

This is copyrighted material. No part of this book may be reproduced, stored in a retrieval system, or transmitted, in any form, or by any means, electronics, mechanical, photocopying, recording or otherwise, without prior permission, in writing, from the authors.

Book is quite student friendly, which makes points easy to remember and recollect with easy memory words.

This is a sample chapter which has been uploaded and all other chapters are similarly written.

Please do give your valuable comments and if you want to purchase the book then give your call to nearest retailer or give a call to us.

Authors:

Jignesh Chheda – 09819068991

Harshal Vora – 09773504845

CHAPTER - 2

SYSTEM DEVELOPMENT LIFE CYCLE METHODOLOGY

Particulars	Page No
2.1 Systems Development Process	
2.1.1 Achieving System Development Objectives	
2.1.2 System Development Team	
2.2 Systems Development Methodology	
2.2.1 The Traditional / Waterfall Approach / Sequential Approach	
2.2.2 The Prototyping Model	
2.2.3 The Incremental Model	
2.2.4 Spiral Model	
2.2.5 Rapid Application Development (RAD)	
2.2.6 Agile Methodologies	
2.3 System Development Life Cycle (SDLC)	
2.4 The Preliminary Investigation	
2.4.1 Identification of Problem	
2.4.2 Identification of Objective	
2.4.3 Delineation of Scope	
2.4.4 Feasibility Study	
2.4.5 Reporting Results to Management	
2.5 System Requirement Analysis	
2.5.1 Facts Finding Techniques	
2.5.2 Analysis of the Present System	
2.5.3 Systems Analysis of Proposed Systems	
2.5.4 System Development Tools	
2.5.5 System Specification	
2.5.6 Roles Involved in SDLC	
2.6 System Design	
2.6.1 Architectural Design	
2.6.2 Design of Data / Information flow	
2.6.3 Design of Database	
2.6.4 Design of User-Interface	
2.6.5 Physical Design	
2.6.6 Design of the Hardware / System Software Platform	
2.7. System Acquisition	
2.7.1 Acquisition Standards	
2.7.2 Acquiring Systems Components from Vendors	
2.8. Development: Programming Techniques And Languages	
2.8.1 Program Coding Standards	
2.8.2 Programming Language	
2.8.3 Program Debugging	
2.8.4 Test the program	
2.8.5 Program Documentation	
2.8.6 Program Maintenance	
2.9. System Testing	
2.9.1 Unit Testing	
2.9.2 Integration Testing	
2.9.3 System Testing	
2.9.4 Final Acceptance Testing	
2.10. Systems Implementations	
2.10.1 Activities during Implementation Stage	

2.10.2 Activities involved in conversion	
2.11. POST Implementation Review And Systems Maintenance	
2.11.1 Post Implementation Review	
2.11.2 System Maintenance	
2.12. Operation Manuals	
2.13. Auditor's Role In SDLC [As Per New Module]	

2.1 SYSTEMS DEVELOPMENT PROCESS: System development is a process of examining a business situation with the intent of improving it through better procedures and methods. Two major components of system development are:

- **System Analysis:** It is the process of gathering facts, diagnosing problems, and recommending improvements to the system.
- **System Design:** It plans a new business system or to replace an existing system.

2.1.1 Achieving System Development Objectives: There are many reasons why organization fails to achieve system development objectives and some of them are as follows:

[NRI – LS i.e., NRI Lacks Style]

- **New technologies:** With advanced technology achieving system development objectives is more difficult because users would not be familiar with that technology.
- **Resistance to change:** People have a natural tendency to resist change. When employee perceives that the project will result in personnel cutbacks, then the development project is doomed to be a failure.
- **Inadequate testing and user training:** New systems must be tested before installation for its effective operation. Users must also be trained effectively for the new system.
- **Lack of senior management support and involvement:** Employees and Developers follow those project where senior management are involved because they consider those projects as important and act accordingly by shifting their efforts to those project and therefore other project receive less attention.
- **Lack of standard project management and systems development methodologies:** Some organizations do not formalize their project or system development methodologies thereby making it very difficult to complete projects on time or within budget.
- **Lack of user participation** Users must participate in the development effort and must feel free to take ownership for project's success, etc. This will reduce resistance to change.
- **Shifting user needs.** User requirements are many and keep on changing. So it becomes difficult for developers to develop system as originally stated.

2.1.2 System Development Team: Several people in the organization are responsible for systems development. In large systems, the worth of a project is typically decided by a top management level steering committee. The steering committee ensures that systems development activities are consistently aimed at satisfying the requirements of managers and users within the organization. System analysts are subsequently assigned to determine user requirements, design the system and assist in development and implementation activities. 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.

2.2 SYSTEMS DEVELOPMENT METHODOLOGY: A system development methodology is a formalized, standardized, documented set of activities used for system development project. Each methodology is best suited for specific kinds of projects. The methodology is characterized by the following: **[TDTDP - Total Dil To Dil Policy & Procedure available]**

- The system must be **tested** thoroughly prior to implementation to ensure that it meets users' needs.
- Each project should be **divided** into a number of identifiable processes and each process has a starting and an ending point. The division of the project facilitates both project

planning and project control.

- User must be provided with adequate **training** on the new system.
- Specific reports and other **documentations** called Deliverables are generated periodically during system development so that execution of system development is proper.
- Users, managers must **participate** in the project to provide approvals for system developed which are often called signoffs.
- A **post-implementation** review must be performed to assess the effectiveness and efficiency of the new system.

Approaches to System Development: Since organizations vary significantly in the way they automate their business procedures and since each new type of system differs from the other, several different system development approaches are used within an organization. All these approaches are not mutually exclusive, which means that it is possible to perform some prototyping while applying the traditional approach. These approaches are as follows:

1. **Waterfall:** Linear framework type
2. **Prototyping:** Iterative framework type
3. **Incremental:** Combination of linear and iterative framework type
4. **Spiral:** Combination linear and iterative framework type
5. **Rapid Application Development (RAD):** Iterative Framework Type
6. **Agile Methodologies**

2.2.1 The Traditional / Waterfall Approach / Sequential Approach: The waterfall approach is a traditional development approach in which each developer in a development team works in different phases. It is used on small projects because it eliminates testing to identify problems early in the process. In the traditional approach of system development, activities are performed in sequences (i.e., step by step). Unless earlier step is completed next step is not undertaken under traditional approach.

Framework type: Linear

Basic Principles: [DEC – deck]

- Project is **divided** into sequential phases, with some overlap and splash back acceptable between the phases.
- **Emphasis** is on planning, time scheduling, target dates, budgets and implementation of an entire system at one time.
- Tight **control** is maintained over the life of the project through the extensive use of written documentation.

Strengths: [COMIc]

- **Conserves** resources.
- An **orderly sequence** of development steps and design ensures the quality, reliability, etc of the developed software.
- Progress of system development is **measurable**.
- **Ideal** for less experienced project teams or where a composition team fluctuates.

Weaknesses:[SFLPCE – Slow Forwarding of Little Problem makes Excessive Changes]

- It is **slow**, costly, and cumbersome due to tight controls maintained over the project.
- Project progresses **forward** with little scope to go back.
- **Little** room for use of iteration, which used will reduce manageability.
- **Problems** cannot be discovered due to lack of system testing.
- Difficult to respond to changes. **Changes** occurring later in the system are more costly and hence discouraged.
- **Excessive** documentation at each stage is time-consuming and also difficult for the users to read it regularly.

2.2.2 The Prototyping Model: The goal of prototyping approach is to develop a small or pilot version of a system called prototype. A prototype is built quickly at a lesser cost and with the intention of being modifying or replacing it. As users work with the prototype, they make suggestions for improvements which are then incorporated and finally a prototype satisfying all the users requirements is developed which is turned into the final system or it is scrapped.

Framework type: Iterative.

Basic Principles: Prototyping can be viewed as a series of four steps:

Step 1 - Identify Information System Requirements: Under prototype approach, the design team must identify only fundamental system requirements to build the initial prototype i.e., initial requirement of users.

Step 2 - Develop the Initial Prototype: In this step, the designers create an initial base model and emphasizes on system characteristics such as simplicity, flexibility, and ease of use which enables users with data entry display screens, menus, prompts inputs, and source documents.

Step 3 - Test and Revise: After finishing the initial prototype, the designers demonstrate the model and then give it to users and ask them to record their likes and dislikes. Using their feedbacks, the design team modifies the prototype and then resubmits the revised model. Thus the process continues until the users are satisfied.

Step 4 - Obtain User Signoff of the Approved Prototype: At the end of Step 3, users formally approve the final version of the prototype, which describes about what the system will, and will not, do or provide.

Strengths: [IQ DEKhI]

- Easily **identify** confusing or difficult functions and missing functionality.
- Provides **quick implementation** of incomplete, but functional application (i.e., system is not complete but incomplete system can still function properly).
- Users' needs and requirements are **defined** properly because it requires extensive user involvement in system development.
- Since users of the system experiment with each version of the prototype, **errors** are detected and eliminated early in the developmental process. As a result, the system should be more reliable and less costly to develop.
- **Knowledge** gained in an earlier iteration is used in development of later iterations.
- Encourages **innovation** and flexible designs.

Weaknesses: [BAD-QR i.e., BAD ?R i.e. bad questionnaire]

- Prototyping may cause **behavioral problems** with the system users if system developers are unable to meet all the user requirements as well as users will grow impatient when they have to go through too many interactions with the prototype.
- **Approval** process and control are not strict.
- To make prototype successful users should be willing to **devote** their significant time for experimenting with the prototype and provide suggestions for improvements.
- Designers design prototype **quickly** without analyzing users' requirements sufficiently resulting in an inflexible design further limiting future system potential.
- **Requirements** of users changes frequently.

2.2.3 The Incremental Model

Framework Type: It is a combination of Linear and Iterative.

Basic Principles: In Incremental model, system model is designed, implemented and tested incrementally (a little more is added each time) until the product is finished. Here project is divided into parts and each part is built separately. Each part is delivered to the client for review when it is complete. Project is defined as finished when it satisfies all the users' requirements.

Strengths: [KFMCG – Knowledge of FMCG sector is must]

- **Knowledge** gained in an earlier iteration is used in development of later iterations.
- It is more **flexible** and less costly to change the scope and requirements.
- Helps to **mitigate** integration and architectural risks early in the project.
- **Concrete** evidence of project status can be given throughout the life cycle.
- **Gradual implementation** helps to monitor the effect of incremental changes and make adjustments before the organization is negatively impacted.

Weaknesses: [MP3]

- Some **modules** may be completed earlier than the other modules and so well-defined interfaces are required.
- Each **phase** of an iteration is rigid and do not overlap each other.
- **Problems** relating to system architecture design may arise because all users' requirements for entire system are not obtained all together.
- System designers may have tendency to **push** difficult problems for future for demonstrating early success to management.

2.2.4 Spiral Model

Framework Type: It is a combination of Linear and Iterative.

Basic Principles: Also known as the Spiral Lifecycle, it is a Systems Development Method (SDM) which combines the features of the prototyping model and the waterfall model. The spiral model is intended for large, expensive and complicated projects. It combines advantages of both top down and bottom up concepts.

- The new system requirements are defined in detail which involves interviewing a number of external or internal users.
- A preliminary design is created for the new system where all possible alternatives that can help in developing a cost effective project are analyzed and decided. This phase identify and resolve all the possible risks. If risks indicate any kind of uncertainty, prototyping may be used to find out possible solution.
- A first prototype of the new system is constructed and tested by designer himself and knows how new prototype is prepared with additional functions based on knowledge obtained from earlier prototype and all these steps are repeated again and again until final product is ready.
- A second prototype is evolved by a fourfold procedure :
 - ✓ evaluating the first prototype in terms of its strengths, weaknesses, and risks;
 - ✓ defining the requirements of the second prototype;
 - ✓ planning and designing the second prototype;
 - ✓ developing and testing the second prototype.

Strengths: [EIS]

- **Enhance** risk avoidance.
- Can **incorporate** Waterfall, Prototyping, and Incremental methodologies in the framework as special case and provides guidance as to which combination will give best software iteration based on project risk.
- Useful in **selecting** best methodology for development of software based on project risk.

Weaknesses: [EICC]

- Requires **experienced** project manager who can determine which methodology to apply in given project.
- No firm deadlines, so there is an **inherent risk** of project not meeting budget or schedule.
- To determine exact **composition** for development of each iteration is a challenge.
- Highly **complex** for each project and therefore has limiting reusability.

2.2.5 Rapid Application Development (RAD)

Framework Type: Iterative.

Basic Principles: It is a type of software development which uses minimum planning for developing a prototype which results in development of software much faster and makes it easier to change requirements.

- Key objective is to develop software fast with high quality at relatively low cost,
- Attempts to reduce inherent project risk by dividing a project into smaller segments.
- Aims to produce high quality systems quickly, primarily through the use of iterative Prototyping, along with active user involvement, etc.
- Key emphasis is on fulfilling the business need.
- Project control involves prioritizing development and defining delivery deadlines or “timeboxes.” If the project starts to slip, emphasis is on reducing requirements to fit the timebox, not in increasing the deadline.
- Active user involvement is compulsory i.e., it includes Joint Application Development (JAD).
- Produces production software as opposed to a throwaway prototype.
- Produces documentation for future development and maintenance.

Strengths: [SPCI-RICO i.e., special reco A/c]

- **Saves** time, money and human efforts.
- System is **produced** more quickly with lesser cost.
- Rapidly **change** system design as per requirement of users.
- Quick initial **reviews** is possible.
- Constant **integration** isolates problems and encourages customer feedback.
- **Concentrates** on development of system elements from user point of view.
- The **operational version** is available much easier as compared to Waterfall, Incremental, or Spiral frameworks.

Weaknesses: [QEMIPD - ? all EMI's are Paid]

- More speed and low cost leads to a lower system **quality**.
- Project will **end up** with more requirements than needed.
- Some **modules** may be completed earlier than others and so well-defined interfaces are required.
- **Inconsistent** system is developed because system is produced quickly.
- System designers may have tendency to **push** difficult problems for future for demonstrating early success to management.
- **Difficult** to reuse module in future systems.

2.2.6 Agile Methodologies: It is based on iterative and incremental approach where requirements and solutions evolve through collaboration between self organizing and cross functional teams. It promotes adaptive planning, evolutionary development and delivery and encourages rapid and flexible response to change. It is a conceptual framework that promotes interactions throughout the development of the life cycle.

Basic Principles: Following are the key principles of this methodology:

- Customer satisfaction by rapid delivery of software,
- Changes in the requirements are welcomed even late in development,
- Working software is delivered frequently (weeks rather than months),
- Sustainable development i.e., able to maintain constant pace,
- Daily co-operation between the users and developers,
- Face to face conversation which is the best form of communication,
- Continuous attention to technical improvement and good design,
- Simplicity,

- Regular adaption to changing circumstances, and
- Self – organized teams.

Strengths: [DADFH- DAD ask How did you Failed?]

- The team **does not** have to invest time and effort and finally find that the requirement of the user has changed by the time they deliver the product.
- Agile methodology has the concept of **adaptive** team which responds to the changing requirements,
- The **documentation** is crisp but to the point to save time,
- The end result is the **high quality** software in least possible time which results in satisfying user,
- **Face to face** communication and continuous inputs from user representatives leaves no space to decide result by guessing.

Weaknesses: [TELL – OI i.e., TELL to Isca]

- Agile increases potential **threats** to business continuity because they have less documentation as they focus on verbal communication rather than documents.
- In some of the software deliverables especially large ones, it is difficult to assess the **efforts** required at the beginning of the software development.
- There is **lack of emphasis** on necessary designing and documentation.
- Due to **lack of long term planning**, re-work is often required on Agile projects when the various components of the software are combined and forced to interact.
- The project can easily get **off track** if the representative is not clear about what they want.
- Agile lacks attention for outside **integration** because they do not invest time in identifying integration points in advance and therefore need for an integration point can become last minute surprise.

2.3 SYSTEM DEVELOPMENT LIFE CYCLE (SDLC): It is a framework which provides system designers and developers to follow a sequence of activities. It consists of steps 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 unless a proper document is prepared. This feature of the SDLC is critical for the successful management of an IS project.

The **advantages** of this system are as follows: **[HQDP- Helps sdlc in Quality Documentation Phase]**

- **Helps** in better planning and control for project managers.
- Ensure better **quality** because it has to comply with set standards.
- **Documentation** is an important feature which is also an important measure for communication and control.
- **Phases** are important milestones as it helps the project manager and user to review the system and for signoff.

From the perspective of the IS Audit, the following are the possible advantages: **[DR –ETc ie., Doctor ETc]**

- Based on **document** of each phase, IS auditor can have clear understanding of the various phases of the SDLC
- Based on his examination, he can **report** whether IS managers have complied with standards set by management, if any.
- The auditor can **evaluate** methods and techniques used in various development phases of SDLC.
- If IS Auditor has a **technical knowledge**, he can guide various phases of SDLC.

Risks Associated with SDLC: Some of the shortcomings of the SDLC are as follows:

- Development team may find it difficult to build

- The users may find that the end product will not sustain for a long time.
- Rigidity in approach may prolong the duration of project
- It may not be suitable for small and medium sized projects.

PHASES INVOLVED IN SDLC:-

PHASE NO	PHASE NAME	NATURE OF ACTIVITY
1	Preliminary Investigation	Determine and evaluate strategic benefits of the system and ensure that the solution fits the business strategy. Includes cost-benefit analysis of the proposed system.
2	Systems Requirements Analysis	Analyzing the system on the basis of the users requirements.
3	Systems Design	Design the system on the basis of the requirement by developing the system flowcharts, system and data flow diagrams, screens and reports.
4	Systems Development/Programming	Programming the system as designed and conduct the continuous testing and debugging
5	Systems Testing	Testing is conducted before the developed system is implemented.
6	Systems Implementation	After final testing and acceptance by management and user, new system is accepted.
7	Post Implementation Review and Maintenance	Continuous evaluation of the system as it functions in the live environment. Maintenance includes continuous evaluation of the system as it functions in the live environment and its updation.

2.4 THE PRELIMINARY INVESTIGATION

Objective: Objective of preliminary investigation is to determine the strategic benefits to be derived after implementing the system. The steps involved in the preliminary investigation phase are as follows:

- Identification of problem
- Identification of objective
- Delineation of scope
- Feasibility study

The following issues are typically addressed in the Feasibility Study:

- Determine whether the solution is as per the business requirements.
- Determine whether the existing system can rectify the situation without a major modification.
- Define the time frame for which the solution is required.
- Determine the approximate cost to develop the new system.
- Determine whether the vendor product offers a solution to the problem.

Document / Deliverable: A preliminary investigation report/ feasibility study for management.

2.4.1 Identification of Problem

The first step in a system development is to define the problem clearly after discussions with the user group. A problem which has a considerable impact on the organization is likely to receive immediate management's attention. Thus to identify a problem system analyst is assigned to make a preliminary investigation. After investigation he submits all alternative proposals for the given problem which might be most beneficial to the organization. Thus purpose of the preliminary investigation is to evaluate the project request.

System analyst on preliminary investigation should accomplish the following objectives: [Qn : what does system analyst do?]

- Clarify and understand the project request.

- Determine size of the project.
- Determine alternative approaches.
- Assess costs and benefits of alternative approaches.
- Report findings to the management with recommendation for acceptance or rejection of the proposal.

2.4.2 Identification of Objective: After identifying the problem, system analyst must work out for the objectives of the proposed solution. For instance, inability to provide a convenient reservation system, for a large number of intending passengers was the problem of the Railways. So its objective was “to introduce a system wherein intending passengers could book a ticket from source to destination, faster than in real-time.”

2.4.3 Delineation of Scope: The scope of a solution should be clear and comprehensible stating what solution will be provided to the users and what solution it will not provide. Outlining scope is essential right from the beginning.

The following questions should be answered while stating the scope:

- **Functionality requirements:** What functions will be delivered by solution?
- **Data to be processed:** What data will be required to achieve these functions?
- **Control requirements:** What are the control requirements for this application?
- **Performance requirements:** What level of response time, execution time is required?
- **Constraints:** What are the conditions the input data has to conform to?
- **Interfaces:** Does the application have to interface with any other hardware/software?
- **Reliability requirements:** Reliability means whether the application will remain uncorrupted from misuse.

While **eliciting information to delineate the scope**, few aspects need to be kept in mind:

- Different users will represent the problem and the solution in different ways. So, the system developer should take the need from the initiator as the base of the project.
- While the initiator of the project may be a member of the senior management but the actual users may be from the operating level. So, understanding his profile helps in designing appropriate system.
- While presenting the proposed solution for a problem, the development organization has to clearly quantify the economic benefits to be derived to the user organization.
- It is also necessary to understand the impact of the solution on the organization. Solutions which have a wide impact are likely to meet with greater resistance.
- While economic benefit is a critical consideration when deciding on a solution, there are several other factors that have to be given weightage too. These factors have to be considered from the perspective of the user management and resolved.

Two methods that help to define of the scope of the project can be analyzed as follows:

(i) Reviewing internal documents: Before conducting the investigation the analysts must first learn about the organization involved in the project by examining organization charts and studying written procedures, etc.

(ii) Conducting Interviews: Written documents tell the analyst how the systems should operate but does not give enough details on what basis decision are taken. To learn these details analysts must conduct interviews which describes the requirement of the project.

2.4.4 Feasibility Study: After possible solutions are identified project feasibility is carried out i.e., whether the system will be useful for the organization is determined. A feasibility study is carried out by the system analysts to evaluate alternative systems so that the most feasible system can be selected for development.

The following issues are typically addressed in the Feasibility Study:

- Determine whether the solution is as per the business requirements.

- Determine whether the existing system can rectify the situation without a major modification.
- Define the time frame for which the solution is required.
- Determine the approximate cost to develop the new system.
- Determine whether the vendor product offers a solution to the problem.

The Feasibility Study of a system is evaluated under following dimensions: **[what are the different types of feasibility study?]**

1. Technical: Is the technology needed available?
2. Financial: Is the solution viable financially?
3. Economic: Is return on Investment economical?
4. Operational: How will the solution work?
5. Behavioral: Is the solution going to bring any adverse effect on quality of work life?
6. Legal: Is the solution valid in legal terms?
7. Schedule / Time: Can the system be delivered on time?
8. Resources: Are human resources reluctant towards the solution?

1. Technical Feasibility: It is concerned with the issues pertaining to hardware and software. An analyst analyze whether the proposed system is feasible with the existing hardware and software technology. The technical issues usually raised during the feasibility stage of investigation include the following:

- Does the necessary technology exist to do what is suggested (and can it be acquired)?
- Can the proposed application be implemented with existing technology?
- Will the proposed system provide adequate responses to inquires?
- Can the system be expanded if developed?
- Are there technical guarantees for accuracy, reliability, ease of access, and data security?

2. Financial Feasibility: The solution proposed may be prohibitively costly for the user organization i.e., proposed solution should be within our cost

3. Economic Feasibility/Cost-Benefit Analysis: It includes an evaluation of all costs and benefits of the proposed system if implemented. The analysts must determine the amount of time and money required. 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 benefits in the form of reduced costs or fewer costly errors.
- The cost if nothing changes (i.e., the proposed system is not developed)

Estimating costs and benefits: After possible solution options are identified, each solution should be analyzed based on costs and benefits. **[What are the costs – benefits, system analyst must determine after solution are identified for system development?]**

Cost: System costs can be sub divided into Development, Operational and Intangible costs.

- **Development costs** for a system include costs of the system development process such as - salaries of the system analysts and computer programmers; cost of converting and preparing data files; cost of launching new computer facilities; cost of testing and documenting the system, cost of training employees, other start up costs; etc.
- **Operating costs** of a system include - salaries of computer operators and other data processing personnel; salaries of system analysts and computer programmers; cost of input data preparation; and Cost of maintaining proper physical facilities including power, light, heat, air conditioning, building, etc.
- **Intangible costs** are costs that cannot be easily measured. For example, reduction in customer sales or loss of goodwill due to errors in the development of a new system of an organization. Such costs are difficult to measure.

Benefits: The benefits which result from developing new or improved information systems can be divided into tangible and intangible benefits. Tangible benefits are those that can be accurately measured and are directly related to the introduction of a new system, such as decrease in data processing cost. Intangible benefits such as improved business image are harder to measure and define.

4. Operational Feasibility: It is concerned with ascertaining the views of workers, employees, etc about the use of computer facility. A system can be highly feasible in all respects except the operational and fails miserably because of human problems. Some of the questions which help in testing the operational feasibility of a project are stated below:

- Is there sufficient support for the system from the management as well as from the 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 any harm?
- Will individual performance poorer after implementation than before?

5. Behavioral Feasibility: It refers to the systems which is designed and whether it can 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.

6. Legal Feasibility: Legal feasibility is concerned with the conflict between a proposed system and the organization's legal obligations. Any system which violates the local legal requirements should also be rejected.

7. Schedule or Time Feasibility: Schedule feasibility involves estimating how long it will take for a new or revised system to become operational and communicating this information to the steering committee. For example, if a project takes 16 months to become fully functional then it may be rejected in favor of a simpler alternative which can be implemented in a shorter time frame.

8. Resources Feasibility: This focuses on human resources i.e., whether skilled personnel will be available to operate the proposed software solutions.

2.4.5 Reporting Results to Management: After the analyst identifies the problem and its scope, he provides alternative solution and estimates the cost and benefits of each alternative, and report the results of the same to the management. The report should recommend further procedures. From the analyst's report, management should determine what to do next i.e., which alternative to be selected or whether a combination of two alternatives is required or to conduct another round of analysis, etc.

2.5 SYSTEM REQUIREMENT ANALYSIS

Objectives: This phase includes a thorough understanding of the current system, identification of the areas that need modifications; requirements of users are analyzed and a fair idea about various systems development tools. The following activities are performed in this phase: [**CA-GMD i.e., Consult Analysis and Gather data to Model and Document activities**]

- To **consult** the stake owners to determine their expectations and resolve their conflicts.
- To **analyze** requirements and determine priorities.
- To **gather data** or find facts using tools like - interviewing, research/document collection, questionnaires, observation.
- To **model** activities such as developing models to document Data Flow Diagrams, E-R Diagrams.
- To **document** activities such as interview, questionnaires, reports etc. and development of a system (data) dictionary to document the modeling activities.

Document/Deliverable: A systems requirements report.

2.5.1 Fact finding Techniques: Every system is built to meet some set of needs. To analyze such needs analysts must interact with the users who will be benefited from the system and also to determine what are their actual requirements. Various fact-finding techniques are discussed below: **[DO - IQ]**

- **Documents:** Document means manuals, input / output forms, diagrams of how the current system works. Documents are also for procedure manuals, program codes of the current system, etc. Documents are a very good source of information about the user needs and the current system.
- **Observation:** Observation plays a central role in requirement analysis in prototype approaches. By observing how users react to prototypes of a new system, system can be successfully developed.
- **Interviews:** Users and managers should be interviewed to extract information in depth. The data gathered through interviews often provide complete picture of the problems and opportunities.
- **Questionnaires:** Through questionnaires large amount of information can be obtained from variety of users quickly. If the questionnaire is skillfully drafted, responses can be rapidly analyzed with the help of a computer.

2.5.2 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.

The following areas should be studied in depth:

- **Review historical aspects:** A brief study of history of the organization is a starting point for an analysis of the present system. The system analyst should review what system changes have been occurred in the past and whether it was successful or not.
- **Review data files maintained:** The analyst should investigate the data files maintained by each department, where they are located, who uses them number of times they are used, etc. He should also review all on-line and off-line files maintained in the organization.
- **Review 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 steps to accomplish a job. The analyst must analyze data-communications network in the present system to identify whether there is need to revise the network in new system or not.
- **Review internal controls:** An investigation is not complete until internal control is reviewed because it identifies weaknesses that should be removed from the new system.
- **Analyze inputs:** A detailed analysis of present inputs is important since they are basic source of data. Analysis should also be conducted to determine what is the source input of present system. Further analysis should be done to check whether output of one area is the input of other. Analyst must also understand the flow of input.
- **Analyze outputs:** The output should be scrutinized carefully by the system analysts to determine how well it meets the organization's needs. The analysts must understand what information is needed and why. Many times information obtained from reports are a carry-over from earlier days which have little or no relevance and therefore such reports should be eliminated.
- **Model the existing physical system:** All the important items in the existing system should be reviewed and analyzed and this process must be properly documented which allows a thorough understanding of numerous problems in the current system.
- **Undertake overall analysis of present system:** It is the final phase of analysis which includes thorough analysis of present work volume; user requirements; etc.

2.5.3 Systems Analysis of Proposed Systems: After analyzing each functional area of the present system along with its strengths and short comings, proposed system should be clearly defined which should be in conformity with the project's objectives are as follows:

- Database maintained should be accessible anytime i.e., online processing capabilities.
- Input data are prepared directly from original source documents for processing in the system.
- Methods and procedures that show the relationship of inputs and outputs to the database must utilize data communications wherever appropriate.
- Work volumes and timings should be considered carefully taking peak periods into consideration.

Qn: Why output – input process is recommended?]

Output must be the starting point for compiling these specifications because after outputs are determined, it is possible to infer what inputs, database, methods, procedures and data communications must be employed. The output-to-input process is therefore recommended since outputs are related directly to the objectives of the organization.

2.5.4 System Development Tools: Many tools and techniques have been developed to improve current information systems and to develop new ones. Such tools help end users and systems analysts to -

- Clarify, document and communicate the activities and resources involved in the organization;
- Analyze present business operations, management decision, etc of the organization;
- Propose and design new or improved information systems to solve business problems.

The major tools used for system development can be grouped into four categories based on the systems features. These are as follows:

System components and flows: They help the analyst to document the data flow among the major activities of a system. Eg: System flow charts are used to show the flow of data among hardware devices.

User interface: System analyst has to design the new system which can interface between end users and the computer system. Eg: Dialogue flow diagrams analyze the flow of dialogue between computers and people.

Data attributes and relationships: When data resources are defined, catalogued are designed by this category of tools.

- A Data Dictionary catalogs the description of the attributes (characteristics) of all data elements and their relationships with each other as well as with external systems.
- File layout forms document the type, size and names of the 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.

Detailed system process: These tools contain detailed procedures and processes to design a computer system. Eg: Decision trees and decision tables use a network or tabular form to document the complex conditional logic involved in choosing among the alternatives in a system.

We will now describe some of these tools in detail.

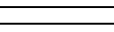
A). Structured English: It also known as Program Design Language (PDL) or Pseudo Code which uses English language with the syntax of structured programming. The Structured English aims at getting the benefits of both the programming language and natural language i.e., it is a combination of technical and English statements. Structured English consists of the following elements:

- Operation statements written as English phrases are executed from top down.

- Conditional statements such as IF, THEN, and ELSE.
- Repetition statements such as DO, WHILE, and UNTIL.

B) Flowcharts: Flowchart is a graphical technique used by analysts to represent the inputs, outputs and processes of a business in a pictorial form. They are used in analyzing, designing, documenting or managing a process or program in various fields.

C) Data Flow Diagrams: A Data Flow Diagram uses few simple symbols to illustrate the flow of data among external entities i.e., how external entities will interact with the system. Four symbols are combined to show how data are processed, they are as follow:

Symbol	Name	Explanation
	Data Sources and Destinations	The user and organizations that send and receive data from the system are represented by square boxes called Data destinations.
	Data flows	The flow of data into or out of a process is represented by curved or straight lines with arrows.
	Transformation Process	Data transformations from inputs to outputs are represented by circles.
	Data stores	The storage of data is represented by two horizontal lines.

D) Decision Tree: A Decision Tree (or tree diagram) helps in decision making with their possible consequences like chances of outcomes, costs, utility, etc. Decision tree is mainly used in decision analysis to identify a strategy which helps to reach goal and to calculate probabilities. It is a tree diagram or tree like graph.

E) Decision Table: A Decision Table is a table which may accompany a flowchart defining the possible contingencies which may be considered within the program and the appropriate course of action for each contingency. The four parts of the decision table are as follows:

1. Condition Stub - It lists the conditions;
2. Action Stub – It lists the actions to be taken;
3. Condition entries - It lists in its columns the possible permutations of answer to the questions in the 1. (i.e., conditions stub); and
4. Action entries - It lists in its columns corresponding to the 3. (i.e., condition entries) the actions contingent upon set of answers to the questions of that column.

F) CASE Tools: The data flow diagram and system flow charts that users review are commonly generated by using the on-screen drawing modules found in CASE (Computer-Aided-Software Engineering) software packages. CASE refers to the automation of anything that humans do to develop systems (i.e., it co-ordinate everything that human do to develop the system) and support virtually all phases of development process. An ideal CASE system would have an integrated set of tools and features to perform all aspects in the life cycle.

G) System Components matrix: It documents the resources used, the activities performed and the information produced by the system. It can be used as an information system by both systems analysts and system designers. It highlights how the basic activities are accomplished in an information system and how resources are converted into information products.

Table illustrated the use of a system component matrix to document the basic components of a sales processing and analysis system in an organization.

Information system Activities	Hardware Resources		Software Resources		People Resources		Data Resources	Information Products
	Machines	Media	Programs	Procedures	Specialists	Users		
Input	POS Terminals	Bar tags, Mag Stripe Cards	Data Entry program	Data entry procedures		Sales clerks customers	Customer data, product data	Data entry Displays
Processing	Mainframe Computer		Sales processing program, sales analysis program	Sales transaction procedures	Computer operators	Sales clerks managers	Customer, inventory and sales databases	Processing status displays
Output	POS Terminals, Management Workstations	Paper reports and receipts	Report generator program, Graphic program	Output use and distribution procedures		Sales Clerks managers, customers		Sales receipts, sales analysis reports and displays
Storage	Magnetic Disk Drives	Magnetic disk packs	Database management system program		Computer operators		Customer, inventory and sales database	
Control	POS Terminals, Management Workstations	Paper documents and control reports	Performance monitor program, security monitor program	Correction procedures	Computer operators control clerks	Sales Clerks managers, customers	Customer, inventory and sales database	Data entry display, sales receipts, Error display and signals

H) Data Dictionary:

- A data dictionary is a computer file that contains descriptive information about the data i.e., various data entering into the data warehouse; source of data; various uses of data that are stored in data dictionary; the identity of the computer programs; individuals permitted to access the data; individuals not permitted to access the data item etc.
- When new data is added then computer records it in the data dictionary. Similarly, when new programs are created then existing data is updated with this new program by data dictionary. Finally, when data fields are deleted from the records then their corresponding records are also dropped from the data dictionary.
- Accountants and auditors can also make good use of a data dictionary. For example, a data dictionary can help establish an audit trail because it can identify the input sources of data items, the computer programs that modify particular data items, and the managerial reports on which the data items are output. When an accountant is participating in the design of a new system, a data dictionary can also be used to plan the flow of transaction data through the system.

I) Layout form and Screen Generator, Menu Generator, Report generator, Code Generator:

1. **Layout form and Screen Generator:** They are for printed reports which are used to format or "paint" the desired layouts without entering complex formatting information.

2. **Menu Generator:** It outlines the functions which the system should accomplish. They may be linked to other submenus which enables users to understand how the screens and sub-screens will be used for data entry.
3. **Report Generator:** Report generator has capacity of performing similar functions as found in screen generators i.e., used to format or “paint” the desired layouts without entering complex formatting information and further indicates totals, paging and breaks in creating samples of the desired report.
4. **Code Generator:** They allow analyst to form modular units of the source code from the high level of specifications provided by the system analyst and helps in systems development process.

2.5.5 System Specification: At the end of analysis phase, system analyst prepares a document called Systems Requirement Specifications (SRS) document which is submitted to the management. A SRS contains the following:

- Introduction: Goals and Objectives for development of a software
- Information Description: Like description of problem; information contents, structure of the Hardware, software; human interfaces for external system; etc.
- Functional Description: Diagrammatic representation of functions; processing of each function; interplay between functions; etc.
- Behavioral Description: Response to external events and internal controls
- Validation Criteria: Class of tests to be performed to validate functions.
- Appendix: Data flow / Object Diagrams; Tabular Data; Detailed description of charts, graphs; etc.
- SRS Review: The development team makes a presentation and hands over the SRS document to management who reviews the same and if satisfied then signs the document.

2.5.6 Roles Involved in SDLC

1. Steering Committee: Some of the functions of Steering Committee are as follows:

- To provide overall direction.
- To conduct a regular review of progress of the project in the meetings of steering committee.
- Taking corrective actions, if necessary.

2. Project Manager: A project manager is normally responsible for more than one project. He is also responsible for completing the project within allocated time and budget and to conduct periodically review of the progress of the project.

3. Project Leader: The project leader has to ensure completeness and fulfillment of the project along with its objectives and also reviews the project position more frequently.

4. Systems Analyst / Business Analyst: The systems analysts' conducts interviews with the users and understand their requirements and then converts the users' requirements in the system.

5. Module Leader / Team Leader: A project is divided into several manageable modules and the development of each module is given to Module Leaders. They are also responsible for the delivering modules within the stipulated time and cost.

6. Programmer / Coder / Developer: Programmers converts design into programs. He also tests the program for debugging activity.

7. Database Administrator (DBA): The DBA handles multiple projects; ensures the integrity and security of information stored in the database. Inclusion of new data has to be done only with the approval of the database administrator.

8. Quality Assurance: This team sets the standards for development and checks compliance with these standards on a periodic basis.

9. Tester: Tester is junior level quality assurance personnel who tests programs as per the plan given by the module leader and prepare test reports.

10. Domain Specialist: Whenever a project team has to develop an application which is new to them, they take the help of a specialist of that field. For example, if a team undertakes application development in Insurance, about which they have little knowledge, they may seek the assistance

of an Insurance expert at different stages. This makes it easier to anticipate user needs. A domain specialist need not have knowledge of software systems.

11. IS Auditor: IS Auditor should be involved at the Design as well as the final Testing Phase to ensure the existence and the operations of the controls in the new software.

2.6 SYSTEMS DESIGN

After the completion of requirements analysis for a system, design of a system of that alternative which has been selected by management takes place.

Objective: Objective is to designs a system that best suites the users' requirements. It describes the parts of the system and their interaction; how the system shall be implemented, security plan and control point to prevent uncontrolled access, etc.

Activities: This phase includes activities like describing inputs and outputs, determining the processing steps, stating rules for the new solution, determining database system design; preparing the program specifications, etc.

Document / Deliverable: Blueprint or the design which will be used in system design shall be created

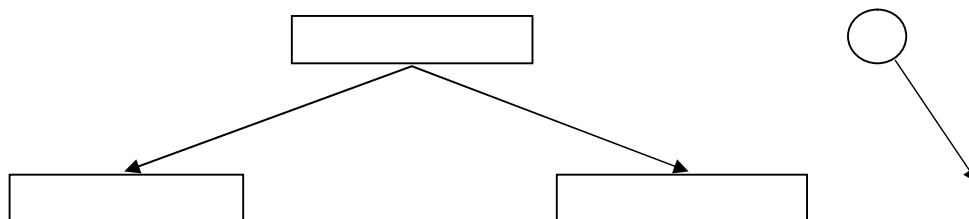
Qn: What are the steps involved in design of a new system?

System design involves first logical design and then physical construction of a system. The logical design describes the main features of the system and how they are interacted with one another. Physical construction produces program software, files and a working system. Design specifications instruct the programmers about what the system should do. Once the detailed design is completed, the design is then given to the system developers for coding. **The design phase involves following steps:**

- Architectural Design / Sub System / Decomposition;
- Design of the Data / Information Flow;
- Design of the Database;
- Design of the User-interface;
- Physical Design; and
- Design and acquisition of the hardware/system software platform

2.6.1 Architectural Design: Architectural design deals with the organizing the applications in terms of modules and sub -modules. Here we identify - major modules; function of each module; interface of each module; data received from / sent to / modified in other modules are also identified.

The architectural design is made with the help of Functional Decomposition. In this module is represented by a box and connected with arrows. Couple is data element that moves from one module to another which is shown by an arrow with circular tail.



2.6.2 Design of Data / Information flow: It is a major step in design of a new system. It defines the data flow for the proposed system, the inputs required for existing data; problems with the present system; and objective of the new system. All these have been identified in the analysis phase and documented in Software Requirements Specification (SRS) document.

2.6.3 Design of Database: It determines the scope ranging from local to global structure. The scope is decided on the basis of interdependence among organizational units. Greater the interdependence, greater will be the need for a global structure to prevent sub-optimization by subunits. The design of the database involves four major activities which are as follows:

Design Activity	Explanation
Conceptual Modeling	These describe the application domain via entities, attributes of these entities and their relationships.
Data Modeling	Conceptual Models need to be translated into data models so that they can be accessed by both high-level and low level programming languages
Storage Structure Design	Decisions must be made on how to partition the data so that it can be stored on some device. Eg: tuples (row) in a relational data model must be assigned to records, and relationships among records might be established via symbolic pointer addresses.
Physical Layout Design	Decisions must be made on how to distribute the storage structure across specific storage media and locations.

2.6.4 Design of User-Interface: Design of user interface means determining how the users will interact with a system like source documents to capture raw data; hard-copy output reports; graphic and color displays; special input/output device; etc.

Designing System Outputs: One of the most important features of a new system for users is what output system will generate. The right output must be developed so that users will find the system easy and effective to use.

Input Objectives: Input design consists of developing specifications for data preparation, developing steps to put data into a usable form for processing, and data-entry, i.e., the activity of putting the data into the computer for processing.

Output Objectives: The output from a system must accomplish one or more of the following objectives:

- Convey information about past activities, current status or future projections.
- Signal important events, opportunities or problems.
- Trigger an action.
- Confirmation of an action.

Important factors in Input / Output design: There are certain important factors listed in Table which should be considered by the system analyst while designing user input/ output forms

Characteristic	Definition	Input Design	Output Design
Content	It means actual data required to produce required output	The analyst should analyze types of data needed or to be gathered to generate desired user outputs.	The contents of a weekly output report to a sales manager might consist of sales person's name, sales made by each sales person during the week, etc.
Timeliness	Timeliness means when does users require outputs, i.e., daily, weekly, monthly, quarter or annually.	A plan must be developed to determine when to enter input in system, so that required output is achieved whenever required. i.e., based on output, input is entered.	A sales manager may require weekly or monthly sales report. Other users, such as airline agents, require both real-time information and rapid response times to render better client service.
Format	Input format means physical arrangement of data.	Input formats are designed on the basis of few constraints	Output format means how contents are organized so

	Output format means arrangement of data output on a printed form i.e., on report or in a display screen.	like - the type and length of each data field and any other special characteristics.	that it assists in decision-making, identifying and solving problems, etc.
Media	Input-output medium refers to the physical device used for input, storage or output.	Input media means on which device data should be entered. Various input device are magnetic tapes / disks, key-boards, pen-drives, voice input etc. A suitable medium should be selected based on requirements.	A variety of output media are paper, video display, microfilm, magnetic tape/disk and voice output.
Form	Form refers, the way information is inputted in the input form and the way it is presented to users in various output forms - quantitative, non-quantitative, text, graphics, video and audio.	Forms are pre-printed papers that require people to fill in a standardized way.	The form of the output should be decided based on the requirements of the concerned user. For example – Text, audio, video, etc.
Input Volume / Output Volume	Input volume means amount of data that has to be entered in the computer system at any one time. Output volume means amount of data required at any one time.	In some decision-support systems and real-time processing systems, input volume is light whereas in batch-processing transaction systems, input volume could be heavy.	If output volume is heavy it is better to use high-speed printer or a rapid-retrieval display unit

2.6.5 Physical Design: For the physical design, the logical design is transformed into units which are further decomposed into implementation units such as programs and modules. During physical design, the auditor's primary concern is efficiency and effectiveness issues. The auditor should also evidence that designers follow some type of structured approach like – CASE tools.

Design Principles: [SMART]

- The design should be as per laid down **standards**.
- The design should be **modular**.
- The design should be based **after analysis**.
- The design should be **related** to business activities.
- There is a **tendency** to develop only one design and consider it as final. However it is recommended to develop two or three design and selects the best out of them.

Modularity: A module is a manageable unit containing data and instructions to be performed in well-defined task. Interaction among modules is based on well-defined interfaces. Modularity is measured by two parameters: Cohesion and Coupling.

Cohesion means the manner in which elements within a module are linked.

Coupling refers to interconnection between the modules. In a good modular design, cohesion will be high and coupling low.

2.6.6 Design of the Hardware / System Software Platform: In some cases, the new system requires hardware and system software not currently available in an organization which has to be developed to support the application system. If different hardware and software are not able to communicate with each other then the subsequent changes will have to be made and resources should be extended in trying to make the hardware and software compatible to each other.

2.7. SYSTEM ACQUISITION: After system is designed the next phase of system development is acquisition of hardware, software and services.

2.7.1 Acquisition Standards: Acquisition standards should focus on -

- Ensuring security, reliability, and functionality of existing system.

- Reviews of vendor's contract, license and acquiring products compatible with existing systems.
- Including invitations-to-tender and request-for-proposals. Invitations-to-tender involve soliciting bids from vendors for acquiring hardware or software. Request-for-proposals involve soliciting bids when acquiring third-party developed software.

2.7.2 Acquiring Systems Components from Vendors: At the end of the design phase, the organization will get an idea of the type of hardware, software and services required for the system development.

Management should also decide whether the hardware is to be purchased, leased or to be rented.

- **Hardware Acquisition:** While procuring hardware, management normally relies on the time tested selection techniques and the objective selection criteria can be delegated to the technical specialist.
- **Software Acquisition:** After input and output designs are finalized, the systems analyst must determine the type of software required. This determination helps the systems developers in deciding about the hardware or software which will be most suited for desired output. They should also determine whether the application software should be created in-house or acquired from a vendor.
- **Contracts, Software Licenses and Copyright Violations:** Contracts between an organization and a vendor should clearly describe the rights and responsibilities of the parties to the contract. The contracts should be in writing and in detail.
Software license is a license that grants permission to do things with computer software i.e., it authorizes activities which are prohibited for others.
Copyright laws protect proprietary as well as open-source software. An organization will be exposed to litigation if it uses unlicensed software or violates licensing agreement.
- **Validation of Vendors' proposals:** The contracts and software license consists of evaluation and ranking of various proposals submitted by various vendors which are quite difficult and time consuming because each vendor offers variety of configurations which has to be rigorously evaluated based on the following factors:
 - ✓ Performance of each proposed system in relation to its costs;
 - ✓ Costs - Benefits analysis of each proposed system;
 - ✓ Maintainability of each proposed system;
 - ✓ Compatibility of each proposed system with existing systems; and
 - ✓ Vendor Support i.e., after sales service.
- **Methods of validating the proposal:**
Some of the validation methods are as follows: [**CP = PBT i.e. Cost Price = Profit Before Tax**]
 - ✓ **Checklists:** It is the most simple and subjective method for validation and evaluation. The various criteria are put in check list in the form of questions which has to be responded by various vendors.
 - ✓ **Public Evaluation Reports:** Several consultancy agencies compare their hardware and software performance with various other manufacturers and publish their reports. For those where published reports are not available, resort would have to be made to other methods of validation. This method is useful when buying staff does not have adequate knowledge of facts.
 - ✓ **Point-Scoring Analysis:** It is an objective mean of selecting a final system. There are no absolute rules in the selection process, only guidelines should match with users needs. Here points are allotted to each criteria or question.

For example – Table illustrates a Point Scoring Analysis list.

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

- ✓ **Bench marking program for vendor's proposals:** Benchmarking program for vendors' proposals are sample programs which have been designed to represent proposed requirements. Benchmarking is a technique which tests, whether the design or computer offered by the vendor meets the requirements of the buyer.
- ✓ **Test problems:** They disregard actual job mix (i.e., current configuration) and are devised to test the true capabilities of the hardware, software or system. For example, test problems may be developed to evaluate the time required to translate the source code (high level language) into the object code (machine language), response time for two or more jobs in multi-programming environment, length of time required to execute an instruction, etc. The results, achieved by the machine can be compared and judgment can be made.

2.8. DEVELOPMENT: PROGRAMMING TECHNIQUES AND LANGUAGES

Objective: To convert the specification into a functioning system.

Activities: Application programs are written, tested and documented to conduct system testing.

Document / Deliverable: A fully documented system.

A good coded program should have the following characteristics:

- **Reliability:** It means program should be consistent over a period of time. However poor setting of parameters or hard coding of some data could result in the failure of a program after some time.
- **Robustness:** It refers to process of taking inputs and outputs in least likely situations.
- **Accuracy:** It refers to what program should do and should also take note of what it should not do.
- **Efficiency:** It means performance should not be affected due to increase in input values.
- **Usability:** It means user-friendly interface and easy-to-understand for any program.
- **Readability:** It refers to maintenance of the program even in the absence of the program developer.

SOFTWARE DEVELOPMENT LIFE CYCLE

2.8.1 Program Coding Standards: Different programmers write a program using different sets of rules but gives the same results and therefore, coding standards should be defined which provides simplicity, efficient utilization of the storage and least processing time.

2.8.2 Programming Language: Application programs are coded in the form of statements or instructions which is converted by the compiler to binary machine, so that computer can understand the instruction and execute. The programming languages commonly used are as follows COBOL, C++, JAVA, JAVA Script, etc.

Choice of Programming Language: The choice of programming language should be decided on the basis of application area; environment in which software has to be executed; data structure complexity; knowledge of software development staff; and capability of in-house staff for maintenance; etc.

2.8.3 Program Debugging: It is a testing activity which corrects programming language syntax and diagnoses errors so that program compiles easily. A clean compile successfully converts the program from the source code into machine language instructions. Debugging can be a tedious task consisting of following four steps:

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

2.8.4 Test the program: A careful testing of each program should be done so that system can be installed successfully. The programmer should plan the testing considering all possible exceptions.

2.8.5 Program Documentation: Managers and users should carefully review documentation to ensure that the software or system functions as indicated in the document. If not, then documentation should be revised.

2.8.6 Program Maintenance: The requirements of business data processing continuously changes and this calls for modification of the various programs which should be properly maintained.

2.9. SYSTEM TESTING: Testing is a process used to identify the correctness, completeness and quality of developed computer software. Testing should uncover all errors with minimum amount of time and with minimum efforts.

Different levels of Testing are as follows:

2.9.1 Unit Testing: Unit testing is a software verification and validation method in which programmer tests whether the individual units are fit for use. The goal of unit testing is to isolate each part of the program and show that the individual parts are correct and works as intended.

There are **five categories** of tests that a programmer typically performs on a program unit:

[FPSSP – Failed Percentage Student Should Pass]

- **Functional Tests:** It checks whether programs do what they are supposed to do or not. The programmer also checks whether the actual and expected result matches or not.
- **Performance Tests:** It verifies response time, execution time, throughput, traffic rates on data channels, etc. It also compares new system performance with similar system using well defined bench marking.
- **Stress Tests:** It determines stability of a given system. Testing is done beyond normal capacity in order to observe the results. They are performed during peak hours when there is large volume of data and thus test its performance.
- **Structural Tests:** Structural Testing examines the internal processing of a software system.
- **Parallel Tests:** In Parallel Testing same test data is used in both old and new system and output results are compared.

Types of Unit Testing

(I) Static Analysis Testing

Some important Static Analysis Tests are as follows:

- **Desk Check:** This is done by the programmer himself and checks for syntax errors and deviation from coding standards.
- **Structured walk-through:** The application developer leads other programmers through the text of the program and explanation.
- **Code inspection:** The program is reviewed by the formal committee. Review is done with formal checklists.

(II) Dynamic Analysis Testing

- **Black Box Testing:** Black Box Testing takes an external view to derive test cases. These tests can be functional or non-functional. The designer selects both valid and invalid inputs and derives correct output. This method of testing is applicable to all the levels of the software testing: unit, integration, functional and system testing. The higher the level, greater the need of black box testing to simplify testing. However, one cannot be sure that all existent paths are tested. If a module performs a function which it is not supposed to do, then the black box testing cannot identify it.
- **White Box Testing:** White box testing takes an internal view to derive test cases. The tester selects inputs to exercise paths and derives the appropriate outputs through codes. It is applicable at the unit, integration and system levels of the testing. Normally tests paths within a unit but it can also test paths between the units during integration and also between subsystems during a system level testing. This test ensures working of internal operation with specifications.
- **Gray Box Testing:** Gray box testing uses a combination of black box testing and white box testing. Here, tester applies a limited number of test cases to the internal workings of the software under test and in the remaining part applies black box approach.

2.9.2 Integration Testing: Integration testing is an activity of software testing where individual modules are combined and tested as a group. It occurs after unit testing and before system testing. It evaluates whether information received by the module is the same which has been sent by earlier module. It is carried out in the following manner:

- **Bottom-up Integration:** Bottom-up integration is the traditional strategy used to integrate the components of a software system into a functioning as a whole. It consists of unit testing, followed by sub-system testing, and then testing of the entire system. The disadvantage is that testing of major decisions is deferred to a later period.
- **Top-down Integration:** Top-down integration starts with the main routine and stubs are substituted for the modules, directly subordinate to the main module. Once the main module testing is complete, stubs are substituted with real modules one by one, and these modules are tested with stubs. The difficulty in the top-down method is that high-level modules are tested from stubs and not with real outputs from subordinate modules.
- **Regression Testing:** Each time a new module is added as part of integration testing, the software changes and this causes problems with functions that previously worked flawlessly. In the context of the integration testing, the regression tests ensure that changes or corrections have not introduced new errors. The data used for the regression tests should be the same as the data used in the original test.

2.9.3 System Testing: System testing is a process where software and other systems are tested as a whole. The purpose of system testing is to ensure that the new or modified system functions properly i.e., it checks whether all of them work together properly and also simultaneously. **The types of testing that might be carried out are as follows:**

- **Recovery Testing:** It tests how well application can recover from crashes, hardware failures and other similar problems. Recovery testing is the forced failure of the software to verify whether recovery procedure can be properly performed.
- **Security Testing:** It determine whether the system protects data and maintains functionality as intended or not. This testing technique also ensures the existence and proper execution of access controls in the new system.
- **Stress Testing:** It determines stability of a given system. Testing is done beyond normal capacity in order to observe the results. They are performed during peak hours when there is large volume of data and thus test its performance.
- **Performance Testing:** It verifies response time, execution time, throughput, traffic rates on data channels, etc. It also compares new system performance with similar system using well defined bench marking.

2.9.4 Final Acceptance Testing: Final Acceptance Testing is conducted when the system is ready for implementation. System testing is conducted by a company on the system supplied by the system developer. This testing ensures that the new system satisfies the quality standards of the business and also satisfies requirements of the users. It has two major parts:

- **Quality Assurance Testing:** It ensures that the new system satisfies the prescribed set standards of the quality and is developed as per the organization's requirements.
- **User Acceptance Testing:** It ensures that the functional aspects required by the users are been properly executed in the new system.

2.10. SYSTEMS IMPLEMENTATION:

Objective: To implement the new system i.e. put it into operation.

Activities: The activities involved in System Implementation are as follows:

- Conversion of data to the new system files.
- Training of end users.
- Proper documentation.
- System changeover.
- Evaluation of the system at regular intervals.

Document / Deliverable: A full functional / documented system in its operational environment.

This process ensures that new system is operational and also allows users to take over its operation for use is called Systems Implementation.

2.10.1 Activities during Implementation Stage

The activities involved in system implementation stage are as follows:

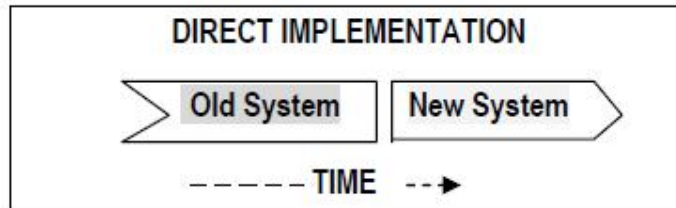
(I) Equipment Installation: Hardware required to support the new system should be selected prior to the implementation phase. For installation adequate time should be given to allow completion of the following activities:

- **Site Preparation:** An appropriate environment for operating equipment must be found which should be as per vendor's temperature, humidity and dust control specifications.
- **Installation of new hardware / software:** The equipment must be physically installed by the manufacturer with proper connection to the power and wired to communication lines, if required.
- **Equipment check out:** The equipment must be tested thoroughly under normal operating conditions by both vendor and the implementation team to ensure that equipments are proper and in working condition.

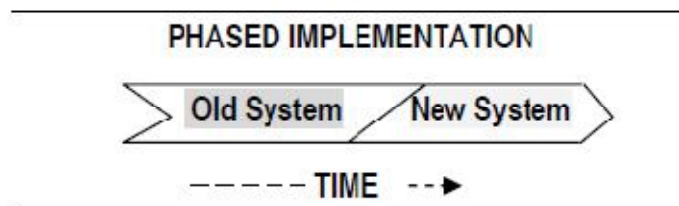
(II) Training Personnel: A system can succeed or fail depending on the way it is being operated and used. Therefore, the proper training should be given to the personnel involved with the system for successful implementation of system.

(III) System Implementation Conversion Strategies: Conversion or changeover is the process of changing from the old system to a new system. It requires careful planning for actual changeover. The four types of implementation strategies are as follows:

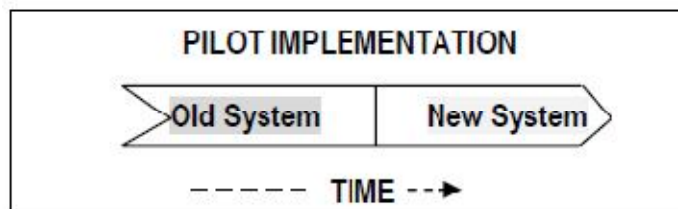
- **Direct Implementation / Abrupt change-over:** In this strategy, the changeover is done in one operation itself i.e., completely replacing the old system at once or on a set date. Direct Implementation usually takes place on a set date often during a break in production or a holiday period so that new system is installed without causing too much disruption.



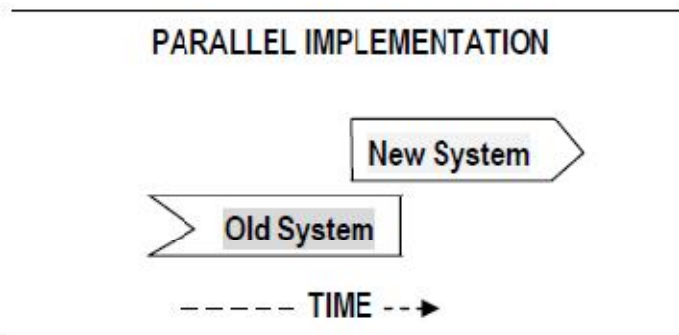
- **Phased implementation:** Under this strategy old system is converted into new system in phased manner i.e. after one phase is successfully converted next phase begins until last phase is successfully completed up till the new system fully replaces the old one.



- **Pilot implementation:** In this strategy the new system replaces the old one at once but only on a small scale so that any errors can be rectified or further any benefits can be incorporated throughout the whole system with least disruption. For example - it might be tried out in one branch of the company or in one location. If successful then the pilot is extended until it eventually replaces the old system completely.



- **Parallel running implementation:** This is considered as the most secure method where the both systems run parallel for a short period of time and both of them are operated independently. If all goes well then the old system is being replaced by new system.



2.10.2 Activities involved in conversion: Conversion includes all those activities which must be completed to successfully convert from the previous system to the new system. These activities are as follows:

- **Procedure conversion:** Operating procedures should be completely documented for the new system that applies to both computer-operations and functional area operations. Information on input, data files, methods, procedures, output, and internal control must be presented in clear, concise and understandable terms for an average reader.
- **File conversion:** This phase should be started long before because large files of information must be converted from one medium to another. In order for the conversion to be as accurate as possible, file conversion programs must be thoroughly tested. The existing computer files should be kept for a sufficient period of time until files are properly backed up.
- **System conversion:** After on-line and off-line files have been converted and the reliability of the new system has been confirmed, daily processing can be shifted from the existing information system to the new one. All transactions are then processed on the new system. Simultaneously the old system should be operated for some more time to check the total results of both systems.
- **Scheduling personnel and equipment:** Scheduling data processing operations of a new 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.

2.11. POST IMPLEMENTATION REVIEW AND SYSTEMS MAINTENANCE

Objective: To assess and review the complete working of a new system after implementation.

Activities: Some of the Systems maintenance activities are as follows:

- Adding new data elements;
- Modifying reports;
- Adding new reports; and
- Changing calculations.

Document / Deliverable: A document should state any scope for improvements, if any like-

- Could further training can improve the degree of benefit being generated?
- Are there further functional improvements or changes that would deliver greater benefit?
- Are specific improvements required in procedures, documentation, support, etc?
- What learning points are there for future projects?

2.11.1 Post Implementation Review: A Post Implementation Review should answer the most important question i.e. “Did we achieve what we set out to do in business terms?” It should further ascertain the success of the project, particularly in achieving its objectives and should also measure improvements which can further be incorporated. A Post-Implementation Review should be conducted some time after the new system is deployed, say ranging from six weeks to six months. There are two basic dimensions of Information system that should be evaluated. The first dimension is concerned with whether the 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. The review should be able to answer the following question:

- **Development evaluation:** Due to the uncertainty and mystique associated with system development, very few system are developed within scheduled time and given budget and therefore the development process is primarily concerned with whether the system was developed on or before schedule date and within given budget.
- **Operation evaluation:** This evaluation pertains to answer whether the hardware, software and personnel are capable to perform their duties. Operation evaluation is relatively straightforward if evaluation criteria are established in advance.
- **Information evaluation:** A system should also be evaluated in terms of information it provides i.e., what information is being provided by the new system should be evaluated over here. It should be able to provide information that supports the organizational decision system.

2.11.2 System Maintenance: Maintaining the system is an important aspect of SDLC. Maintenance can be categorized in the following ways: **[Case Study Qn:]**

- **Scheduled maintenance:** Scheduled maintenance is anticipated and can be planned for. For eg., the implementation of a new inventory coding scheme can be planned in advance.
- **Rescue maintenance:** Rescue maintenance refers to previously undetected errors that were not anticipated but require immediate solution. A system that is properly developed and tested should have few occasions of rescue maintenance.
- **Corrective maintenance:** Corrective maintenance deals with fixing bugs (i.e., fixing solution) in the code or defects found. A defect can result from design errors, logic errors; coding errors, data processing and system performance errors.
- **Adaptive maintenance:** Adaptive maintenance consists of adapting software as per changes in the environment, such as the hardware or the operating system. The term environment in this context refers to the totality of all conditions from outside of the system, for example, business rule, government policies, etc.
- **Perfective maintenance:** Perfective maintenance mainly deals with accommodating to new or changed user requirements and activities to increase the system's performance or to enhance its user interface.
- **Preventive maintenance:** Preventive maintenance activities aim at increasing the system's maintainability, such as updating documentation, adding comments, and improving the modular structure of the system.

2.12. OPERATION MANUALS: A user's guide is commonly known as an Operation Manual which is a technical communication document that gives assistance to the user on how to use a particular system. It is usually written by a technical writer. Operation manuals are most commonly associated with electronic goods, computer hardwares and softwares. The section of an operation manual includes the following:

- A cover page, a title page;
- A preface containing details of information on how to use the user's guide;
- A contents page;
- A guide on how to use the main functions of the system;
- A troubleshooting section stating possible errors and problems that may occur, along with solution;
- A FAQ (Frequently Asked Questions);
- Where to find further help along with contact details;
- A glossary and for larger documents an index.

2.13. Auditors Role in SDLC: The audit of system under development can have three main objectives:

- To provide an opinion on the efficiency, effectiveness and economy of project management,
- To assess the extent to which the system being developed provides adequate audit trails and controls, to ensure the integrity of data processed; and
- To assess the controls being provided for the system's operation.

For the first objective to achieve, an auditor will have to attend project and steering committee meetings; examine project documentation and conduct interviews. This is required to ensure what project control standards are to be complied with and extent of compliance achieved.

For the second objective, auditor can examine system documentation to arrive at opinion on controls. The auditor's opinion will be based on the degree to which the system satisfies the general control objectives.

For the third objective, the auditor should provide list of standard controls for operational concerns such as response time, CPU usage, etc

An auditor may adopt a rating system like 1 to 10 to the various phases of SDLC. Eg in rating a feasibility study, an auditor can review feasibility study report and can also interview personnel who have conducted this feasibility study. Based on this an auditor can arrive at a rating (10 being

the best). After deriving such a rating for all the phases, an auditor can form his overall opinion about the SDLC phases. In order to audit technical work products such as physical design, an auditor may opt to include a technical expert to seek his opinion. However, an auditor will have to give control objectives, directives, etc. Some of them are:

- Documented policy and procedures;
- Established project team with all infrastructure and facilities;
- Development is carried over as per standards, functional specifications;
- Separate test environment for development / test production / test plans;
- Business owners testing and approval before system goes alive;
- Adequate audit trails are provided in system; and
- Appropriateness of methodology selected.

Further, post implementation review is carried out to determine whether the system meets the business requirements. An auditor should determine whether the expected benefits of the new system are realized and whether users are satisfied with the new system. Further auditor should also review which of the SDLC phases have not met the desired objectives and whether any corrective action was taken. If differences exist between expected and actual results then the auditor should determine the reasons for the same.

SELF STUDY QUESTIONS:

1. Write short notes on the following:
 - a) System Manual
 - b) Systems Flow Chart
 - c) CASE Tools
 - d) Technical Feasibility
 - e) Audit Trails
 - f) Preventive Controls
 - g) Point-Scoring Analysis.
 - h) Factors for validation of vendor's proposals
 - i) Data Flow Diagram
 - j) Data Dictionary
 - k) Economic Feasibility
 - l) Crypto systems
 - m) Public Key Infrastructure
 - n) Guidelines for input form design
2. The top management of company has decided to develop a computer information system for its operations. Is it essential to conduct the feasibility study of system before implementing it? If answer is yes, state the reasons. Also discuss three different angles through which the feasibility study of the system is to be conducted.
3. Explain the System Development Life Cycle.
4. What are the Systems Development Tools? Explain briefly.
5. What do you understand by feasibility study? Explain various types of feasibility studies.
6. State and explain any five approaches to System Development.
7. Discuss in detail, how the analysis of present system is made by the system analyst.

OR

A Company is offering a wide range of products and services to its customers. It relies heavily on its existing information system to provide up to date information. The company wishes to enhance its existing system. You being an information system auditor, suggest how the investigation of the present information system should be conducted so that it can be further improved upon.

8. What do you understand by "Requirement analysis"? What is the significance of analyzing the present system and how is it carried out? Explain briefly.

OR

What are the facts finding techniques used by a system analyst?

9. If you are the Project Manager of a Software Company with the responsibility for developing a break-through product, combining state of the art hardware and software, will you opt for prototyping as a process model for a product meant for the intensely competitive entertainment market?
10. Describe briefly four categories of the major tools that are used for system development.
11. Bring out the reasons as to why the organizations fail to achieve their Systems Development Objectives?

12. Describe any five functional areas of a system which needs to be analyzed by system analyst for detailed investigation of the present system.
13. What are the various tangible and intangible benefits that can result from the development of a computerized system?
14. Write short note on top down approach of system development.
15. What are the project management items associated with an I.T. project system failures? Give the elements to be included in the adopted framework to avoid such failures.
16. Explain different conversion strategies used for conversion from manual to computerized system. Enumerate the advantages and disadvantages of each strategy.
17. Discuss briefly, various activities that are involved for successful conversion of an existing manual system to a computerized information system.
18. ABC Technologies Ltd. is in the development of application software for various domains. For the development purposes, the company is committed to follow the best practices suggested by SDLC. SDLC provides the guidelines to follow a sequence of activities. It consists of a set of steps and phases in which each phase of the SDLC uses the results of the previous one. The SDLC is document driven which means that at crucial stages during the process, documentation is produced. A phase of the SDLC is not complete until the appropriate documentation or artifact is produced. These are sometimes referred to as deliverables.
19. State main objectives of system development tools. Briefly describe the major categories of documentation tools that are used for system development with any one simple illustrative example for each.
20. What are the major factors to be considered in designing User inputs? Explain.
21. What are the six important factors which should be considered while designing the user outputs?
22. As a system analyst, you need to assess the successful implementation and stakeholder's actual requirements of an enterprise system in a retail chain organization across its branches to provide the following features:
 - Lower operational costs,
 - Better information for managers, and
 - Smooth operations for users or better levels of service to customers.
 Justify your answer with the necessary techniques used to determine the requirements of a system.
23. What are the cost elements that need to be considered for cost estimate of a system? What are the major benefits that a system can provide?
24. Discuss the system development approach whose components are time boxing, incremental prototyping and clean rooms.

RTP QUESTIONS:

RTP NOV - 10

1. Describe the reasons why organizations fail to achieve their systems development objectives.
2. Explain the strengths and weaknesses of the Waterfall model.
3. What are the aspects that should be kept in mind while eliciting information to delineate the scope?
4. Discuss the areas which should be studied in depth, in order to fully understand the present system and its related problems.
5. Explain the characteristics of a good coded program.
6. Describe various activities involved in conversion from previous system to the new information system.

RTP MAY 11

1. Explain the strengths and weaknesses of Prototyping model.
2. 'Requirements Analysis phase includes a thorough and detailed understanding of the current system, identifications of the areas that need modifications to solve the problem, the determination of user/managerial requirements and to have fair idea about various systems development tools'. Briefly discuss the activities, which are performed in this phase.
3. Describe physical design principles in brief.
4. 'Management should establish acquisition standards that address the same security and reliability issues as development standards.' What are the areas that should be focused by acquisition standards?
5. Explain various categories of system maintenance.
6. 'Large organizations would naturally tend to adopt a sophisticated and objective approach to validate the vendor's proposal'. What are the validation methods that may be used for this purpose?
7. Write short notes on the following:
 - (a) Strengths of Incremental Model

- (b) Program Debugging
- (c) System Testing

RTP NOV 11

1. Why organizations fail to achieve their systems development objectives; explain in brief.
2. Discuss the weaknesses of Rapid Application Development (RAD) approach.
3. 'While eliciting information to delineate the scope, few aspects need to be kept in mind during the preliminary investigation of the development life cycle'. Explain these aspects, in brief.
4. 'Management should establish acquisition standards that address the same security and reliability issues as development standards'. What are the areas that need to be focused by Acquisition Standards?
5. Discuss the major characteristics of a good coded program.

RTP MAY 12

1. What is Waterfall Model? Describe its strengths and weaknesses in brief.
2. Discuss basic principles of Prototyping Model in brief.
3. What are the major activities that are performed in the requirements analysis phase of the SDLC?
4. What is Integration testing? Explain bottom-up and top-down integration.
5. Discuss System testing along with its other associated testing techniques.
6. Write short notes on the following:
 - (a) Technical Feasibility
 - (b) Final Acceptance Testing
 - (c) Program Debugging

EXAMS QUESTIONS:

Nov – 10

1. As a person in charge of System Development Life Cycle, you are assigned a job of developing a model for a new system which combines the features of a prototyping model and the waterfall model. Which will be the model of your choice and what are its strengths and weaknesses? 8
2. From the perspective of IS audit, what are the advantages of System Development Life Cycle? 4

May - 11

1. What do you mean by system requirements analysis? What are the activities to be performed during system requirement analysis phase? 5
2. Discuss in brief the various functional areas to be studied by a system analyst for a detailed investigation of the present system. 8

Nov – 11

1. What areas are required to be studied in order to know about the present system? Write the problems that the ABC Udyog is presently facing. 5
2. What kind of training you will recommend to enrich the human resources for effective utilization of the proposed new system and standards? 5
3. At the end of analysis phase, the System Analyst prepares a document called "Systems Requirement Specifications (SRS)". Write the contents of SRS. 4
4. What is the significance of Post Implementation Review? How it is performed? 4
5. Following are involved in the System Development Life Cycle (SDLC). Discuss their roles: 4
 - (i) Project Manager.
 - (ii) System Analyst.
 - (iii) Database Administrator (DBA).
 - (iv) IS Auditor.
6. Draw the flowchart to find the sum of first 50 even numbers, starting from 2. 4
7. Write short notes on Objectives of an Operating System 4
8. Write short notes on Information System Maintenance 4

May - 12

1. What are tangible and intangible benefits that can result from the development of a computerized system? 6
2. What are the major activities involved in the design of a database? 4
3. Write short notes on Data Dictionary 4

[illegible]