

2.1. Definition :-

- ◆ System Development Process is a process of examining a business problem -
 - with the intentions of improving it,
 - through better procedures and methods
- ◆ There are 2 components of system development process:
(i) System Analysis (ii) System Design
- ◆ *System Analysis*: Gathering, processing, and interpreting the facts related to existing system, diagnosing problem to recommend improvement to the system.
- ◆ *System Design*: Process of preparing new business system or improving an existing one.
- ◆ The job of system development is done by ***System Analyst***.

2.2. Why System development fails :-

There are many reasons why organizations fail to achieve their system development objectives. Some of them are as follows -

1. Lack of management support and involvement in system development.
2. Frequently shifting user's need.
3. Development of strategic system.
4. Unfamiliarly of development personnel to new technology.
5. Lack of standard methodology.
6. Over worked and under trained development staff.
7. People's resistance to change.
8. Lack of user's participation in development.
9. Inadequate testing and user training.

2.3. System Development Team -

- ◆ Top management level *Steering committee* consisting of group of key IS users is formed to review system plan. If project is worth while, the responsibility of development of system is on information systems department.
- ◆ A ***Project Management team*** is generally appointed consisting of both computer professional and key user to coordinate development activities of the system.

2.4. Accountant's involvement in development:-

- ◆ Accountant can combine knowledge of IT, business accounting, internal control, behaviour and communication.
- ◆ They can provide assistance in setting up internal controls.
- ◆ They may perform the analysis of proposed system's cost and benefits.

2.5. System Development Methodology:-

SDM, also known as SDLC, is a formalized, standardized and well documented set of activities used to manage a system development project. Characteristics of a good SDM includes -

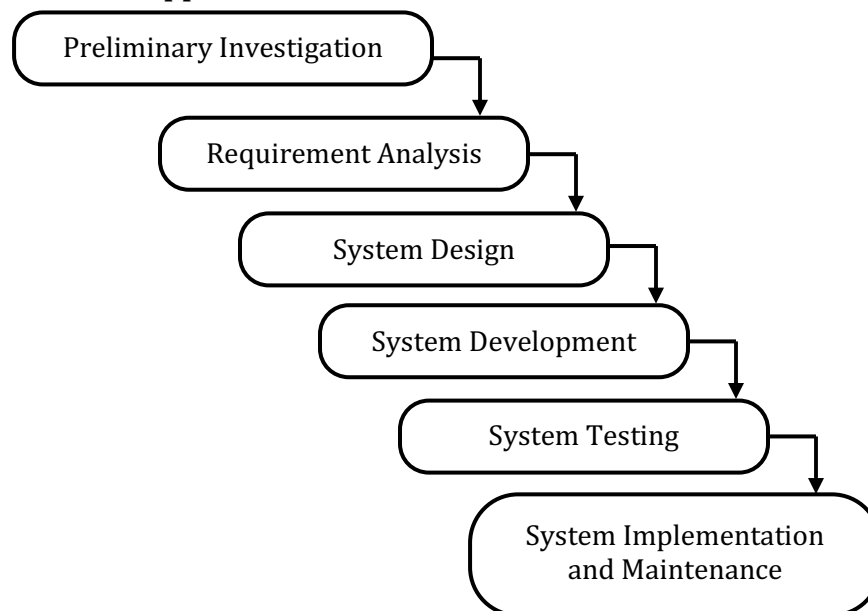
- ◆ SDM visualizes a project as divided into number of identifiable process. Each process comprises several activities, deliverables and management checkpoints.
- ◆ SDM ensures various reports and other documents (Deliverables) are prepared & submitted periodically during system development. These reports make development personnel accountable for faithful execution of system development tasks. Management monitors the development process by reviewing these deliverables.
- ◆ SDM suggests that the users, managers and auditors are required to participate in the project to provide approvals (Sign off).
- ◆ SDM makes sure that thorough system testing is done prior to implementation to ensure that it meets user's need.
- ◆ SDM also checks for an appropriate training program is developed for users.
- ◆ SDM calls for post implementation review to access efficiency & effectiveness.
- ◆ SDM checks that formal program change control are established to preclude unauthorised changes to computer program during system modification.

2.6. Approaches to System development:-

There are several approaches to the system development. Some of them are as follows -

- | | |
|---------------------------------|---------------------------------|
| [1] <i>Waterfall approach</i> | [2] <i>Prototyping approach</i> |
| [3] <i>Incremental approach</i> | [4] <i>Spiral approach</i> |
| [5] <i>RAD approach</i> | [6] <i>Agile Methodologies</i> |

2.7. Waterfall approach :-



2.7.1 Basic Principles

- ◆ Project is divided into sequential phases.
- ◆ Emphasis is on one Planning, Scheduling, Budgeting and Implementation.
- ◆ Tight control is maintained over the life of the project through the use of extensive written documentation.

2.7.2 Strengths

- ◆ Ideal for supporting less experienced project teams and project managers.
- ◆ An orderly sequence of development steps and design reviews help ensure the quality, reliability, adequacy and maintainability of the system.
- ◆ Progress of system development is measurable.
- ◆ Conserves resources.

2.7.3 Weaknesses

- ◆ Inflexible, slow, costly and cumbersome.
- ◆ Little room for use of iteration, which can reduce manageability.
- ◆ Depends upon early identification and specification of requirements.
- ◆ Requirement inconsistencies and unexpected development needs are often discovered during design and coding.
- ◆ Problems are often not discovered until system testing.
- ◆ System performance cannot be tested until the system is almost fully coded.
- ◆ Difficult to respond to changes.
- ◆ Produces excessive documentation which is time-consuming.
- ◆ Technical specifications are often difficult for users to read and thoroughly appreciate.
- ◆ Increases the gap between users and developers with clear vision of responsibility.

2.8. Prototyping Approach

- ◆ In this approach a small version or prototype of the system is prepared and implemented.
- ◆ User while working with prototype make suggestions.
- ◆ Suggestions are incorporated to make it fully operational system.

2.8.1 Basic Principles

1. Identify information system requirement -

- a. Analyst identify only fundamental system requirement to build the initial prototype.
- b. The process of determination could be formal and time consuming.

2. Develop the initial prototype -

- a. Using 4GL tools or CASE tools designers creates a base model.
- b. The goals here *axe-Rapid development. Low cost. Simplicity, Flexibility and Ease of use.*

3. Test and Revise -

- a. Analyst demonstrate the model to user and then give it to them to use.
- b. Users are asks to record their liking and disliking about the system while using it and recommend changes.
- c. This iterative process of "***modification- revaluation***" continues until users are satisfied.

4. Obtain user's approval -

- a. Finally Analyst obtain the formal user's approval of the system.

2.8.2 Strengths

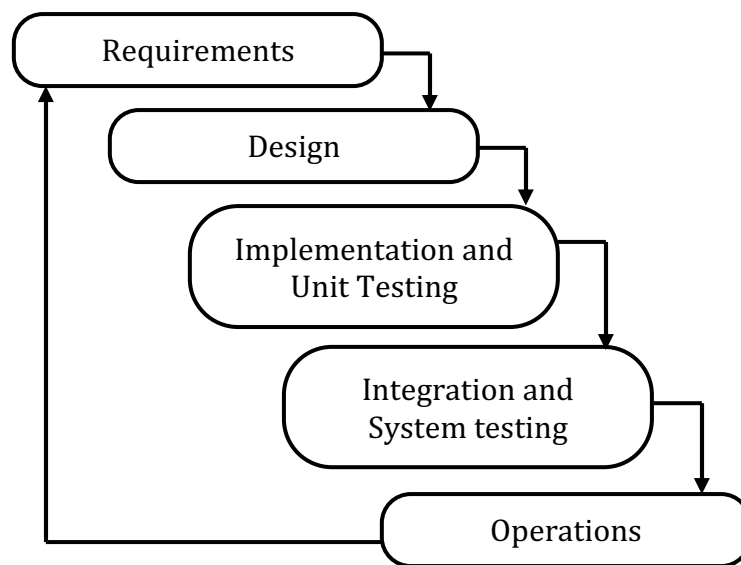
- ◆ Improves user participation in system development.
- ◆ Especially useful for resolving unclear objectives; developing user requirements; or experimenting various design solutions.
- ◆ Potential exists for exploiting knowledge gained in an early iteration.
- ◆ Helps to easily identify difficult functions and missing functionality.
- ◆ Encourages innovation and flexible designs.
- ◆ Provides quick implementation.
- ◆ Prototyping results in a better definition of the users' needs and requirements.
- ◆ Short time period (generally a week) is normally required to develop and start experimenting with a prototype.
- ◆ Errors are detected and eliminated early in the developmental process.

2.8.3 Weaknesses

- ◆ Approval process and control are not strict.
- ◆ Incomplete or inadequate problem analysis may occur, resulting in current inefficient practices being easily built into the new system.
- ◆ Requirements may frequently change significantly.
- ◆ Designers may prototype too quickly, without sufficient up-front user needs analysis.
- ◆ Prototype may not have sufficient checks incorporated.
- ◆ Prototyping can only be successful if the system users are willing to devote significant time in experimenting with prototype.
- ◆ The system developers are frequently, tempted to minimize the testing and documentation process.
- ◆ Prototyping may cause behavioral problems with system users that leads to dissatisfaction.

2.9. The Incremental Model**2.9.1 Basic Principles**

- ◆ In this method of development, a model is designed, implemented and tested incrementally (a little more is added each time) until the product is finished.
- ◆ This method combines the elements of waterfall model and iterative philosophy of prototyping.
- ◆ The product is decomposed, into a number of components, each of which are designed and built separate. Each component is delivered to the client when it is complete. This allows partial utilization of product and avoids a long development time.
- ◆ A series of mini-waterfalls are performed, where all phases of the waterfall development model are completed for a small part of the system, before proceeding to the next increment.

**2.9.2 Strengths**

- ◆ Potential exists for exploiting knowledge gained in an early increment on later increments.
- ◆ Moderate control is maintained over the life of the project through the use of documentation, the formal review and approval / signoff.
- ◆ Stakeholders can be given concrete evidence of project status.
- ◆ More flexible - less costly to change scope and requirements.
- ◆ Helps to mitigate integration and architectural risks.
- ◆ Allows delivery in parts and can go into production more quickly as incremental releases.
- ◆ Gradual implementation provides the ability to monitor the effect of incremental changes.

2.9.3 Weaknesses

- ◆ A lack of overall consideration of the business problem and technical requirements.
- ◆ Each phase of an iteration is rigid and do not overlap each other.
- ◆ Not all requirements are gathered upfront for the entire software life cycle.
- ◆ Since some modules will be completed much earlier than others, well-defined interfaces are required

2.10. Spiral Model

2.10.1 Basic Principles

Also known as the Spiral Life cycle, it is a systems development method which combines the features of the Prototyping model and the Waterfall model. The spiral model is intended for large, expensive and complicated projects.

- ◆ The new system requirements are defined in as much detail as possible by interviewing a number of users.
- ◆ A preliminary design is created for the new system. This phase is the most important part of "Spiral Model" in which all possible alternatives are analysed, that can help in developing a cost effective project.
- ◆ This phase has been added specially in order to identify and resolve all the possible risks in the project and If risks indicate any kind of uncertainty, prototyping may be used to proceed.
- ◆ A first prototype of the new system is constructed from the preliminary design. This represents an approximation of the characteristics of the final product.
- ◆ A second prototype is evolved by a fourfold procedure:
 - evaluating the first prototype in terms of its strengths, weaknesses, and risks;
 - defining the requirements of the second prototype;
 - planning and designing the second prototype;
 - constructing and testing the second prototype.

2.10.2 Strengths

- ◆ Enhance risk avoidance.
- ◆ Useful in helping to select the best methodology to follow based on project risk.
- ◆ Can incorporate Waterfall Prototyping and Incremental methodologies in the framework. For example, a project with low risk of not meeting user requirements but high risk of missing budget or schedule targets would essentially follow a linear Waterfall approach. If the risk factors were reversed, the Spiral methodology could yield an iterative prototyping approach.

2.10.3 Weaknesses

- ◆ Challenges to determine the exact composition of development methodologies.
- ◆ Highly customized to each project, & thus is quite complex, limiting reusability.
- ◆ Skilled and experienced project manager required to determine how to apply it to any project.
- ◆ No established controls for moving from one cycle to another cycle.
- ◆ No firm deadlines.

2.11. Rapid Application Development (RAD)**2.11.1 Basic Principles**

Rapid Application Development (RAD) refers to a type of software development methodology which uses minimal planning in favor of rapid prototyping. The "planning" of software developed using RAD is interleaved with writing the software itself.

- ◆ Key objective is fast development and high quality system at a relatively low investment cost.
- ◆ Breaking a project into smaller segments.
- ◆ Aims to produce high quality systems quickly, through the use of computerized development tools. Graphical User Interface (GUI) builders, Computer Aided Software Engineering (CASE) tools, Database Management Systems (DBMS), Fourth generation languages, etc.
- ◆ Key emphasis is on fulfilling the business need while technological or engineering excellence is of lesser importance.
- ◆ Project control involves prioritizing development and defining delivery deadlines or time boxes. The emphasis is on reducing requirements, not in increasing the deadline.
- ◆ It includes Joint Application Development (JAD), where users are intensely involved in design.
- ◆ Active user involvement is imperative.
- ◆ Produces documentation necessary to facilitate future development and maintenance.

2.11.2 Strengths

- ◆ The operational version of an application is available much faster.
- ◆ This approach tends to produce systems at lower cost.
- ◆ It encourage customer feedback.
- ◆ Users are seen as gaining more of a sense of ownership of a system, while developer are seen as gaining more satisfaction from producing successful systems quickly.
- ◆ Concentrates on essential system elements from user viewpoint.
- ◆ Ability to rapidly change system design as demanded by users.
- ◆ Produces a tighter fit between user requirements and system specifications.
- ◆ Generally produces a dramatic savings in time, money and human effort.

2.11.3 Weaknesses

- ◆ Lower overall system quality.
- ◆ Danger of misalignment between system and the business due to missing information.
- ◆ Project may end up with more requirements than needed.
- ◆ Potential for feature creep where more and more features are added.
- ◆ Potential for inconsistent designs within and across systems.
- ◆ Potential for violation of ramming standards.
- ◆ Lack of re-usability.
- ◆ Lack scalability.
- ◆ Lack of attention to later system administration needs.
- ◆ Formal reviews and audits are more difficult.
- ◆ Tendency for difficult problems to be pushed to the future, to show early success to management.

2.12. Agile Methodologies

- ◆ All the methodologies described before are based on the premise that any software development process should be predictable and repeatable. More emphasis is on following procedures and preparing documentation. They are considered to be heavy weight.
- ◆ Agile methods attempt to minimize risk by developing software in short time boxes called iterations. Each iteration is like a miniature software project of its own, and includes all of the tasks necessary to release the mini-increment of new functionality.
- ◆ Hence the agile methodologies advocate the principle "Build Short Build Often", i.e. the given project is broken up into sub projects and each subproject is developed and integrated in to the already delivered system. The subprojects have short delivery cycles, usually 3 to 4 weeks.
- ◆ Some of the Characteristics of Agile Methodology are as follows:
 - Fast verifications and corrections.
 - Time bound iterative cycles.
 - Modularity at development process level.
 - People oriented.
 - Collaborative and communicative working style.
 - Minimizes risks and facilitates functional additions.
- ◆ Some of the popular agile methodologies are - Scrum, FDD (Feature - Driven Development), Crystal and XP (Extreme Programming).

2.13. System Development Life Cycle (SDLC)

- ◆ The System Development Life Cycle (SDLC) framework provides system designers and developers to follow a sequence of activities. It consists of a set of steps or phases to be followed sequentially to develop a system.
- ◆ The SDLC is document driven methodology where a phase is not complete until the appropriate documentation is produced. These are sometimes referred to as deliverables.
- ◆ The advantages of SDLC are as follows :
 - Better planning and control by project managers.
 - Compliance to prescribed standards ensuring better quality.
 - Documentation is a measure of communication and control.
 - The phases help the project manager and the user for to review & signoff.
- ◆ From the perspective of the IS Audit the following are the possible advantages:
 - The IS auditor can have clear understanding of the various phases because of the detailed documentation.
 - The IS Auditor can report about the compliance, by IS management, of the procedures set by the management.
 - The IS Auditor can be a guide during the various phases of SDLC.
 - The IS auditor can provide an evaluation of the methods used through the various development phases of the SDLC.
- ◆ Risks associated with SDLC
 - The development team may find it cumbersome.
 - The users may find that the end product is not visible for a long time.
 - It may prolong the duration of many projects.
 - It may not be suitable for small and medium sized projects.
- ◆ The SDLC starts when management realizes that a business system needs improvement.
 - 1] Preliminary Investigation
 - 2] System/Requirement Analysis
 - 3] System Design
 - 4] System Development/Programming
 - 5] System Testing
 - 6] System Implementation
 - 7] Post Implementation Review and Maintenance

2.14. The Preliminary Investigation

The purpose of the preliminary investigation is to evaluate the project request. It is carried out by system analyst under the directions of steering committee.

The following issues are typically addressed in the Feasibility Study:

- (i) Determine whether the solution is as per the business strategy.
- (ii) Determine whether the existing system can rectify the situation without a major modification.
- (iii) Define the time frame for which the solution is required.
- (iv) Determine the approximate cost to develop the system.
- (v) Determine whether the vendor product offers a solution to the problem.

2.14.1 Objective/Activities:-

Preliminary investigation must accomplish the objectives -

- ◆ Clarify and understand the project request
- ◆ Determine the size of project
- ◆ Conduct feasibility test
- ◆ Report findings to the management.
- ◆ Assess costs and benefits of approaches.

2.14.2 Identification of problem:-

To understand and define problem clearly several rounds of discussion with the user group are organized. The following methods of investigation are used -

- 1. Review internal document
- 2. Conducting Interviews

2.14.3 Delineation of Scope :-

The scope defines the boundary of the project i.e. what will be addressed by the solution and what will not. The following requirements should be considered while stating the scope:

- (i) Functionality requirements:** What functionalities will be delivered through the solution?
- (ii) Data requirements:** What data is required to achieve these functionalities?
- (iii) Control requirements:** What are the control requirements for this application?
- (iv) Performance requirements:** What level of response time, execution time and throughput is required?
- (v) Constraints requirements:** What are the conditions the input data has to conform to? For example, what is the maximum number of characters that a name can have in a database?

(vi) Interface requirements: Is there any special hardware/software that the application has to interface with? For example – Payroll application may have to capture from the attendance monitoring system that the company has already installed. Then the solution developer has to understand the format of data, frequency mode of data transfer and other aspects of the software.

(vii) Reliability requirements: Reliability of an application is measured by its ability to remain uncorrupted in the face inadvertent/deliberate misuse. The reliability required for an application depends on its criticality and the user profile.

While delineating the scope, few aspects need to be kept in mind, such as:

- ◆ Different users will represent the problem and required solution in different ways. The system developer should confirm the need from the executive sponsor of the project
- ◆ The profile of the actual users may be used in designing appropriate user interface.
- ◆ While presenting the proposed solution for a problem, clearly quantify the economic benefits to the user organization.
- ◆ It is also necessary to understand the impact of the solution on the organization.
- ◆ Besides economic benefit, there are several other factors, such as security, have to be given weight age too.

2.14.4 Testing Project's Feasibility :-

- ◆ After possible solution option are identified, project feasibility i.e. the likelihood that the system will be useful for the organization, is determined.
- ◆ Feasibility study refers to a process of evaluating alternative systems through cost/benefit analysis so that the most feasible and desirable system can be selected for development.
- ◆ It is conducted by system analyst from eight angles: Technical, Financial, Economical, Schedule/ Time, Resource, Operational, Behavioral and Legal.

2.14.4.1 Technical Feasibility

Testing technical feasibility of an alternative is about answering the following questions -

- ◆ Whether necessary technology exists or not?
- ◆ If not, whether can be acquired or not?
- ◆ If yes, what is its technical capacity in terms of storage and processing?
- ◆ What is its Response time, Accuracy, Expandability, Ease of use, Security etc.

2.14.4.2 Financial Feasibility

The cost of the solution provided is affordable to the user organization or not?

2.14.4.3 Economic Feasibility

This test calls for evaluation of all incremental costs and benefits expected from the proposed system. Following is the classification of cost and benefit that are likely to occur from the new system.

The financial and economic questions raised by analysis during the preliminary investigation are for the purpose of estimating the following:

- (i) The cost of conducting a full systems investigation.
- (ii) The cost of hardware and software for the class of applications being considered.
- (iii) The benefits in the form of reduced costs or fewer costly errors.
- (iv) The cost if nothing changes (i.e. the proposed system is not developed)

Cost: Cost of new system development can be classified into three categories–

a. Development Cost

- ◆ Cost of conducting a full system investigation.
- ◆ Salaries of system analyst and programmers.
- ◆ Cost of file conversion
- ◆ Cost of hardware
- ◆ Cost of software development or acquisition
- ◆ Cost of documentation and training
- ◆ Cost of Infrastructure

b. Operating Cost

- ◆ Hardware and software rental or depreciation charges
- ◆ Salaries of programmer, operator and data processing personnel
- ◆ Cost of data preparation and control
- ◆ Cost of consumables
- ◆ Utility charges

c. Intangible Cost

- ◆ Disruption in the routine organizational activities
- ◆ Loss of employee productivity
- ◆ Loss of good will
- ◆ Cost if nothing changes

Benefits: The benefits which results from developing new information system can be subdivided into Tangible and Intangible benefits.

a. Tangible

- ◆ Increase in sales or profit
- ◆ Decrease in data processing cost
- ◆ Decrease in operating cost (Inventory carrying cost)

b. Intangible

- ◆ Improved information availability
- ◆ Improved accuracy in computation and analysis
- ◆ Improved customer service
- ◆ Improved management decision making
- ◆ Improved employee morale
- ◆ Improved corporate image

2.14.4.4 Schedule Feasibility

It involves estimating how long it will take proposed system to become operationalised and communicating this information to steering committee.

2.14.4.5 Resources Feasibility

This focuses on whether the skilled human resources required for implementing the system will be available or not.

2.14.4.6 Operational Feasibility

It is concerned with ascertaining the views of workers, employees, customers and suppliers about the use of computer facility. The support or lack of support that the firm's employee are likely to give to the system is a critical aspect of feasibility. Testing operational feasibility of an alternative is about answering the following questions -

- ◆ What is the level of support from top management?
- ◆ What is the level of user involvement in system development?
- ◆ What are the possible side effect of the proposed system? E.g. Loss of Control.
- ◆ What is the user's attitude towards current business system?

2.14.4.7 Behavioral feasibility

It refers to whether the system behaves in acceptable manner or not? In other words, can the system be designed to process required data and produce the desired output.

2.14.4.8 Legal Feasibility

It involves testing whether there will be any conflict between proposed system and the organizations legal obligations.

2.14.5 Reporting to Management:-

- ◆ Scope of the problem, Alternative solution, estimated cost and benefits of each alternative is reported to management. The report should also include the recommendations regarding further procedures.
- ◆ Management decides what to do next.

2.15. System Requirement Analysis

2.15.1 Objective :-

- ◆ Determining user's need and expectations.
- ◆ Studying application area in depth to assess the strength and weakness of present system.
- ◆ Developing specification of proposed system.
- ◆ Reporting to management.

2.15.2 Determining user's need :-

Following fact finding techniques are used to determine user's need -

- ◆ *Review Document* - System manual, Input form, Report formats, flowcharts, structure charts etc are generally reviewed by the analyst.
- ◆ *Questionnaires* - Users & managers are asked questions about system.
- ◆ *Interview* - Users and managers are interviewed to obtain exact information.
- ◆ *Observation* - Analyst visit the system's site to watch how system works.

2.15.3 Analysis of Present system:-

If the present system needs just a few adjustment, instead of a complete overhauling the system should be studied in depth. If management decide to replace the current system, then there is no need to waste time in studying it thoroughly. The following aspect of the system need to be analysed -

2.15.3.1 Review historical aspect

- ◆ Identify major decisions and milestones organization has achieved that influencing the growth of the organization and what is the role of present system in that.
- ◆ Investigate what system changes have occurred.
- ◆ Review of Annual report can give real information about the system.

2.15.3.2 Analyse Input

Following informations are studied by analyst in order to analyse input -

- | | |
|----------------------------|--------------------------|
| ◆ Identify various sources | ◆ Nature of each form |
| ◆ Contents of Form | ◆ Origin of information |
| ◆ Distribution of form | ◆ Authorisation of Input |

2.15.3.3 Review Data file maintained

Following informations are analyzed about present system's data files -

- | | |
|--|--------------------------|
| ◆ Investigate data file maintained by each department. | |
| ◆ Number of file and their size | ◆ Location of data files |
| ◆ User of data files | ◆ Activity Ratio |

2.16. System Development Tools

There are several tools and techniques available that help analyst in developing system. They help analyst -

- ◆ To identify the activities and resources involved in the system.
- ◆ To analyse present business operation
- ◆ To design new system

The major tool used for system development can be grouped into four categories based on the system features each document has:

- (a) *Component & Flow of system* : To identify the flow of data among many resources. For e.g.- System Flowchart, Data Flow Diagram, System Component Matrix.
- (b) *User Interface* : To design interface between user and computer system. For e.g.-layout chart, customized screen, Dialog flow diagram.
- (c) *Data attribute & Relationship* : In Define attributes of data and their relationship. For e.g. - Data Dictionary, File layout form, Entity-Relationship chart.
- (d) *Detailed system flow* : To develop detailed process of the system. E.g. - Decision Table, Program Flowchart, Structured chart.

Some of these tools are -

2.16.1 Structured English :-

Also known as Program Design Language (PDL) or Pseudo code, Structured English is the use of the English language with the syntax of structured programming. Structured English consists of the following elements:

- (i) Operation statements written as English phrases executed from the top down.
- (ii) Conditional blocks indicated by keywords such as IF, THEN, and ELSE.
- (iii) Repetition blocks indicated by keywords such as DO, WHILE, and UNTIL.

Some of the keywords that may be used are as follows:

START, BEGIN, END, STOP, DO, WHILE, FOR, UNTIL, REPEAT, END, IF, THEN, ELSE, SO, CASE, EQUAL, LE, GT, GE, NOT, TRUE, FALSE, AND, OR, XOR, GET, WRITE, PUT, UPDATE, CLOSE, OPEN, CREATE, DELETE, EXIT, FILE, READ, EOF, EOT.

Example: A bank will grant loan under the following conditions:

1. If a customer has an a/c with the bank and had no loan outstanding, loan will be granted.
2. If a customer has an a/c with the bank but some amount is outstanding from previous loans then loan will be granted if special approval is needed.
3. Reject all loan applications in all other cases. Write the above conditions in structured language.

Solution : IF customer has a 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

ENDIF

ELSE

Reject

ENDIF

2.16.2 System Flowchart :-

- ◆ Diagrammatic representation of flow of data and steps of information processing taking place in an information system.
- ◆ It is used to analyze, design or document a process or program.
- ◆ Generally, flowcharts are divided into four major categories -
 - Document flowchart : showing a document flow through systems.
 - Data flowchart : showing data flows in a system.
 - System flowchart : showing controls at a physical or resource level.
 - Program flowchart : showing the controls in a program within a system.

◆ **Benefits -**

Communication: They are better way of communicating the logic of a system to all concerned.

Effective analysis: Problem can be analyzed in more effective way.

Proper documentation : Program flowcharts serve as a good program documentation.

Efficient Coding: It act as a guide during the analysis and program development phase.

Proper Debugging: The flowchart helps in debugging process.

Efficient Program Maintenance: The maintenance of program becomes easy.

♦ **Limitations -**

Complex logic: When the program logic is quite complicated, flowchart becomes complex.

Modifications: If modifications are required the flowchart may require re-drawing completely.

Reproduction: As the flowchart symbols cannot be typed, re-production of flowchart becomes problem.

The essentials of what is done can easily be lost in the technical details of how it is done.

2.16.3 Data Flow Diagram :-

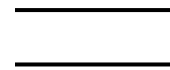
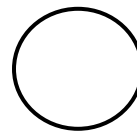
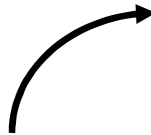
DFD's are used to graphically describe the flow of data within the organization. It is used to document existing system and plan & design the new system. There are 4 elements of DFD :

(a) Data source and Data destinations

(c) Dataflow

(b) Transform process

(d) Data store



*Data Source and
Data Destination*

Data Flow

*Transform Process
Or
Bubble*

Data Store

Data source and Destination

It is in the shape of rectangle and used to represent an organization or individual that sends or receives data used or produced by the system.

Data Flow

- ♦ Represents flow of data between processes, data stores and data sources & destination.
- ♦ Data flow arrows must be labeled to indicate the type of data being passed.
- ♦ If data elements are always more show then with multiple lines; if only some times more show with single line.

Process

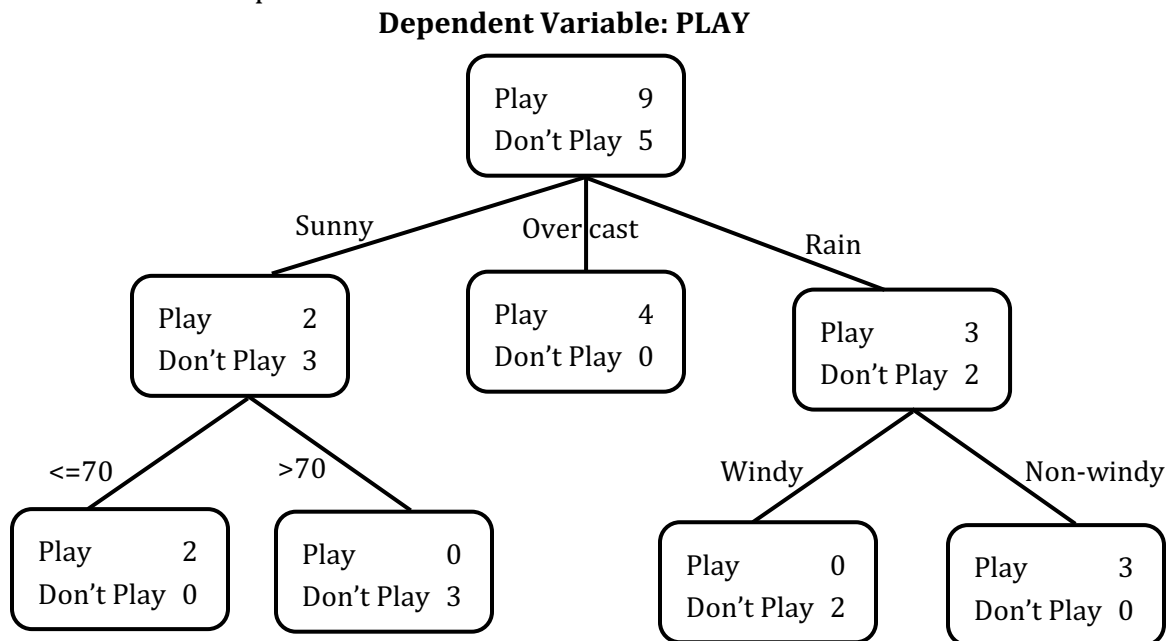
It is in the shape of circle and represent the process that transform data from input to output. They are also referred to as bubble.

Data Stores

- ♦ Represents temp, or permanent repository of data.
- ♦ It does not show the physical storage medium.
- ♦ It must have descriptive data store name.

2.16.4 Decision Tree :-

A Decision Tree is a tree-like graph or model of decisions, commonly used in operations research, specifically in decision analysis, to help identify a strategy most likely to reach a goal and to calculate conditional probabilities. For example –

**2.16.5 Decision Table :-**

A Decision Table is a table which may accompany a flowchart, defining the possible contingencies that may be considered within the program and the appropriate course of action for each contingency. Decision tables are necessitated by the fact that branches of the flowchart multiply at each diamond (comparison symbol) and may easily run into hundreds. If, therefore, the programmer attempts to draw a flowchart directly, he is liable to miss some of the branches. The four parts of the decision table are as follows:

1. Condition Stub - which comprehensively lists the comparisons or conditions;
2. Action Stub - which lists the actions to be taken along the various program branches;
3. Condition entries - which list in its various columns the possible permutations of answer to the questions in the conditions stub; and
4. Action entries - which lists, in its columns the actions contingent upon conditions.

2.16.6 CASE Tools :-

- ◆ CASE refers to Computer Aided Software Engineering.
- ◆ It is a set software packages which is used to automate all manual tasks done to develop systems and support virtually all phases of system development process.
- ◆ An ideal CASE system would have an integrated set of tools and features to perform all aspects in the life cycle.
- ◆ Data Dictionary; Computer aided Diagramming Tools; Report generator; Code Generation; and Reverse Engineering are some of the CASE tools.

2.16.7 System - Component Matrix :-

- ◆ It views system as matrix of components where all activities are placed in rows and components are written on the columns.
- ◆ This show how activities of input, output, processing, storage are accomplished using what types of resources.
- ◆ It also show how the use of hardware, software and people resources can convert data into information.

2.16.8 Data Dictionary :-

- ◆ It is a computer file that contains descriptive information about data stored in other files.
- ◆ It contains information like-
 - (a) Data item's length
 - (b) Data type
 - (c) Range of value
 - (d) Identity of source document from where it is recorded
 - (e) Name of file that stores them
 - (f) Name of program that modify them
 - (g) Identify program & people authorised to access them
- ◆ It is used as a documentation aid to analyst and programmer. They can study, correct or enhance either the database or the computer.
- ◆ It is also useful for file security. It can be used to prohibit certain employees from gaining access to sensitive payroll data.
- ◆ It can serve as audit trail. It can identify the input sources of data items, the computer programs that modify particular data items and the managerial reports on which the data items are output.
- ◆ It is of a great help in investigation of a system controls.

2.16.9 Layout forms & Screens; Menu :-

- ◆ These consists of preprinted form or electronic displays on which the size and placement of title, heading, data & information can be designed.
- ◆ These are used to design source documents, Records structure, File layout, Display and Printed reports.

2.17. System Specification

At the end of the analysis phase, the systems analyst prepares a document called "Systems Requirement Specifications (SRS)" with the following contents -

- ◆ **Introduction:** Goals and Objectives of the software.
- ◆ **Information Description:** Problem description; Information content, flow and structure; Hardware, software, human interfaces etc.
- ◆ **Functional Description:** Diagrammatic representation of functions; Interplay among functions; Design constraints etc.
- ◆ **Behavioral Description:** Response to external events and internal control.
- ◆ **Validation Criteria:** Tests to be performed to validate functions.
- ◆ **Appendix:** Data flow Diagram; Tabular Data; Detailed description of algorithms charts.
- ◆ **SRS review :** It contains the following -
 - The development team makes a presentation and then hands over the SRS document to be reviewed by the user or customer.
 - The review reflects the development team's understanding of the existing processes. Only after ensuring that the document represents process accurately, should the user sign the document.

2.18. Roles involved in SDLC

i) Steering Committee

- ◆ To provide overall direction.
- ◆ To be responsible for all cost and timetables.
- ◆ To conduct a regular review of progress of the project.
- ◆ Taking corrective actions like re-scheduling, re-staffing, change in the project objectives.

ii) Project Manager

A project manager responsible for more than one project and liaisons with the client. He is responsible for delivery of the project within the time and budget and periodically review the Progress of the project with the project leader.

iii) Project Leader

Dedicated to a project, project leader is responsible to ensure its completion and fulfillment of objectives. The entire project team reports to him.

iv) Systems Analyst/Business Analyst

The systems analysts' main responsibility is to conduct interview with users and understand their requirements. He is a link between the users and the programmers who converts the users requirements in the system requirements. He plays a pivotal role in the Requirements analysis and Design phase.

v) Module Leader I Team Leader

A project is divided into several manageable modules. For example, while developing a financial accounting application - Treasury, Accounts payable, Accounts receivable can be identified as separate modules. The development responsibility for each module is assigned to Module Leaders.

vi) Programmer / Coder / Developer

Programmers convert design into programs by coding using programming language. He also tests the program for debugging activity.

vii) Database Administrator (DBA)

The DBA ensures the integrity and security of information stored in the database. DBA handles multiple projects and also helps the application development team in database performance issues. Inclusion of new data elements has to be done only with the approval of the database administrator.

viii) Quality Assurance

This team sets the standards for development, and checks compliance with these standards. Any quality assurance person who has participated in the development process shall not be viewed as "independent" to carry out quality audits.

ix) Tester -

Tester tests programs and subprograms as per the plan given by the module/project leaders.

x) Domain Specialist

Whenever a project team has to develop an application in a field that's new to them, they take the help of a domain specialist. For example, if a team undertakes application development in Insurance, they may seek the assistance of an insurance expert at different stages. A domain specialist need not have knowledge of software system.

xi) IS Auditor

As a member of the team, IS Auditor ensures that the application development also focuses on the control perspective. He is involved at the Design Phase and the final Testing Phase.

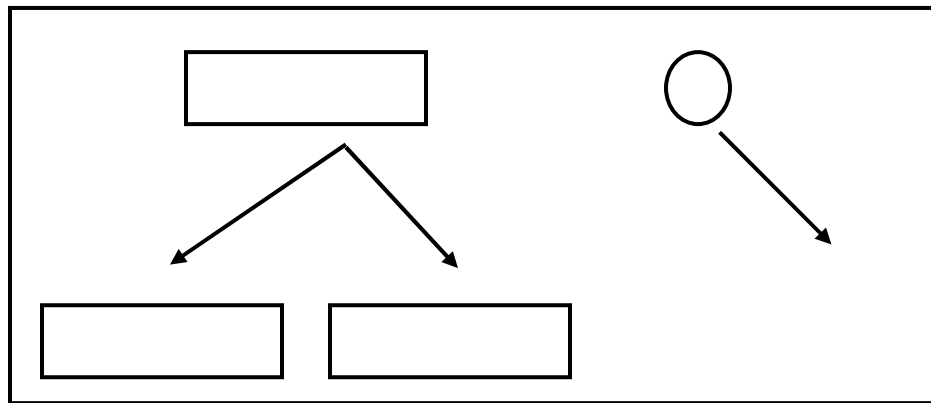
2.19. System Design

- ◆ **Objective:** Designs an Information system that best satisfies the user/managerial requirements.
- ◆ **Activities:** Describing inputs and outputs; Determining the processing steps and computation rules; Determining data file or database system file design; Preparing the program specifications; and Internal / external controls.
- ◆ **Deliverable:** This phase creates a "blueprint" for the design with the necessary specifications for the hardware, software, people and data resources. System design involves first logical design and then physical construction of a system. Once the detailed design is completed, the design is then distributed to the system developers for coding.

- ◆ **Steps :** The design phase involves following steps:.
 - (i) Architectural Design;
 - (ii) Design of the Data / information Flow;
 - (iii) Design of the Database;
 - (iv) Design of the User-interface;
 - (v) Physical Design; and
 - (vi) Design and acquisition of the hardware/software platform.

2.19.1 Architectural Design

- ◆ Architectural design deals with hierarchy of modules and sub-modules. At this stage, we identify - major modules; function and scope of each module; interface features of each module; Data received from/sent to other modules.
- ◆ The architectural design is made with the help of a tool called Functional Decomposition which has three elements - Module; Connection; and Couple.
- ◆ The module is represented by a box and connection between them by arrows. Couple is shown by an arrow with circular tail.



2.19.2 Design of Data / Information flow

The major step in the conceptual design of the new system is designing of the data/information flow for the proposed system, the inputs that are required are -

- existing data/information flows,
- problems with the present system; and
- objective of the new system.

2.19.3 Design of Database

Design of the database involves determining its scope of use and design of architecture. The design of the database involves four major activities -

- ◆ **Conceptual Modeling:** Describing entities of the database; attributes of the entities; constraints on the entities and their relationship.

- ♦ **Data Modeling:** Translating conceptual model into data model so that they can be accessed and manipulated by programming languages.
- ♦ **Logical layout design:** Partitioning the data structure on common column value so that it can be accessed together. For example relationships among records might be established via symbolic pointer addresses.
- ♦ **Physical layout design :** Decisions must be made on how to distribute the storage structure across specific storage media and locations - for example, the cylinders, tracks, and sectors on a disk and the computers in a LAN or WAN.

2.19.4 Design of User-Interface

- ♦ Design of user-interface involves determining the ways in which users will interact with a system. The points that need to be considered are –
 - Source documents to capture raw data;
 - hard-copy output reports;
 - screen layouts for input;
 - inquiry screens;
- ♦ Designing System Input/Outputs
 - Input design consists of developing specifications and procedures for data preparation, collection and data entry.
 - One of the most important feature of an information system for users is the output it generates. It involves designing of both visual and printed output. The output should accomplish one or more of the following objectives -

Convey some information.

Signal important events, opportunities, problems or warnings.

Trigger an action.

Confirmation of an action.

- ♦ Important factors in Input / Output design:

Characteristic	Definition	Input Design	Output Design
Content	Refers to the actual pieces of data to be gathered to produce the required output to be provided to users.	The analyst is required to consider the types of data that are needed to be gathered to generate the desired user outputs. New documents for collecting such information may be designed.	The contents of a weekly output report to a sales manager might consist of sales person's name, sales calls made by each sales person during the week, and the amount of each product sold by each salesperson to each major client category.

Timeliness	Timeliness refers to when users need outputs, which may be required on a regular, periodic basis – perhaps daily, weekly, monthly, at the end of quarter or annually.	Data needs to be inputted to computer in time because outputs cannot be produced until certain inputs are available. Hence, a plan must be established regarding when different types of inputs will enter the system.	A sales manager, may be requiring a weekly sales report. Other users, such as airline agents, require both real time information and rapid response times in order to render better client service.
Format	Input format refers to the manner in which data are physically arranged. Output format refers to the arrangement referring to data output on a printed report or in a display screen.	After the data contents and media requirements are determined, input formats are designed on the basis of few constraints like – the type and length of each data field as well as any other special characteristics (number decimal places etc.)	Format of information reports for the users should be so devised that it assists in decision making, identifying and solving problems, planning and initiating corrective action and searching.
Media	Input-output medium refers to the physical used for input, storage or output.	This includes the choice of input media and subsequently the devices on which to enter the data. Various user input alternatives may include display workstations, magnetic tapes, magnetic disks, key-boards, optical character recognition, pen-based computers and voice input etc. A suitable medium may be selected depending on the application to be computerised.	A variety of output media are available in the market these days which include paper, video display, microfilm, magnetic tape/disk and voice output.

Form	Form refers to the way the information is inputted in the input form and the content is presented to users in various output forms – quantitative, non-quantitative, text, graphics, video and audio.	Forms are pre-printed papers that require people to fill in responses in a standardized way. Forms elicit and capture information required by organisational members that often will be input to the computer. Through this process, forms often serve as source documents for the data entry personnel.	The form of the output should be decided keeping in view the requirements for the concerned user. For example – Information on distribution channels may be more understandable to the concerned manager if it is presented in the form of a map, with dots representing individual outlets for stores.
Input Volume /Output Volume	Input volume refers to the amount of data that has to be entered in the computer system at any one time. The amount of data output required at any one time is known as output volume.	In some decision-support systems and many real-time processing systems, input volume is light. In batch-oriented transaction processing systems, input volume could be heavy which involves thousands of records that are handled by a centralized data entry department using key-to-tape or key-to-disk systems.	It is better to use high speed printer of a rapid retrieval display unit, which are fast and frequently used output devices in case the volume is heavy.

2.19.5 Physical Design -

- ◆ For the physical design, the logical design is transformed into units, which in turn can be decomposed further into implementation units such as programs and modules.
- ◆ Some of the issues addressed in physical design are - type of hardware for client and server; Operating systems to be used; Type of networking; Type of Processing - Batch, Online, Real-time; and Frequency of Input, output.
- ◆ **Design Principles -**
 - Design two or three alternatives and choose the best one on pre-specified criteria.
 - The design should be based on the analysis.
 - The software functions should be directly relevant to business activities.
 - The design should follow standards laid down.
 - The design should be modular.
- ◆ **Modularity** : A module is a manageable unit containing data and instructions to perform a well-defined task. Modularity is measured by two parameters: Cohesion and Coupling.
- ◆ **Cohesion** refers to the manner in which elements within a module are linked. Coupling is a measure of the interconnection between modules. It refers to the number and complexity of connections between "calling" and "called" modules.
- ◆ In a good modular design, Cohesion will be high and coupling low.

2.19.6 Design of the Hardware / Software Platform

- ◆ The new system some times requires hardware and system software not currently available in the organization. Therefore, the new hardware and system software platform that can support the application system will have to be designed.
- ◆ Auditors should be concerned about the extent to which modularity & generality are preserved in the design of the hardware/system software platform.

2.20. System Acquisition and Development

This phase of the systems development relates to the acquisition of hardware, software and services.

2.20.1 Acquisition Standards:-

Management should establish acquisition standards that should focus on -

- ◆ Ensuring security, reliability, and functionality already built into a product.
- ◆ Ensuring vendor, contract, and licensing reviews and compatibility with existing systems.
- ◆ Including invitations-to-tender and request-for-proposals.
- ◆ Establishing acquisition criterias.

2.20.2 Acquiring Systems Components from Vendors :-

At the end of the design phase, the organization gets a reasonable idea of the types of hardware, software and services it needs for the system being developed.

2.20.2.1 Hardware Acquisition

- ◆ In case of procuring such machinery as transportation vehicle, air conditioning equipment etc., the management can normally rely on the objective selection criteria.
- ◆ But complex internal structure, versatility and compatibility with other devices makes purchasing of computer hardware very difficult.

2.20.2.2 Software

- ◆ Once user output and input designs are finalized, the nature of the application software requirements can be assessed by the systems analyst. The analyst also determine the degree of processing that the system need to handle.
- ◆ At this stage, the system developers must determine whether the application software should be created in-house or acquired from a vendor.

2.20.2.3 Contracts, Software Licenses and Copyright Violations

- ◆ Contracts between an organization and a software vendor should clearly describe the rights and responsibilities of the parties in the contract.
- ◆ The contracts Should be in writing with sufficient details to provide assurances for performance, source code accessibility, software and data security etc.
- ◆ Software license is a license that grants permission to do things with computer software. The usual goal is to authorize activities which are prohibited by default by copyright law, Paten I: law, Trademark law and any other Intellectual property right.

2.20.2.4 Validation of Venders proposal

It is the process of evaluating & ranking proposals on the basis of same criterias. Some of the commonly considered factors for ranking are -

- ◆ *Economic Factors*- Cost, Return, Method of acquisition (Terms & conditions) etc.
- ◆ *Hardware Factors* - Memory and Processor requirement.
- ◆ *Software Factors* - Performance, Reliability, Language, Ease of use, modification.
- ◆ *Service Factors* - Checking new system, Training, Maintenance.
- ◆ *Reputation* - Financial stability, Record of Keeping promises etc.

2.20.2.5 Methods of validation –**1. Check list -**

It is very simple and a subjective method of evaluation. A list of questions is prepared to be answered by the vendors proposal. This method can be applied to select both hardware and software.

2. Point scoring Analysis -

- ◆ Prepare list of criteria
- ◆ Assign criteria values (Possible points)
- ◆ Response of vendors are assigned a response value
- ◆ Calculate response score (total)
- ◆ Analyse the results

	CV	RV-1	RV-2
Meets all mandatory req.?	10	09	07
Contains adequate controls?	10	06	08
Is it user friendly?	09	08	07
Is it maintainable?	08	07	09
Response Score		30	31

3. Public Evaluation Reports -

- ◆ Several consultancy agencies compare hardware, software performance for various manufactures and publish their report in this regard.
- ◆ On the basis of such reports we can select a vendor.
- ◆ This method is particularly useful where the buying staff has inadequate knowledge of computer facts.

4. Bench marking problem -

- ◆ It is a sample problem that represents a part of buyer's primary work load.
- ◆ It is oriented towards testing whether a computer offered by the vendor meets the requirement of the job of the buyer.
- ◆ Since it is a sample of all the user's job, it comprises of long job, short job, tape job, disk job, mathematical job, I/O loads etc. i.e. a complete job mix.
- ◆ Although bench marking problems helps in choosing a computer system but. it needs considerable time & efforts to select the true bench marking problem.
- ◆ In addition to it conducting bench marking test requires existence of operational hardware, software and Services.

5. Test Problem -

- ◆ It is also a sample problem used to check the computer system.
- ◆ Unlike bench marking problem it disregards the actual job mix and is devised to test the true capabilities of hardware and software.
- ◆ It includes problems such as Problem to test translation speed and Problem to test response time in multiprogramming environment.

2.20.3 System Development: Programming Techniques and Languages

- ◆ **Objective:** To convert the system specification into a fully functioning system.
- ◆ **Activities:** Application programs are written, tested and documented, conduct system testing.
- ◆ **Deliverable:** A fully functional and documented system.
- ◆ **Characteristics:** A good coded program should have the following characteristics -
 - *Reliability:* It refers to the consistence which a program provides over a period of time.
 - *Robustness:* It refers to taking into account all possible inputs and outputs of a program.
 - *Accuracy:* It refers not only to what program is supposed to do, but also what it should not do.
 - *Efficiency:* It refers to the performance which should not be unduly affected with the increase in input values.
 - *Usability:* It refers to a user-friendly interface and easy-to-understand manuals.
 - *Readability:* It refers to the ease of maintenance of program.

2.20.3.1 Program Coding Standards

- ◆ The logic of the program outlined in the flowcharts is converted into program statements at this stage. Different programmers may write a program using different sets of instructions but each giving the same results. Therefore, the coding standards are defined.
- ◆ These standards serves as a method of communication between teams, amongst the team members and users, thus working as a good control.
- ◆ Coding standards minimize the system development setbacks due to programmer turnover. Coding standards helps achieve programming objectives such as - simplicity, efficient utilization of storage and least processing time.

2.20.3.2 Programming languages

Programming languages are the medium of communication between user and computer. The programming languages commonly used are:

- ◆ High level general purpose programming language such as COBOL & C language.
- ◆ Object Oriented languages such as C++, JAVA etc.
- ◆ Scripting language like JAVA Script, VBScript.
- ◆ Decision Support or Expert System languages like PROLOG.

2.20.3.3 Choice of Programming Language

Selection of programming language should be based on algorithmic complexity; environment in which software has to be executed; performance consideration; data structure complexity; knowledge of software development staff etc.

2.20.3.4 Program Debugging

It is the process of correcting programming language syntax and diagnostic errors so that the program compiles cleanly. Debugging can be a tedious task consisting of following four steps -

- ◆ Inputting the source program to the compiler,
- ◆ Letting the compiler find errors in the program,
- ◆ Connecting lines of code that are erroneous, and
- ◆ Resubmitting the corrected source program as input to the compiler.

2.20.3.5 Test the program

The programmer should plan the testing of all possible exceptions. The program test plan should be discussed with the project manager and/or system users. A log of test results and all conditions successfully tested should be kept.

2.20.3.6 Program Documentation

The writing of narrative procedures and instructions for people who will use software is called documentation. Managers and users should carefully review documentation in order to ensure that the system behaves as the documentation indicates. User documents should be prepared in such a way that the user can clearly understand the instructions.

2.20.3.7 Program Maintenance

The users' requirements from the system are subject to continual change. Modifying the program according to the user's requirement is called program maintenance. There are, usually, separate categories of programmers called maintenance programmers who are entrusted with this task.

2.21. System Testing

Testing is a process used to identify the correctness, completeness and quality of developed computer software. Testing uncovers different classes of errors in a minimum amount of time and effort. However, testing does not show the absence of defect; it only shows that software defects are present. Different levels of testing are as follows :-

2.21.1 Unit Testing

A unit is the smallest testable part of an application which may be an individual program, function, procedure, or module. The goal of unit testing is to isolate each part of the program and show that the individual parts are correct. There are five categories of tests that a programmer typically performs on a program unit:

Functional Tests : Functional Tests check "whether programs do what they are supposed to do or not". The test plan specifies input values and expected results, and as per this plan the programmer checks whether the actual result and expected result match.

Performance Tests : Performance Tests should be designed to verify the response time, the execution time, the throughput, memory utilization and the traffic rates on data channels.

Stress Tests : Stress testing is a form of testing that is used to determine the stability of a given system when operated beyond normal operational capacity. The purpose of a stress test is to determine the limitations of the program.

Structural Tests : These are concerned with examining the internal processing logic of a software.

Parallel Tests : In Parallel Tests, the same test data is used in the new and old system and the output results are then compared.

Methods of Unit Testing

2.20.3.7.1.1.1.1 Static Analysis Testing

- ◆ **Desk Check:** This is done by the programmer himself. He checks for logical syntax errors, and deviation from coding standards.
- ◆ **Structured walk-through:** This is done by other programmers.
- ◆ **Code inspection :** The program is reviewed by a formal committee. Review is done with formal checklists.

2.20.3.7.1.1.1.2 Dynamic Analysis Testing

- ◆ **Black Box Testing:** The test designer selects valid and invalid inputs and determines the correct output. This testing checks whether the system can do what it is supposed to do. If a module performs a function which is not supposed to, the black box test does not identify it.
- ◆ **White Box Testing:** White box testing design test cases based on internal structure of the system and aim to test what the program can do. It requires programming skills and the tester chooses test case inputs to exercise paths through the code and determines the appropriate outputs. It is applicable at the Unit, Integration and System levels of the testing. After obtaining a clear picture of the internal workings of a product, tests can be conducted.
- ◆ **Gray Box Testing:** In gray box testing, the tester applies a limited number of test cases to test the internal workings of the software. In the remaining part of the gray box testing, one takes a black box approach in applying inputs to the software and compare the outputs.

2.21.2 Integration Testing

- ◆ Integration testing is an activity of software testing in which individual software modules are combined and tested as a group. It occurs after unit testing and before system testing with an objective to evaluate the connection of two or more components that pass information from one area to another.
- ◆ Integration testing tests those modules that have been unit tested, groups them in larger aggregates, applies tests defined in an integration test plan, and delivers its output. This is carried out in the following manner:

Bottom-up Integration: It consists of unit testing, followed by sub-system testing, and then testing of the entire system. Bottom-up testing is easy to implement, however, its disadvantages is that testing of major control points is defined to a later period.

Top-down Integration: Top-down integration starts with the main routine, and stubs are substituted, for the module directly subordinate to the main module. Once the main module testing is complete, stubs are substituted with real modules one by one, and these modules are tested with stubs.

The difficulty arises in the top-down method, because the high-level modules are tested, not with real outputs from subordinate modules but from stubs.

2.21.3 Regression Testing

- ◆ Each time a new module is added to the system the software changes. These changes may cause problems with functions that previously worked flawlessly.
- ◆ The regression tests ensure that changes or corrections have not introduced new errors. The data used for the regression tests should be the same as the data used in the original test.

2.21.4 System Testing

System testing is a process in which all the system elements are tested as a whole. System testing begins when the software is operational. These test procedures are often performed in a non- production test environment. The types of testing that might be carried out are as follows:

Recovery testing: This is the testing of "how well the application is able to recover from crashes, hardware failures and other similar problems". Recovery testing is the forced failure of the software in a variety of ways.

Security Testing: This is the process to determine that an Information System protects data and maintains functionality as intended. The six basic security concepts that need to be covered by security testing are - confidentiality, integrity, authentication, authorization, availability and non-repudiation. This testing technique also ensures the existence and proper execution of access controls in the new system.

Stress or Volume Testing: Stress testing is a form of testing that is used to determine the stability of a given system when operated beyond normal operational capacity.

Performance Testing: This testing is used to determine the speed or effectiveness of a computer, network, software program or device. This testing technique compares the new system's performance with that of similar systems using well defined benchmark.

2.21.5 Final Acceptance Testing

Final Acceptance testing is conducted when the system is just ready for implementation. It is used to ensure that the new system satisfies the quality standards and satisfies the users. The final acceptance testing has two major parts:

Quality Assurance Testing: It ensures that the new system satisfies the prescribed quality standards.

User Acceptance Testing: It ensures that the functional aspects expected by the user have been well addressed in the new system. There are two types of the user acceptance testing -

- **Alpha Testing:** This is the first stage, often performed by the users within the organization.
- **Beta Testing:** This is the second stage, generally performed by the external users. This involves sending the product outside the development environment for real world exposure.

2.22 Systems Implementation

The process of ensuring that the information system is operational and then allowing users to takeover its operation for use and evaluation is called Systems implementation.

- ♦ Objective : To put the new system into production.
- ♦ Activities : The activities involved in system implementation are as follows:
 - Conversion of data to the new system files.
 - Training of end users.
 - Completion of user documentation. System changeover.
 - Evaluation of the system at regular intervals.
- ♦ Deliverable : A full functional and documented system in its operational environment.

2.22.1 Activities during implementation stage

Following activities are carried out in system implementation.

2.22.1.1 Equipment Installation :-

Equipment here means hardware required to operationalise the new system. Since these equipment are already purchased in the previous phase, they are installed now according to the guidelines given by the vendor. The following 3 activities are performed -

i. Site preparation

An appropriate location must be found, to provide an operating environment for the equipment, that will meet the vendor's specification. A proper layout of the space required should be prepared considering following factors-

- ♦ Space for equipment
- ♦ Space for people
- ♦ Space for movement

ii. Equipment installation

It refers to physical installation of equipment assembling the computer and connecting it to power mains.

iii. Equipment checkout

It is switch on the equipment and check them with the configuration ordered. It doesn't require to check equipments thoroughly as they are already checked during acquisition phase.

2.22.1.2 Training Personnel :-

The success or failure of a system depends upon the way it is being operated and used. Therefore the quality of training received by the personnel involved with the system is of utmost importance. There are two kind of people associated with system: Operators and End-Users. Their training must cover –

i. Operators

- ◆ How to handle all possible operations.
- ◆ How to turn the equipment ON & use it.
- ◆ What common malfunctions may occur.
- ◆ How to recognize them.
- ◆ Steps to be taken when they arise.
- ◆ To whom to contact.

ii. End-Users

- | | |
|--|-------------------------|
| ◆ Capturing data | ◆ Preparing data |
| ◆ Editing data | ◆ Formulating inquiries |
| ◆ Equipment operation | ◆ How to prepare disk |
| ◆ Loading paper, changing ribbon of printer. | |

2.22.1.3 Conversion strategies :-

These are the strategies for replacing old system with the new one. There are 5 conversion strategies-

(i) Direct Implementation/Abrupt change-over

In this, on a specified date old system is dropped and the new is put in use. It is adopted only when extensive testing is done. Risks involved in this strategy are -

- ◆ Long delays when error occurs
- ◆ No possibility of comparing result
- ◆ User is forced to use unfamiliar system

(ii) Parallel Implementation

Here both old and new systems are used parallel for a fixed period of time and results are compared. Old is dropped when the pre-determined period is over. Risks involved are -

- ◆ Double cost
- ◆ Double work load
- ◆ User will always prefer old.

(iii) Phased Implementation

It is combination of the above two strategies. Both the systems work parallelly but one part of the work is get done by the old system and the other part by the new one. Volume of transactions is gradually increase for new system. The only risk involved in this is that it takes lot of time in implementation of the system.

(iv) Pilot Implementation

When installations are to be done at many places then this strategy is used. In this strategy, at one place the new system is implemented using any of the above approaches and when results are satisfactory then at all other location, most often, direct changeover strategy is used.

2.22.1.4 Activities involved in conversion

Following activities are performed during system conversion –

a. Procedure Conversion :

- ◆ Operating procedure must be completely documented.
- ◆ Procedure must be communicated out to all the personnels.
- ◆ Written procedures must be supplemented in training.
- ◆ Brief meetings must be held of all employees.

b. File Conversion :

- ◆ It should begin long before programming & testing is completed.
- ◆ It may include conversion from one medium to another or one organization to another.
- ◆ Adequate controls must be executed during file conversion.
- ◆ Old files must be kept for certain period to be on the safer side.

c. System Conversion :

- ◆ Daily processing is shifted from existing to the new system.
- ◆ A cut of date is decided and transactions after then will be processed by the new system.
- ◆ Old system be dropped as soon as user group is satisfied from the new one.

d. Scheduling :

- ◆ System manager, in conjunction with departmental manager, prepares the schedule.
- ◆ It is difficult to prepare schedule for the new system due to lack of familiarity.
- ◆ Monthly schedule should be prepared reflecting daily schedule.
- ◆ Schedule should include time necessary for rerun, program testing, special non recurring reports etc.
- ◆ Time log book of machine should be kept for realistic schedule.
- ◆ Just as the machine must be scheduled so must be personnel who operate them.

2.23 Post Implementation Review and Systems Maintenance

2.23.1 P.I.R.

- ◆ A Post Implementation Review answers the question "Did we achieve the objectives we set out before the system development?"
- ◆ Post-Implementation Review should be scheduled some time after 6 weeks to 6 months since the solution has been deployed. The two basic dimensions of Information system that should be evaluated are : (i) whether the newly developed system is operating properly, (ii) whether the user is satisfied with the information system with regard to the reports supplied by it.
- ◆ PIR provides feedback to the management to access value of information and performance of personnels & equipments. System is evaluated form three angles -

2.23.1.1 Development Evaluation

It refers to evaluating whether the system is completed - *Within time* and *Within budget* or not. This evaluation is done by the *top management level steering committee*.

2.23.1.2 Operation Evaluation

This evaluation of the system is done by *System Analyst*. It refers to checking whether the system has the following characteristics or not -

- | | |
|---------------------|-----------------|
| ◆ Timely processing | ◆ Accuracy |
| ◆ User friendliness | ◆ Response time |
| ◆ Storage capacity | |

2.23.1.3 Information Evaluation

- ◆ It refer to checking the extent to which information supplied by MIS is used to decision making. It is done by the system users.
- ◆ Manager's satisfaction is the criteria for evaluation. It is considered that the more frequently they ask for information the more satisfied the look.

2.23.2 System Maintenance

- ◆ Information systems are subject to frequent changes as business requirement keep changing periodically.
- ◆ Modification after implementation is called system maintenance. It may involves adding new data items, modifying report formats, changing calculations etc.
- ◆ Maintenance can be classified in to the following categories -
 - **Scheduled maintenance:** It is anticipated and can be planned for.
 - **Rescue maintenance:** It refers to removing undetected malfunctions that were not anticipated but require immediate solution.
 - **Corrective maintenance:** It deals with fixing bugs in the code or defects found. A defect can result from design errors, logic errors; coding errors, processing errors etc.
 - **Adaptive maintenance:** It consists of adapting changes in the environment and making adjustive corrections.
 - **Perfective maintenance:** It mainly deals with accommodating new or changed user requirements and concerns functional enhancements to the system.
 - **Preventive maintenance:** It is concerns with activities aimed at increasing the system's maintainability, such as updating documentation, adding comments, and improving the modular structure of the system.

2.24 Operation Manuals

- ◆ A user's guide also commonly known as an Operation Manual, is a technical communication document intended to give assistance to people using a particular system.
- ◆ It is usually written by a technical writer, although user guides are written by programmers. Operation manuals usually include the following:
 - A cover page, a title page and copyright page;
 - A preface, containing details of related documents and information on how to navigate the user guide;
 - A contents page;
 - A guide on how to use at least the main functions of the system;
 - A troubleshooting section detailing possible errors or problems that may occur, along with how to fix them;
 - A FAQ (Frequently Asked Questions);
 - Where to find further help, and contact details;
 - A glossary and, for larger documents, an index;

2.25 Organizational Structure of IT department :-

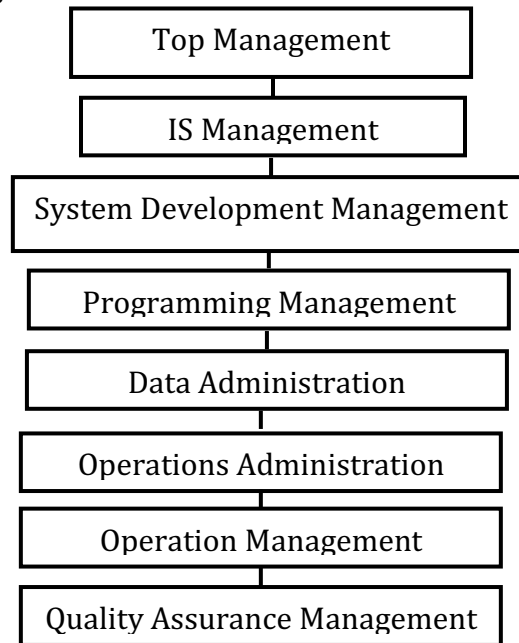
The organizational structure of Management of IT department can be classified into 2 categories:

(i) *Line Management*

(ii) *Project Management*

2.25.1 Line Management structure

In the line management structure, IT department in an organizational subsystem which attempt to ensure that the development, implementation, operation and maintenance of information system proceed in a planned and controlled manner. Corresponding to the organizational hierarchy, several levels of management have been identified in this structure.

**2.25.1.1 Top Management**

Top management of the organization must ensure that the data processing installation is well managed. It is responsible primarily for long run policy decisions on how computers will going to benefit the organization.

2.25.1.2 IS Management

IS management has overall responsibility for planning and controlling of all computer activities. It also provides inputs to top management's long run policy decision making and translates long run policies into short run goals and objectives.

2.25.1.3 System Development Management

SDM is responsible for the design, implementation and maintenance of application systems.

2.25.1.4 Programming Management

They are responsible for the control and use of an organization's data including the database and library of application file.

2.25.1.5 Data Administration

Data administration is responsible for the control and use of an organization's data including the database and library of application system files.

2.25.1.6 Security Administration

Security administration is responsible for the physical security of the data processing and IS programs.

2.25.1.7 Operations Management

Operations Management controls the day-to day operations of data processing systems. It is responsible for -

- Data preparation - Dataflow
- Production run - Maintenance of hardware
- File library facilities - Sometimes maintenance of program

2.25.1.8 Quality Assurance Management

Quality assurance management undertake an in-depth quality assurance review of data processing in each application system. This review involves a detailed check of the -

- | | | | |
|--------------|------------|---|-----------------|
| Authenticity | Accuracy | & | Completeness of |
| Input | Processing | & | Output |

2.25.2 Project Management Structure

In project management, project requests are submitted to and prioritized by the steering committee. The project manager, who may be a non IS staff member, should be given complete operational control of the project. IS auditor may be included in the project team as control advocate and experts.

The structure of an IT department in Project management structure is divided into tow main areas of activity:

- ◆ System Development
- ◆ Information Processing

System Development is concerned with the development, acquisition and maintenance of computer application systems and performs System Analysis and Programming functions.

Information Processing or IP is primarily concerned with the operational aspect of the information processing environment and often include Computer Operation, System Programming, Telecommunication and Librarian functions.

Duties and Responsibilities of some of the personnels working under Project Management structure are -

2.25.2.1 Data Entry

- ◆ The data entry supervisor is responsible for ensuring whether the data is authorized, accurate and complete when entered into the system.
- ◆ Responsible for bringing "information" into a system. The information takes two forms: first, it may be raw data to be processed and second it may be instruction to execute particulars processes.
- ◆ DEO validates both type of information input and any errors detected is investigated and corrected so that the input resubmission is authentic, complete, unique and timely.

2.25.2.2 File library

- ◆ File librarian is responsible for recording, issuing, receiving and safeguarding all programs and data files that are maintained in the library on computer disks.
- ◆ Managing the organization's library of machine readable files involves three functions :
 - (i) File must be used only for the purposes intended. Control must be exercised over program files, data files, and procedure files.
 - (ii) The storage media used for files must be maintained in correct working order.
 - (iii) A file backup strategy and file retention strategy must be implemented.
 - (iv) Files should be stored in a secure room.
 - (v) Files storage room must have a stable environment, constant temperature, no dust etc.
 - (vi) Files should be issued only in accordance with a schedule or on the basis of an authorised requisition.
 - (vii) To keep a record of all event that occur to files, a log must be maintained.

2.25.2.3 Control Group

- ◆ The control group manages the flow of data and is responsible for the collection, conversion and control of input, and balancing the distribution of output to the user community.
- ◆ The input/output control group should be in a separate area where only authorised personnel are permitted. In other words control and operation should be separate functional area.

2.25.2.4 Operations

- ◆ Operation management is responsible for the daily running of hard ware and software facilities so that the production application system can accomplish their work.
- ◆ Though there arc some variations across the organization, the operations group within the IT department undertakes following functions -
 - Computer operations
 - Data preparation
 - File library.
 - Performance monitoring
 - Communication network control
 - Production work flow control
 - Documentation library

2.25.2.5 Security Administration

- ◆ The security administration in a data processing organization is responsible for matters of physical security, hi Other words, the security administrator attempts to ensure that the physical facilities in which the system are developed, implemented, maintained and operated are safe from threats that affect the continuity of operation.
- ◆ The security administration is supposed to provide two types of protection to system -
 - (i) **Physical security:** A complete reliable protection scheme must take into account the possibility of physical attacks on the system, such as person entering into the computer room and steal the thing. Protecting can be provided by appointing a *Security guard* who will personally verify the persons entering into the computer room. *Door access lock, Biometric* devices are the other security measures.
 - (ii) **Data security:** Database management system often provide control over data definition and data manipulation facility. In the environment where DBMS combines with online transaction processing, access to the database objects such as table or view can be controlled through internal database mechanism, by way of *Password, User view(Sub-schema)* and *Access privileges* which limits, what transaction can do. Further *monitoring and performance tools, should be restricted* to appropriate personnel.
- ◆ In order to provide regular security, organizations prepare a security program under the guidance of security administrator. A security program is a series of ongoing regular, periodic evaluation conducted to ensure that the physical facilities of an information system are safeguarded adequately.
- ◆ The first security evaluation conducted may be a major, security administrator has to consider an extensive list of possible threat to the organization, prepare an inventory of assets, evaluate the adequacy of control, implement new control etc.
- ◆ Subsequent security evaluation may focus only on changes that have occurred, perhaps due to purchase of new hardware, new software or a new threat etc. Security evaluation need to be repeated periodically to determine whether these changes have necessitate modification to controls.
- ◆ Various steps involved in evaluation are -
 - Preparation of a project plan
 - Identification and valuation of assets
 - Threats identification
 - Exposure analysis
 - Control adjustment
 - Report preparation

2.25.2.6 Production work flow control

The PWF Control Section manages the flow of data between users and the information systems and between data preparation and the computer room. It makes it more difficult for operators and data preparation personnel to collude and to perpetrate a fraud.

2.25.2.7 Quality Assurance

- ◆ QA group is responsible for testing and verifying whether the programs, program changes and documentation adhere to standard before the programs are moved into production.
- ◆ The quality control section check to see that the input is in order by scanning it for reasonableness and completeness. If the input passes this quality assurance check, it is entered into a log and dispatched either to the computer room if it is already in machine readable form or to data preparation if it must be keyed in.

2.25.2.8 System Analysis

SA are responsible for interpreting the need of the user, and design system based on the need. S A performs the following functions:

- ◆ Determining user need
- ◆ Identifying alternative solution
- ◆ Conducting feasibility test
- ◆ Logical designing of proposed system

2.25.2.9 Auditor

- ◆ Auditors act as a participant in the system development process.
- ◆ From system effectiveness view point the auditor is concerned with whether the design meets strategic requirements.
- ◆ From efficiency view point the auditor is concerned with the resources that will be needed to run the system.
- ◆ From safeguarding access and data integrity view point the auditor is concerned with the controls designed into the system.
- ◆ During system design phase the auditor also evaluates the ongoing audit ability of the system. Auditor may deem it necessary to build certain audit capabilities into the system in the form of audit module. These modules capture data or examine conditions of interest to the auditor.
- ◆ When evaluating the information processing system design phase, the auditor must examine six major activities –
 - Elicitation of detailed requirement
 - Design of the data / information flow
 - Design of the database
 - Design of the user interface
 - Physical design
 - Design of the hardware / software configuration.

2.25.2.10 Application Programming

- ◆ Application programmers are responsible for developing system as per analyst's design and monitoring system in production.
- ◆ They should work in a test only environment and should not move test version into the production environment.
- ◆ AP should not have access to system program libraries.

2.25.2.11 System Programming

- ◆ System Programmers are responsible for maintaining the systems software including the operating system.
- ◆ System software is a shared resource thus error in the software may be propagated through any application system that use it.
- ◆ Furthermore, system software often must operate in a privileged mode if it is to be able to perform its functions.
- ◆ Access privileges are sometimes can be abused, but controlling system programmers is a difficult task. They are highly skilled individuals who often work alone or in a small group.
- ◆ Moreover, they often work in crisis situations where the need to get a job running overrides the need to maintain control procedures.

2.25.2.12 LAN Administrator

- ◆ LA is responsible for Technical and Administrative control over the LAN. This is include ensuring transmission links are functioning correctly, backups of the system are occurring regularly and purchase of LAN hardware and software.
- ◆ In smaller installation this person may be responsible for security administration over the LAN. The LAN administrator should have no application responsibilities but may have end-user responsibility.

2.25.2.13 Help desk Administrator

- ◆ The HDA is responsible for monitoring, improving and controlling system performance.
- ◆ The HDA may be useful when data entry is not based upon a dedicated source document. If users are uncertain about the nature or format of the data to be entered into a particular field, they may ask the HDA to provide information to assist them.
- ◆ It may also describe the validation rules that apply to item. The HDA facility is especially important if inexperienced or novice users will submit data to the system.

SYSTEMS DEVELOPMENT PROCESS: Systems development refers to the process of examining a business situation with the intent of improving it through better procedures and methods. • System Analysis • System Design	REASONS FOR FAILURE: • Lack of senior management support for and involvement in information systems development • Shifting user needs • Development of strategic systems • New technologies • Lack of standard project management and systems development methodologies • Overworked or under-trained development staff • Resistance to change • Lack of user participation • Inadequate testing and user training	THE CHARACTERISTICS OF SYSTEM DEVELOPMENT METHODOLOGY: • The project is divided into a number of identifiable processes • Deliverables must be produced • Users, managers and auditors are required to participate in the project to provide sign-offs. • The system must be tested thoroughly • A training plan is developed • Formal program change controls are established • A post-implementation review of all developed systems must be performed
APPROACHES TO SYSTEM DEVELOPMENT • Waterfall: Linear framework type • Prototyping: Iterative framework type • Incremental: Combination of linear and iterative framework type • Spiral: Combination linear and iterative framework type • Rapid Application Development (RAD): Iterative Framework Type • Agile Methodologies	SYSTEM DEVELOPMENT LIFE CYCLE (SDLC): framework provides system designers and developers to follow a sequence of activities. It consists of a set of steps or phases in which each phase of the SDLC uses of results of the previous one.	THE PHASES INVOLVED IN THE SDLC: (<u>I</u> <u>R</u>equire <u>A</u> <u>D</u>esign <u>D</u>eveloper <u>T</u>o <u>I</u>mplement and <u>M</u>aintain) • Preliminary <u>I</u>nvestigation • Systems <u>R</u>equirements <u>A</u>nalysis • Systems <u>D</u>esign • Systems Acquisition & <u>D</u>evelopment • Systems <u>T</u>esting • Systems <u>I</u>mplementation • Post Implementation Review and <u>M</u>aintenance

STAGE-I: PRELIMINARY INVESTIGATION – OBJECTIVES Objective: To determine and analyze the strategic benefits in implementing the system through evaluation and quantification of – productivity gains; future cost avoidance; cost savings and intangible benefits like improvement in morale of employees.	1. IDENTIFICATION OF PROBLEM • Clarify and understand the project request • Determine the size of the project • Determine the technical and operational feasibility of alternative approaches • Assess costs and benefits of alternative approaches • Report findings to the management with recommendation outlining the acceptance or rejection of the proposal 2. IDENTIFICATION OF OBJECTIVE After the identification of the problem, it is easy to work out the objectives of the proposed solution.	3. DELINEATION OF SCOPE • Functionality requirements • Data to be processed • Control requirements • Performance requirements • Constraints • Interfaces • Reliability requirements Methods with the help of which the scope of the project can be analyzed are as follows: • Reviewing internal documents • Conducting interviews
4. FEASIBILITY STUDY: (LOBSTER Fry) • <u>L</u>egal • <u>O</u>perational • <u>B</u>ehavioural • <u>S</u>chedule/Time • <u>T</u>echnical • <u>E</u>conomical • <u>R</u>esources • <u>F</u>inancial	ESTIMATING COST & BENEFIT COSTS: • Development • Operational • Intangible BENEFITS: • Tangible • Intangible	

STAGE-II: SYSTEMS REQUIREMENTS ANALYSIS FACT FINDING TECHNIQUES (DIQO) <u>Documents</u> <u>Interviews</u> <u>Questionnaires</u> <u>Observations</u>	ANALYSIS OF PRESENT SYSTEM • Review <u>Historical</u> Aspects • Analyze <u>Inputs</u> • Review <u>Data</u> Files Maintained • Review <u>Methods</u>, Procedures & data Communications • Analyze <u>Output</u> • Review <u>Internal Controls</u> • <u>Model</u> the Existing Physical & Logical System • Undertake <u>Overall</u> Analysis	SYSTEMS ANALYSIS OF PROPOSED SYSTEMS: After each functional area of the present information system has been carefully analysed, the proposed system specifications must be clearly defined.
SYSTEM DEVELOPMENT TOOLS: 1. System Components & Flows • System Flow Chart (SFC) • Data Flow Diagram (DFD) - Data Source & destination - Data Flows - Transformation Process - Data Stores • System Components Matrix • CASE Tools 2. User Interface 3. Data Attributes & Relationship 4. Detailed System Process	STRUCTURED ENGLISH: Structured English, also known as Program Design Language (PDL) or Pseudo Code, is the use of the English language with the syntax of structured programming. Thus, Structured English aims at getting the benefits of both the programming logic and natural language. Program logic that helps to attain precision and natural language that helps in getting the convenience of spoken languages.	FLOWCHARTS: Flowcharting is a graphic technique that can be used by analysis to represent the inputs, outputs and processes of a business in a pictorial form. It is a common type of chart, that represents an algorithm or process showing the steps as boxes of various kinds, and their order by connecting these with arrows. Flowcharts are used in analyzing, designing, documenting or managing a process or program in various fields.

TYPES OF FLOW CHARTS • Document flowchart • Data flowchart • System flowchart • Program flowchart	BENEFITS OF FLOWCHART: • Communication • Effective analysis • Proper documentation • Efficient Coding • Proper Debugging • Efficient Program Maintenance	LIMITATIONS OF USING FLOWCHARTS: • Complex logic • Alterations and Modifications • Reproduction • The essentials of what is done can easily be lost in the technical details of how it is done.
DECISION TREE: A Decision Tree (or tree diagram) is a support tool that uses a tree-like graph or model of decisions and their possible consequences, including chance event outcomes, resource costs, and utility. DECISION TABLE: A Decision Table is a table which may accompany a flowchart, defining the possible contingencies that may be considered within the program and the appropriate course of action for each contingency Condition Stub – which comprehensively lists the comparisons or conditions; • Condition Stub • Action Stub • Condition entries • Action entries	DATA DICTIONARY: It is a computer file that contains descriptive information about the data items in the files of a business information system. In other words, it is a computer file about data. Uses: • Aids in documentation – To programmers & analysis • File Security • For Accountant – Planning flow of transaction data • For Auditors – Establish audit trail • Aids in investigation/documenting internal control procedures	<u>Data Dictionary Contains:</u> • Data item's length, type & range • Identify of source document used to create data item • Names of computer file storing data item • Names of computer programs that modify data item • Identify of individual permitted to access • Identify of individual not permitted to access

LAYOUT FORM AND SCREEN GENERATOR, MENU GENERATOR, REPORT GENERATOR, CODE GENERATOR • Layout form and Screen Generator • Menu Generator • Report Generator • Code Generator	SYSTEM SPECIFICATION At the end of the analysis phase, the systems analyst prepares a document called “Systems Requirement Specifications (SRS)”, it contains: • Introduction • Information Description • Functional Description • Behavioural Description • Validation Criteria • Appendix • SRS Review	ROLES INVOLVED IN SDLC • Steering Committee • Project Manager • Project Leader • Systems Analyst/Business Analyst • Module Leader/Team Leader • Programmer/Coder/Developer • Database Administrator (DBA) • Quality Assurance • Tester • Domain Specialist • IS Auditor
STAGE – III: SYSTEM DESIGN System design involves first logical design and then physical construction of a system. Design specifications instruct programmers about what the system should do. The programmers, in turn, write the programs that accept input from users, process data, produce the reports, and store data in the files.	THE DESIGN PHASE INVOLVES: • Architectural Design • Design of the Data/Information Flow • Design of the Database • Design of the User-interface • Physical Design • Design and acquisition of the hardware/system software platform	DESIGN OF DATABASE • Conceptual Modeling • Data Modeling • Storage Structure Design • Physical Layout Design

DESIGN OF USER-INTERFACE Output Objectives: • Convey info about past, current, future • Signal important events, opportunities, warnings • Trigger an action • Confirmation of action • Meets needs of organisation & users	Important factors in Input/Output design: (<u>CM</u>'s <u>F</u>raud on <u>TV</u>) • <u>C</u>ontent • <u>M</u>edia • <u>F</u>orm • <u>F</u>ormat • <u>T</u>imeliness • <u>V</u>olume	PHYSICAL DESIGN: Design Principles- • The recommended procedure is to design two or three alternatives & choose the best one on pre-specified criteria. • The design should be based on the analysis. • The software functions designed should be directly relevant to business activities. • The design should follow standards laid down. • The design should be modular.
Stage – IV – Part 1: SYSTEM ACQUISITION Acquisition Standards • Ensuring security, reliability and functionality already built into a product. • Ensuring managers complete appropriate vendor, contract, and licensing reviews. • Including invitations-to-tender and request-for-proposals. • Establishing acquisition standards to ensure functional, security and operational requirements to be accurately identified and clearly detailed in request-for-proposals.	ADVANTAGES OF APPLICATION SOFTWARE: (<u>R</u>oyal <u>C</u>hallengers <u>Q</u>uit <u>L</u>eague) • <u>R</u>apid implementation • <u>C</u>ost • <u>Q</u>uality • <u>L</u>ow risk METHODS OF VALIDATING PROPOSAL • Checklists • Point Scoring Analysis • Public Evaluation Reports • Benchmarking problem for vendor's proposal • Test problems	Validation of Vendors' proposals: (<u>V</u>ice-<u>C</u>hairman <u>M</u>anages <u>C</u>ompany's <u>P</u>ortfolio) • <u>V</u>endor Support • <u>C</u>ompatibility with Existing Systems • <u>M</u>aintainability of the proposed system • <u>C</u>ost benefits of the proposed system • <u>P</u>erformance rating of the proposed system in relation to its cost

STAGE – IV – Part 2: DEVELOPMENT (PROGRAMMING TECHNIQUES AND LANGUAGES) Objective: To convert the specification into a functioning system. Activities: Application programs are written, tested and documented, conduct system testing. Document/Deliverable: A fully functional and documented system.	Characteristics Of A Good Coded Program: • Reliability • Robustness • Accuracy • Efficiency • Usability • Readability	Program Debugging Debugging refers to correcting programming language syntax and diagnostic errors so that the program compiles cleanly. It consists of: • Inputting the source program to the compiler, • Letting the compiler find errors in the program, • Correcting lines of code that are erroneous, and • Resubmitting the corrected source program as input to the compiler.
STAGE – V: SYSTEM TESTING UNIT TESTING Categories of tests that a programmer typically performs on a program unit: • Functional Tests • Performance Tests • Stress Tests • Structural Tests • Parallel Tests	TYPES OF UNIT TESTING • Static Analysis Testing: - Desk Check - Structured walk-through - Code inspection • Dynamic Analysis Testing: - Black Box Testing - White Box Testing - Gray Box Testing	INTEGRATION TESTING • Bottom-up Integration • Top-down Integration • Regression Testing
SYSTEM TESTING • Recovery Testing • Security Testing • Stress or Volume Testing • Performance Testing	FINAL ACCEPTANCE TESTING • Quality Assurance Testing • User Acceptance Testing: - Alpha Testing - Beta Testing	

STAGE – VI: SYSTEM IMPLEMENTATION EQUIPMENT INSTALLATION • Site Preparation • Installation of New Hardware/Software • Equipment Checkout TRAINING PERSONNEL • System Operators Training • Users Training	CONVERSION PROCEDURE: <u>Activities for successful conversion:</u> (Fifa Played in South Africa) • <u>F</u>ile conversion • <u>P</u>rocedure conversion • <u>S</u>ystem conversion • <u>S</u>cheduling personnel and equipment • <u>A</u>lternative plans in case of equipment failure	<u>System Implementation Conversion Strategies:</u> • Direct Implementation/Abrupt change-over • Phased implementation • Pilot implementation • Parallel running implementation
STAGE VII – POST-IMPLEMENTATION REVIEW & MAINTENANCE • Development Evaluation • Operation Evaluation • Information Evaluation	SYSTEM MAINTENANCE • Scheduled maintenance • Rescue maintenance • Corrective maintenance • Adaptive maintenance • Perfective maintenance • Preventive maintenance	OPERATION MANUALS • A cover page, a title page and copyright page; • A preface and information on how to navigate the user guide; • A contents page; • A guide on how to use at least the main functions of the system; • A troubleshooting section; • A FAQ (Frequently Asked Questions); • Where to find further help, and contact details; • A glossary and an index.
