ACQUISITION, DEVELOPMENT AND IMPLEMENTATION OF INFORMATION SYSTEMS

- Information systems play a vital role in the success of any functional system today. It may reckoned as the symbiosis of IT hardware and software

Business Process Design

Business process design means structuring or restructuring the tasks, functionalities and activities for improvising a business system

Business Process Design involves a sequence of steps as follows:-

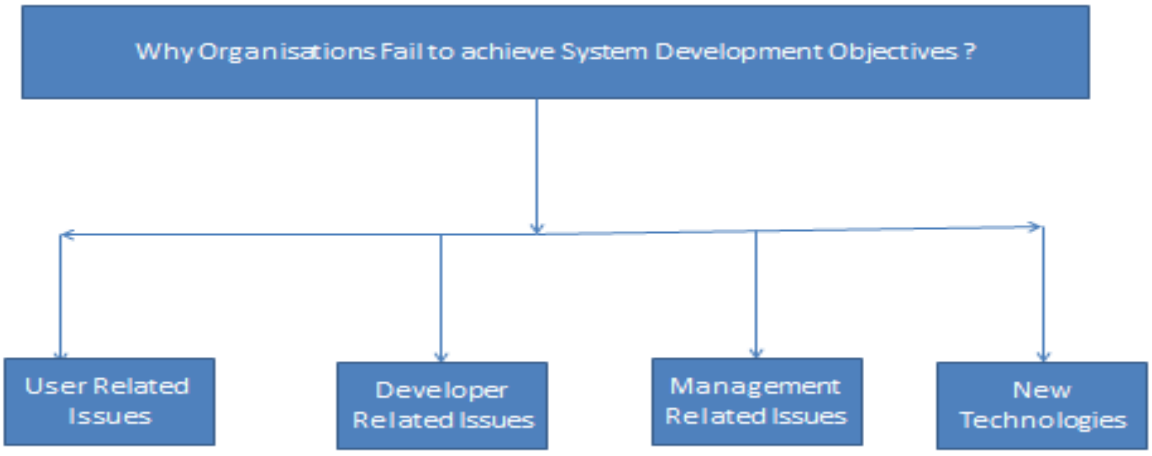
- Present Process Documentation:-Present business process is analysed and documented
- Proposed Process Documentation:-Design the new process requirements for the system.
- Implementation of New Process:-Implement largely the new as well as modified processes at the entity.

System Development

System development refers to the process of

- Examining a business situation with the intent of
- Improving through better procedure and methods
- Two Components of System Development are
    - System Analysis:- Gathering and interpreting facts
    - System Design:- planning and structuring a new business system

Achieving System Development Objectives



User related issues:-

- Shifting user needs:-User requirements are constantly changing-more requirements for IS development and more development projects
- Resistance to change:-People have tendency to resist changes
- Lack of user participation:- User must participate in development efforts
- Inadequate Testing and User training:- New systems must be tested before installation

Developer Related Issues

- Lack of standard project management and system development methodologies:-Some organisations do not formalise the same
- Overworked or undertrained development staff:- System developers often lack educational background and art skills

Management Related Issues

- Lack of Senior Management Support and involvement:-Developers and users watch senior management to determine which system development projects are important
- Development of Strategic Systems:- Strategic Decision Making is unstructured-requirements, specifications and objectives for such development projects are difficult to determine

New Technologies

Organisation tries to create a competitive advantage by applying advanced technologies, it finds it is difficult as personnel are not familiar with technology

System Development Team

- Project to be decided by a top management level steering committee
- Steering committee consists of group of key users of IT

Accountant's role in the Development work

Accountants can combine IT and internal control, also have specialised skills like accounting and auditing-can be applied to development project. For the analysis, the following data is to be generated

- Return on Investment:- Defines what an entity shall earn on a particular investment. This financial data is consideration for any expenditure company wishes to incur. For analysis the following is to be generated
    - Cost:-Estimate of the cost involved in the development which are
        - Development Cost:-E.g. Salaries of Developers
        - Operating Cost:- E.g. rental or depreciation charges, salaries of operators
        - Intangible Cost:- Cannot be easily measured, development of a new system may cause a loss of employee productivity or morale
    - Benefits:- Includes tangible and intangible benefits

- Computing the cost of IT implementation and cost benefit analysis:-For analysis of ROI, accountants need the cost and returns from the system development efforts
- Skills expected from an accountant:- Accountant, being an expert, must possess the skills to understand the system development efforts

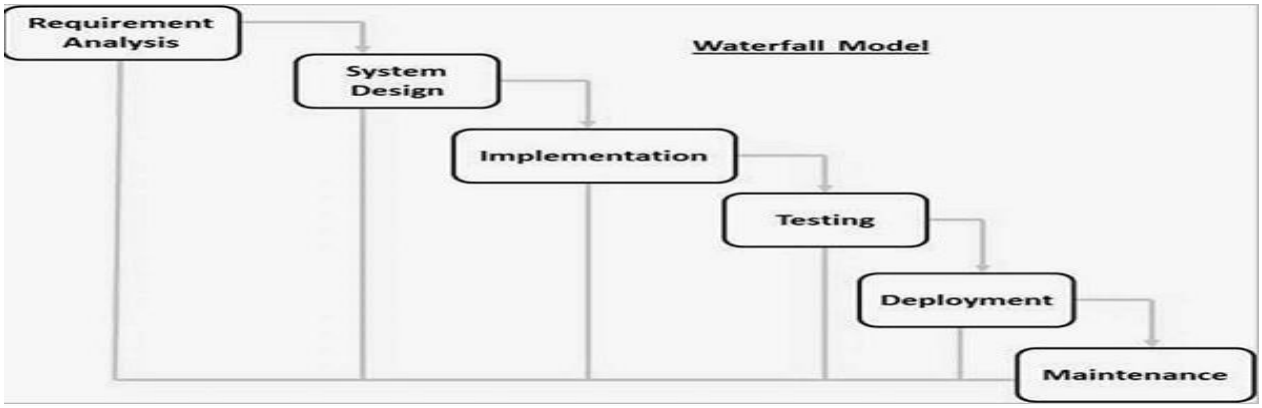
System Development Methodology

- It is a formalised, standardised, well organised and documented set of activities used to manage a system development project
- The methodology is characterised by the following
    - Project is divided into a number of identifiable processes –each having a starting and an end point
    - Specific reports and other documents called deliverables must be produced periodically
    - Users, managers and auditors are required to participate in the project
    - System must be tested thoroughly prior to implementation
    - Training plan developed for those who will operate and use the new system
    - Formal program change controls are established too preclude unauthorised changes
    - Post implementation review of all developed systems to be performed to assess efficiency and effectiveness
- Approaches for the System Development are
    - Waterfall Model
    - Prototyping Model
    - Incremental Model
    - Spiral Model
    - Rapid Application Development Model
    - Agile Model

Waterfall Model

- The waterfall Model illustrates the software development process in a linear sequential flow.
- This means that any phase in the development process begins only if the previous phase is complete. In this waterfall model, the phases do not overlap.
- The best examples of waterfall projects are huge undertakings that have to be completed in an all or nothing fashion, such as airplanes, automobiles, bridges.
    - An airplane has to be designed to every little detail upfront. This is possible mostly because we have been building airplanes for a lot longer than we've been building software. Furthermore, because any two airplanes share perhaps 80% of their DNA and any two software applications share perhaps 20% of their DNA.
    - Put another way, you could probably design 80% of an airplane using existing tried-and-tested components and only have to innovate 20% of

the design. On the other hand, you could probably design only 20% of a non-commodity software application or non-standard house using existing components and would have to innovate 80% of the design.



Characteristics of the Model

- Project is divided into sequential phases, with some overlap and splash back acceptable between phases
- Emphasis on Planning, time schedules, target dates, budgets and implementation of an entire system at one time
- Tight controls are maintained over the life of the project through extensive documentation, as well as formal reviews

Strengths

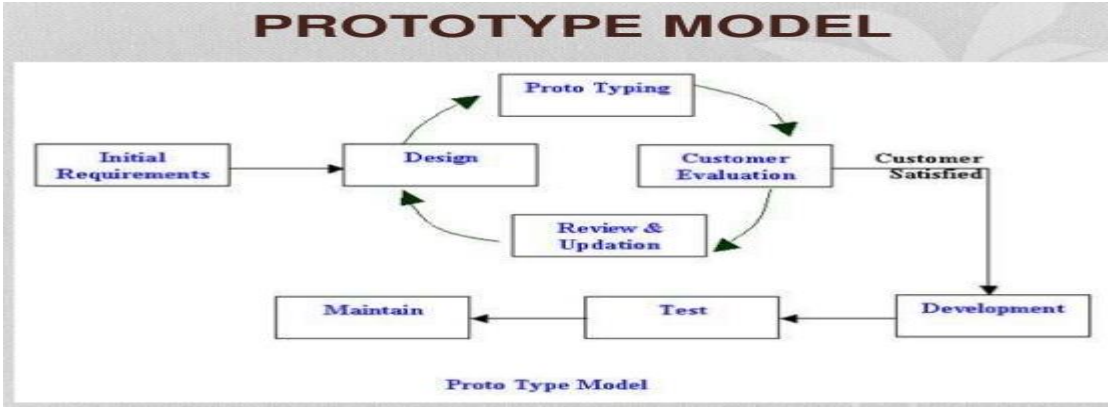
- Ideal for supporting less experienced project teams
- Orderly sequence helps to ensure quality, reliability, adequacy and maintainability of the developed software
- Progress of system development is measurable
- It enables to conserve resources

Weakness

- Criticised to be inflexible, slow, costly and cumbersome
- Project progresses forward with only slight movement backward
- Problems are not discovered until system testing
- It is difficult to respond to changes, which may occur in the later stages of system development life cycle

The Prototyping Model

- The goal of prototype is to develop a small or pilot version called a prototype of part or all of a system.
- Prototype is a usable system or system component that is built quickly and at a low cost
- Generic Phases of this model are:-
    - Identify Information System Requirements
    - Develop the initial prototype
    - Test and Review
    - Obtain user signoff of the approved prototype



- Prototype model should be used when the desired system needs to have a lot of interaction with the end users.
- Typically, online systems, web interfaces have a very high amount of interaction with end users, are best suited for Prototype model. It might take a while for a system to be built that allows ease of use and needs minimal training for the end user.

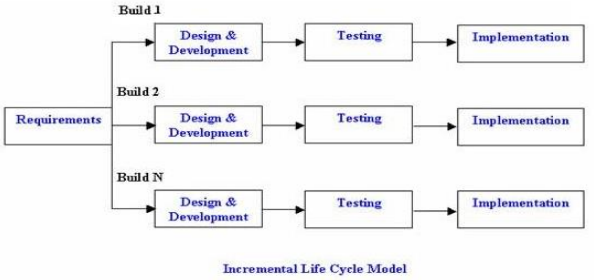
Strengths

- Improves user participation in system development
- Helps to easily identify, confusing or difficult functions and missing functionality
- Encourages innovation and flexible designs
- Very short time is required for developing and start experimenting with a prototype

Weaknesses

- Approval process and control are not strict
- Requirements may frequently change significantly
- Identification of non-functional elements is difficult to document
- Prototype may cause behavioural problems with system users

Incremental Model



Incremental Life Cycle Model

- The incremental build model is a method of software development where the product is designed, implemented and tested incrementally (a little more is added each time) until the product is finished.
- It involves both development and maintenance.
- The product is defined as finished when it satisfies all of its requirements
- E.g. Microsoft Office provides various products like Word, PowerPoint, Excel. Every version of MS Word are not same and doesn't have same features. Engineers will keep on updating features on the software for every release to make productivity. Engineers uses incremental model to develop MS Word, because each and every version of MS Word changes based on adding or removing features.

Strength

- Potential exists for exploiting knowledge gained in an early increment
- More flexible and less costly to change the scope and requirements
- Helps to mitigate integration and architectural risks earlier in the project
- Gradual implementation provides the ability to monitor the effect of incremental changes

Weakness

- Each phase is rigid and not overlap each other
- Problems may arise pertaining to the system architecture because not all requirements are gathered upfront
- Some modules completed earlier than others, well defined interfaces are needed
- Difficult to demonstrate the early success to management

Spiral Model

- Spiral model is a combination of sequential and prototype model.
- This model is best used for large projects which involves continuous enhancements.
- There are specific activities which are done in one iteration (spiral) where the output is a small prototype of the large software.
- The same activities are then repeated for all the spirals till the entire software is build.
- A spiral model has 4 phases:

  1. Planning phase
  2. Risk analysis phase
  3. Engineering phase
  4. Evaluation phase.

- Activities which are performed in the spiral model phases are shown below:

Phase Name	Activities performed	Deliverables / Output
Planning	-Requirements are studied and gathered. - Feasibility study - Reviews and walkthroughs to streamline the requirements	Requirements understanding document Finalized list of requirements.
Risk Analysis	Requirements are studied and brain storming sessions are done to identify the potential risks Once the risks are identified , risk mitigation strategy is planned and finalized	Document which highlights all the risks and its mitigation plans.
Engineering	Actual development and testing if the software takes place in this phase	Code Test cases and test results Test summary report and defect report.
Evaluation	Customers evaluate the software and provide their feedback and approval	Features implemented document

Spiral Model is used in the following scenarios:

- When the project is large.
- Where the software needs continuous risk evaluation.
- Requirements are a bit complicated and require continuous clarification.
- Software requires significant changes.
- Where enough time frames is there to get end user feedback.
- Where releases are required to be frequent.

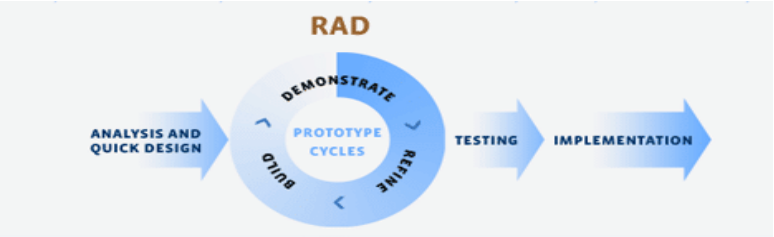
Strengths

- Development is fast
- Larger projects / software are created and handled in a strategic way
- Risk evaluation is proper.
- Control towards all the phases of development.
- More and more features are added in a systematic way.
- Software is produced early.
- Has room for customer feedback and the changes are implemented faster.

Weaknesses

- Risk analysis is important phase so requires expert people.
- Is not beneficial for smaller projects.
- Spiral may go infinitely.
- Documentation is more as it has intermediate phases.
- It is costly for smaller projects.

Rapid Application Development Model



- Rapid application development (RAD) describes a method of software development which heavily emphasizes rapid prototyping and iterative delivery.
- The RAD model is, therefore, a sharp alternative to the typical waterfall development model, which often focuses largely on planning and sequential design practices.
- Key Features
    - Fast Development and delivery of a high quality system at a relatively low investment cost
    - Attempts to reduce inherent product risk
    - Aims to produce high quality systems quickly
- Strengths
    - Operational version available much earlier than waterfall, increment or spiral frameworks
    - Quick initial reviews are possible
    - Constant integration isolates problems and encourages customer feedback
    - Provides for the ability to rapidly change system design as demanded by the users
    - Leads to a tighter fit between user requirements and system specifications
- Weaknesses

- Fast speed and lower cost may affect adversely the system quality
- Project may end up with more requirements than needed
- May lead to inconsistent designs within and across systems
- May call for violation of programming standards related to inconsistent naming conventions and inconsistent documentation

Agile Model

- Agile software development describes an approach to software development under which requirements and solutions evolve through the collaborative effort of self-organizing cross-functional teams and their customer(s)/end users(s).
- It advocates adaptive planning, evolutionary development, early delivery, and continuous improvement, and it encourages rapid and flexible response to change.

- The Manifesto for Agile Software Development is based on twelve principles
  - Customer satisfaction by early and continuous delivery of valuable software
  - Welcome changing requirements, even in late development
  - Working software is delivered frequently (weeks rather than months)
  - Close, daily cooperation between business people and developers
  - Projects are built around motivated individuals, who should be trusted
  - Face-to-face conversation is the best form of communication (co-location)
  - Working software is the primary measure of progress
  - Sustainable development, able to maintain a constant pace
  - Continuous attention to technical excellence and good design
  - Simplicity—the art of maximizing the amount of work not done—is essential
  - Best architectures, requirements, and designs emerge from self-organizing teams
  - Regularly, the team reflects on how to become more effective, and adjusts accordingly

- Strengths of Agile Model

  - It has the concept of adaptive team, which enables to respond to changing requirements
  - Team does not have to invest time and efforts and finally find that by the time they have delivered the product, the requirement of customer has changed
  - Documentation is crisp and to the point to save time
  - End result is high quality software in least possible time duration and satisfied customer
- Weakness of Agile Model
  - In case of some software deliverables, especially the large ones, it is difficult to assess the efforts required at the beginning of the software development life cycle
  - There is lack of emphasis on necessary designing and documentation
  - Agile increases potential threats to business continuity and knowledge transfer. By nature Agile projects are extremely light on documentation because the team focuses on verbal communication with the customer rather than on documents and manuals

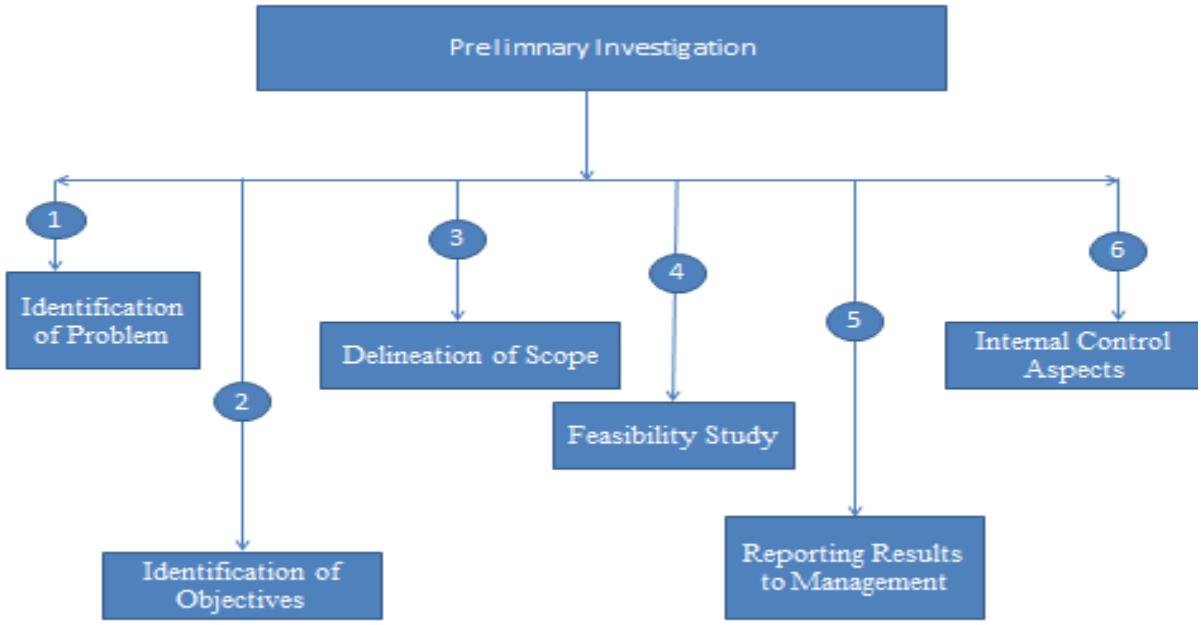
SYSTEM DEVELOPMENT LIFE CYCLE

- The System Development Life Cycle provides system designers and developers to follow a sequence of activities
- Advantages of this system are as follows
  - Better Planning and control by project managers
  - Compliance to prescribed standards ensuring better quality
  - Documentation that SDLC stresses on is an important measure of communication and control
  - Phases are important milestones and help the project manager and user for the review and signoff
- Perspective of IS Audit, following are the possible advantages
  - IS auditor can have a clear understanding of the phases of SDLC on the basis of documentation
  - IS auditor can state in his report about the compliance by the IS management of the procedures
  - IS auditor , having technical knowledge and ability of areas of SDLC, can be a guide during various phases of SDLC
  - IS auditor can provide an evaluation of the methods and techniques used through various development phases of SDLC
- Shortcomings and anticipated risks associated with SDLC are
  - Development team may find it cumbersome
  - Users find that the end product is not visible for long time
  - Rigidity of approach may prolong the duration of many projects
  - May not be suitable for small and medium sized projects

Activities of Phases involved in SDLC

- *Prelimnary investigation:-* Determining and evaluating strategic feasibility of system
‘
- *System Requirement Analysis:-* Analysing the system requirements
- *System Design:-* Design system in terms of user interface, data storage and processing function
- *Systems Acquisition:-* Acquisition of Operating Infrastructure
- *System Development:-* Developing the system as per the system designed
- *System Designing:-* Requisite testing to ensure the valid and reliable implementations
- *System implementation:-* Operationalisation of the developed system for acceptance
- *Post Implementation review and maintenance:-* Continous evaluation of the system as it functions in the live environment

Prelimnary Investigation



Identification of Problem

The first step in application development is to define the problem clearly and precisely, which is done only after the critical study of the existing system and several rounds of discussion with the user group

Analyst working in the Prelimnary investigation must accomplish the following objectives

- Clarify and understand the project request
- Determine the size of the project
- Determine the technical and operational feasibility of alternate approaches
- Assess Costs and benefits of the alternate approaches

- Report findings to the management with recommendation outlining the acceptance or rejection of the proposal

Identification of Objectives

After identifying the problem, it is easy to work out and specify the objectives of the proposed solution

Delineation of Scope

Scope should be clear and comprehensible to the user management stating the extent and what will be addressed by the solution and what will not be. Typical scope determination may be performed in the following dimensions

- Functionality requirements:- What functionalities will be delivered through the solution
- Data to be processed:- What data is required to achieve these functionalities
- Control Requirements:- What are the control requirements for these applications
- Performance Requirements:- What level of response time, execution time and throughput is required
- Constraints:- What are the conditions input data has to confirm to
- Interfaces:- Is there any special hardware/software that the application has to interface with
- Reliability Requirements:-Ability of system to remain uncorrupted in the face of deliberate misuse

While drawing out information for delineating the scope, aspects to be kept in mind are:-

- Different users present the problem and solution in different ways-developer to draw out information from the initiator of project- the need
- Understanding the profiles of the users-who may be from the operating levels
- Development organisation has to not only present the proposed solution but also the economic benefits to the organisation
- Also necessary to understand the impact of the solution on the organisation
- Other factors other than the economic benefits are also to be considered while deciding on a solution

Primary Methods by which scope of a project can be analysed are :-

- Reviewing Internal Documents:-Learn about the organisation- how does a department work and who are the persons responsible
- Conduct Interviews:- How the systems should operate

Feasibility Study

- It refers to the process of evaluating alternative systems through cost/benefit analysis so that the most feasible and desirable system can be selected for development

- Feasibility study is evaluated under the following dimensions
    - Technical:- is technology needed available
    - Financial:- Solution Viable financially
    - Economic:- Return on investment
    - Schedule/Time:- Can the system be delivered on time
    - Resources:- Human resources reluctant to the solution
    - Operation:- Will the solution work
    - Behavioural:- Solution going to bring any adverse effect on the quality of work life
    - Legal:- Is the solution valid in legal terms
- Technical Feasibility
    - Does the necessary technology exist to do what is suggested
    - Does proposed equipment have the technical capability to hold the data required to use the new system
    - Can the proposed application be implemented with the existing technology
    - Will the proposed system provide adequate response to enquiries
    - Can the system be expanded if developed
    - Are there technical guarantees of accuracy, reliability, ease of access and data security
        - Technical Issues to be Considered
            - Communication channel configuration
            - Communications channel
            - Communications network
            - Computer programs
            - Data storage medium
            - Input Medium
            - Operations
            - Printed Output
- Financial Feasibility
    - Cost considerations of the organisation-may be acceptable for the organisations with large turnover but not for others
- Economic Feasibility
    - It includes evaluation of the costs and benefits expected if the proposed system is implemented
    - The financial and economic questions raised by analysts during the prelimnary investigation are for the estimation of the following
        - Cost of conducting a full system investigation
        - Cost of hardware and software for the class of applications considered
        - Benefits in the form of reduced costs or fewer costly errors
        - The cost if nothing changes
- Schedule or time feasibility
    - How long a new system will take to become operational and communicating this information to the steering committee

- Resources Feasibility
    - Focusing on human resources
- Operational Feasibility
    - Concerned with ascertaining the views of the workers, employees, customers and suppliers of the use of computer facility
    - Some questions helping in testing the operational feasibility of a project may include the following
        - Support for the system from the top management and users
        - Current business methods acceptable to users
        - Users involved in the planning and development of the project
        - Will proposed system cause harm, will it produce poorer results
        - Individual performance be poorer after implementation than before
- Behavioural Feasibility
    - Refers to systems which is to be designed to process data and to produce the desired outputs
- Legal Feasibility
    - Whether any conflict between the newly proposed system and the organisation's legal objectives

Reporting Results to the Management

- Report to be accompanies by the covering letter summarising the results and makes recommendation regarding the further procedures

Internal Control Aspects

- Management is responsible for the implementation of the internal control system, controls to be in place during the development of the system
- Key Control Queries include
    - Problem definition is proper
    - All feasibility studies have been properly done
    - Results of feasibility studies have been properly documented

System Requirement Analysis

This phase includes the understanding of the current system, identifying the areas that need modification to solve the problem, determination of requirements and to have fair ideas about system development tools

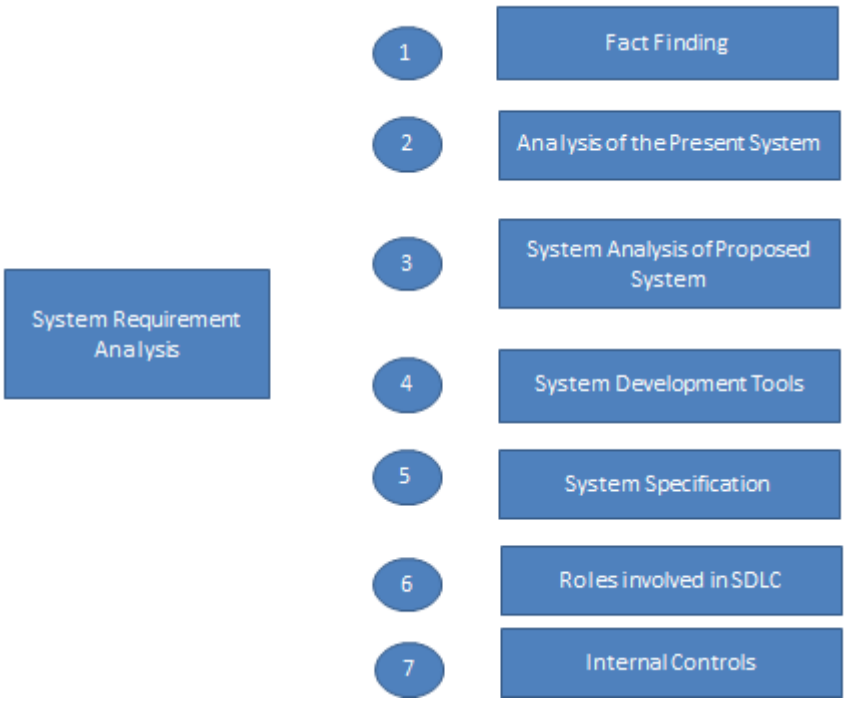
Following objectives are performed in this phase

- Identify and consult stakeholders to determine their expectations
- Analyse the requirements to detect and correct conflicts
- Gather data or find facts using tools like interviewing, research/document collection, questionnaires and observation
- Verify that the requirements are complete, consistent, unambiguous, verifiable

- Model activities such as developing models to document Data Flow Diagrams
- Document activities such as interview, questionnaires, reports etc.

Steps to achieve the above mentioned objectives are as follows

Generic Set of Processes here are as follows:-



Fact Finding

Every system is built to meet some set of needs and to assess these, the analysts interact with people. Various fact finding techniques are as follows

- Documents:- Manuals, input forms, output forms, diagrams of how the system works
- Questionnaires:-users are asked to complete a questionnaire
- Interviews:- Users and managers may be interviewed to extract information in depth
- Observation:- How the users react to the prototypes

Analysis of the Present System

Collecting, organising and evaluating facts about the system and the environment in which it operates. The following areas are to be studied in depth

- *Reviewing Historical Aspects*:- Review of the annual reports and organisation charts can identify the growth of the management levels as well as development of various functional areas and departments

- *Analysing inputs*:- System analyst to be aware of the sources from which the data is captured –outputs for one area may serve as inputs for the other
- *Reviewing Data Files* :- Analyst to check the systems and procedural manuals to find the data files maintained by each department, where they are located and who uses them
- *Reviewing Methods, Procedures and Data Communications*:- Ways in which a job is accomplished, equipment utilised and the actual location of the operations. Basic purpose is to eliminate unwanted tasks
- *Analysing outputs*:- Outputs/reports to be carefully scrutinised by the system analysts to determine how well they will meet the organisation needs
- *Reviewing internal controls*:- Locate control points, examination of the current internal control system will help in locating the weaknesses to be removed in the new system
- *Modelling the existing system:*- Processes to be documented after the review of inputs, methods, procedures, data files and data communications
- *Undertaking overall analysis of the existing system*:- Analysis of the present work volume, current personnel requirements, present cost-benefits of each of these

System Analysis of the Proposed Systems

Proposed system specifications are defined, determined from the desired objectives

Required system specification must be in conformity with the projects objectives

- Outputs are produced with emphasis on managerial reports
- Databases are maintained with the great accent
- Input data is prepared from the original source documents
- Methods and procedures- utilise communications where deemed appropriate
- Work volumes and timings are carefully considered for present and future periods

System Development Tools

Tools for system development specification can be classified into four categories as follows

- System Components and Flows :- help the system analysts to document the data flow among the major resources and activities of an information system
- User Interface:- Designing the interface between the end users and computer system is major consideration for system analyst while designing the new system
- Data Attributes and Relationships:- Data resources in information system are defined, catalogued and designed by this category of tools
- Detailed system Processes:- Tools help the programmer to develop detailed procedures and processes required in the design of a computer system

Set of tools are as follows

- Structured English/PDL(Program Design Language):- use of the English language with the syntax of structured programming Structured English aims at getting the benefits of both the programming logic and natural language
- Flowcharts :- Pictoral representation of the inputs, outputs and processes of a business processes
- Data Flow Diagrams:- Uses few simple symbols to illustrate the flow of data among external entities
- Decision Tree:- uses a tree like graph or model of decisions and their possible consequences
- Decision Table:- Accompany a flowchart and defining the possible contingencies that may be considered within the program and the appropriate course of action for each contingency
- CASE Tools:- CASE refers to the Computer Aided Software Engineering software packages. It refers to the automation of anything that humans do to develop systems and support virtually all phases of traditional system development process
- System Component Matrix:-Provides a matrix framework to document the resources used, activities performed and the information produced by an information system
- Data Dictionary:- Contains descriptive information about the data items in the files of a business information system
- User Interface layout and forms:- Several layout forms for both soft and hard copy are used to model input/output components of an automated information system. Prominent ones are as follows
    - Layout form and screen generator:- For printed report used to format or print the desired layout
    - Menu Generator:-Outlines the functions, which the system is aimed to accomplish
    - Report Generator:- Indicate totals, paging, sequencing and control breaks
    - Code Generator:-Allowing the analyst to generate modular units of source code

<u>System Specification</u>

At the end of the analysis phase, the system analyst prepares a document called System Requirement Specification –consisting of the following

- Introduction:-goals and objectives
- Information Description:- Problem description, information content, flow and structure
- Functional Description:- Diagrammatic description of functions
- Behavioural Description:- Response to external events and internal controls
- Validation Criteria:- Classes of test to be performed to validate functions, performance and constraints
- Appendices:- Data flow/object diagrams

- SRS review:- Development team makes an presentation and hands over the SRS document to be reviewed by the user or customer

Roles involved in SDLC

- Steering Committee:- Committee of experts to accord approvals for go ahead and implementations
- Project Manager:- Normally responsible for more than one project and liasonsing with the client or the affected functions
- Project Leader:- Dedicated to a project, who has to ensure its completion and the fulfilment of objectives
- System Analyst or Business Analyst:- Responsible to conduct interviews with users and understand their requirements
- Module Leader or Team Leader:- Project is divided into modules and each module is assigned to Module Leaders
- Programmer or Developers:- Programmers is the mason of the software industry, who converts design into programs by coding using programming language
- Database administrator:- Data in the database has to be maintained by a specialist in database administration so as to support the application program
- Quality Assurance:- Team setting the standards for development and checks the compliance with these standards
- Testers:- Junior level quality assurance personnel attached to a project
- Domain specialist:- Project team has to develop an application in a field that is new to them, they take the help of a domain specialist
- IS Auditor:- IS Auditor ensures that the application development also focuses on the control perspective

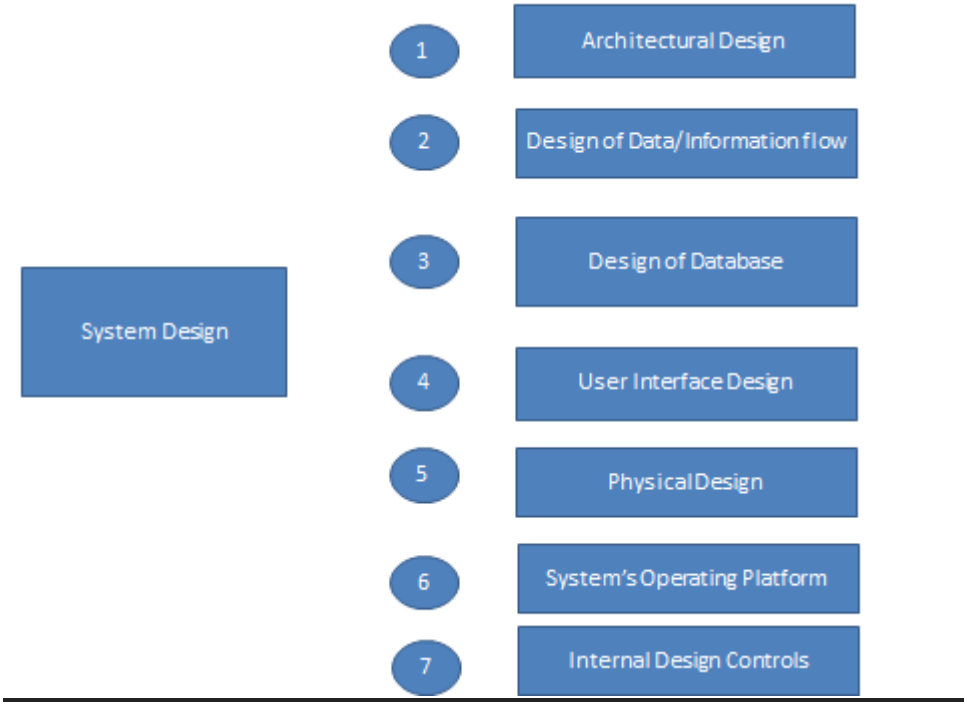
Internal Controls

Some of the key aspects here are:-

- Whether the present system analysis has been properly done
- Whether appropriate domain and expert was engaged
- Whether all user requirements of proposed system was considered
- Whether SRS document has been properly made and vetted by users, domain experts and system analysts

SYSTEM DESIGNING

- Objective is to design an information system that best satisfies the users/managerial requirements
- It sets out how the system shall be implemented using the chosen hardware, software and network facilities



Architectural Design

- Deals with the organisation of applications in terms of hierarchy of modules and sub-modules

Design of Data/Information flow

- Inputs required here are existing data/information flows, problems with the present system, and objective of the new system

Design of Database

- It involves the determining the scope ranging from the local to global structure
- Major Activities involved here are
    - Conceptual Modelling:- Describe the application domain, attributes and their relationships
    - Data Modelling:- Conceptual models to be translated into data models so that they can be accesses and manipulated
    - Storage Structure Design:-Decisions to be made how the data is to be partitioned so that it can be stored on some device
    - Physical layout design:- How to distribute the storage structure across specific storage media and locations

User Interface Design

- It involves how the users will interact with a system.

- Points to be considered are- source documents to capture raw data, hard copy-output reports, screen layouts, inquiry screens for database interrogation
- Important factors in Input/output design are as follows:-
    - Content:- -Consider the inputs to generate the outputs
    - Timeliness:- When users need outputs-Plan to be devised when the different types of inputs will enter the system
    - Format:- Manner in which the data is to be arranged-formats are designed on the basis of length of each field
    - Media:- Medium used for input, storage and output-various media like magnetic tape, voice input, etc.
    - Form:- Way information is inputted in the input form and presented to users-forms elicit and capture information required for the organisational personnel –Forms serve as source documents for data entry personnel
    - Input Volume/Output Volume:- amount of data that has to be entered into the system- DSS and real time processing systems, Input Volume is light whereas in Batch oriented systems, input volume could be high

Physical Design

- Logical design here is transformed into units, which can be decomposed further into implementation units such as programs and modules
- During this process the primary concern of the auditor is effectiveness and efficiency

System's Operating Platform

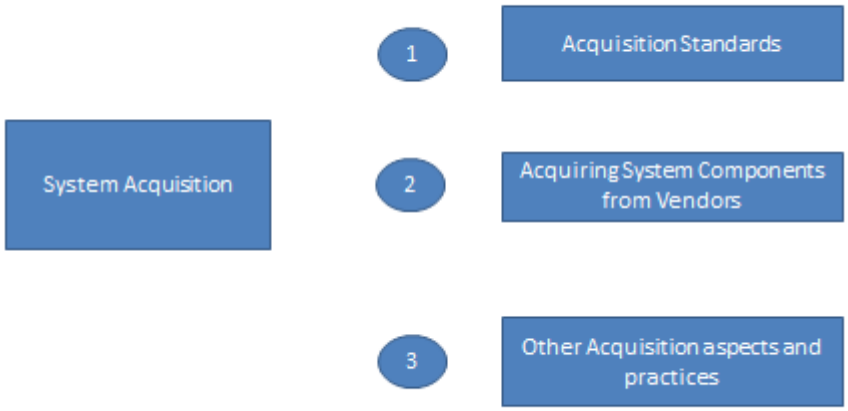
- New system may require an operating platform –hardware, network and system software which is not currently available in an organisation
- DSS may require a high quality graphics output not supported by existing hardware and software

Internal Control Designs

Controls aspects at this stage include the following

- Management reports were referred by the System Designer
- All Control aspects have been properly covered
- Controls put in place in system, appear in the documentation done at this stage
- Separate review of design document has been done by the internal auditor

SYSTEM ACQUISITION



Acquisition Standards

- Management to establish the acquisition standards that address the security and reliability issues
- Acqusition standards to focus on
    - Ensuring safety, reliability and functionality already built into a project
    - Ensuring managers complete appropriate vendor contract and licensing reviews and acquire products compatible with current systems
    - Invitation to tender-soliciting bids from the vendors for acquiring hardware or software
    - Request for Proposals-soliciting bids when acquiring off the shelf or third party developed software
    - Establishing acquisition standards- to ensure functional , security and operational requirements to be accurately identified and clearly detailed in RFP

Acquiring System Components from Vendors

- Acquiring the appropriate hardware and software is critical for the success of the whole project
- The following considerations are valid for both the acquisition of hardware and software
    - *Vendor Selection*: Vendor selection to be done before RFP. Vendor selection to be based on the parameters like background and location advantage, market feedback, price and services
    - *Geographical Location of Vendor*:- Local support persons whether are available with vendor

- *Presentation by selected vendors*: -Vendors whose proposals are selected are allowed to make presentation to system acquisition team
- *Evaluation of users feedback:-* analyse the feedback from users
- Other Considerations
  - Benchmark tests to be done for the proposed machine
  - Software considerations-current application programs or new programs that have been designed
  - Benchmarking processes to test whether computer offered meets requirements of the job on hand of the buyer
  - Tests to enable the buyer to evaluate performance

Other Acquisition Aspects and Practices

- Hardware Acquisition:-
  - Management depends on the vendor for the support services , system design, education and training-so it is not just about buying machine
- Software Acquisition
  - Nature of software application requirements to be assessed by the system analyst
  - What type of application software programs is needed and degree of processing that the system needs to handle
- Contracts, software licences and copyright violations
  - Contracts lay down the rights and responsibilities of party to a contract
  - Contracts to be in writing-goal is to provide authorisation
  - Copyright laws protect the proprietory as well as software
- Validation of Vendor Proposals
  - Evaluating and Ranking the proposals submitted by the vendors is quite difficult
  - Consider the following factors
    - Performance capability of each system
    - Costs and Benefits of each system
    - Maintainability of each system
    - Comparability of the proposed system with current system
    - Vendor Support

- Methods of Validating the Proposal
  - *Checklists*:- Questions put for which the responses of various vendors are validated
  - *Point Scoring Analysis*
    - It Provides a means of selecting a final system. No absolute rules in the selection system, online guidelines
    - Even a small business , evaluators may consider these issues

Software Evaluation Criteria	Possible Points	Vendor A	Vendor B	Vendor C
Does the software meet all mandatory specifications?	10	7	9	6
Will program modifications, if any, be minimal to meet company needs?	10	8	9	7
Does the software contain adequate controls?	10	9	9	8
Is the performance (speed, accuracy, reliability, etc.) adequate?	10	7	8	6
Are users satisfied with the software?	8	6	7	5
Is the software well documented?	10	8	8	7
Is the software compatible with existing company software?	10	7	9	8
Is the software user friendly?	10	7	8	6
Can the software be demonstrated and test driven?	9	8	8	7
Does the software have an adequate warranty?	8	6	7	6
Is the software flexible and easily maintained?	8	5	7	5
Is online inquiry of files and records possible?	10	8	9	7
Will the vendor keep the software up to date?	10	8	8	7
Totals	123	94	106	85

- *Public Evaluation Reports:-* Consultancy agencies compare and contrast the hardware and software performance for various manufacturers and publish their reports in this regard
- *Benchmarking Problems related to vendors solution*:-Accomplished by sample programs representing a part of the buyer's primary work load and include considerations
- *Testing Problems*:- Test problem results achieved by the machine can be compared and price performance judgements can be made

SYSTEM DEVELOPMENT: PROGRAMMING TECHNIQUES AND LANGUAGES

A Good Coded application and programs should have the following characteristics

- Reliability:- Consistency with which the program operates over a period of time
- Robustness:- Applications strength to uphold its operations in adverse situations
- Accuracy:- take care of what it should do and what it should not do
- Efficiency:- Performance per unit cost with respect to the relevant parameters
- Usability:- Refers to user friendly interface and easy to understand internal /external documentation
- Readability:-Ease of maintenance program even in the absence of program developer

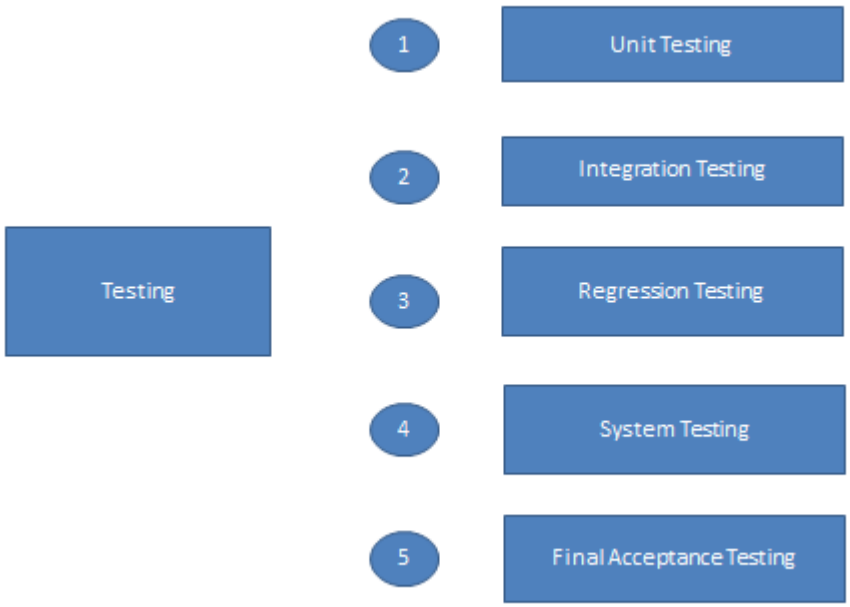
Other related aspects of this phase

- Program Coding Standards:-Logic of the program is converted into program statements or instructions at this stage

- Programming Language:-Application programs are coded in the form of statements or instructions and the same is converted by the complier to object code for the computer to understand and execute
- Program Debugging:- Correcting the programming language syntax and diagnostic errors so that the program complies cleanly
- Testing the Programs:- Programmer to plan testing to cover all the possible exceptions
- Program Documentation:- Writing of narrative procedures and instructions for people who will use the software and system behave as document indicates
- Program Maintenance:- Requirements of business data processing are subject to periodic change

SYSTEM TESTING

- Testing is a process used to identify the correctness, completeness and quality of developed computer software.



Unit Testing

- Unit testing is a software verification and validation method in which a programmer tests if individual units of source code a refit for use
- A unit is a smallest testable part of an application, which may be an individual program, function, procedure etc.
- Five categories of test performed by a programmer on the program unit. They are
    - Functional test:- Check whether programs do what they are supposed to do

- Performance Tests:- Verify the response time, execution time, throughput
- Stress Tests:- Used to determine the stability of a given system. It involves testing beyond the normal capacity
- Structural tests:- Examining the internal processing logic of a software system
- Parallel Tests:- Same test data is used in the new and old system and the output results are then compared

Terms of techniques, unit testing is classified as Static Analysis Testing and Dynamic Testing

- Static Testing: - are conducted on source programs and do not require the executions in operating conditions. IT includes the following
  - Desk Check:- Done by the programmer himself, checking logical syntax errors and deviation from the coding standards
  - Structured Walk through:- Application developer leads other programmers to scan through text of program and explanation to uncover errors
  - Code Inspection: - Program is reviewed by a formal committee. Review is done with formal checklists
- Dynamic Analysis Testing:- conducted through execution of programs in operating conditions
  - Let us take an example of a car to understand it in a better way. If you go to a showroom of a car to buy it, you might sit in the car to see if is comfortable and see what sound the doors make – this would be static analysis because the car is not being driven. If you take a test drive, then you would check that how the car performs when it is in the running mode e.g. the car turns right when you turn the steering wheel clockwise or when you press the break then how the car will respond and can also check the oil pressure or the brake fluid, this would be dynamic analysis, it can only be done while the engine is running.

  - Typical techniques for dynamic testing and analysis include
    - Black Box Testing
    - White Box Testing
    - Grey Box Testing
  - Black Box Testing:-
    - Black box testing is the Software testing method which is used to test the software without knowing the internal structure of code or program
  - White Box Testing
    - White box testing is the software testing method in which internal structure is being known to tester who is going to test the software.
  - Gray Box Testing
    - Gray box testing is the testing of software application using effective combination of both White box testing & Black box testing method. \
    - The white box testing means tester is aware of internal structure of code but the black box tester is not aware the internal structure of code.

        - In the **Gray box testing** tester is usually having knowledge of limited access of code

Integration Testing

- It is an activity of software testing in which individual software modules are combined and tested as a group
- This is carried out in the following two manners
    - **Top-down testing** is an approach to integrated testing where the top integrated modules are tested and the branch of the module is tested step by step until the end of the related module.
    - **Bottom-up testing** is an approach to integrated testing where the lowest level components are tested first, then used to facilitate the testing of higher level components. The process is repeated until the component at the top of the hierarchy is tested.

Regression Testing

- Regression testing is a type of software testing that ensures that previously developed and tested software still performs the same way after it has changed or interfaced with other software.
- It ensures that the changes or corrections have not introduced new faults

System Testing

- System testing of software or hardware is testing conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements.
- The types of testing that may be carried out are as follows
    - Recovery Testing
        - Recovery testing is a type of non-functional testing technique performed in order to determine how quickly the system can recover after it has gone through system crash or hardware failure.
    - Security Testing
        - Security testing is a process intended to reveal flaws in the security mechanisms of an information system that protect data and maintain functionality as intended.
    - Stress or Volume Testing
        - Used to determine the stability of a given system or entity-testing beyond capacity
    - Performance Testing: Used to determine the speed or effectiveness of a computer network, software program or device

Final Acceptance Testing

- Conducted when a system is just ready for implementation.

- During the testing it is ensured that the new system satisfies the quality standards adopted by the business and system satisfies the users
- Final Acceptance testing has two parts
    - Quality Assurance Testing:- New system satisfies the prescribed quality standards
    - User Acceptance Testing:- Ensures that the functional aspects are well addressed in the new system Two types of user acceptance testing
        - Alpha Testing:- **Alpha Testing** is a type of testing conducted by a team of highly skilled testers at development site
        - Beta Testing: - Beta Testing is done by customers or end users at their own site.

Internal Testing Controls

Generic key control aspects to be addressed by responses to following queries

- CASE tools have been used
- Test results have been documented
- Tests have been performed in the correct order
- Modifications needed based on test results have been done

SYSTEM IMPLEMENTATION

In terms of output, a fully functional as well as a documented system is a perquisite

Generic activities involved in system implementation include the following

- 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

Equipment Installation

- Hardware required to support the new system us elected prior to the implementation phases
- Adequate time to be scheduled to allow completion of the following activities
    - Site Preparation:- Appropriate location or ambiance is required
    - Installation of New Hardware /Software:- Equipment must be physically installed by the manufacturer, connected to the power source and wired to communication lines
    - Equipment Checkout:- The equipment must be turned on for testing under normal conditions

Training Personnel

- Quality of the training received by the personnel helps or hinders the successful implementation of the information system

System Changeover strategies

- Direct implementation/Abrupt Changeover:-
    - In a direct changeover, the entire system is replaced in an instant.
    - Basically, as soon as the new system is powered up, the old system is shut down.
    - This type of changeover carries the most risk because, if something goes wrong, reverting back to the old system usually is impossible

- Phased Changeover:-
    - Phased implementation is a method of changing from an existing system to a new one.
    - Phased implementation is a changeover process that takes place in stages.
        - As an example, think of a supermarket. In this supermarket the checkout system is being upgraded to a newer version. Imagine that only the checkout counters of the vegetable section are changed over to the new system, while the other counters carry on with the old system. If the new system does not work properly, it would not matter because only a small portion of the supermarket has been computerised. If it does work, staff can take turns working on the vegetable counters to get some practice using the new system
- Pilot Changeover
    - Changing over in a small part of the company to start with.
    - Only when the system is deemed satisfactory will it be rolled out to the rest of the organisation.
    - A supermarket introducing a new 'self-scanning' system might choose to introduce it in two or three stores at first. This is a pilot changeover.
- Parallel Changeover
    - Running both the old and new system until you are certain the new system is working correctly.
    - Parallel running is likely to be the most expensive as it involves doing the work twice for a period of time.
    - However, it is the safest.
    - If there are any bugs in the new system, you can always go back to the old system while the problems are corrected.

Changeover or conversion needs activities which must be completed to successfully convert from the previous system to the new information. These technical activities can be classified as follows

- Procedure Conversion:-

        - Operating procedure to be completed with sufficient documentation for the new system
    - File Conversion
        - Large number of files to be converted therefore this process is to be started long before programming and testing are completed
    - System Conversion
        - After on-line and off-line files have been converted and reliability of new system has been confirmed for a functional area, daily processing can be shifted from existing information system to the new one
    - Scheduling Personnel and equipment
        - Schedules to be set up by the system manager in conjunction with the departmental managers of operational units serviced by the equipment

POST IMPLEMENTATION REVIEW AND SYSTEM MAINTENANCE

Post implementation review answers question whether the organisation achieved what it did set in business terms

Post implementation review to be checked after sometime after the system has been implemented Typical evaluations include

- Development Evaluation:- System was developed on schedule and within budget
- Operational Evaluation:- Whether the hardware software and personnel are capable to perform their duties
- Information Evaluation:-Information System to be evaluated in terms of the information it provides or generates

System Maintenance

Maintenance can be characterised in the following ways

- Scheduled Maintenance:- Can be planned for in advance
- Rescue Maintenance:- Refers to previously undetected malfunctions which were not anticipated but require immediate troubleshooting solution
- Corrective Maintenance:- Fixing Bugs in the code or defects found during executions
- Adaptive Maintenance:- Adapting software to changes in the environment
- Perfective Maintenance:-Accommodating to the new or changed user requirements and
- Preventive Maintenance:- Activities aimed at increasing the system's maintainability, such as updating documentation, adding comments, and improving modular structure of the system

Auditor's Role in SDLC

Control Consideration for the auditor are as follows

- Documented policy and procedures
- Established project team with all infrastructure and facilities
- Developers/IT Managers are trained on the procedures
- Separate test environment for development/test/production/test plans
- Version Control on Programs
- Source code is properly secured