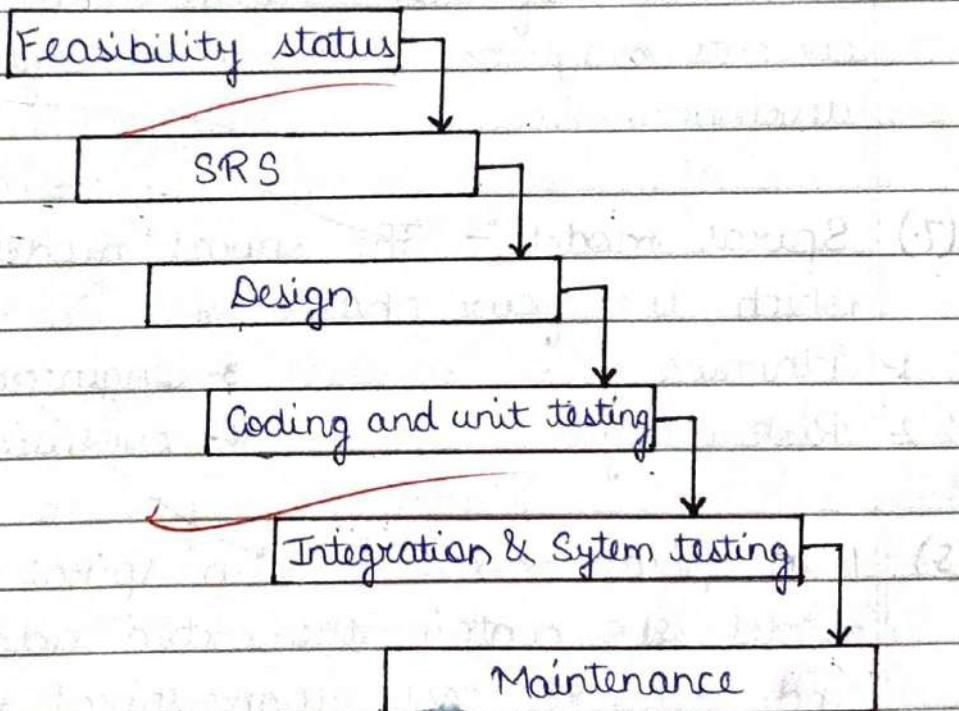
- 1- Planning - The planning of software production can be done by the software developer as per the requirement of client/user. It also plan the cost estimation for software.

- 2- Requirement analysis — The requirement analysis of client software can be made with the help of software engineering terms that is known as feasibility study.
- 3- Design — After gathering all the user information to prepare a client software the web page designer will be prepare a static as well as dynamic web page to improve the productivity and efficiency of software. For each webpages the coding phase will be started by the developer. The designing of webpages can be made with high security measures.
- 4- Implementation — The implementation phase contain the two important factor which is source code and object code of design structure.
- 5- Post implementation &
- 5- Maintenance — The software maintenance will be prepared by software test engineer with the help of specified tools i.e. manual testing and automated testing. If any bug find out during the maintenance of software, the test engineer will fix the bugs and that eliminated by some specific queries.

## \* Software Models :-



(1) Waterfall model :- Waterfall model is a software model in a linear sequential flow. In the waterfall model the outcome of one phase act as an input for the next phase.



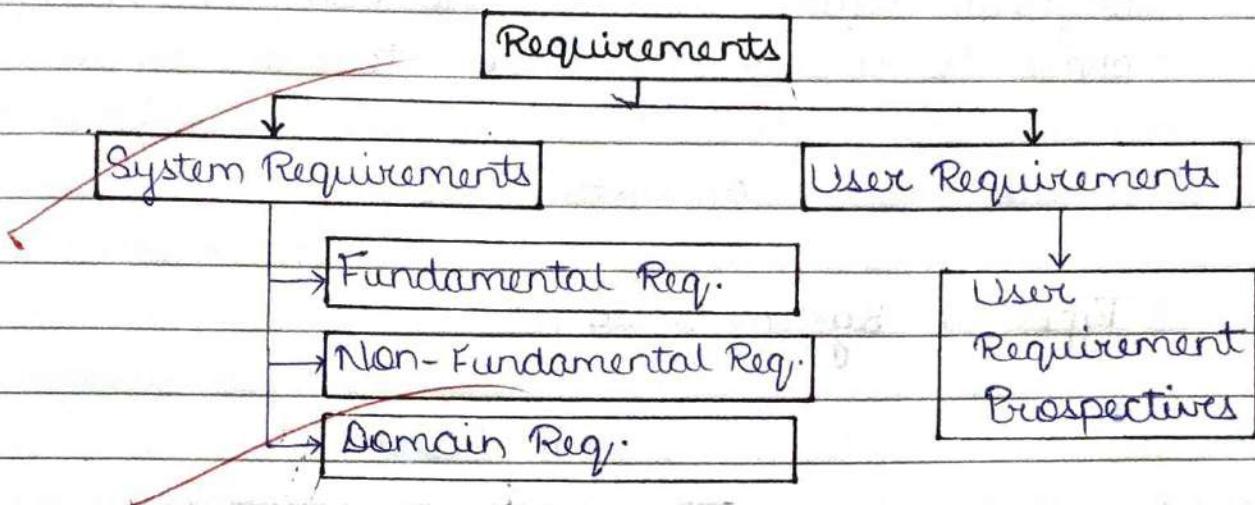
(2) Prototyping model :- Prototyping is the development approach of activities during the software development.

In a prototype model a web software is tested coding wise and unit wise.

- (3.) Development model :- The development model is achieved through application of interactive process of a system and the analysis design of every model done within this model.
- (4.) Interactive model :- In interactive model the exhibition phase of the prepared software will be implemented.
- (5.) Expository model :- In expository model some new contents about software will be searched for the implementation in the coding phase.
- (6.) Throw away model :- It is a temporary storage model of software related contents in which the contents may be used on emergency by a software developer.
- (7.) Spiral model :- The spiral model is a model in which the four phases will be use.
- 1- Planning                                  3- Engineering  
2- Risk analysis                            4- Evolution
- (8.) Win spiral model :- Win spiral model is a model who contain the extra additional features and software tools than spiral models.

\* Requirements :- A requirement is simply the statements of what the system must do and what are the characteristics. During the analysis phase, requirements are written in the form of business environment and also the requirements in the designing phase are written in form by the developer prospectuses.

### Types of Requirements -



\* Requirement determination :- The requirement determination activity is the most difficult part of system analysis. The requirement determination address gathering the documentation to make a system software as per the requirement of client.

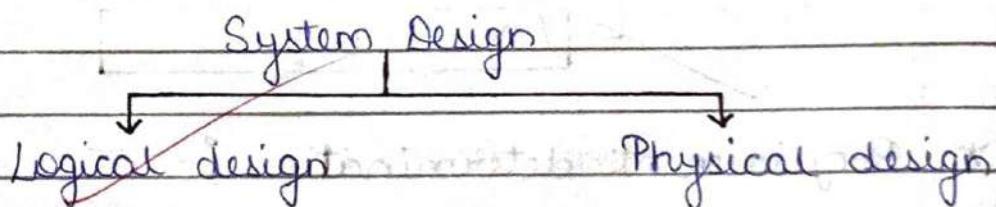
The ~~requirement~~ determination involves and studies the current business system to find out how it works and where improvement should be made. Hence, we can say that requirement determination is the ~~data gathering~~ activities done during the requirement analysis. It contains four sub-activities -

- 1- Requirement anticipation
- 2- Requirement elicitation
- 3- Requirement assurance
- 4- Requirement specification

\* System Design :- The system design is the process of art of defining the architectures, components, modules, interface and the data for a system to satisfy the specified requirement. The system design phase starts after the requirement analysis will be over.

(The system analyst must understand the logical design of the system before beginning the physical design of any one component). The designing phase is second important step in System Development Life Cycle. The final step in system design is to prepare a system design specification.

### Types of System Design -



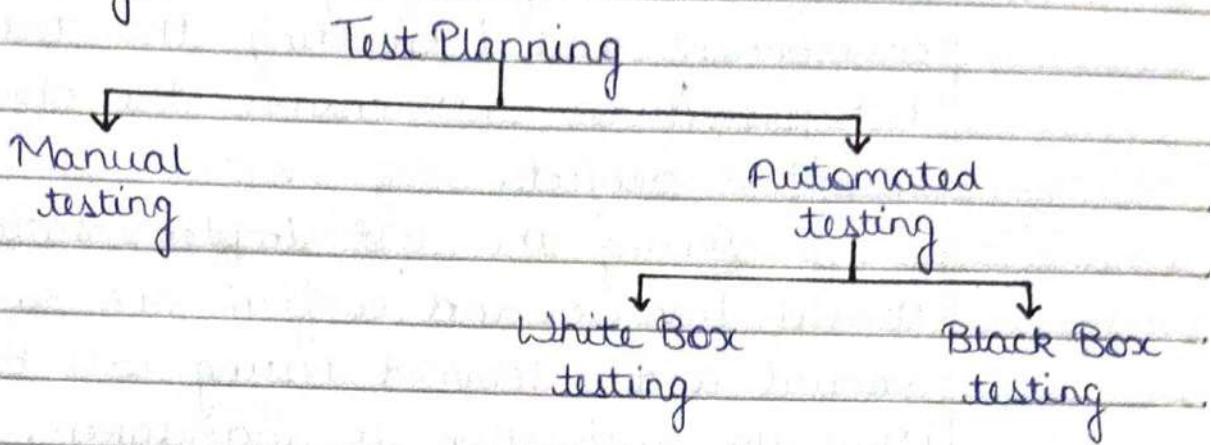
1- Logical design - In the logical design, the programmer and developer implemented their strategies and plans against the designing module. The logical design define the functions and features of system and relationship among its components.

The logical design define what must take place and it include the output that must be produce by the systems and the input needed by the client.

2- Physical design - The physical design is made up on the system logical design and describe the

specific requirement and their implementation.

## \* Test planning :-



The system testing is considered as a part of implementation process. Using the test data, the testing can be done by two types -

- 1- Program test (Automated testing)
- 2- System test (Manual testing)

1- Program test - When the program have been coded, compiled, they must be individually tested with the prepared test cases at any undesirable error, finding out and fix the bugs.

2- System test - After carrying out the program test for each of the program of the system and errors <sup>are</sup> removed then system test is done. (The test planning involve the scheduling and estimating the system testing process and establishing the standard and describe the test that should be carried out.

The test planning is particularly important in large software development as well as small software development processes.

✓ V good  
W  
21/10/20

★

Test implementation :- Test implementation includes and organises the test cases into the test procedures and finalising the test data and testing environment and formatting the test cases. The test procedures will ensure the objective of identified project.

During the test implementation, the tester should finalise and confirm the order in which manual and automated testing will be take place.) When the automation is undertaken, the test implementation include the creation of test procedure and testing script. The tester should carefully check the mapping constraints that might required to be run in a particular order.)

The dependency on the test environment or the test data must be known and checked by a developer. During the test implementation, the tester must ensure that both responsible for the creation and maintenance of the testing environment and all the test supporting tools and associated procedures are ready for use.

This include the configuration management, test logging and their management. In addition, the tester must verify the procedures that gather the information, data and the test result reporting to the senior software test engineer. In this case, some percentage of test execution efforts is allocated to the testing team which does not follow the predetermined script.

\* Performance Evaluation:- Evaluating a system include the hardware and software as a unit. Hardware and software selection require an analysis of several performance categories -

- 1- System requirements - When the system will be available for hardware and software installation for the purpose of analyse the specific requirement by the development team.
- 2- Cost - In this, cost estimation, the maintenance and the operational cost will be identified.
- 3- Performance - What are the capacity and throughput of the system.
- 4- Usability - How easy is to program modifying and operate the system.

For the software evaluation, the following factors are to be considered -

- (a) Programming language and its suitability for the application.
- (b) Easy of installation and training.

In addition to hardware/ software evaluation, the quality of vendor services should be examined.

\* Interviewing:- Interviews are the commonly used and most popular method for requirement determination. The bidding analyst uses the interview skills to collect information from the individuals and from the group.

In this method, the analyst and the engineer of requirement engineering process discuss our organisation specification to the different clients to obtain the different projects.

### \* Group Dynamics / Facilitating Group :-

While working in a joint application development (JAD), a system analyst and the bidding analyst required these skills. In this approach, there is a group of members and system analyst work with that group during the software development.) These analyst can be asked to work and their progress, and it also conduct the design review about the running project.

There are following guidelines which is given below:-

- 1-The purpose of project development should be clear.

- 2-Make sure that the group members understand what is the expectation of our client.
- 3-Be a good listener.
- 4-Reward the group members for their participation with thanks and respect.

### \* Presentation Skills for project development :-

At many points during the system development, the project analyst (bidding analyst) makes the documentation and the progress report of the project and communicate that with the group members. This communication takes the following forms-

- 1. Meeting agenda

2. Minutes of meeting
3. Project summary
4. Project schedules
5. Description
6. Request for new proposal from the new vendors and new contractors.

### \* Structure Walkthrough :-

It is the review of the findings from investigation and of the model which is designed by the development team because the bidding analyst have the review process into a set of procedures.

The objective of the structure walk through is to find errors or problems. Its purpose is to ensure that the model which is developed by development team is correct. The fundamental concept is one of to making the documentation of gathered requirement.

A structure walkthrough is not a performance review, the manager should be involve in this structure analysis basically the review of a bidding analyst work not for any other company person.

One of the major responsibilities of the project manager is to ensure the quality of the final system, offer by the project manager to the bidding team. To complete the work within specific deadline of the project completion, structure walkthrough can be perform to validate the gathered information and give the project summary to the every member of the team.

### Process of Structure Walkthrough

There are following points which are required to complete the process of structure walkthrough:

1. For verification
2. For validation
3. Preparation
4. Execution of project
5. Follow-up

\* Prototyping :- To prepare a working model of the system in order to find out that how the system is working and what are its prototyping.

The traditional approach sometime may take years to analyse, design and implement a system. In order to avoid such delay, the organisation are using prototyping technique to develop smaller system such as decision support system, management information system and expert system.

The goal of prototyping approach is to develop an extended version of the previous prototype. A prototype is a usable system or system component that is prepared quickly at lower cost.

When a prototype is developed, that satisfy all the user requirement either it is refined and turned into the final system.

Experimenting with a prototype help user to identify additional requirement. Prototyping can also be used in the development of transaction processing system. It is most commonly used during the system design.

For example - To develop the mock screen for input.

DATE : / 20
PAGE NO

### Advantages of Prototyping -

The prototyping requires the involvement by the system user. System user experiment with each version of the prototype through an interactive process. Hence the error in the project development are easily detected and eliminated.

### Disadvantages of Prototyping -

The user may not be able for willing to spend the amount of time required under the prototyping approach. Prototyping may have behaviour problem with system user, these problems include the dissatisfaction by the user if the system developers are unable to meet all the user demand for improvement in the project bidding.

\* Database design :- During the actual design, of the database these details and features are evaluated with the different phase of database design which are involved during the design of any database.

There are different phases of database design which is given below -

1- Requirement gathering - It is the process of conducting meetings and interviews with customer, end user, other individual in the company. For that the following information will be required -

- (a) Business rules and process
- (b) Information about the current database.
- (c) Future need of the business as related to the database.

2- Data modelling - It is the process of representing the data for business and prepare the business model to connect the business model with any database management system. The following are the few model is to be require -

- (a) Hierarchical model
- (b) Network model
- (c) Object-oriented model

3- Database design with normalisation - It is a phase in which the business model is converted into the physical model. The normalisation with the tables performed by the different database is -

- (a) Primary key
- (b) Foreign key
- (c) Candidate key
- (d) Super key etc.

4- Database schema's - The database provide the schema for the overall designing of the database in the structural manner. There are following types of schemas available with database which is given below

- (a) Conceptual schema (Logical schema)
- (b) Physical schemas
- (c) Temporary schemas

\* Software Quality :- Software quality is the degree to which a system component or process specifies requirement and meets customer or user needs or expectation.

The quality of the software is accessed by a number of variables. These variables can be divided into two criteria -

1. External quality criteria
2. Internal quality criteria

1- External quality criteria - External quality is what a user experience when running the software in its operational mode. Some of external quality are

- (a) Features
- (b) Speed
- (c) Space
- (d) Network uses
- (e) Robustness
- (f) Security
- (g) Power consumption

Stability

End-of-use

Back compatibility

2- Internal quality criteria - Internal quality refers to aspect that are code dependent that are not visible to the end user. The external quality is critical

to the user while internal quality is meaningful to the developer only.

Some of the internal qualities are -

- (a) Test coverage
- (b) Portability
- (c) Testing ability
- (d) Threat-determination
- (e) Documentation
- (f) Skill ability
- Scalability

#### \* Software evaluation and package :-

Package is a software rather than application program specifically written for an installation is becoming more readily available and certainly comfortable for use.

The evaluation of software can be done with the help of software and hardware cost. Specifically there are six categories on behalf of which a software may be graded -

Performance

1. Effectiveness

2. Performance efficiency

3. Easy to use

4. Flexibility

5. Quality of documentation

6. Manufacturer support

✓ good

✓ 12/10/2020

## UNIT - 2

### Information Requirement Analysis

#### \* Requirement Analysis :-

Requirement analysis is very important and essential activity after gathering all the requirement from the client. This activity review all the requirements and may provide a graphical view of the entire system.

Requirement analysis is done to obtain a clear understanding of the need of the client and the user. All the requirement have been specified at the proper level of abstraction.

#### • Problem during the analysis -

1- Completeness and consistency - One major problem is how to organise the information obtained so the information can be effectively evaluated for completeness and consistency.

2 Contradictions - Another major problem is resolving the contradiction that exist in the information from the different parties.

\* Requirement problems - Most important requirement problems that the analyst has to identify and eliminate in the problem. Once the analyst detect any inconsistency and incompleteness in the gathered requirement, the requirement analysis and the negotiation process goes under the process.

An incomplete requirement is one where some of the requirement have been overlooked.

### \* Requirement Analysis Principle :-

It categorized in four properties which is given below -

1. Model of data domain
2. Model function
3. Model behaviour
4. Partition the model
5. Model absence

### \* Information Modelling :-

Information modelling is a representation of concept relationship, constraints, rules and operations (to specify data semantics). It can provide the sharable, stable and organised structure of information requirement. Once all the requirements have been gathered and analysed, any computer base system can be model these requirements.

Using a representation of input processing output user-interface, a software system engineer can create a model of system components.

#### • Types of information models :-

Information model can be divided into following two categories -

- Logical models

- Physical models

Logical models - The logical model shows that

what a system can do any task and which type of logical and mathematical application will be involved. The logical model include essential models, conceptual model and business model.

2- Physical model - The physical model show not only what a system is do mathematically and logically but it checked how the system is physically and technically implemented. They are implementation dependent because they reflect by technology choice

### • Tools for Information Modelling :-

1. Process modelling tools
2. Logic modelling tools
3. Data modelling tools

1- Process modelling tools - Process modelling is a technique for organising and documenting the structure and flow of data through a system process and what procedure to be implemented by a process or developing team.

2- Logic modelling tools - The logical process model are used to document an information system for the implementation of new technology. In the

3- Data modelling tools - In the data process model, it is a process model and the abstract representation of development process. A development process is a set of activities, together with an ordering relationship between the activities. The main tool The main tool in the information

modelling is data flow diagram.

(Ans)  
12/02/20

### \* Data Flow Diagram :-

A DFD is a "graphical representation" of the flow of data through an information system. DFD can also be used for the visualisation of the data processing.

The DFD are commonly used during the problem analysis. The DFD are quite general and are not limited to the problem analysis for software requirement specification. DFD are very useful in understanding a system and can be effectively used during the analysis.

A DFD show the flow of data through a system, it view a system as a function that transform the input into the desired output. Any complex system will not perform this transformation in a single step and data will typically undergo a series of transformation before it become the output.

### • DFD Rules -

No process can have only outputs, if an object has only output then it must be source.

No source and process can have only input. If an object has only input then it must be a sink.

A process has a limited source of output. In AFD, the delete and changes are the updation

process.

- **Developing DFD :-** The following steps are used to develop a DFD using a top-down approach.

**Step 1 -** Make a list of business activities to use it to determine the various properties.

- External entities
- Data flow
- Process
- Data store

**Step 2 -** Create a context diagram which show external entities and data flow to form the system. Do not show any detailed process or data stored.

**Step 3 -** Check for error and make sure the labels are assigned to each process and data flow are meaningful.

**Step 4 -** Develop a physical data flow diagram from the logical data flow diagram.

**Step 5 -** Partition the physical data flow diagram by separating or grouping parts of the diagram in order to facilitate programming and implementation.

- **Advantages of DFD :-**

1- **Early implementation -** Freedom from committing to the technical implementation of the system is to early.

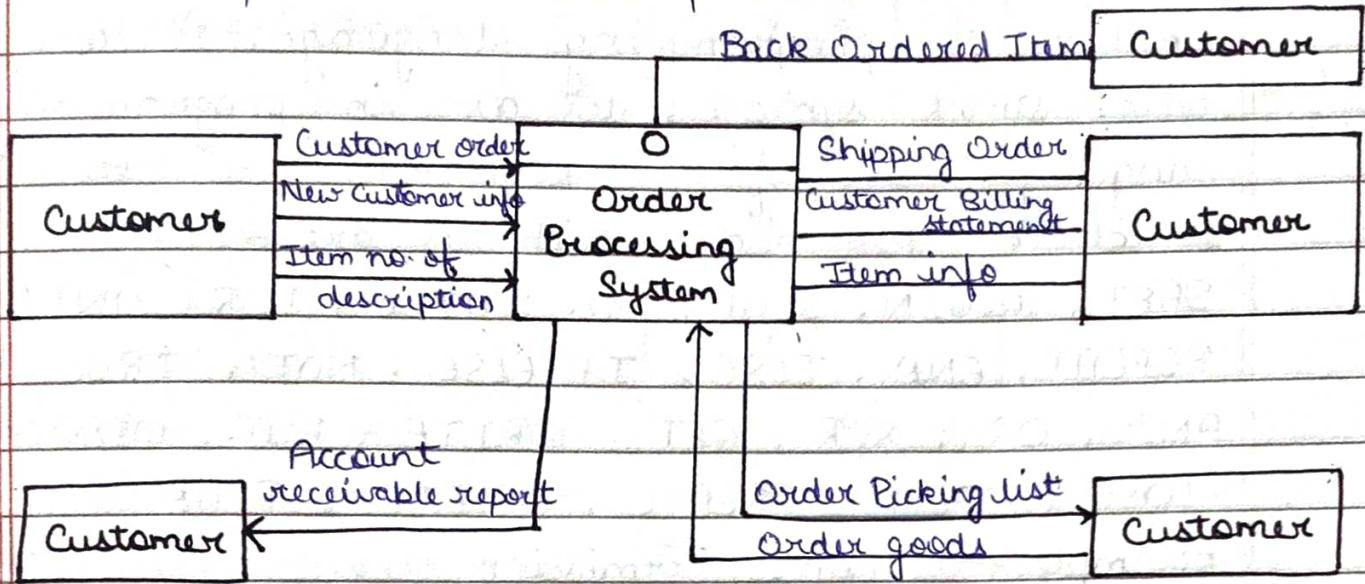
- 2- Study independence - Further understanding of the whole process of the system and its sub-system.
- 3- Analysis - Analysis of a proposed system to determine if the necessary data and the process have been defined.
- 4- Tool for communication - It can be used as a tool to interact with user. An interesting use of DFD is to show them to user and the system analyst. The system analyst check the accuracy of the DFD.
- 5- Reduced cost - They allow the system analyst to describe each component used in the DFD. The analysis can then be performed to ensure that all necessary output may be obtained from the input data and that processing the logic reflected in the diagram.  
Detecting and correcting errors and design the error diagram from input data.

### • Disadvantages of DFD -

- 1- During the implementation of development module within the DFD there are no chance to remove the error from the running stage.  
For Example - Some input information is missing or is incorrect then it is not possible to overcome by the system analyst.
- 2- A DFD model does not specify the order in

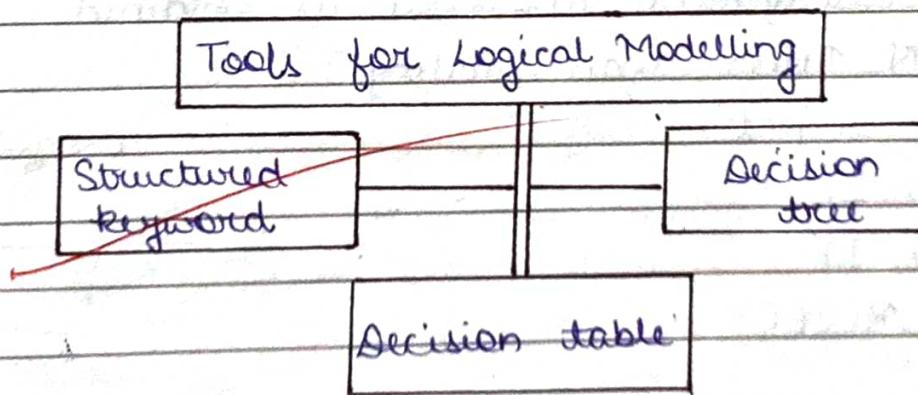
which the different process is to be executed.

- 3- Decomposition methodology to arrive at the successive level of DFD. Due to this reason several alternative DFD representation are possible.



### Zero- Level DFD

### \* Logical Modelling Tools :-



DFA through our versatile tool but not provide 100% accurate result for their user and for modelling all the complexities of the system.

Logic modelling involve representing the internal structure and the functionality of

the process in the AFD.

- Structured keyword - When the process logic involves formulas or iteration, structured decisions are not complex. The structured keyword is similar to programming language. It does not have strict syntax rules as in programming language. Some of the keywords are used in structured keywords which are -  
 START, BEGIN, STOP, DO, WHILE, FOR, UNTIL, REPEAT, END, ELSE, IF ELSE, NOT, TRUE, FALSE, AND, OR, XOR, GET, WRITE, PUT, UPDATE, CLOSE, OPEN, CREATE, DELETE, READ.

Example - Loan Permission

IF customer has an Bank account

THEN

IF Customer has no dues from previous account

THEN Allow loan facility

ELSE

IF Management approval is obtained

THEN Allow loan facility

ELSE reject

END IF

END IF

ELSE REJECT

- Decision Table - Decision table is a matrix representation of the logic of a decision which specify the possible condition and resulting the action. The decision tables are a convenient way to organise information in a systematic manner. The major drawback of a decision tree is

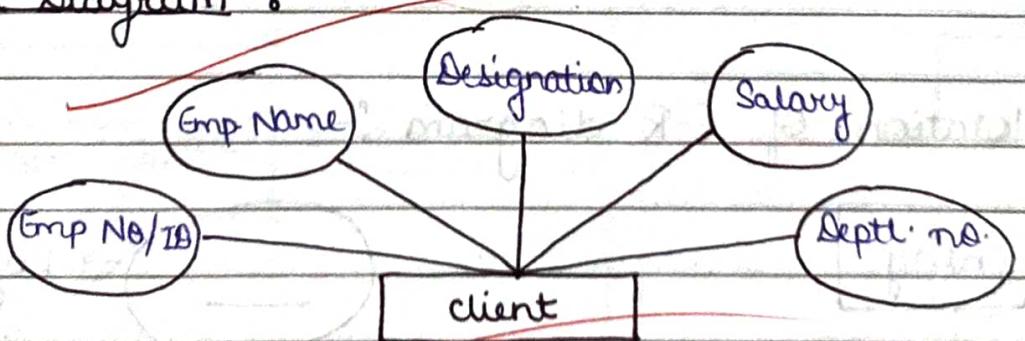
the lack of information in its unit gives format.  
A decision table is a single representation of the relationship between condition and action.

Carriage Evaluation	1	2	3	4
Pass exam	Y	Y	N	N
Complete course work	Y	N	Y	N
Pass	X			
Refer		X	X	
Fail				X

- Decision tree—

Decision tree is a graphic representation of a decision process. The decision making involves several stages and at every stage each of the choice result in a different outcome. In other words, a decision tree is a predictive model hence the decision tree are simple to use for knowledge representation for a user and developer of application program.

### \* E-R Diagram :-



The basic E-R diagram was introduced by Peter for study the large database. E-R diagram or data schema is another means of capturing the data for their organisation. The ERD capture the entities,

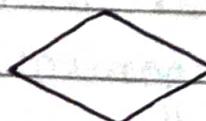
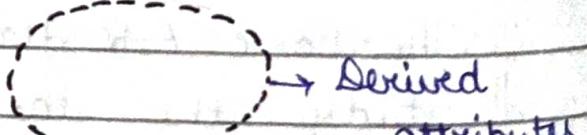
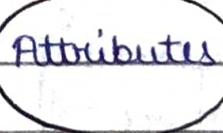
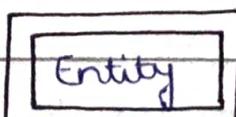
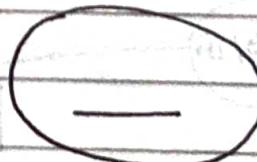
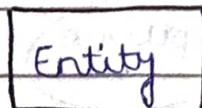
their attributes and the relationship between the entities.

ERD represent the relationship of different databases with the help of different E-R diagram notations.

- Objectives of E-R D :- Benefits

1. Straight-forward relational representation — Having designed an E-R diagram for a database application, the relational representation of the database model becomes relatively straight-forward.
2. Easy conversion of E-R diagram to other data model — The conversion of E-R diagram to a network or hierarchical model can be done easily.
3. Graphical representation for better understanding — A E-R model gives graphic and diagrammatical representation of various entities, its attributes and relationship between entities.

- Notation of E-R diagram :-



→ Relationship

From the above E-R diagram, it is clear that these notations can make the databases strong because of multiple assumption with multiple notation will be encapsulated. The relationship notation diagram show the database mapping constraints which is given below.

- 1- One to one <sup>(1:1)</sup> mapping constraints
- 2- 1:M One to many mapping constraints
- 3- M:M Many to many mapping constraints
- 4- M:1 Many to one mapping constraints

✓ Ngeo

M  
05/03/20

## \* Feasibility Study :-

### Types of Feasibility study

Technical feasibility	Economic feasibility
Operational feasibility	Political feasibility
Legal feasibility	

\* The base of requirement engineering is feasibility study. (This activity is performed to support the decision of whether a new development should be started or not.)

The feasibility is the analysis of risk caused and benefit relating to the economic technology and user. The problem to be automated is analysed in sufficient detail to ensure that all aspects of feasibility are evaluated.

(Its goal is to provide a feasibility study document which present different scenario and alternative solution to the software engineer. The feasibility study offers to resolve the decision for the software development. After the feasibility study, the project management team not be ensure for further decision during the software preparation.)

(The feasibility study tries to anticipate future scenario of software development. Its result is a document which is known as feasibility study report.)

- a) Operational feasibility — Will the solution fulfill the user's requirement.
- b) Technical feasibility — Is the solution technically practical?
- c) Economic feasibility — Is the solution cost effective?
- ) Schedule feasibility — Can the solution be designed and implemented within an acceptable time period?

## \* Cost estimation :-

The accuracy of cost estimation result is important in any software development project. The information system defined as interaction between people, process, data and technology. The important question in software development project is how to complete a project in specific time, budget and resources. In order to measure these attributes are achieved when a person who involved in estimation process especially project manager need to ensure all the requirements are considered when defined added project schedule is one of the main contributors to the project failure.

The accuracy of cost estimation is depending upon how software development communities defined the resources needed and the quantity of resources.

Software cost estimation process begins during the planning phase in SDLC.

Cost of project is derived from the estimate of efforts and size but from the other parameters such as hardware, travel, expense, training and tele-communication cost.

For example - In developing a personal website in order to develop the module, project manager needs to list the type of hardware and software.

In this case, the web developer or web designer or web programmer the overall cost of these above mentioned employees estimate the cost of salaries of these employees.

## \* System Design:-

System design is the transformation of an analysis model into a system design model during the system design, developer define the design goals of the projects and decompose a system into smaller sub-system that can be realised by individual teams.

Developers select the strategy for building the system such as hardware software strategies. The result of system design is a model that include sub-system decomposition and a clear description of each of these strategies.

(System design is the process or art of defining the architecture, components, modules, interface and data for a system to satisfy the specified requirement.) The purpose of system design is to create a technical solution that satisfy the functional requirement for the system.

(The challenge is to translate all these information into technical specification that accurately describe the design of the system and that can be used as input to system construction.)

## \* Design of output and control in system design:-

Output generally refers to the result and information that are generated by the system for many end users. Output is the main reason for developing the system and the

basis on which they will evaluate of the multiple application. Most end-users will not actually operate the information system through the different workstation (server) but they will use the output from the system.

The arrangement of information on a display is termed as layout. (The output design is specified on layout forms, sheets that describe the characteristics of display page such as length, type and format of sheet.) Computer output is the most important and direct source of information to the user. A major form of output is the hardcopy which is produced by different printers or single printer.

(The output can be produced in a variety of ways such as printing, audio, display screen, electronic output etc. Each technology has different speed and cost and affect the end user differently. The output control our design to ensure accurate result and to restrict the distribution of output to authorize recipients.)

There are two major types of output controls are -  
Magnetic tape and disk output control.

- 1- Printed output control such as visual display
- 2- control output.

## \* File Design and Database Design :-

File Design :- A file is a collection of related records generally. All the records in a file are of same size and record types. The record in a file

may be of fixed length or variable length depending upon the size of record contained in a file. File is two-dimensional table summarizing the multiple instances of a set of fields and entity.

The file is based on their multiple attributes which is given below -

- 1- Name
- 2- Identifier
- 3- Type
- 4- Location
- 5- Size
- 6- Protection
- 7- Time, date and user identification

There are multiple method to organise a file.

These methods are given below -

- (a) Sequential organisation
- (b) Direct access organisation
- (c) Index organisation

To find a specific record with file management system. The file should be search with the help of their extension. These methods are useful method for the seperation of files with the different database management system.

### \* Software Constructors :-

The following principles constructor that should guide the software design process with the help of following points which is given below-

- 1- **Modularity and Partitioning** — Each system should consist of a hierarchy of modules. Lower level modules are generally smaller in scope and size compared to high level modules.
  
- 2- **Coupling** — Coupling between two modules is the strength of interconnection between modules or a measure of interdependency among modules. There are following types of coupling which is listed below —
  - (a) Data coupling
  - (b) Stamp coupling
  - (c) Control coupling
  - (d) Common coupling
  - (e) Content coupling
 Because of there are multiple coupling techniques are used for the process software construction and it is the highest form of coupling.
  
- 3- **Cohesion** — (Cohesion is a measure of functional strength of a module). (A cohesive module performs a single task within a software procedure which requiring little interaction with procedures being performed in other parts of a program.)  
 Cohesion of a module represent the tightly bound the internal element of the module are to one or another. (Cohesion of a module gives the designer an idea about the different element of a module belong together in the same module so the cohesion and coupling are clearly related.)
  
- 4- **Span of control** — Span of control refers to the

number of subordinate module which is controlled by a calling module. On the other hand excessive span of control meaning a high number of subordinate module which create a new module under the certain conditions and establishing the calling sequence to pass data and receive results.

- 5- Size — The number of instructions contain in a module should be limited so that module size is generally small. Some organisation have establish the rules to manage the module size.
- 6- Shared use — Sharing modules minimize the amount of software that must be designed and written. It minimize the number of changes that must be made during the system maintenance and it have a single shared modules that reduce the chance of errors.

### \* Documentation —

Documentation describe an information system and help the user, project manager and IT staff who must interact with it. The accurate documentation can be reduce the system down-time, cut the cost and speed-up the maintenance task.

The software documentation is made for the developer of the system and it include all the paper work has done. The documentation is essential for successful system operation and their maintenance. In addition to supporting a

PAGE No

system user, the accurate documentation is essential for IT staff member who must modified the system and add a new feature in it.

- Importance of Documentation -

1. Communication
2. Maintenance
3. Trouble shooting
4. Decision-making
5. Control management
6. Evaluation

## UNIT - 4

### Application Development Methodology and Case Tools

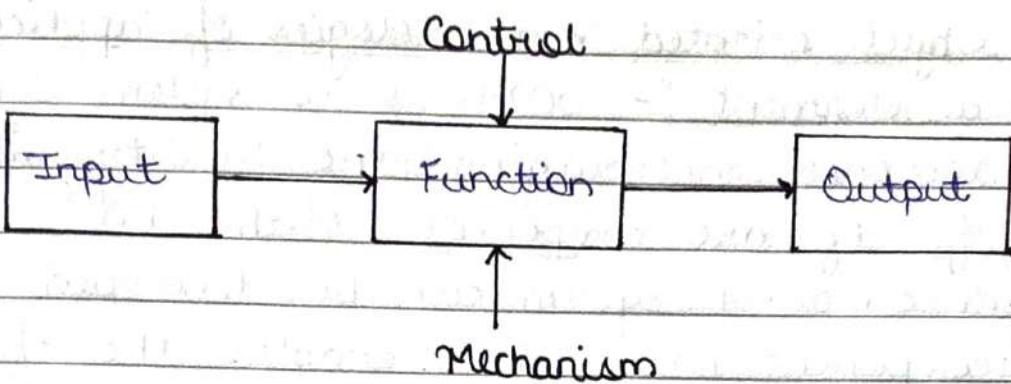
\* Information Engineering :- Information engineering is the engineering discipline that deals with the generation and distribution analysis and used of information data and knowledge in the system. The component of information engineering include more theoretical such as machine learning, AI, CT and signal processing and many more applied field such as natural language, bio medical images computing robotics. The fields of the information engineering is based heavily on mathematics such as probability calculus, algebra, optimization and complex analysis.

#### Tools of Information Engineering -

In the past, some area in information engineering such as signal processing used for Analog engineering, but nowadays most information engineering is done with digital computer. Many tasks in the information engineering can be parallelized and now a days the information engineering is carried out using CPU's and AI accelerators. There are also used quantum computers.

\* Structured Analysis and System design in information engineering :- SAFT is a system engineering and software engineering methodology for describing system as a hierarchy system. It is structured analysis modelling language which uses the two types of the diagram -

- 1- Activity models and
- 2- Data models



It is a system engineering and software engineering methodology for describing system as a hierarchy of function. SAFT is a modelling language. It was developed in 1960. The SAFT's representation is the following -

- 1- A main box where the name of process or the action is specified.
- 2- On the left hand side of this box, incoming arrow that is input and output.
- 3- On the upper part, the incoming arrows - means used the action.
- 4- Authors or developer of SAFT models.
- 5- Comments who review the author work.
- 6- Readers or users of SAFT models.
- 7- Expert, who can advise the authors.
- 8- Technical committee or reviewers of the SAFT models in details.

- 9- Project manager who governs the system analysis and design.
- 10- Monitor or chief analyst to assist SADT and developers and users.
- 11- Instructor to train SADT developer and user of the data models.

#### \* Object oriented methodologies of application development :-

OODM is a system development approach encouraging the facility of reuse of software component. With this methodology a computer system can be developed on a component basis which enables the effective reuse of existing component and facilitates the sharing of its component by other system.

#### \* Document of OODM — These document provides a brief overview of the ~~OODM~~. It benefits the process and some of the major techniques used in OODM. These are following —

- 1- OODM procedures manual — This document describes the comm<sup>n</sup> process structures and procedures involved in conducting the commercial project.
- 2- OODM documentation manual — This document define the deliverables for the project using the OODM. It describes the purpose content and preparation guidelines for each deliverables.

Object oriented design techniques are

widely accepted due to following reasons -

- (i) Simplicity due to abstraction.
- (ii) Easily decomposed into sub-problem.
- (iii) Easily maintenance.
- (iv) Reuse of code and design.
- (v) Improvement in the productivity.

OOD is a group of methodologies that sees real world entities as objects and classes.

For example - hospital is the real entity.

### \* Steps of object oriented system development -

- 1- Define the use case model.
- 2- During the system analysis phase drawing UML syntax.
- 3- Counting in the analysis phase develop class diagram.
- 4- Still in the analysis draw state diagram.
- 5- Develop and document the system.
- 6- Begin system design by modifying the UML diagram

### • Characteristics of Object Oriented System :-

The basic characteristics of OOD are classes, object, method and message.

- 1- Dynamic binding refers to the ability of an object oriented system to differ the data types of object until run time.
- 2- Polymorphism means having the ability to take to take on the multiple form.

### • Key activities in an object oriented design process :-

There are five key activities in an object oriented design -

Design the system architecture.

Identify the principal objects

Develop the design model

Specify the triggers interface.

### \* Data modelling :-

(Data modelling is the process of creating a data model for the data to be stored in the database.) This data model is a conceptual representation of data objects. The association between different data objects and their rules.

Data modelling helps in the visual representation of data and enforces the business rules regulatory companies and the government policies on the data. Data models ensure consistency and in naming conventions, default values and security while ensuring quality of data.

Data model emphasis on what data is needed and how it should be organised instead of what operations need to be performed on data. Data model is like architecture of building plan which help to build a conceptual model and set the relationship between the data item.

There are two types of data model -

1- Entity Relationship (E-R) model.

2- UML ( Unified Modelling Language )

The primary goal of using data model are -

(a) Ensure that all data objects required by the database are actually represented.

- (b) A data model helps design the database at the conceptual, physical and logical levels.
- (c) Data model structure helps to define the relation tables, primary and foreign key and stored procedure.
- (d) It provides a clear picture of the base data and can be used by the database developer to create a physical database.
- (e) It is also helpful to identify missing and redundant data.
- (f) Though the initial creation of the data model is labour and time consuming and it makes the IT infrastructure upgrade and maintenance cheaper and faster.

\* Process model :- Process models are processes of the same nature that classified together into a model. Then a process model is a description of process at the type level. The same process model is used repeatedly for development of a many applications and installation.

• Process modelling tools :- Techniques to model business process such as the flowchart, functional flow, block diagram, control flow diagram, PERT diagram bars emerged since the beginning at the 20th century.

Develop a model using a analogy, or abstract representation to describe a scientific principle to design solution, develop and used the model describe or predict the phenomena. Develop a diagram or simple physical prototype to convey a proposed

object tools or process.

- **Advantage of Process Modelling :-** There are hundred of different process model eg - waterfall, fix, spiral, rapid, prototyping model and agile model. All these above model have different characteristics with different software environment.

There are following key advantages of process model is -

1. Quality of conservation
2. Mutual understanding
3. Documentation
4. Transparency
5. Alignment of business processes with business and IT infrastructure.
6. Business process improvement and quality.
7. Standardization of best practices
8. Business system requirements
9. Rapid prototyping united process
10. Extreme language programming environment
11. Visual model are thing like flow chart, picture etc.

#### \* Use of computer aided SW engineering case tools :-

The software engineering case tools are used for developing high quality decision tree and maintainable software. Case software is often to associated with method for the development of the information system together with the automated tool, that can be used in the software development process.

It is the domain of software tools used

To design and implement application. Case tools are similar to and were partially inspired by CAD tools for designing the hardware product.)

1. The case tools are support specific task in the SDLC.
2. It combine the two or more tools focused on a specific part of software life cycle.
3. Software environment combine two or more tools and support the complete software life cycle.
4. Business and analysis modelling, graphical modelling tools. Eg- E-R modelling.
5. Development, design and construction phase of the life cycle, debugging environment like IISE.
6. Verification and validation and construction phase of life cycle debugging environment.
7. Control the check-in and check-out of repository object.
8. Analysing the code for complexity, modularity and performance etc.
9. It manage the project plan, task assignment, scheduling.

Another common way to define case tools to user defined like upper case and lower case tools.

### \* Implementation of Information System :-

System implementation is the process of defining how the information system should be build (physical system design) ensuring that the information system is operational and used, ensuring that the information system meets quality standard.

## • Stages of implementation of Information System :-

The system described in the guide include the following -

1. Exploration

2. Installation

3. Full implementation

4. Initial implementation

5. Expansion and scale up

Each stage has specific step and associated activities to produce a system that is easy to read, code and maintain the following -

1. Factoring decomposition

2. Span of control

3. Reasonable size

4. Coupling - minimize the inter-module dependency

5. How the various program parts/module of an information system are physically organised.

6. How the modules communicate with each other through data couple (data exchange) and flag (control / message).

7. How the modules are related to each other in term of sequence selection and repetition.

## \* Connecting AFD to structure chart :- The following assumption are required for implementation of information system.

1. Locate the central transform / transaction center.

2. Find the co-ordinating module for the top of the chart.

3. Identify the primary input and output data flow.

- 4- Draw a top-level chart.
- 5- Refine the chart until the data origin, system function and output disposition are defined.

Management Info System

\* Implementation of MIS :- The following points are required for the implementation of MIS -

- 1- Implementation plans - It is the series of action oriented step planned for making the implementation smooth.
- 2- Organising the MIS department.
- 3- Selection and procurement of hardware.
- 4- Creating the database.
- 5- Training of users.
- 6- Creating physical infrastructure.
- 7- Transition of new system

Implementation is the process that turns strategies and plans into action in order to accomplish goals. The critical action move strategic plans from a document that sits on the shelf to action that drive business growth.

Implementation is the carrying a unit, execution or practice of a plan, a method or any decision idea, model, specification standard or policy for doing something. Implementation is the action that must follow any in order for something to actually happen.

## Design and Implementation of OO Platform

\* Object Oriented Analysis :- (Object-oriented analysis and design is a technical approach for analysing and designing an application, system or business by applying object-oriented programming as well as using visual modelling throughout the software development process.)

(OOA is the process of planning a system of interacting object for the purpose of solving a software problem. It is the main approach of software design.)

An object contains encapsulated data and procedure, grouped together to represent the entity. An object-oriented program is described by the integration of these objects. (OOD is the discipline of defining the objects and their interaction to solve a problem that was identified and documented during object oriented analysis) OOD is a method of design on comprising the process of object-oriented decomposition and notation for depicting both logical and physical as well as static and dynamic model of the system under the design.

The input for OOD is provided by the output of OOA. It relies that an output Both analysis and design can be performed incrementally. Some typical input for object-oriented design are -

1. Conceptual model
2. Use case
3. System sequence diagram
4. User interface
5. Relational data model
6. Object-relational mapping model
7. Information hiding
8. Interface
9. Polymorphism

### \* Object Modelling Techniques:-

The object modelling techniques is methodologies of OOA, design and implementation that focuses on creating a model of object from the real world and then to use this model to develop object oriented software.

Object modelling techniques (OMT) was developed by James Rumbaugh. It is primarily used by system and software developers to support full life cycle development while targeting object-oriented implementation. OMT has proven itself easy to understand. It is very successfull in many application domain - telecommunication, transportation, compilers etc. The popular object modelling techniques are used in many real world problem.

- Phases of OMT :- The OMT methodology covers the full SDLC. The methodology has following phase-

1. Analysis
2. System design
3. Object design
4. Implementation

- 1- Analysis — Analysis is the first phase of OMT methodology. The aim of analysis phase is to build a model of the real world situation to show its important properties and domain.
- 2- System Design — The system design phase comes after the analysis phase. System design phase determines the overall system architecture using sub-system architecture, concurrent task and data storage. During system design, the high level structure of system is designed.
- 3- Object Design — The object design phase comes after the system design phase is over. Here the implementation phase is described. OOA is fully
- 4- Implementation — Implementation phase of the OMT is a matter of translating the design into a programming language construct. It is the important to have good software engineering practice so that the design phase is smoothly translated into the implementation phase such as —
  - (1) To increase flexibility
  - (2) For the design traceability
  - (3) To increase efficiency

### \* Dynamic Modelling :-

Dynamic system modelling (DSM) is used to describe and predict the interaction over time between multiple component of a phenomenon that it

viewed as a system. It focus on the mechanism of how the component and the system evolve across time. Although DSM is just emerging in the field of communication. DSM has a rich history in many academic field originating in mathematics and physics before its adoption in the life social and the natural science.

"The dynamic model are generally models that contain or depend upon an element of time especially allowing for interaction between variable over the time.

There are many kind of dynamic model -

- 1- Linear model
- 2- Non-linear model

The component of the dynamic model are their states. The dynamic model include the activity diagram, state diagram, sequence diagram.

#### \* Functional object-oriented design :-

Functional programming is a way of thinking about software construction by creating pure function. It avoid concept of shared state mutable data is observed in object oriented programming.

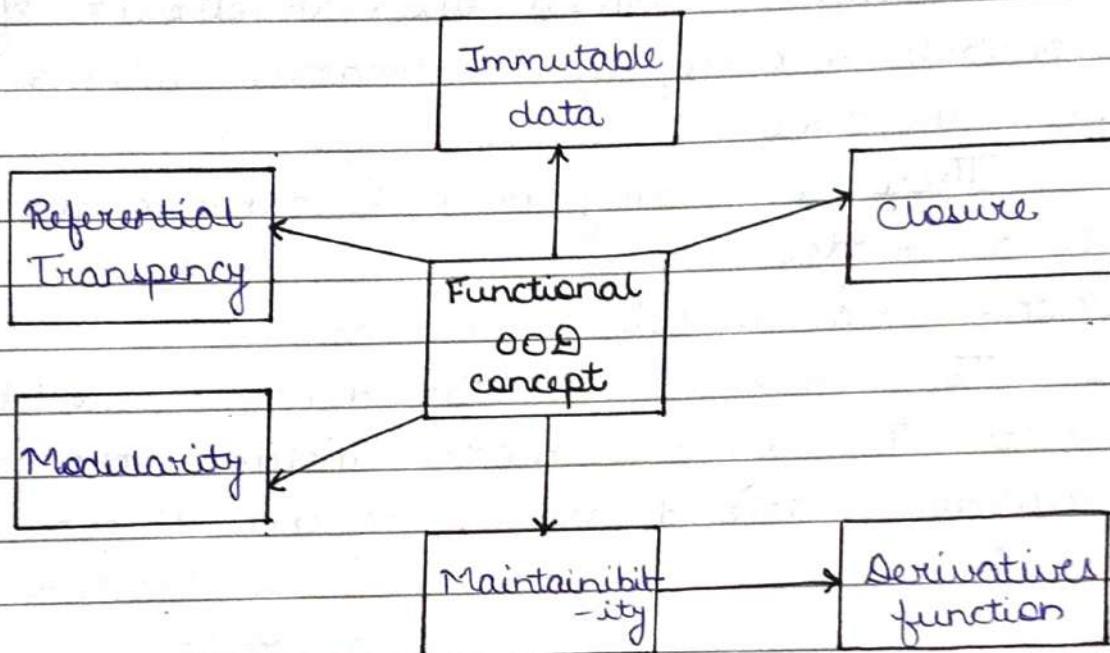
"Functional language emphasizes on expression and declaration rather than execution of statement.

#### \* Characteristics of functional object-oriented design :-

There are following characteristics of functional programming which is given below -

- 1- Functional design focus on result not on process.  
The data is immutable.
- 2- Functional programming decompose the problem into functions.
- 3- It does not support iteration like loop statement and conditional statement.

Block diagram of Functional design :-



\* Difference between object oriented programming and function object-oriented design :-

- OO Programming :- Object-oriented programming (oop) is a programming paradigm based on the concept of "object" which are data structure that contain the data in the field of fields often known as attributes and code in the form of procedure which often known as method.
- Functional OO programming :- Functional programming

is a programming paradigm, a style of building the structure and element of computer program that treat computation as the evaluation of the mathematic function and avoid the changing state and mutable data.

Both OOP and FP have the shared goal of creating understandable flexible program that are free from bugs.

In all program, there are two primary component-

- 1- The data and data behaviours.
- 2- OOPS say that bringing together data and its associated behaviour is the OOD which makes it easier to understand how a program work.

Example -

```
employee = [  
    Employee.new("Bob", 100,000.0),  
    Employee.new("Jane", 125,000.0),  
]
```

#### \* Object oriented programming system for implementation:-

Implementing an object oriented design generally involves using a standard object oriented programming language or mapping object design to database.

#### \* Implementation using programming language :- Usually the task of transforming an object design into code is a standard process. Any object-oriented programming language like C++, Java, C# and python include provision for representing classes.

The following figure show that representation of the class circle using C++.

Circle - X - Coord - Y - Coord # radius  + find Area + find Volume + scale ()	class circle { private: double X - Coord ; double Y - Coord ;  protected: double radius;  public: double find Area(); double find Volume(); void scale(); }
--	--

\* Implementing association of object-oriented programming  
**association :-** Most programming language do not provide construct to implement association directly.

The association may be either unidirectional or bidirectional. Besides each association may be either one to one, one to many, many to many.

\* Implementing Constraints :- Constraints in classes restrict the range and types of value that the attributes may take. In order to implement constraints, a valid default value is assigned to the attributes when an object is initialized from classes.

**Example -** Consider a class employee where age is an attribute that may have value in the range of 18 to 60. The following C++ code incorporate it -

```
class Employee {
private: char* name;
```

```

int age;
// other attributes
public
Employee() {
strcpy(name, " ");
age = 18;
}
class Age_Error {
void change_Age(int a) {
if (a < 18 || a > 60)
throw Age_Error();
age = a;
}
}

```

\* Object-Oriented Databases :- An object database is a database management system in which information is represented in the form of object as used in the form of object as used in OOP. Object database are different form of relational database which are tabulated. It combine the database capabilities with object-oriented programming.

It allow the programmers to develop the product, store them as an object and replicate or modifying existing object to make the new object within OODBMS.

Because the database is integrated with programming language the programmer can maintain the consistency within an environment. The relational DBMS project by that way of contrast maintain a clearer division between the database model and the application.

Some of object-oriented database are designed to work with object-oriented programming languages such as Python, JavaScript, C#, Visual Basic, .Net, C++ etc.

Example -

Object - 1	Instances
Date	01-12-01
Activity code	24
Route No.	J-95
Daily Production	2.5
Hours	6.0
Labour Hours	6.0

Object 2: Maintenance Activity

Activity code	
Activity name	
Production unit	

Most object database also offer some kind of the query language, allowing object to be find using a very declarative programming approach.

Access to data can be faster because an object can be retrieved directly without a search by the following pointers -

Multimedia application are facilitated because the class method associated with the data are responsible for its correct interpretation.

The efficiency of such database is greatly be improved in areas which demand the massive amount of data about one time.

For example - A banking institution could get the user account information and provide them efficiency

with the extensive information such as transaction, account information entities etc.

The object data management group was constituted of object database and relational mapping vendor member of academic community. Its goals were to create a set of specification that would be allowed for portable application that store the object into the Database management system.

## UNIT - 6

### Management Issues in Software Projects

#### \* Introduction of Software market :-

The software market consist of sales of software product by entities companies (organization), sale, traders that develop market and distribute the software product for commercial and personal use. Software product are usually sold in the form of licenses to use the software for specified period or permanently.

The market include the operating system and the productivity software, database, storage and backup software, business analyst and enterprise software, video game, software design and editing software.

#### \* Software Product Market Segmentation :- The global software product is further segmented based on type and geographically.

The software product market is segmented into various type of software product. The global software product is segmented into the North America, Europe, Africa, largest share in global software product market.

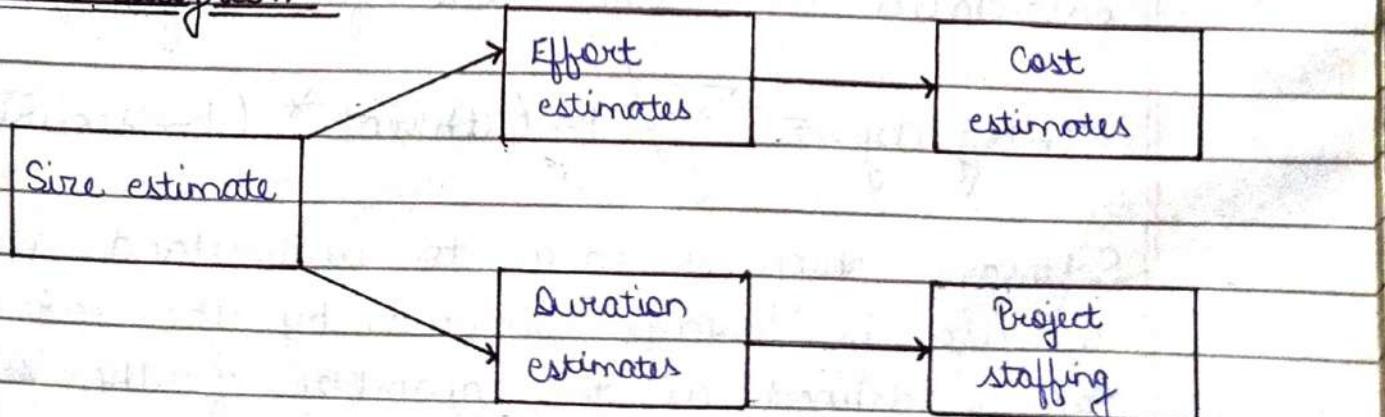
#### \* Trends in Software product market :-

Increase in the number of mergers and acquisition in the software product industry. Introduction of high performance cloud computing are the major trends witness in global software product market.

\* Planning of Software Projects :- Once a project is found to be possible, computer code project managers undertake project designing. The project designing is undertaken and completed even before any development activity start. The project designing consist of subsequent essential activities. The attributes of the project -

- 1- Project Size - What going to be downsize quality in terms of the trouble and time needed to develop the project.
- 2- Cost - What propagation is it reaching to value to develop the project.
- 3- Duration - However long is it reaching to want complete the development.
- 4- Effort - What proportion effort would be required. The effectiveness of the following designing activities is-
  - (a) Planning force and alternative resources
  - (b) Risk identification, analysis and designing
  - (c) Quality assurance planning
  - (d) Configuration management and arrange etc.

Block diagram -



## \* Measurement of Software Quality and Productivity:

The aim of the software developer is to develop high quality software within a specified time and budget. To achieve this, software should be developed according to the functional and performance requirement and document standard. The private matrices are collected by software engineers and then estimated to achieve project level to measure.

The main aim at the project level to measure both the error and defects from software.

Many measures have been proposed for assessing software quality such as functionality and so on.

It has been observed that reliability, integrity and usability are most useful as they provide valuable indicators to the project team, which are given below -

- 1- Reliability
- 2- Correctness
- 3- Maintainability
- 4- Integrity

For measuring integrity of software attributes such as threat and security are to be used.

The threat can be defined as the probability of a particular attack at a given point of time.

Using these two attributes integrity can be calculated by using the following equation -

$$\text{Integrity} = \sum / 1 - (\text{threat} * (1 - \text{security}))$$

Software which is easy to understand and easy to use, is always preferred by the user. Usability can be defined as the capability of the software.

to be understood, learned and used under specified condition. The software which accomplished all the user requirement but is not easy to use it often destined to fail.

### \* Defect Removal Efficiency (DRE) :-

DRE can be defined as the quality matrices. The quality assurance and control activities that are applied throughout software development are responsible for detecting errors. The ability to detect errors is measured with the help of DRE which can be calculated by using following equation -

$$\text{DRE} = \frac{E}{(E + P)}$$

where E = number of error found software is delivered

P = number of defects found after software is to be declared to the user.

If there are no defect in the software, as the value of E increase for a given value of P, The overall value of DRE stored with new development of approach. DRE can also used at different phase of software development.

### \* ISO and capability maturity models for the organisational growth :-

The CMM is methodology used to develop and refine the organisational software development process. The ISO 9000 standard specify an effective quality system for

manufacturing and service industries. ISO 9001 deals with specifically with software development and maintenance.

The CMM was developed and is promoted by the SEI (Software Engineering Institute) which is research and development center. SEI was founded in 1984 to address to fund the software engineering issues.

- CMM five maturity levels of software process:-

- 1- At initial level, processes are disorganized even chaotic and documented to allow them to be replicated.
- 2- At repeatable level, basic project management techniques are established and success could be repeated.
- 3- At define level, an organisation has developed its own standard software process through the documentation and integration.
- 4- At managed level, an organisation monitors and control its own processes through data collection.
- 5- At optimizing level, the processes are to be constantly being improved through monitoring feedback.