

5. Acquisition, Development and Implementation of Information Systems

5.1 Business Process Design

- Business process design means structuring or restructuring the tasks, functionalities and activities for improvising a business system.
- Business Process Design is a critical step to understand the requirements of the system. It needs a lot of intellectual capability from team of developers doing the same.
- Business Process Design is largely based on nature of system i.e. whether it is typically integrated, automatic and manual. Each nature of system needs a special design consideration.
- Business process design involves a sequence of the steps described briefly in the following sections.

1) Present Process Documentation

- In this step, the present business process is analyzed and documented.
- The key deliverable of this step includes the well-defined short-comings of the present processes and the overall business requirements.
- This step includes the following activities:
 - Understanding the business and the objectives for which it exists;
 - Documenting the existing business processes; and
 - Analysis of the documented processes.

2) Proposed Process Documentation

- This step is to design the new process requirements for the system. The design is based on the new system requirements and the changes proposed.
- The activities include the following:
 - Understanding of the business processes necessary to achieve the business objectives;
 - Designing the new processes; and
 - Documentation of the new process, preferably using of CASE tools.

3) Implementation of New Process

- This step is to implement largely the new as well as modified processes at the entity.
- The critical activities may include the following:
 - Validating the new process;
 - Implementing the new process; and
 - Testing the new process.

5.1.1 Case Studies –

To understand these aforementioned facts in a more practical way, some case studies are given as follows:

- **Case 1: Manual Billing System –**

- ✦ The billing clerk checks the approved price list of the products as is applicable on the date of billing, checks for discounts and bills the products to customer.
- ✦ In the above process, the key issue being that the billing clerk needs to possess, applicable and authorized price list with him.
- ✦ Business Process Design shall need to address the following critical issues:
 - Availability of approved price lists with billing clerk.
 - How it shall be ensured that the billing has been done on applicable rates only?
 - How the approval is accorded to price lists?
 - Whether the price list updating process is pre-defined or need based?
 - How exceptions to listed price shall be documented?
 - How the exceptions shall be submitted to management?

- **Case 2: Automatic Billing System –**

- ✦ The billing clerk selects the product from product lists and the system automatically picks up the price, as it is already available in data base.
- ✦ There is no option available with billing clerk to modify price at time of billing. The key processes that need to be controlled include, ensuring the system price list is the approved price list.
- ✦ Business Process Design shall need to address the following issues:
 - Availability of approved price lists in system.
 - How it shall be ensured that the price lists in system cannot be modified?
 - How the approval is accorded to price lists?
 - Whether the price list updating process is pre-defined or need based?
 - How the said price lists are updated in system?
 - How the correctness of updates is validated?
 - How exceptions, where the billed price is different from price in system authorized?
 - How the exceptions to price reported to management?
 - Does system provide a separate report for the same?

- **Case 3: Integrated Systems –**

- ✦ Integrated system means a system, which has been tightly interfaced with the business processes of entity. This shall require greater intellectual inputs to the process of business process modeling.
- ✦ The key issue to be addressed being the interface between the business process and objective.
- ✦ The issue to be addressed in addition to those discussed above shall include following:
 - How the business objective change is built into business process change?
 - How the business processes shall be documented?

5.2 System Development

- Systems development refers to the process of examining a business situation with the intent of improving it through better procedures and methods.
- System development can generally be thought of as having two major components as follows:
 - i). **System Analysis** is the process of gathering and interpreting facts, diagnosing problems, and using the information to recommend improvements to the system.
 - ii). **System Design** is the process of planning and structuring a new business system or one to replace or complement an existing system.

5.2.1 Achieving System Development Objectives

- An analysis on 'why organizations fail to achieve their systems development objectives' reveals bottlenecks. Some of the most notable ones are as follows:

1) User Related Issues –

It refers to those issues where user/customer is reckoned as the primary agent. Some of the aspects with regard to this problem are mentioned as follows:

i). Shifting User Needs –

- ✦ When user requirements for IT changes occur during development process, the development team faces the challenge of developing systems whose very purpose might change since the development process began.

ii). Resistance to Change –

- ✦ When personnel perceive that the project will result in personnel cutbacks, threatened personnel will dig in their heels, and the development project is doomed to failure.

iii). Lack of User Participation –

- ✦ Users must participate in the development efforts to define their requirements, feel ownership for project success, and work to resolve development problems.

iv). Inadequate Testing and User Training –

- ✦ New systems must be tested before installation to determine that they operate correctly. Users must be trained to effectively utilize the new system.

2) Developer Related Issues –

It refers to the issues and challenges with regard to the developers. Some of the critical bottlenecks are mentioned as follows:

i). Lack of Standard Project Management and System Development Methodologies –

- ✦ Some organizations do not formalize their project management and system development methodologies, thereby making it difficult to complete projects on time or within budget.

ii). Overworked or Under-Trained Development Staff –

- ✦ In many cases, system developers often lack sufficient educational background and requisite state of the art skills.

3) Management Related Issues –

It refers to the bottlenecks with regard to organizational set up, administrative & overall management to accomplish the system development goals. Some of bottlenecks are as follows:

i). Lack of Senior Management Support and Involvement –

- ✦ Developers and users of information systems watch senior management to determine ‘which systems development projects are important’ and act accordingly by shifting their efforts away from any project, which is not receiving management attention.

ii). Development of Strategic Systems –

- ✦ Because strategic decision making is unstructured, the requirements, specifications, and objectives for such development projects are difficult to define.

4) New Technologies –

- ✦ When an organization tries to create a competitive advantage by applying advance technologies, it generally finds that attaining system development objectives is more difficult because personnel are not as familiar with the technology.

5.2.2 System Development Team

- Several people in the organization are responsible for systems development. In large systems, the worth of a particular project is typically decided by a top management level steering committee.
- Such committee usually consists of a group of key users of Information Systems services. The steering committee ensures that ongoing systems development activities are consistently aimed at satisfying the information requirements of managers and users within the organization.
- A project management team consists of both computer professional & key users. In any organization, systems designers take a lead role during the design, development and implementation stages.
- In end-user developed systems, the end-user is ultimately responsible for the system. Generally, the end-user seeks guidance from information centre personnel while developing the system.

5.2.3 Accountants’ Involvement in Development Work

- Most accountants have specialized skills - such as accounting and auditing - that can be applied to the development project.
- An accountant can help in various related aspects during system development, which are as follows:

i). Return on Investment (ROI) –

- ✦ ROI defines the return, an entity shall earn on a particular investment i.e. capital expenditure. The analysis ideally needs to be done before the start of the development efforts for better decision making by management.
- ✦ For this analysis following data needs to be generated.

a) Cost –

This includes estimates for typical costs involved in the development, which are as follows:

- **Development Costs** – Development Costs for a computer based information system include costs of the system development process, like salaries of developers.
- **Operating Costs** – Operating Costs of a computer based information system including hardware/software rental or depreciation charges; salaries of computer operators.
- **Intangible Costs** – Intangible Cost that cannot be easily measured.

b) Benefits –

- The benefits, which result from developing new or improved information systems that can be subdivided into tangible and intangible benefits. A post implementation analysis is also done to see how the system development effort has benefitted an organization.

ii). Computing Cost of IT Implementation and Cost Benefit Analysis –

- ✦ For analysis of ROI, accountants need the costs and returns from the system development efforts. Accountants shall be the person to whom management shall look for the purpose.

iii). Skills expected from an Accountant –

- ✦ An accountant, being an expert in accounting field must possess skills to understand the system development efforts and nuances of the same.

5.3 Systems Development Methodology

- A System Development Methodology is a formalized, standardized, well-organized and documented set of activities used to manage a system development project.
- A System Development Methodology refers to framework that is used to structure, plan and control the process of developing an information system. Each of the available methodologies is best suited to specific kinds of projects, based on various technical, organizational, project & team consideration.
- The methodology is characterized by the following:
 - ✦ The project is divided into a number of identifiable processes, and each process has a starting point and an ending point. Each process comprises several activities, one or more deliverables, and several management control points. The division of the project into these small, manageable steps facilitates both project planning and project control.
 - ✦ Specific reports and other documentation, called Deliverables must be produced periodically during system development to make development personnel accountable for faithful execution of system development tasks.
 - ✦ Users, managers, and auditors are required to participate in the project, which generally provide approvals, often called signoffs, at pre-established management control points. Signoffs signify approval of the development process and the system being developed.
 - ✦ The system must be tested thoroughly prior to implementation to ensure that it meets users' needs as well as requisite functionalities.
 - ✦ Formal program change controls are established to preclude unauthorized changes to computer programs.

- ✦ A training plan is developed for those who will operate and use the new system.
- ✦ A post-implementation review of all developed systems must be performed to assess the effectiveness and efficiency of the new system and of the development process.
- These approaches are established as models and include Waterfall, Prototyping, Incremental, Spiral, Rapid Application Development (RAD) and Agile Methodologies models.

5.3.1 Waterfall Model

- The waterfall approach is a traditional development approach in which each phase is carried in sequence or linear fashion.
- These phases include requirements analysis, specifications and design requirements, coding, final testing, and release.
- Some of the key characteristics are the following:
 - ✦ Project is divided into sequential phases, with some overlap and splash back acceptable between phases.
 - ✦ Emphasis is on planning, time schedules, target dates, budgets and implementation of an entire system at one time.
 - ✦ Tight control is maintained over the life of the project through the use of extensive written documentation, as well as through formal reviews and approval/signoff.
- **Strengths –**
 - ✦ It is ideal for supporting less experienced project teams and project managers or project teams, whose composition fluctuates.
 - ✦ The orderly sequence of development steps and design reviews help to ensure the quality, reliability, adequacy and maintainability of the developed software.
 - ✦ Progress of system development is measurable.
 - ✦ It enables to conserve resources.
- **Weaknesses –**
 - ✦ It is inflexible, slow, costly and cumbersome due to significant structure and tight controls.
 - ✦ Project progresses forward, with only slight movement backward.
 - ✦ There is a little to iterate, which may be essential in situations.
 - ✦ It depends upon early identification and specification of requirements, even if the users may not be able to clearly define 'what they need early in the project'.
 - ✦ Requirement inconsistencies, missing system components 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 and under capacity may be difficult to correct.

- ✦ It is difficult to respond to changes, which may occur later in the life cycle, and if undertaken it proves costly and are thus discouraged.
- ✦ It leads to excessive documentation.
- ✦ Written specifications are often difficult for users to read and thoroughly appreciate.
- ✦ It promotes the gap between users and developers with clear vision of responsibility.

5.3.2 The Prototyping Model

- A prototype is a usable system or system component that is built quickly and at a lesser cost, and with the intention of modifying/replicating or even replacing it by a fully operational system. The goal of prototyping approach is to develop a small or pilot version called a prototype of part or all of a system.
- As users work with the prototype, they learn about the system criticalities and make suggestions about the ways to manage it. These suggestions are then incorporated to improve the prototype.
- Finally, when a prototype is developed that satisfies all user requirements, either it is refined and turned into the final system or it is scrapped. If it is scrapped, the knowledge gained from building the prototype is used to develop the real system.
- The generic phases of prototyping model are explained as follows:

i). Identify Information System Requirements –

- Under prototype approach, the design team needs only fundamental system requirements to build the initial prototype, the process of determining them can be less formal and time consuming than when performing traditional systems analysis.

ii). Develop the Initial Prototype –

- The designers create an initial base model and emphasize system characteristics such as simplicity, flexibility, and ease of use. These characteristics enable users to interact with tentative versions of data entry display screens, menus, input prompts, and source documents.

iii). Test and Revise –

- After finishing the initial prototype, the designers first demonstrate the model to users and then give it to them to experiment. Using this feedback, the design team modifies the prototype and then resubmits the revised model to users for review. Thus iterative process of modification and reevaluation continues until the users are satisfied.

iv). Obtain User Signoff of the Approved Prototype –

- Users approve the final version of the prototype, which commits them to the current design and establishes a contractual obligation about what the system will, and will not, do or provide.

• Strengths –

- ✦ It improves both user participation in system development and communication among project stakeholders.
- ✦ It helps to easily identify, confusing or difficult functions and missing functionality.

- ✦ It is especially useful for resolving unclear objectives; developing and validating user requirements; experimenting with or comparing various design solutions.
- ✦ Potential exists for exploiting knowledge gained in early iteration as later iterations are developed.
- ✦ It enables to generate specifications for a production application.
- ✦ It encourages innovation and flexible designs.
- ✦ It provides for quick implementation of an incomplete, but functional, application.
- ✦ It typically results in a better definition of these users' needs and requirements than does the traditional systems development approach.
- ✦ A very short time period is normally required to develop and start experimenting with a prototype.
- ✦ Since system users experiment with each version of the prototype through an interactive process, errors are hopefully detected and eliminated early in the developmental process.
- **Weaknesses –**
 - ✦ Approval process and control are not strict.
 - ✦ Inadequate problem analysis may occur whereby only the most obvious and superficial needs will be addressed, resulting in current inefficient practices being easily built into the new system.
 - ✦ Requirements may frequently change significantly.
 - ✦ Identification of non-functional elements is difficult to document.
 - ✦ Designers may prototype too quickly, without sufficient upfront user needs analysis, resulting in an inflexible design with narrow focus that limits future system potential.
 - ✦ Prototype may not have sufficient checks and balances incorporated.
 - ✦ Prototyping can only be successful if the system users are willing to devote significant time in experimenting with the prototype and provide the system developers with change suggestions.
 - ✦ The interactive process of prototyping causes the prototype to be experimented with quite extensively. Inadequate testing can make the approved system error-prone.
 - ✦ Prototyping may cause behavioral problems with users. These problems include dissatisfaction by users if system developers are unable to meet all user demands as well as dissatisfaction and impatience by users when they have to go through too many interactions of the prototype.

5.3.3 The Incremental Model

- The Incremental model is a method of software development where the model is designed, implemented and tested incrementally until the product is finished. The product is defined as finished when it satisfies all of its requirements.
- The product is decomposed into a number of components, each of which are designed and built separately. Each component is delivered to the client when it is complete. This allows partial utilization of product and avoids a long development time.
- This model combines the elements of waterfall model with the iterative philosophy of prototyping.

- A few pertinent features are listed as follows:
 - ✦ 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.
 - ✦ Overall requirements are defined before proceeding to evolutionary, mini – Waterfall development of individual increments of the system.
 - ✦ The initial software concept, requirement analysis, and design of architecture and system core are defined using the Waterfall approach, followed by iterative Prototyping, which culminates in installation of the final prototype (i.e. working system).
- **Strengths –**
 - ✦ Potential exists for exploiting knowledge gained in an early increment as later increments are developed.
 - ✦ Moderate control is maintained over the life of the project through the use of written documentation and the formal review and approval/signoff.
 - ✦ Stakeholders can be given concrete evidence of project status throughout the life cycle.
 - ✦ It is more flexible and less costly to change scope and requirements.
 - ✦ It helps to mitigate integration and architectural risks earlier in the project.
 - ✦ It allows the delivery of a series of implementations that are gradually more complete and can go into production more quickly as incremental releases.
 - ✦ Gradual implementation provides the ability to monitor the effect of incremental changes, isolated issues and make adjustments before the organization is negatively impacted.
- **Weaknesses –**
 - ✦ When utilizing a series of mini-waterfalls for a small part of the system before moving onto the next increment, there is usually a lack of overall consideration of the business problem and technical requirements for the overall system.
 - ✦ Each phase of an iteration is rigid and do not overlap each other.
 - ✦ Problems may arise pertaining to system architecture because not all requirements are gathered up front for the entire software life cycle.
 - ✦ As some modules will be completed much earlier than others, well-defined interfaces are required.
 - ✦ It is difficult to demonstrate early success to management.

5.3.4 Spiral Model

- The Spiral model is a software development process combining elements of both design and prototyping-in-stage. It tries to combine advantage of top-down and bottom-up concept. It combines the features of the prototyping model and the waterfall model.
- The spiral model is intended for large, expensive and complicated projects. Game development is a main area where the spiral model is used and needed.

- A list of pertinent characterizing features includes the following:
 - ✦ The new system requirements are defined in as much detail as possible. This involves interviewing a number of users representing all the external or internal user and other aspect of existing system.
 - ✦ 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 analyzed and strategies are decided to use them. This phase has added specially in order to identify and resolve all possible risks in project development.
 - ✦ A first prototype of the new system is constructed from the preliminary design. This is usually a scaled-down system, and represents an approximation of the characteristics of the final product.
 - ✦ A second prototype is evolved by a fourfold procedure by 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; and constructing and testing the second prototype.
- **Strengths –**
 - ✦ It enhances the risk avoidance.
 - ✦ It is useful in helping for optimal development of a given software iteration based on project risk.
 - ✦ It can incorporate Waterfall, Prototyping, and Incremental methodologies as special cases in the framework, and provide guidance as to which combination of these models best fits a given software iteration, based upon the type of project risk.
- **Weaknesses –**
 - ✦ It is challenging to determine the exact composition of development methodologies to use for each iteration around the Spiral.
 - ✦ It may prove highly customized to each project, and thus is quite complex and limits reusability.
 - ✦ A skilled and experienced project manager is required to determine how to apply it to given project
 - ✦ No established controls exist for moving from one cycle to another cycle. Without controls, each cycle may generate more work for the next cycle.
 - ✦ There are no firm deadlines, cycles continue with no clear termination condition leading to, inherent risk of not meeting budget or schedule.

5.3.5 Rapid Application Development (RAD) Model

- Rapid Application Development (RAD) refers to a type of software development methodology; which uses minimal planning in favor of rapid prototyping. The lack of extensive pre-planning generally allows software to be written much faster, and makes it easier to change requirements.
- Key features include the following:
 - ✦ Key objective is fast development and delivery of a high quality system at a relatively low investment cost,
 - ✦ Attempts to reduce inherent project risk by breaking a project into smaller segments and providing more ease-of-change during the development process.

- ✦ Aims to produce high quality systems quickly, primarily through the use of iterative Prototyping (at any stage of development), active user involvement, and computerized development tools.
- ✦ Key emphasis is on fulfilling business need while technological excellence is of lesser importance.
- ✦ Project control involves prioritizing development and defining delivery deadlines or “timeboxes.”
- ✦ It includes Joint Application Development (JAD), where users are involved in system design, either through consensus building in structured workshops or electronically facilitated interaction.
- ✦ Active user involvement is imperative.
- ✦ Iteratively produces production software, as opposed to a throwaway prototype.
- ✦ Produces documentation necessary to facilitate future development and maintenance.
- ✦ Standard systems analysis and design techniques can be fitted into this framework.
- **Strengths –**
 - ✦ The operational version of an application is available much earlier than with Waterfall, Incremental or Spiral frameworks.
 - ✦ Because RAD produces systems more quickly and to a business focus, this approach tends to produce systems at lower cost.
 - ✦ Quick initial reviews are possible.
 - ✦ Constant integration isolates problems and encourages customer feedback.
 - ✦ It holds a great level of commitment from stakeholders, both business and technical, than Waterfall, Incremental, or spiral frameworks.
 - ✦ It concentrates on essential system elements from user viewpoint.
 - ✦ It provides for the ability to rapidly change system design as demanded by users.
 - ✦ It leads to a tighter fit between user requirements and system specifications.
- **Weaknesses –**
 - ✦ Fast speed and lower cost may affect adversely the system quality.
 - ✦ The project may end up with more requirements than needed (gold-plating).
 - ✦ Potential for feature creep where more and more features are added to the system over the course of development.
 - ✦ It may lead to inconsistent designs within and across systems.
 - ✦ It may call for violation of programming standards related to inconsistent naming conventions and inconsistent documentation,
 - ✦ It may call for lack of attention to later system administration needs built into system.
 - ✦ Formal reviews and audits are more difficult to implement than for a complete system.
 - ✦ Tendency for difficult problems to be pushed to the future to demonstrate early success to management.
 - ✦ As some modules will be completed much earlier than others, well-defined interface are required.

5.3.6 Agile Model

- This is an organized set of software development methodologies based on the iterative and incremental development, where requirements and solutions evolve through collaboration between self-organizing, cross-functional teams. It promotes adaptive planning, evolutionary development and delivery; time boxed iterative approach and encourages rapid and flexible response to change.
- It is a conceptual framework that promotes foreseen interactions throughout development life cycle.
- **Agile Manifesto is based on following 12 features:**
 - ✦ Customer satisfaction by rapid delivery of useful software;
 - ✦ Welcome changing requirements, even late in development;
 - ✦ Working software is delivered frequently (weeks rather than months);
 - ✦ Working software is the principal measure of progress;
 - ✦ Sustainable development, able to maintain a constant pace;
 - ✦ Close, daily co-operation between business people and developers;
 - ✦ Face-to-face conversation is the best form of communication (co-location);
 - ✦ Projects are built around motivated individuals, who should be trusted;
 - ✦ Continuous attention to technical excellence and good design;
 - ✦ Simplicity;
 - ✦ Self-organizing teams; and
 - ✦ Regular adaptation to changing circumstances.
- **Strengths –**
 - ✦ It has the concept of an adaptive team, which enables to respond to the changing requirements.
 - ✦ The team does not have to invest time and efforts and finally find that by the time they delivered the product, the requirement of the customer has changed.
 - ✦ Face to face communication and continuous inputs from customer representative leaves a little space for guesswork.
 - ✦ The documentation is crisp and to the point to save time.
 - ✦ The end result is the high quality software in least possible time duration and satisfied customer.
- **Weaknesses –**
 - ✦ 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.
 - ✦ Agile requires more re-work and due to the lack of long-term planning and the lightweight approach to architecture.
 - ✦ The project can get taken off track if the customer representative is not clear about final outcome.
 - ✦ Agile lacks the attention to outside integration.

5.4 System Development Life Cycle (SDLC)

- The System Development Life Cycle consists of a generic sequence of steps or phases in which each phase of the SDLC uses the results of the previous one.
- The SDLC is document driven. A phase of the SDLC is not complete until the appropriate documentation or artifact is produced. These are sometimes referred to as logical phase deliverables.
- A deliverable may be a substantial written document, a software artifact, a system test plan or even a physical object such as a new piece of technology that has been ordered and delivered.
- The System Development Life Cycle method can be thought as a set of activities that analysts, designers and users carry out to develop and implement an information system.
- **The advantages of this system are given 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; and
 - ✦ The phases are important milestone and help the project manager and user for review and signoff.
- **From the perspective of the IS Audit, the following are the possible advantages:**
 - ✦ The IS auditor can have clear understanding of various phases of the SDLC on the basis of the detailed documentation created during each phase of the SDLC.
 - ✦ The IS Auditor on the basis of his/her examination, can state in his/her report about the compliance by the IS management of the procedures, if any, set by the management.
 - ✦ The IS Auditor, if has a technical knowledge and ability of different areas of SDLC, can be a guide during the various phases of SDLC.
 - ✦ The IS auditor can provide an evaluation of the methods and techniques used through the various development phases of the SDLC.
- **Some of the shortcomings and anticipated risks associated with the SDLC are as follows:**
 - ✦ The development team may find it cumbersome.
 - ✦ The users may find that the end product is not visible for a long time.
 - ✦ The rigidity of the approach may prolong the duration of many projects.
 - ✦ It may not be suitable for small and medium sized projects.
- Table describes the activities of all the phases involved in the System Development Life Cycle.

S No	Phase Name	Nature of Activity
1	Preliminary Investigation	Determining and evaluating the strategic feasibility of the system and ensure that the solution fits the business strategy.
2	Systems Requirements Analysis	Analyzing the typical system requirements, in view if its functionalities, deliverables etc.

3	Systems Design	Designing the system in terms of user interface, data storage and data processing functions on the basis of requirement phase by developing the system flowcharts, system & data flow diagram, screens & reports.
4	Systems Acquisition	Acquisition of Operating infrastructure including hardware, software and services.
5	Systems Development	Developing the system as per the system designed in view of its adequate implementation to lead to fulfillment of requirements to the satisfaction of all the stakeholders.
6	Systems Testing	Requisite testing to ensure the valid and reliable implementations.
7	Systems Implementation	Operationalization of the developed system for the acceptance by management and user before migration of the system to the live environment and data conversion from legacy system to new system.
8	Post Implementation Review & Maintenance	Continuous evaluation of the system as it functions in the live environment and its updation / maintenance.

5.4.1 Preliminary Investigation

- The purpose of the preliminary investigation is to evaluate the project request feasibility. It is neither a designed study nor it includes the collection of details to completely describe the business system.
- Rather it relates to collection of information that permits committee members to evaluate the merits of the project request and make an informed judgment about the feasibility of the proposed project.
- The deliverable of preliminary investigation includes a report including feasibility study observations.
- The analyst working on the preliminary investigation should accomplish the following objectives:
 - 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; and
 - Report findings to the management with recommendation outlining the acceptance or rejection of the proposal.
- ✦ The steps involved in the preliminary investigation phase are as follows –
 - i). Identification of Problem
 - ii). Identification of Objectives
 - iii). Delineation of Scope
 - iv). Feasibility Study
 - v). Reporting results to Management
 - vi). Internal Control Aspects

These are describes in detail as follows –

1) Identification of Problem –

- ✦ The first step in an 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 discussions with the user group.

2) Identification of Objectives –

- ✦ After the identification of the problem, it is easy to work out and precisely specify the objectives of the proposed solution.

3) Delineation of Scope –

- ✦ The scope of a solution defines its typical boundaries. It should be clear and comprehensible to the user management stating the extent and 'what will be addressed by the solution and what will not'
- ✦ The typical scope determination may be performed on 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 this application?
 - **Performance Requirements –**
 - What level of response time, execution time and throughput is required?
 - **Constraints –**
 - What are the conditions the input data has to conform to?
 - **Interfaces –**
 - Is there any special hardware/software that the application has to interface with?
 - **Reliability requirements –**
 - Reliability of an application is measured by its ability to remain uncorrupted in the face of inadvertent / deliberate misuse and probability of failure-free operations.
- ✦ Moreover, while eliciting information to delineate the scope, few aspects needs to be kept in mind:
 - Different users may represent the problem and required solution in different ways. The system developer should elicit the need from the initiator of the project alternately called champion or executive sponsor of the project, addressing his concerns should be the basis of the scope.
 - While the initiator of the project may be a member of the senior management, the actual users may be from the operating levels in an organization.
 - It is necessary to understand the impact of the solution on the organization- its structure, roles & responsibilities. Solutions, which have wide impact are likely to meet with greater resistance.

- While presenting the proposed solution for a problem, the development organization has to clearly quantify the economic benefits to the user organization. The information required has to be gathered at this stage.
 - While economic benefit is a critical consideration when deciding on a solution, there are several other factors that have to be given weightage too.
- ✦ **Two primary methods with the help of which the scope of project can be analyzed are as follows:**
- i). **Reviewing Internal Documents –**
 - The analysts conducting the investigation first try to learn about the organization involved in, or affected by, the project.
 - Analysts can usually learn these details by examining organization charts and studying written operating procedures.
 - ii). **Conducting Interviews –**
 - Written documents tell the analyst how the systems should operate, but they may not include enough details to allow a decision to be made about the merits of a systems proposal.
 - To learn these details, analysts use interviews. Interviews allow analysts to know more about the nature of the project request and the reasons for submitting it.

4) Feasibility Study –

- A feasibility study is carried out by the system analysts, which 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.
- The Feasibility Study of a system is evaluated under following dimensions described as follows:
 - ✦ **Technical:** Is the technology needed available?
 - ✦ **Financial:** Is the solution viable financially?
 - ✦ **Economic:** Return on Investment?
 - ✦ **Schedule/Time:** Can the system be delivered on time?
 - ✦ **Resources:** Are human resources reluctant for the solution?
 - ✦ **Operational:** How will the solution work?
 - ✦ **Behavioral:** Is the solution going to bring any adverse effect on quality of work life?
 - ✦ **Legal:** Is the solution valid in legal terms?

The detailed description of each dimension is given as follows:

a) Technical Feasibility –

- It may try to answer, whether implementation of the project viable using current technology?
- It is concerned with issues pertaining to hardware and software. An analyst ascertains whether proposed system is feasible with existing/ expected computer hardware & software technology.

- The technical issues usually raised during feasibility stage of investigation include the following:
 - ✦ Does the necessary technology exist to do what is suggested (and can it be acquired)?
 - ✦ Does the proposed equipment have the technical capacity to hold the data required to use the new system?
 - ✦ Can the proposed application be implemented with existing technology?
 - ✦ Will the proposed system provide adequate responses to inquiries, regardless of the number or location of users?
 - ✦ Can the system be expanded if developed?
 - ✦ Are there technical guarantees of accuracy, reliability, ease of access, and data security?

b) Financial Feasibility –

- The solution proposed may be prohibitively costly for the user organization.
- Example, monitoring stock through VSAT network connecting multiple locations may be acceptable for an organization with high turnover. But may not be a viable for smaller ones.

c) Economic Feasibility –

- It includes an evaluation of all the incremental costs and benefits expected if the proposed system is implemented.
- The financial and economic questions raised by analysts during the preliminary investigation are for the purpose of estimating the following:
 - ✦ The cost of conducting a full systems investigation;
 - ✦ The cost of hardware and software for the class of applications being considered;
 - ✦ The benefits in the form of reduced costs or fewer costly errors; and
 - ✦ The cost if nothing changes (i.e. the proposed system is not developed).

d) Schedule or Time Feasibility –

- Schedule feasibility involves the design team's estimating how long it will take a new or revised system to become operational and communicating this information to the steering committee.

e) Resources Feasibility –

- This focuses on human resources. Implementing sophisticated software solutions becomes difficult at specific locations because of reluctance of skilled personnel to move to such location.

f) Operational Feasibility –

- It is concerned with ascertaining the views of workers, employees, customers and suppliers about the use of computer facility.
- Some questions, which help in testing operational feasibility of a project, may include following:
 - ✦ Is there sufficient support for the system from management and from users?
 - ✦ Are current business methods acceptable to users?

- ✦ Have the users been involved in planning and development of the project?
- ✦ Will the proposed system cause harm? Will it produce poorer results in any respect or area?
Will loss of control result in any areas? Will accessibility of information be lost?
- ✦ Will individual performance be poorer after implementation than before?

g) Behavioral Feasibility –

- It refers to the systems, which is to be designed to process data and produce the desired outputs.
- However, if the data input for the system is not readily available or collectable, then the system may not be successful.

h) Legal Feasibility –

- Legal feasibility is largely concerned with whether there will be any conflict between a newly proposed system and the organization's legal obligations.
- Any system, which is liable to violate the local legal requirements, should also be rejected.

5) Reporting Results to Management –

- ✦ After the analyst articulates the problem, defines the same along with its scope, s/he provides one or more solution alternatives and estimates the cost and benefits of each alternative and reports these results to the management.
- ✦ The report should be accompanied by a short covering letter of intent that summarizes the results and makes recommendation regarding further procedures. From the analysts report, management should determine what to do next.

6) Internal Control Aspects –

- ✦ Management implements proper internal control to ensure business objectives. Sec. 217 (2AA), Companies Act, 1956 – Directors are responsible to have proper internal control for a company.
- ✦ For validating control aspects in system, entity may have an internal audit team. Few large software developers engage outside experts for the same.
- ✦ However, the key control queries regarding various aspects at this stage may include the following:
 - Whether problem definition is proper?
 - Whether all feasibility studies have been properly done?
 - Whether results of feasibility studies have been documented?
 - Whether management report submitted reflects the outcome of feasibility studies done?

5.4.2 System Requirements Analysis

- This phase includes a thorough and detailed understanding of the current system, identifies the areas that need modification to solve the problem, the determination of user/managerial requirements and to have fair idea about various systems development tools.

- The following objectives are performed in order to generate System Requirement Specification (SRS):
 - ✦ To identify and consult the stake owners to determine their expectations and resolve their conflict;
 - ✦ To analyze requirements to detect and correct conflicts and determine priorities;
 - ✦ To gather data or find facts using tools like - interviewing, research/document collection, questionnaires, observation;
 - ✦ To verify that the requirements are complete, consistent, unambiguous, verifiable, modifiable, testable and traceable;
 - ✦ To model activities such as developing models to document Data Flow Diagrams, ER Diagrams; and
 - ✦ To document activities such as interview, questionnaires, reports etc. and development of a system dictionary to document the modeling activities.
- In order to accomplish the aforementioned objectives, a series of steps are taken, are as follows –
 - i). Fact Finding
 - ii). Analysis of the Present System
 - iii). System Analysis of Proposed Systems
 - iv). System Development Tools
 - v). Systems Specification
 - vi). Role involved in SDLC

A generic set of process are described as follows:

1) Fact Finding –

- Every system is built to meet some set of needs.
- To assess these needs, the analysts often interact extensively with people, who will be benefited from the system in order to determine ‘what are their actual requirements’.
- Various fact-finding techniques/tools are used by the system analyst for determining these needs/requirements are briefly discussed below:

a) Documents –

- ✦ Document means manuals, input forms, output forms, diagram of how current system works, organization charts showing hierarchy of users & manager responsibilities, job descriptions, procedure manuals, program codes for the applications associated with current system, etc.
- ✦ Documents are a very good source of information about user needs and the current system.

b) Questionnaires –

- ✦ Users and managers are asked to complete questionnaire about the information systems when the traditional system development approach is chosen.
- ✦ The main strength of questionnaires is that a large amount of data can be collected through a variety of users quickly.

c) Interviews –

- ✦ Users and managers may be interviewed to extract information in depth. The data gathered through interviews provide a larger picture of the problems and opportunities.
- ✦ Interviews also give analyst the opportunity to observe and record first-hand user reaction and to probe for further information.

d) Observation –

- ✦ In general, observation plays a central role in requirement analysis. Only by observing how users react to prototypes of a new system, the system can be successfully developed.

2) Analysis of the Present System –

- Detailed investigation of the present system involves collecting, organizing and evaluating facts about the system and the environment in which it operates.
- Survey of existing methods, procedures, data flow, outputs, files, input and internal controls that should be intensive in order to fully understand the present system and its related problems.
- For analysis of the present system, the following areas should be studied in depth:

a) Reviewing Historical Aspects –

- ✦ A brief history of the organization is a logical starting point for an analysis of present system.
- ✦ A review of annual reports and organization charts can identify the growth of management levels as well as the development of various functional areas and departments.
- ✦ The system analyst should investigate 'what system changes have occurred in past including operation' that have been successful or unsuccessful with computer equipment & technique.

b) Analyzing Inputs –

- ✦ A detailed analysis of present inputs is important since they are basic to the manipulation of data. Source documents are used to capture the originating data for any type of system.
- ✦ The system analyst must understand the nature of each form, 'what is contained in it', 'who prepared it', 'from where form is initiated', 'where it is completed', the distribution of form.
- ✦ If the analyst investigates these questions thoroughly, s/he will be able to determine how these inputs fit into the framework of the present system.

c) Reviewing Data Files –

- ✦ The analyst should investigate the data files maintained by each department, noting their number & size, where they are located, who uses them and number of times, these are used.
- ✦ The system analyst should also review all on-line and off-line files, as it will reveal information about data that are not contained in any outputs.

d) Reviewing Methods, Procedures and Data Communications –

- ✦ Methods and procedures transform input data into useful output. A method is defined as a way of doing something; a procedure is a series of logical step by which a job is accomplished.

- ✦ A system analyst needs to review and understand the present data communications used by the organization. S/he must review the types of data communication equipment.
- ✦ The system analyst must understand how the data communications network is used in the present system so as to identify the need to revamp network when new system is installed.

e) Analyzing Outputs –

- ✦ The outputs or reports should be scrutinized carefully by the system analysts in order to determine 'how well they will meet the organization's needs.
- ✦ The analysts must understand what information is needed and why, who needs it and when and where it is needed.

f) Reviewing Internal Controls –

- ✦ Locating the control points helps analyst to visualize essential parts & framework of a system.
- ✦ An examination of the present system of internal controls may indicate weaknesses that should be removed in the new system.

g) Modeling the Existing System –

- ✦ As the logic of inputs, methods, procedures, data files, data communications, reports, internal controls and other important items are reviewed and analyzed in a top down manner; the processes must be properly documented.

h) Undertaking Overall Analysis of the Existing system –

- ✦ The final phase of the detailed investigation includes the analysis of the present work volume; the current personnel requirements; the present costs-benefits of each of these must be investigated thoroughly.

3) System Analysis of Proposed Systems –

- After a thorough analysis of each functional area of the present information system, the proposed system specifications must be clearly defined, which are determined from the desired objectives.
- The required systems specifications should be in conformity with the project's objectives articulated and in accordance with the following:
 - ✦ Outputs are produced with great emphasis on timely managerial reports that utilize the management by exception' principle.
 - ✦ Databases are maintained with great accent on online processing capabilities.
 - ✦ Input data is prepared from original source documents for processing by the computer system.
 - ✦ Methods and procedures that show the relationship of inputs and outputs to the database, utilize data communications as, when and where deemed appropriate.
 - ✦ Work volumes and timings are carefully considered for present and future periods.

4) System Development Tools –

- Many tools have been developed to improve current information systems & to develop new ones.
- Such tools help end users and systems analysts primarily for the following:
 - ✦ To conceptualize, clarify, document and communicate the activities and resources involved in the organization and its information systems;
 - ✦ To analyze present business operations, management decision making and information processing activities of the organization; and
 - ✦ To propose and design new or improved information systems to solve business problems or pursue business opportunities that have been identified.
- **The major tools used for system development specification or representations can be classified into four categories based on the systems features.** These are described briefly as follows:

i). System Components and Flows –

- ✦ These tools help the system analysts to document the data flow among the major resources and activities of an information system.
- ✦ System flow charts are typically used to show the flow of data media as they are processed by the hardware devices and manual activities.
- ✦ A system component matrix provides a matrix framework to document the resources used, the activities performed and the information produced by an information system.

ii). User Interface –

- ✦ Designing the interface between end users and the computer system is a major consideration of a system analyst while designing the new system.
- ✦ Layout form & screen are used to construct the format and content of input/output media & methods. Dialogue flow diagrams analyze the flow of dialogue between computers & people.

iii). Data Attributes and Relationships –

- ✦ A Data Dictionary catalogs the description of the attributes of all data elements and their relationships to each other as well as to external systems.
- ✦ Entity-relationship diagrams are used to document the number & type of relationship among the entities. File layout forms document the type, size & names of data elements in a system.
- ✦ Grid charts help in identifying the use of each type of data element in input/output or storage media of a system.

iv). Detailed System Processes –

- ✦ These tools are used to help the programmer to develop detailed procedures and processes required in the design of a computer program.
- ✦ Structure charts document the purpose, structure and hierarchical relationships of the modules in a program.

- **A number of CASE tools are in use for typical representation, specifications and system modeling. A set of the prominent tools have been already mentioned and some are described as follows:**

a) Structured English –

- ✦ Structured English, also known as Program Design Language (PDL), 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. A better structured, universal and precise tool is referred as pseudo code.

b) Flowcharts –

- ✦ Flowcharting is a pictorial representation technique that can be used by analysts to represent the inputs, outputs and processes of a business process.
- ✦ 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.

c) Data Flow Diagrams –

- ✦ A Data Flow Diagram uses few simple symbols to illustrate the flow of data among external entities, processing activities and data storage elements.
- ✦ A DFD is composed of four basic elements: Data Sources and Destinations, Data Flows, Transformation processes and Data store and are combined to show how data are processed.

d) 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.
- ✦ Decision tree is 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.

e) Decision Table –

- ✦ A Decision Table is a table, which may accompany a flowchart defining possible contingencies that may be considered within program and proper course of action for each contingency.
- ✦ Decision tables are necessitated, since branches of the flowchart multiply at each diamond and may easily run into scores and even hundreds. If the programmer attempts to draw a flowchart directly, some of the branches may be missed.
- ✦ The four parts of the decision table are given as follows:

i). Condition Stub –

This comprehensively lists the comparisons or conditions;

ii). Action Stub –

This comprehensively lists actions to be taken along various program branches;

iii).Condition entries –

This list in its various columns the possible permutations of answer to the questions in the conditions stub; and

iv).Action entries –

This lists in its columns corresponding to the condition entries the actions contingent upon the set of answers to questions of that column.

f) CASE (Computer-Aided-Software Engineering) Tools –

- ✦ CASE refers to the automation of anything that humans do to develop systems and support virtually all phases of traditional system development process.
- ✦ An ideal CASE system would have an integrated set of tools and features to perform all aspects in the life cycle.
- ✦ Some of the features that various CASE products possess are – Repository / Data Dictionary; Computer aided Diagramming Tools; Word Processing; Screen and Report generator; Prototyping; Project Management; Code Generation and Reverse Engineering.

g) System Components Matrix –

- ✦ A System Component Matrix provides a matrix framework to document the resources used, the activities performed and the information produced by an information system.
- ✦ It can be used as an information system framework for both systems analysis and system design and views the information system as a matrix of components that highlights
 - how the basic activities of input, processing, output, storage and controls are accomplished in an information system, and
 - how the use of hardware, software and people resources can convert data resources into information products.

h) Data Dictionary –

- ✦ A data dictionary contains descriptive information about the data items in the files of a business information system. Thus, a data dictionary is a computer file about data.
- ✦ This information may include –
 - the identity of the source document(s) used to create the data item;
 - the names of the computer files that store the data item;
 - the names of the computer programs that modify the data item;
 - the identity of the computer programs or individuals permitted to access the data item for the purpose of file maintenance, upkeep, or inquiry; and
 - the identity of computer programs or individual not permitted to access the data item etc.
- ✦ When an accountant participates in the design of a new system, a data dictionary can also be used to plan the flow of transaction data through the system.

- ✦ As new data fields are added to the record structure of a business file, information about each new data item is used to create a new record in the data dictionary. Similarly, when data fields are deleted from the structure of file records, their corresponding records in the data dictionary are dropped.

i) User Interface Layout and Forms –

- ✦ Several type layout forms for both soft and hard copy are used to model input/output components of an automated information system.
- ✦ Some of the prominent and inevitable ones are described briefly as follows:

i). Layout form and Screen Generator –

- These are for printed report used to format or “paint” the desired layouts and contact without having to enter complex formatting information.

ii). Menu Generator –

- Menu generator outlines the functions, which the system is aimed to accomplish. Menu may be linked to other submenus that will enable the user to understand how the screens and sub-screens will be used for data entry or inquiry.

iii). Report Generator –

- Report generator has similar functions as found in screen generator. Also, it can indicate totals, paging, sequencing and control breaks in creating samples of the desired report.

iv). Code Generator –

- Code generator allows the analyst to generate modular units of source code from the high level specifications provided by the system analyst and play significant role in systems development process.

5) Systems Specification –

- At the end of the analysis phase, the systems analyst prepares a document called Systems Requirement Specifications (SRS).
- A well-documented SRS may normally contains the following sections:

a) Introduction –

- Goals, Objectives, software context, Scope and Environment of the computer-based system.

b) Information Description –

- Problem description; Information content, flow and structure; Hardware, software, human interfaces for external system elements and internal software functions.

c) Functional Description –

- Diagrammatic representation of functions; processing narrative for each function; Interplay among functions; Design constraints.

d) Behavioral Description –

- Response to external events and internal controls.

e) Validation Criteria –

- Classes of tests to be performed to validate functions, performance and constraints.

f) Appendices –

- Data flow/Object Diagrams; Tabular Data; Detailed description of algorithms charts, graphs and other such material.

g) SRS Review –

- The development team makes a presentation and then hands over the SRS document to be reviewed by the user or customer.
- Only, after ensuring that the document represents existing processes accurately, the user should sign the document. This is a technical requirement of the contract between users and development team/organization.

6) Roles Involved in SDLC –

- A variety of tasks during the SDLC are performed by special teams/ committees/ individuals based on requisite expertise as well as skills.
- Some of the generic roles are described as follows:

a) Steering Committee –

- ✦ It is special high power committee of experts to accord approvals and implementations.
- ✦ Some of the functions of Steering Committee are given as follows:
 - To provide overall directions and ensures appropriate representation of affected parties;
 - To be responsible for all cost and timetables;
 - To conduct a regular review of progress of the project in the meetings of steering committee, which may involve co-ordination and advisory functions; and
 - To undertake corrective actions like rescheduling, re-staffing, change in the project objectives and need for redesigning.

b) Project Manager –

- ✦ A project manager is responsible for delivery of the project deliverables within time/ budget and periodically reviews the progress of the project with the project leader and his/her team.

c) Project Leader –

- ✦ The project leader is dedicated to a project, who has to ensure its completion and fulfillment of objectives.

d) Systems Analyst / Business Analyst –

- ✦ The systems analysts' main responsibility is to conduct interviews with users and understand their requirements. S/he is a link between the users and the designers/programmers.

e) Module Leader/Team Leader –

- ✦ A project is divided into several manageable modules, and the development responsibility for each module is assigned to Module Leaders.
- ✦ Module leaders are responsible for the delivery of tested modules within the stipulated time and cost.

f) Programmer/Developers –

- ✦ Programmers is a mason of the software industry, who converts design into programs by coding using programming language. They also test the program for debugging activity to assure correctness and reliability

g) Database Administrator –

- ✦ The database administrator handles multiple projects; ensures the integrity and security of information stored in the database and also helps the application development team in database performance issues.

h) Quality Assurance –

- ✦ This team sets the standards for development, and checks compliance with these standards by project teams on a periodic basis.

i) Testers –

- ✦ Testers are a junior level quality assurance personnel attached to a project, who test programs and subprograms as per the plan given by the project leader & prepare test reports.

j) 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.
- ✦ A domain specialist need not have knowledge of software systems.

k) IS Auditor –

- ✦ As a member of the team, IS Auditor ensures that the application development also focuses on the control perspective.

7) Internal Controls –

- Requirements phase is the most important phases of SDLC. The issue of controls is very important.
- Some of the key control aspects at this stage may be taken care by the following queries:
 - ✦ Whether present system analysis has been properly done?
 - ✦ Whether appropriate domain, were expert was engaged?
 - ✦ Whether all user requirements of proposed system have been considered?
 - ✦ Whether SRS document has been made and vetted by Users, Domain Experts, System Analysts?

5.4.3 System Designing

- The objective of system designing is to design an Information System that best satisfies the users/managerial requirements. It describes the parts of the system and their interaction.
- The key and generic design phase activities include –
 - Describing inputs and outputs such as screen design and reports;
 - Determining the processing steps and computation rules for the new solution;
 - Determining data file or database system file design;
 - Preparing the program specifications for the various types of requirements; and
 - Internal/external controls.
- System design involves first logical design and then physical construction of a system.
 - The logical design of an information system is like an engineering blueprint; it shows major features of the system and how they are related to one another’.
 - Physical construction, the activity following logical design, produces program software, files and a working system.
- The design phase activities includes
 - 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/system software platform.

Which are described briefly as follows:

1) Architectural Design –

- ✦ Architectural design deals with the organization of applications in terms of hierarchy of modules and sub-modules.
- ✦ At this stage, we identify major modules; functions and scope of each module; interface features of each module; modules that each module can call directly or indirectly and Data received from/ sent to/ modified in other modules.
- ✦ The architectural design is made with the help of a tool called Functional Decomposition, which can be used to represent hierarchies. It has three elements – Module, Connection, and Couple.
- ✦ The module is represented by a box and connection between them by arrows. Couple is data element that moves from one module to another and is shown by an arrow with circular tail.

2) Design of Data/Information flow –

- ✦ In designing 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 new system.

3) Design of Database –

- ✦ Design of the database involves determining its scope ranging from local to global structure.
- ✦ The scope is decided on the basis of interdependence among organizational units.

4) User Interface Design –

- ✦ It involves determining the ways in which users will interact with a system.
- ✦ The points that need to be considered while designing the user interface are –
 - source documents to capture raw data,
 - hard-copy output reports,
 - screen layouts for dedicated source-document input,
 - inquiry screens for database interrogation,
 - graphic and color displays, and
 - requirements for special input/output device.

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. During physical design, the primary concern of the auditor is effectiveness and efficiency issues.
- ✦ Some of the generic design principles being applied to develop the design of typical information systems include the following:
 - There is a tendency to develop merely one design and consider it the final product. However, the recommended procedure is to 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 designed should be directly relevant to business activities.
 - The design should follow standards laid down. For instance, the user interface should have consistent color scheme, menu structure, location of error message and the like.
 - The design should be modular, with high cohesion and low coupling.

6) System's Operating Platform –

- ✦ In some cases, the new system requires an operating platform including hardware, network and system software not currently available in an organization.
- ✦ The new hardware/system software platform required to support the application system will then have to be designed for requisite provisions.

7) Internal Design Controls –

- ✦ The key control aspects at this stage include the following:
 - Whether management reports of stage I and stage II, were referred by System Designer?

- Whether all control aspects have been properly covered?
- Whether controls put in place in system, appear in the documentation done at this stage?
- Whether a separate review of design document has been done by internal auditor?

5.4.4 System Acquisition

- After a system is designed either partially or fully, the next phase of the systems development starts, which relates to the acquisition of operating infrastructure including hardware, software and service. Thereby, technical specifications, standards etc. come to rescue.

A. Acquisition Standards –

- ✦ Management should establish acquisition standards that address the security and reliability issues as per current state-of-the art development standards.
- ✦ Acquisition standards should focus on the following:
 - Ensuring security, reliability, and functionality already built into a product;
 - Ensuring managers complete appropriate vendor, contract, and licensing reviews and acquiring products compatible with existing systems;
 - Invitations-to-tender soliciting bids from vendors when acquiring hardware or integrated systems of hardware and software;
 - Request-for-proposals soliciting bids when acquiring off-the-shelf or third-party developed software; and
 - Establishing acquisition standards to ensure functional, security, and operational requirements to be accurately identified and clearly detailed in request-for-proposals.

B. Acquiring Systems Components from Vendors –

- ✦ Acquiring the appropriate hardware and software is critical for the success of the whole project. A sub-committee of experts under steering committee, referred as 'System Acquisition Committee' is constituted. The subcommittee is mandated to ensure timely & effective completion of this stage
- ✦ The next aspect is call for Request For Proposal (RFP) from vendors. RFP, means asking vendors to submit proposals for the requirements mentioned.
- ✦ **The following considerations are valid for both acquisition of hardware and software:**

i). Vendor Selection –

- It is necessary to remember that vendor selection is to be done prior to sending RFP. The result of this process is that 'RFP are sent only to selected vendors'.
- For vendor selection, following things are kept in mind including the background and location advantage of the vendor, the financial stability of vendor, the market feedback of vendor performance, in terms of price, services etc.

ii). Geographical Location of Vendor –

- The issue to look for whether the vendor has local support persons.

- Otherwise, the proposals submitted by vendor not as per RFP requirements need to be rejected, with no further discussion on such rejected proposals.
- This stage may be referred to as 'technical validation', that is to check the proposals submitted by vendors, are technically complying with RFP requirements.

iii). Presentation by Selected Vendors –

- All vendors, whose proposals are accepted after "technical validation", are allowed to make presentation to the System Acquisition Team for evaluation.

iv). Evaluation of Users Feedback –

- The best way to understand the vendor system is to analyze the feedback from present users.
- Present users can provide valuable feedback on system, operations, problems, vendor response to support calls.

✦ Besides these, some specific consideration for hardware and software acquisition are as follows:

- The benchmark tests to be done for proposed machine. For hardware's, there are specified benchmark test based on nature of hardware. These need to be applied to proposed equipment.
- Software considerations that can be current applications programs or new programs that have been designed to represent planned processing needs.
- The benchmarking problems are oriented towards testing whether a computer offered by the vendor meets the requirements of the job on hand of the buyer.
- The benchmarking problems would then comprise long jobs, short jobs, printing jobs, disk jobs, mathematical problems, input and output loads etc., in proportion typical of the job mix.
- If the job is truly represented by the selected benchmarking problems, then this approach can provide a realistic and tangible basis for comparing all vendors' proposals.
- Tests should enable buyer to effectively evaluate cross performance of various systems in terms of hardware performance, compiler language and operating system capabilities, diagnostic message, ability to deal with certain types of data structure & effectiveness of software utilities.
- Benchmarking problems, however, suffer from a couple of disadvantages. It takes considerable time and efforts to select problems representative of the job mix which itself must be precisely defined. It also requires the existence of operational hardware, software & services of systems.
- Nevertheless, this approach is very popular because it can test functioning of vendors' proposal.

C. Other Acquisition Aspects and Practices –

On addition to above, there are several other acquisition aspects and practices, which are as follows:

1) Hardware Acquisition –

- ✦ In case of procuring such machinery as machine tools, transportation equipment, air conditioning equipment, etc., the management can normally rely on the time tested selection techniques and the objective selection criteria can be delegated to the technical specialist.

- ✦ The management depends upon the vendor for support services, systems design, education and training etc., and expansion of computer installation for almost an indefinite period; therefore, this is not just buying the machine and paying the vendor for it but it amounts to an enduring alliance with the supplier.

2) Software Acquisition –

- ✦ This determination helps the systems development team to decide ‘what type of application software products is needed’ and thus, the degree of processing that the system need to handle.
- ✦ This helps the system developers in deciding about the nature of the systems software and computer hardware that will be most suitable for generating the desired outputs.
- ✦ At this stage, the system developers must determine whether the application software should be created in-house or acquired from a vendor.

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 to the contract.
- ✦ The contracts should be in writing with sufficient detail to provide assurances for performance, source code accessibility, software and data security, and other important issues.
- ✦ Software license is a license that grants usage permission to do things with computer software.
- ✦ Copyright laws protect proprietary as well as open-source software. The use of unlicensed software or violations of a licensing agreement expose organizations to possible litigation.

4) Validation of Vendors’ proposals –

- ✦ The contracts and software licensing process consists of evaluating and ranking the proposals submitted by vendors and is quite difficult, expensive and time consuming. This problem is made difficult by the fact that vendors would be offering a variety of configurations.
- ✦ The following factors have to be considered towards rigorous evaluation.
 - The Performance capability of each proposed System in Relation to its Costs;
 - The Costs and Benefits of each proposed system;
 - The Maintainability of each proposed system;
 - The Compatibility of each proposed system with Existing Systems; and
 - Vendor Support.

5) Methods of validating the proposal –

- ✦ Large organizations would naturally tend to adopt a sophisticated and objective approach to validate the vendor’s proposal.
- ✦ Some of the validation methods are given as follows:

i). Checklists –

- It is the most simple and a subjective method for validation and evaluation.

- The various criteria are put in check list in the form of suitable questions against which the responses of the various vendors are validated.

ii). Point-Scoring Analysis –

- Point-scoring analysis provides an objective means of selecting a final system. There are no absolute rules in the selection process, only guidelines for matching user needs with software capabilities.
- Thus, even for a small business, the evaluators must consider issues as the company's data processing need, its in-house computer skill, vendor reputation, software cost and so forth

iii). Public Evaluation Reports –

- Several consultancy as well as independent agencies compare and contrast the hardware and software performance for various manufacturers and publish their reports.
- This method is particularly useful where buying staff has inadequate knowledge of facts.

iv). Benchmarking Problems related Vendor's Solutions –

- Benchmarking problems related to vendors' proposals are accomplished by sample programs that represent at least a part of the buyer's primary work load and include considerations and can be current applications.

v). Testing Problems –

- Test problems disregard the actual job mix and are devised to test the true capabilities of the hardware, software or system.
- The results, achieved by the machine can be compared and price performance judgment can be made. It must be borne in mind, however that various capabilities to be tested would have to be assigned relative weightage.

5.4.5 System Development: Programming Techniques and Languages

- Application programs are written, tested and documented, conduct system testing. Finally it results into a fully functional and documented system.
- **A good coded application and programs should have the following characteristics:**

i). Reliability –

- ✦ It refers to the consistency with which a program operates over a period of time. However, poor setting of parameters and hard coding, then could result in failure of a program after some time.

ii). Robustness –

- ✦ It refers to the applications' strength to uphold its operations in adverse situations by taking into account all possible inputs and outputs of a program in case of least likely situations.

iii). Accuracy –

- ✦ It refers not only to 'what program is supposed to do', but should also take care of 'what it should not do'.

iv). Efficiency –

- ✦ It refers to the performance per unit cost with respect to relevant parameters and it should not be unduly affected with the increase in input values.

v). Usability –

- ✦ It refers to a user-friendly interface and easy-to-understand internal/external documentation.

vi). Readability –

- ✦ It refers to the ease of maintenance of program even in the absence of the program developer.

- Other related aspects of this phase are given as follows:

a) Program Coding Standards –

- ✦ The logic of the program outlined in the flowcharts is converted into program statements or instructions at this stage. For each language, there are specific rules concerning format and syntax. Syntax means vocabulary, punctuation and grammatical rules available in the language manuals that the programmer has to follow strictly and pedantically.
- ✦ Therefore, the coding standards are defined, which 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 provide simplicity, interoperability, compatibility, efficient utilization of resources and least processing time.

b) Programming Language –

- ✦ Application programs are coded in the form of statements or instructions and the same is converted by the compiler to object code for the computer to understand and execute.
- ✦ Language selection is based on of application area; algorithmic complexity; environment in which software has to be executed; performance consideration; data structure complexity; knowledge of software development staff & capability of in-house staff for maintenance.
- ✦ The programming languages commonly used are given as follows:
 - High level general purpose programming languages such as COBOL and C;
 - Object oriented languages such as C++, JAVA etc.;
 - Scripting language such as JavaScript, VBScript; and
 - Decision Support or Logic Programming languages such as LISP and PROLOG.

c) Program Debugging –

- ✦ Debugging is the most primitive form of testing activity, which refers to correcting programming language syntax and diagnostic errors so that the program compiles cleanly.
- ✦ Debugging can be a tedious task consisting of following four steps:
 - Giving input the source program to the compiler,
 - Letting the compiler to find errors in the program,

- Correcting lines of code that are erroneous, and
- Resubmitting the corrected source program as input to the compiler.

d) Testing the Programs –

- ✦ A careful and thorough testing of each program is imperative to the successful installation of any system. The programmer should discuss with the project manager and plan the testing to be performed, including testing of all the possible exceptions. The test plan should require the execution of all standard processing logic based on chosen testing strategy/techniques.
- ✦ A log of test results and all conditions successfully tested should be kept. The log will prove invaluable in finding the faults and debugging.

e) Program Documentation –

- ✦ Managers and users should carefully review both internal and external documentation in order to ensure that the software and system behave as the documentation indicates.
- ✦ If they do not, documentation should be revised. User documentation should also be reviewed for understandability.

f) Program Maintenance –

- ✦ The requirements of business data processing applications are subject to periodic change. There are programmers called maintenance programmers, who are entrusted with this task.

5.4.6 System Testing

- Testing is a process used to identify the correctness, completeness and quality of developed computer software. Testing should systematically uncover different classes of errors in a minimum amount of time with a minimum amount of efforts.
- Different levels/facets of Testing are as follows.
 - i). Unit Testing
 - ii). Integration Testing
 - iii). System Testing
 - iv). Final Acceptance Testing
 - v). Internal Testing Controls

These are described as follows –

1) Unit Testing –

- ✦ In computer programming, unit testing is a software verification and validation method in which a programmer tests if individual units of source code are fit for use.
- ✦ Unit tests are typically written and run by software developers to ensure that code meets its design and behaves as intended.
- ✦ The goal of unit testing is to isolate each component of the program and show that they are correct. A unit test provides a strict, written contract that the piece of code must satisfy.

✦ The five categories of tests that a programmer performs on a program unit, which are as follows:

i). Functional Tests –

- Functional Tests check ‘whether programs do, what they are supposed to do or not’.
- As per this plan, programmer checks by inputting the values to see whether the actual result and expected result match.

ii). Performance Tests –

- Performance Tests should be designed to verify the response time, the execution time, the throughput, primary and secondary memory utilization and the traffic rates on data channels and communication links.

iii). Stress Tests –

- Stress testing is a form of testing that is used to determine the stability of a given system. It involves testing beyond normal operational capacity, often to a breaking point, in order to observe the results. The purpose of a stress test is to determine the limitation of the program.

iv). Structural Tests –

- Structural Tests are concerned with examining internal processing logic of a software system.

v). Parallel Tests –

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

✦ **Unit testing is classified** as Static Analysis Testing and Dynamic Testing. Such typical testing techniques are elaborated as follows:

a) Static Testing –

Static Analysis Tests are conducted on source programs and do not normally require executions in operating conditions. Typical static analysis techniques include the following:

i). Desk Check –

- This is done by the programmer him/herself. S/he checks for logical syntax errors, and deviation from coding standards.

ii). Structured Walk Through –

- The application developer leads other programmers to scan through the text of the program and explanation to uncover errors.

iii). Code Inspection –

- The program is reviewed by a formal committee. Review is done with formal checklists.

b) Dynamic Analysis Testing –

Such testing is normally conducted through execution of programs in operating conditions. Typical techniques for dynamic testing and analysis include the following:

i). Black Box Testing –

- Black Box Testing takes an external perspective of the test object, to derive test cases. These tests can be functional or non-functional, though usually functional.
- The test designer selects inputs including simple, extreme, valid & invalid input-cases and executes to uncover errors. There is no knowledge of the test object's internal structure.
- This method of test design is applicable to all level of software testing i.e. unit, integration, functional testing, system and acceptance. The higher the level, hence the bigger and more complex the box, the more one is forced to use black box testing to simplify.
- While this method can uncover unimplemented parts of the specification, one cannot be sure that all existent paths are tested. If a module performs a function, which is not supposed to, the black box test does not identify it.

ii). White Box Testing –

- It uses internal perspective of the system to design test cases based on internal structure.
- It requires programming skill to identify all paths through the software. The tester chooses test case inputs to exercise paths through the code and determines appropriate outputs.
- Since the tests are based on the actual implementation, if the implementation changes, the tests probably will need to change, too. It is applicable at the unit, integration and system levels of the testing process, it is typically applied to the unit.
- After obtaining a clear picture of the internal workings of a product, tests can be conducted to ensure that the internal operation of the product conforms to specifications and all the internal components are adequately exercised.

iii). Gray Box Testing –

- It is a software testing technique that uses a combination of black box testing and white box testing.
- In gray box testing, the tester applies a limited number of test cases to the internal workings of the software under test.
- In the remaining part of the gray box testing, one takes a black box approach in applying inputs to the software under test and observing the outputs.

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 validity of connection of two or more components that pass information from one area to another.
- ✦ This is carried out in the following manners:

i). Bottom-up Integration –

- It is the traditional strategy used to integrate the components of a software system into a functioning whole. It consists of unit testing, followed by subsystem testing, and then testing of the entire system.
- Bottom-up testing is easy to implement as at the time of module testing, tested subordinate modules are available.
- The disadvantage is that testing of major decision/ control points is deferred to a later period.

ii). Top-down Integration –

- It starts with the main routine, and stubs are substituted, for the modules directly subordinate to the main module.
- An incomplete portion of a program code that is put under a function in order to allow the function and the program to be compiled and tested, is referred to as a stub.
- Once the main module testing is complete, stubs are substituted with real modules and these modules are tested with stubs. This process continues till the atomic modules are reached.
- The top-down strategy emphasizes on major control decision points encountered in the earlier stages of a process and detects any error in these processes.
- 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.

iii). Regression Testing –

- Each time a new module is added or any modification made in the software, it changes.
- New data flow paths are established, new I/O may occur and new control logic is invoked. These changes may cause problems with functions that previously worked flawlessly.
- The regression tests ensure that changes or corrections have not introduced new faults. The data used for the regression tests should be the same as the data used in the original test.

3) System Testing –

- ✦ It is a process in which software and other system elements are tested as a whole.
- ✦ System testing begins either when the software as a whole is operational or when the well-defined subsets of the software's functionality have been implemented.
- ✦ The purpose of system testing is to ensure that the new or modified system functions properly.
- ✦ The types of testing that might be carried out are as follows:

i). Recovery Testing –

- This is the activity of testing '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 to verify that recovery is liable to be properly performed, in actual failures.

ii). Security Testing –

- This is the process to determine that an Information System protects data and maintains functionality as intended or not.
- The six basic security concepts that need to be covered by security testing are – confidentiality, integrity, availability authentication, authorization, and non-repudiation.

iii). Stress or Volume Testing –

- Stress testing is a form of testing that is used to determine the stability of a given system.
- It involves testing beyond normal operational capacity, often to a breaking point, in order to observe the results.

iv). Performance Testing –

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

4) Final Acceptance Testing –

- ✦ It is conducted when the system is just ready for implementation. During this testing, it is ensured that new system satisfies the quality standards adopted by the business and system satisfies users.
- ✦ Thus, the final acceptance testing has two major parts:

i). Quality Assurance Testing –

- It ensures that the new system satisfies the prescribed quality standard and the development process is as per the organization's quality assurance policy, methodology and prescriptions.

ii). User Acceptance Testing –

- It ensures that functional aspects expected by users have been well addressed in new system.
There are two types of the user acceptance testing described as follows:

a) Alpha Testing –

- This is the first stage, often performed by the users within the organization by the developers, to improve and ensure the quality/functionalities as per users satisfaction.

b) Beta Testing –

- This is the second stage, generally performed after the deployment of the system. It is performed by the external users, during the real life execution of the project.
- It normally involves sending the product outside the development environment for real world exposure and receives feedback for analysis and modifications, if any.

5) Internal Testing Controls –

- ✦ Though it varies from one organization to another, some of the generic key control aspects appear to be addressed by the responses to following queries:

- Whether the test-suite prepared by the testers includes the actual business scenarios?
- Whether test data used covers all possible aspects of system?
- Whether CASE tools like 'Test Data Generators' have been used?
- Whether test results have been documented?
- Whether test have been performed in their correct order?
- Whether modifications needed based on test results have been done?
- Whether modifications made have been properly authorized and documented?

5.4.7 System Implementation

- The process of ensuring that the information system is operational and then allowing users to take over its operation for use and evaluation is called Systems Implementation.
- Implementation includes all those activities that take place to convert from the old system to the new. The new system may be totally new, replacing an existing manual or automatic system or it may be a major modification in an existing system.
- Some of the generic activities involved in system implementation stage are as follows:
 - i). Equipment Installation,
 - ii). Training Personnel,
 - iii). System Change – over Strategies
 - iv). Requisite Changeover or Conversion

The generic activities involved in system implementation stage are described as follows –

1) Equipment Installation –

- ✦ The hardware required to support the new system is selected prior to the implementation phase.
- ✦ The necessary hardware should be ordered in time to allow for installation and testing of equipment during the implementation phase.
- ✦ In installations, adequate time should be scheduled to allow completion of the following activities:
 - i). Site Preparation –**
 - An appropriate location must be availed to provide an operating environment for equipment that will meet the vendor's temperature, humidity and dust control specifications etc.
 - ii). Installation of New Hardware / Software –**
 - The equipment must be physically installed by the manufacturer, connected-to the power source and wired to communication lines, if required.
 - iii). Equipment Checkout –**
 - The equipment must be turned on for testing under normal operating conditions. The implementation in-house team should devise and run extensive tests of its own to ensure that equipment functionalities in actual working conditions.

2) Training Personnel –

- A system can succeed or fail depending on the way it is operated and used.
- Therefore, the quality of training received by the personnel involved with the system in various capacities helps or hinders the successful implementation of information system.
- Thus, training is a major component of systems implementation. Such training structure should be highly formalized and be based business process executions with actual data.

3) System Change-Over Strategies –

- Conversion or changeover is the process of changing over or shifting over from the old system to new system. It require careful planning to establish basic approach to be used in actual changeover
- The Four types of popular implementation strategies are described as follows:

i). Direct Implementation / Abrupt Change-Over –

- This is achieved through an abrupt takeover – an all or no approach. With this strategy, the changeover is done in one operation, completely replacing the old system in one go.
- Direct Implementation, which usually takes place on a set date, often after a break in production or a holiday period so that time can be used to get the hardware and software for the new system installed without causing too much disruption.

ii). Phased Changeover –

- With this strategy, implementation can be staged with conversion to the new system taking place gradually.
- If a phase is successful then the next phase is started, eventually leading to the final phase when the new system fully replaces the old one.

iii). Pilot Changeover –

- With this strategy, the new system replaces old one in one operation but only on small scale.
- Any errors can be rectified or further beneficial changes can be introduced and replicated throughout the whole system in good time with the least disruption. If successful then the pilot is extended until it eventually replaces the old system completely.

iv). Parallel Changeover –

- This is considered the most secure method with both systems running in parallel over an introductory period. The old system remains fully operational while new system come online.
- With this strategy, the old and new system are both being able to operate independently. If all goes well, the old system is stopped and new system carries on as the only system.

4) Requisite Changeover or Conversion –

- Such requisite changeover or Conversion includes all those activities, which must be completed to successfully convert from the previous system to the new information system.

- Fundamentally these technical activities can be classified as follows:

i). Procedure Conversion –

- Operating procedures should be carefully completed with sufficient-enough documentation for the new system. It applies to both computer operations and functional area operations.
- Before any parallel or conversion activities can start, operating procedures must be clearly spelled out for personnel in the functional areas undergoing changes.
- Information on input, data files, methods, procedures, output, and internal control must be presented in clear, concise and understandable terms for the average reader.
- Written operating procedures must be supplemented by oral communication during the training sessions on the system change.

ii). File Conversion –

- Because large files of information must be converted from one medium to another, this phase should be started long before programming and testing are completed.
- In order for the conversion to be as accurate as possible, file conversion programs must be thoroughly tested. Adequate control should be required output of the conversion program.
- The existing computer files should be kept for a period of time. This is necessary, when files must be reconstructed from scratch after a "bug" is discovered later in conversion routine.

iii).System conversion –

- After on-line and off-line files have been converted and the reliability of new system has been confirmed, daily processing can be shifted from the existing information system to new one.
- System development team members should be present to assist and to answer any question.
- Consideration should be given to operating the old system for some more time to permit checking, matching and balancing the total results of both systems.

iv).Scheduling Personnel and Equipment –

- Scheduling data processing operations of a new information system for the first time is a difficult task for the system manager. As users become more familiar with the new system, the job becomes more routine.
- Schedules should be set up by the system manager in conjunction with departmental managers of operational units serviced by the equipment.

5.4.8 Post Implementation Review and Systems Maintenance

1) Post Implementation Review –

- A Post Implementation Review answers the question “Did we achieve what we set out to do in business terms?”
- It examines the efficacy of all elements of the working business solution to see if further improvements can be made to optimize the benefit delivered.

- A Post-Implementation Review should be scheduled some time after solution has been deployed. Typical periods range from 6 weeks to 6 months, depending on type of solution & its environment.
- There are two basic dimensions of Information system that should be evaluated.
 - The first dimension is concerned with whether newly developed system is operating properly.
 - The other dimension is concerned with whether the user is satisfied with the information system with regard to the reports supplied by it.

- Typical evaluations include the following:

i). Development Evaluation –

- Evaluation of the development process is primarily concerned with whether the system was developed on schedule and within budget. It requires schedules and budgets to be established in advance and that record of actual performance and cost be maintained.
- Due to the uncertainty and mystique associated with system development, they are not subjected to traditional management control procedures.

ii). Operational Evaluation –

- The evaluation of the information system's operation pertains to whether the hardware, software and personnel are capable to perform their duties.
- It tries to answer the questions related to functional aspects of the system. Such an evaluation is relatively straightforward if evaluation criteria are established in advance.

iii). Information Evaluation –

- An information system should be evaluated in terms of information it provides or generates. This aspect of system evaluation is difficult and cannot be conducted in quantitative manner.
- Therefore, the extent to which information provided by the system is supportive to decision making is the area of concern in evaluating the system.

2) System Maintenance –

- Maintaining the system is an important aspect of SDLC. As key personnel change positions in the organization, new changes will be applied, which will require system updates at regular intervals.
- The need for modification arises from a failure to anticipate/capture all the requirements during system analysis/design and/or from changing organizational requirements.
- Maintenance can be categorized in the following ways:

i). Scheduled Maintenance –

- Scheduled maintenance is anticipated and can be planned for operational continuity and avoidance of anticipated risks.

ii). Rescue Maintenance –

- Rescue maintenance refers to previously undetected malfunctions that were not anticipated but require immediate troubleshooting solution.

iii).Corrective Maintenance –

- Corrective maintenance deals with fixing bugs in the code or defects found during executions.
- A defect can result from design errors, logic errors coding errors, data processing and system performance errors.
- The need for corrective maintenance is initiated by bug reports drawn up by the end users.

iv).Adaptive Maintenance –

- Adaptive maintenance consists of adapting software to changes in the environment, such as the hardware or the operating system.
- The need for adaptive maintenance can only be recognized by monitoring the environment.

v). Perfective Maintenance –

- Perfective maintenance mainly deals with accommodating to the new or changed user requirements and concerns functional enhancements to the system and activities to increase the system's performance or to enhance its user interface.

vi).Preventive Maintenance –

- Preventive maintenance concerns with the activities aimed at increasing the system's maintainability such as updating documentation, adding comments and improving the modular structure of the system.

5.5 Operation Manuals

- It is typical user's guide, also commonly known as Operations Manual.
- It may be a technical communication document intended to give assistance to people using a particular system.
- It is usually written by technical writers, although user guides are written by programmers, product or project managers, or other technical staff, particularly in smaller companies. These are most commonly associated with electronic goods, computer hardware and software.
- The section of an operation manual will 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 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.

5.6 Auditors' Role in SDLC

- The audit of systems under development can have three main objectives, given as follows –
 - i). To provide an opinion on the efficiency, effectiveness, and economy of project management.
 - ii). To assess the extent to which the system being developed provides for adequate audit trails and controls to ensure the integrity of data processed and stored; and
 - iii). The effectiveness of controls being enacted for the management of the system's operation.
- In order to achieve these goals, an auditor has to attend project and steering committee meetings and examine project control documentation and conducting interviews.
- An Auditor may adapt a rating system such as on a scale of 1 to 10 in order to give rating to the various phases of SDLC.
- Moreover, in order to audit technical work products, auditor may opt to include a technical experts to seek his/her opinion on the technical aspects of SDLC.
- Some of the control considerations for an auditor include the following:
 - ✦ Documented policy and procedures;
 - ✦ Established Project team with all infrastructure and facilities;
 - ✦ Developers/ IT managers are trained on the procedures;
 - ✦ Appropriate approvals are being taken at identified mile-stones;
 - ✦ Development is carried over as per standards, functional specifications;
 - ✦ Separate test environment for development/ test/ production / test plans;
 - ✦ Design norms and naming conventions are as per standards and are adhered to;
 - ✦ Business owners testing and approval before system going live;
 - ✦ Version control on programs;
 - ✦ Source Code is properly secured;
 - ✦ Adequate audit trails are provided in system; and
 - ✦ Appropriateness of methodologies selected.
- Further, in post implementation review, auditors need to review which of the SDLC phases have not met desired objectives and whether any corrective actions were taken. If there are differences between expectations and actual results, auditors need to determine the reasons for the same. Such reasons can help auditors to evaluate the current situation and offer guidelines for future projects.