System analysis and design

Answer key

PART – A (ANY SIX QUESTIONS)

1. Write a note on functions of a systems analyst?

A system analyst is responsible for analyzing, designing and implementing systems to fulfill organizational needs. He/she plays a vital role in making operational the management information system. The role of the system analyst has however changed.

An organization requires system analysts as line managers normally do not have an understanding of the kind of information-based solutions that are possible for their business problems. A system analysts bridges this gap as he/she is has a thorough knowledge of both the business systems and business processes. A system analyst is therefore in a position to provide information system based solutions to organizations after having studied the problem that the organization is facing. They understand both business and technology. They study a business problem or opportunity and devise an information system enabled solution for it by detailing the information system specifications. This set of specification that the analyst delivers is in a technical format which is easily understandable to a technical (IT) specialist. The technical specialist might not understand the business issue, if it comes directly from the line managers as he has very little knowledge of business processes. The system analyst then bridges the gap between the two by translating and transforming the business problem/opportunity into a information systems solution and supplying the specification of such a system to the technologist who can then take up the task and build the actual system.

2. Explain testing project feasibility.

Feasibility Study can be considered as preliminary investigation that helps the management to take decision about whether study of system should be feasible for development or not.

- It identifies the possibility of improving an existing system, developing a new system, and produce refined estimates for further development of system.
- It is used to obtain the outline of the problem and decide whether feasible or appropriate solution exists or not.
- The main objective of a feasibility study is to acquire problem scope instead of solving the problem.
- The output of a feasibility study is a formal system proposal act as decision document which includes the complete nature and scope of the proposed system.

Types of Feasibilities

Economic Feasibility

- It is evaluating the effectiveness of candidate system by using cost/benefit analysis method.
- It demonstrates the net benefit from the candidate system in terms of benefits and costs to the organization.

- The main aim of Economic Feasibility Analysis (EFS) is to estimate the economic requirements of candidate system before investments funds are committed to proposal.
- It prefers the alternative which will maximize the net worth of organization by earliest and highest return of funds along with lowest level of risk involved in developing the candidate system.

Technical Feasibility

- It investigates the technical feasibility of each implementation alternative.
- It analyzes and determines whether the solution can be supported by existing technology or not.
- The analyst determines whether current technical resources be upgraded or added it that fulfill the new requirements.
- It ensures that the candidate system provides appropriate responses to what extent it can support the technical enhancement.

Operational Feasibility

- It determines whether the system is operating effectively once it is developed and implemented.
- It ensures that the management should support the proposed system and its working feasible in the current organizational environment.
- It analyzes whether the users will be affected and they accept the modified or new business methods that affect the possible system benefits.
- It also ensures that the computer resources and network architecture of candidate system are workable.

Behavioral Feasibility

- It evaluates and estimates the user attitude or behavior towards the development of new system.
- It helps in determining if the system requires special effort to educate, retrain, transfer, and changes in employee's job status on new ways of conducting business.

Schedule Feasibility

- It ensures that the project should be completed within given time constraint or schedule.
- It also verifies and validates whether the deadlines of project are reasonable or not.

3. What is data dictionary?

A data dictionary is a structured repository of data elements in the system. It stores the descriptions of all DFD data elements that is, details and definitions of data flows, data stores, data stored in data stores, and the processes.

A data dictionary improves the communication between the analyst and the user. It plays an important role in building a database. Most DBMSs have a data dictionary as a standard feature. For example, refer the following table −

Sr.No.	Data Name	Description	No. of Characters
1	ISBN	ISBN Number	10
2	TITLE	title	60

3	SUB	Book Subjects	80
4	ANAME	Author Name	15

4. What are the tools of prototyping?

Prototypes are early samples, models, or releases of products built to test a concept or process. There are many contexts in which semantics can be used, for example, in design, electronics, and software programming. Generally, prototypes are used by system analysts and users to improve the precision of a new design.

Prototyping allows you to build simple, small-scale prototypes of your products, and use them to observe, record, and assess user performance levels or the users' general behavior and reactions to the overall design. Designers can then make appropriate refinements or possible alterations in the right direction.

Prototypes can be of any form, from simple sketches and storyboards to rough paper prototypes and even role-playing prototypes that enact a service offering. They do not need to be complete products – in fact, you can prototype a part of a product to test that part of your solution. Often, prototypes are quick and rough - designed for early-stage testing and understanding – and at times full-formed and detailed – aimed for pilot trials towards the project's final stages.

Figma, InVision Studio, Adobe XD, Webflow, Axure RP, Origami Studio and Justinmind are some popular prototyping tools.

5. What are the categories of computer aided system tools?

CASE tools are set of software application programs, which are used to automate SDLC activities. CASE tools are used by software project managers, analysts and engineers to develop software system.

There are number of CASE tools available to simplify various stages of Software Development Life Cycle such as Analysis tools, Design tools, Project management tools, Database Management tools, Documentation tools are to name a few.

Diagram tools

These tools are used to represent system components, data and control flow among various software components and system structure in a graphical form. For example, Flow Chart Maker tool for creating state-of-the-art flowcharts.

Process Modeling Tools

Process modeling is method to create software process model, which is used to develop the software. Process modeling tools help the managers to choose a process model or modify it as per the requirement of software product. For example, EPF Composer

Project Management Tools

Project management tools help in storing and sharing project information in real-time throughout the organization. For example, Creative Pro Office, Trac Project, Basecamp.

Documentation Tools

Documentation in a software project starts prior to the software process, goes throughout all phases of SDLC and after the completion of the project.

Analysis Tools

These tools help to gather requirements, automatically check for any inconsistency, inaccuracy in the diagrams, data redundancies or erroneous omissions. For example, Accept 360, Accompa, CaseComplete for requirement analysis, Visible Analyst for total analysis.

Design Tools

These tools help software designers to design the block structure of the software, which may further be broken down in smaller modules using refinement techniques. These tools provides detailing of each module and interconnections among modules. For example, Animated Software Design

Configuration Management Tools

An instance of software is released under one version. Configuration Management tools deal with –

- Version and revision management
- Baseline configuration management
- Change control management

Change Control Tools

These tools are considered as a part of configuration management tools. They deal with changes made to the software after its baseline is fixed or when the software is first released. CASE tools automate change tracking, file management, code management and more. It also helps in enforcing change policy of the organization.

Programming Tools

These tools consist of programming environments like IDE (Integrated Development Environment), in-built modules library and simulation tools. These tools provide comprehensive aid in building software product and include features for simulation and testing. For example, Cscope to search code in C, Eclipse.

Prototyping Tools

Software prototype is simulated version of the intended software product. Prototype provides initial look and feel of the product and simulates few aspect of actual product.

6. Write a note on designing of printed outputs.

A design output is a drawing or specification or manufacturing instruction. Design outputs describe all the components, parts, and pieces that go into your medical device. Design outputs describe all assemblies and subassemblies of your product.

Think of it like this. If you were tasked with assembling a medical device from scratch, what documentation would you need to do so? All this documentation that you would need--design outputs.

Design outputs established during product development become the basis of the device master record (DMR) when in production.

Each manufacturer shall establish and maintain procedures for defining and documenting design output in terms that allow an adequate evaluation of conformance to design input requirements. Design output procedures shall contain or make reference to acceptance criteria and shall ensure that those design outputs that are essential for the proper functioning of the device are identified. Design output shall be documented, reviewed, and approved before release. The approval, including the date and signature of the individual(s) approving the output, shall be documented.

7. What are the objective of input design?

Input Design
Input design involves capturing of data as well as inputting it to the computer. According input design consists of data capturing and data validation.

Input Design Objectives
1. Input Design is the process of converting a user oriented description of the input into a computer based system. This design is important to avoid errors in the data input process and show the correct direction to the management for getting correct information from the computerized system.

2. It is achieved by creating user friendly screens for the data entry to handle large volume of data. The goal of designing input is to make data entry easier and to be free from errors. The data entry screen is designed in such a way that all the data manipulates can be performed. It also provides record viewing facilities.

3. Then the data is entered it will check for its validity. Data can be entered with the help of screens propriate messages are provided as when needed so that the user will not be in mail of instant. Thus the objective of input design is to create an input layout that is easy to follow.

8. Explain visual display units.

Visual materials has assisted in the communication process since ages in form of paintings, sketches, maps, diagrams, photographs, etc. In today's world, with the invention of technology and its further growth, new potentials are offered for visual information such as thinking and reasoning. As per studies, the command of visual thinking in human-computer interaction (HCI) design is still not discovered completely. So, let us learn the theories that support visual thinking in sense-making activities in HCI design.

An initial terminology for talking about visual thinking was discovered that included concepts such as visual immediacy, visual impetus, visual impedance, and visual metaphors, analogies and associations, in the context of information design for the web.

As such, this design process became well suited as a logical and collaborative method during the design process.

9. What are the ways of training in systems implementation?

Training

The personnel in the system must know in detail what their roles will be, how they can use the system, and what the system will or will not do. The success or failure of well-designed and technically elegant systems can depend on the way they are operated and used.

Training Methods

Instructor-led training

It involves both trainers and trainees, who have to meet at the same time, but not necessarily at the same place. The training session could be one-on-one or collaborative. It is of two types −

Virtual Classroom

In this training, trainers must meet the trainees at the same time, but are not required to be at the same place. The primary tools used here are: video conferencing, text based Internet relay chat tools, or virtual reality packages, etc.

Normal Classroom

The trainers must meet the trainees at the same time and at the same place. They primary tools used here are blackboard, overhead projectors, LCD projector, etc.

Self-Paced Training

It involves both trainers and trainees, who do not need to meet at the same place or at the same time. The trainees learn the skills themselves by accessing the courses at their own convenience.

10. How will you select hardware and software for system implementations?

The systems come with hardware, software and support. Today, selecting a system is a serious and time-consuming business.
The selection process should be viewed as a project, and a project team should be organized with management support. In larger projects, the team includes one or more user representatives, an analyst and EDP auditor, and a consultant. Several steps make up the selection process.

1. Requirements analysis
2. System specifications
3. Request for proposal(RFP)

4. Evaluation and validation
5. Vendor selection
6. Post-installation review

Software selection:

Software selection is a critical aspect for system development. There are 2 ways of acquiring the software.
Custom -made
Packages

Criteria for Software selection:

Reliability – It is the probability that the software will executed in a specific period of time without any failures. It is important to the professional user. It brings up the concept of modularity, or the ease which a package can be modified.

Functionality – It is the definition of the facilities, performance and other factors that the user requires in the finished product.

Capacity – Capacity refers to the capability of the software package to handle the users requirements for size of files, number of data elements, and reports. All limitations should be checked.

Flexibility – It is a measure of effort required to modify an operational program. One feature of flexibility is adaptability.

Usability – This criteria refers to the effort required to operate, prepare the input, and interpret the output of a program.

PART-B

11. Explain in detail about various categories of information systems.

An information system is a combination of software, hardware, and telecommunication networks to collect useful data, especially in an organisation. Many businesses use information technology to complete and manage their operations, interact with their consumers, and stay ahead of their competition. Some companies today are completely built on information technology, like eBay, Amazon, Alibaba, and Google.

Types of information systems

There are various information systems, and the type of information system a business uses depends on its goal and objective. Here are the four main types of information systems:

1. **Operations support systems** – The first type of information system is the operation support system. Such type of information system mainly supports a specific type of operation in a business. An example is the transaction processing system used in all banks

worldwide. This type of information system enables the service provider to assess a specific process of business.

2. **Management information systems** – This is the second category of information systems, consisting of hardware and software integration allowing the organisation to perform its core functions. They help in obtaining data from various online systems. The data thus obtained is not stored by the system; rather, it is analysed in a productive manner to help in the management of an organisation.
3. **Decision support systems** – An organisation can make an informed decision about its operations using decision support systems. It analyses the rapidly changing information that cannot be determined in advance. It can be used in completely automated systems and human-operated systems. However, for maximum efficiency combination of human and computer-operated systems is recommended.
4. **Executive information systems** – EIS or executive support system is the last category that serves as management support systems. They help in making senior-level decisions for an organisation.

OR

12. Explain in detail about classical systems development lifecycle.

System Development Life Cycle

An effective System Development Life Cycle (SDLC) should result in a high quality system that meets customer expectations, reaches completion within time and cost evaluations, and works effectively and efficiently in the current and planned Information Technology infrastructure.

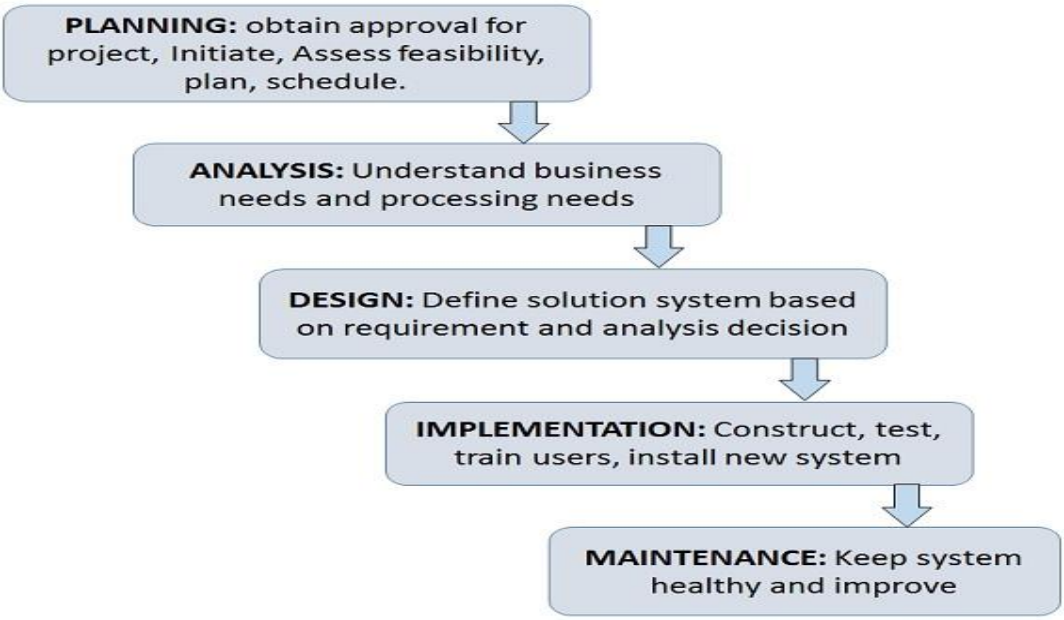
System Development Life Cycle (SDLC) is a conceptual model which includes policies and procedures for developing or altering systems throughout their life cycles.

SDLC is used by analysts to develop an information system. SDLC includes the following activities −

- requirements
- design
- implementation
- testing
- deployment
- operations
- maintenance

Phases of SDLC

Systems Development Life Cycle is a systematic approach which explicitly breaks down the work into phases that are required to implement either new or modified Information System.



Feasibility Study or Planning

- Define the problem and scope of existing system.
- Overview the new system and determine its objectives.
- Confirm project feasibility and produce the project Schedule.
- During this phase, threats, constraints, integration and security of system are also considered.
- A feasibility report for the entire project is created at the end of this phase.

Analysis and Specification

- Gather, analyze, and validate the information.
- Define the requirements and prototypes for new system.
- Evaluate the alternatives and prioritize the requirements.
- Examine the information needs of end-user and enhances the system goal.
- A Software Requirement Specification (SRS) document, which specifies the software, hardware, functional, and network requirements of the system is prepared at the end of this phase.

System Design

- Includes the design of application, network, databases, user interfaces, and system interfaces.
- Transform the SRS document into logical structure, which contains detailed and complete set of specifications that can be implemented in a programming language.
- Create a contingency, training, maintenance, and operation plan.
- Review the proposed design. Ensure that the final design must meet the requirements stated in SRS document.
- Finally, prepare a design document which will be used during next phases.

Implementation

- Implement the design into source code through coding.
- Combine all the modules together into training environment that detects errors and defects.
- A test report which contains errors is prepared through test plan that includes test related tasks such as test case generation, testing criteria, and resource allocation for testing.

- Integrate the information system into its environment and install the new system.

Maintenance/Support

- Include all the activities such as phone support or physical on-site support for users that is required once the system is installing.
- Implement the changes that software might undergo over a period of time, or implement any new requirements after the software is deployed at the customer location.
- It also includes handling the residual errors and resolve any issues that may exist in the system even after the testing phase.
- Maintenance and support may be needed for a longer time for large systems and for a short time for smaller systems.

13. what do you mean by data flow analysis? Explain in detail.

Data-flow analysis is a technique for gathering information about the possible set of values calculated at various points in a computer program. A program's control-flow graph (CFG) is used to determine those parts of a program to which a particular value assigned to a variable might propagate. The information gathered is often used by compilers when optimizing a program. A canonical example of a data-flow analysis is reaching definitions.

A simple way to perform data-flow analysis of programs is to set up data-flow equations for each node of the control-flow graph and solve them by repeatedly calculating the output from the input locally at each node until the whole system stabilizes, i.e., it reaches a fixpoint. This general approach, also known as Kildall's method, was developed by Gary Kildall while teaching at the Naval Postgraduate School

Data-flow analysis is the process of collecting information about the way the variables are defined and used in the program. It attempts to obtain particular information at each point in a procedure. Usually, it is enough to obtain this information at the boundaries of basic blocks, since from that it is easy to compute the information at points in the basic block. In forward flow analysis, the exit state of a block is a function of the block's entry state. This function is the composition of the effects of the statements in the block. The entry state of a block is a function of the exit states of its predecessors.

Data Flow Diagrams (DFD) or Bubble Chart

It is a technique developed by Larry Constantine to express the requirements of system in a graphical form.

- It shows the flow of data between various functions of system and specifies how the current system is implemented.
- It is an initial stage of design phase that functionally divides the requirement specifications down to the lowest level of detail.
- Its graphical nature makes it a good communication tool between user and analyst or analyst and system designer.
- It gives an overview of what data a system processes, what transformations are performed, what data are stored, what results are produced and where they flow.

Basic Elements of DFD

DFD is easy to understand and quite effective when the required design is not clear and the user wants a notational language for communication. However, it requires a large number of iterations for obtaining the most accurate and complete solution.

The following table shows the symbols used in designing a DFD and their significance −

Symbol Name	Symbol	Meaning
Square		Source or Destination of Data
Arrow		Data flow
Circle		Process transforming data flow
Open Rectangle		Data Store

Types of DFD

DFDs are of two types: Physical DFD and Logical DFD. The following table lists the points that differentiate a physical DFD from a logical DFD.

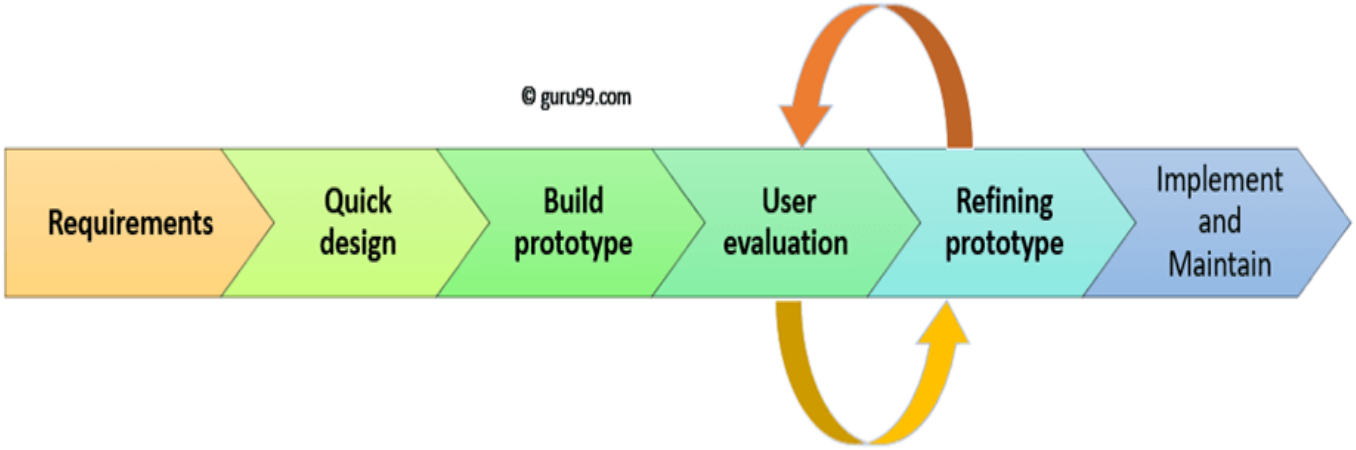
Physical DFD	Logical DFD
It is implementation dependent. It shows which functions are performed.	It is implementation independent. It focuses only on the flow of data between processes.
It provides low level details of hardware, software, files, and people.	It explains events of systems and data required by each event.
It depicts how the current system operates and how a system will be implemented.	It shows how business operates; not how the system can be implemented.

OR

14. Explain various steps involved in uses of prototypes.

Prototyping Model is a software development model in which prototype is built, tested, and reworked until an acceptable prototype is achieved. It also creates base to produce the final system or software. It works best in scenarios where the project's requirements are not known in detail. It is an iterative, trial and error method which takes place between developer and client.

Prototyping Model Phases



Prototyping Model has following six SDLC phases as follow:

Step 1: Requirements gathering and analysis

A prototyping model starts with requirement analysis. In this phase, the requirements of the system are defined in detail. During the process, the users of the system are interviewed to know what is their expectation from the system.

Step 2: Quick design

The second phase is a preliminary design or a quick design. In this stage, a simple design of the system is created. However, it is not a complete design. It gives a brief idea of the system to the user. The quick design helps in developing the prototype.

Step 3: Build a Prototype

In this phase, an actual prototype is designed based on the information gathered from quick design. It is a small working model of the required system.

Step 4: Initial user evaluation

In this stage, the proposed system is presented to the client for an initial evaluation. It helps to find out the strength and weakness of the working model. Comment and suggestion are collected from the customer and provided to the developer.

Step 5: Refining prototype

If the user is not happy with the current prototype, you need to refine the prototype according to the user's feedback and suggestions.

This phase will not over until all the requirements specified by the user are met. Once the user is satisfied with the developed prototype, a final system is developed based on the approved final prototype.

Step 6: Implement Product and Maintain

Once the final system is developed based on the final prototype, it is thoroughly tested and deployed to production. The system undergoes routine maintenance for minimizing downtime and prevent large-scale failures.

15. Explain various computer aided system tools in detail.

CASE tools are set of software application programs, which are used to automate SDLC activities. CASE tools are used by software project managers, analysts and engineers to develop software system.

There are number of CASE tools available to simplify various stages of Software Development Life Cycle such as Analysis tools, Design tools, Project management tools, Database Management tools, Documentation tools are to name a few.

Diagram tools

These tools are used to represent system components, data and control flow among various software components and system structure in a graphical form. For example, Flow Chart Maker tool for creating state-of-the-art flowcharts.

Process Modeling Tools

Process modeling is method to create software process model, which is used to develop the software. Process modeling tools help the managers to choose a process model or modify it as per the requirement of software product. For example, EPF Composer

Project Management Tools

These tools are used for project planning, cost and effort estimation, project scheduling and resource planning. Managers have to strictly comply project execution with every mentioned step in software project management. Project management tools help in storing and sharing project information in real-time throughout the organization. For example, Creative Pro Office, Trac Project, Basecamp.

Documentation Tools

Documentation in a software project starts prior to the software process, goes throughout all phases of SDLC and after the completion of the project.

Documentation tools generate documents for technical users and end users. Technical users are mostly in-house professionals of the development team who refer to system manual, reference manual, training manual, installation manuals etc. The end user documents describe the

functioning and how-to of the system such as user manual. For example, Doxygen, DrExplain, Adobe RoboHelp for documentation.

Analysis Tools

These tools help to gather requirements, automatically check for any inconsistency, inaccuracy in the diagrams, data redundancies or erroneous omissions. For example, Accept 360, Accompa, CaseComplete for requirement analysis, Visible Analyst for total analysis.

Design Tools

These tools help software designers to design the block structure of the software, which may further be broken down in smaller modules using refinement techniques. These tools provides detailing of each module and interconnections among modules. For example, Animated Software Design

Configuration Management Tools

An instance of software is released under one version. Configuration Management tools deal with –

- Version and revision management
- Baseline configuration management
- Change control management

CASE tools help in this by automatic tracking, version management and release management. For example, Fossil, Git, Accu REV.

Change Control Tools

These tools are considered as a part of configuration management tools. They deal with changes made to the software after its baseline is fixed or when the software is first released. CASE tools automate change tracking, file management, code management and more. It also helps in enforcing change policy of the organization.

Programming Tools

These tools consist of programming environments like IDE (Integrated Development Environment), in-built modules library and simulation tools. These tools provide comprehensive aid in building software product and include features for simulation and testing. For example, Cscope to search code in C, Eclipse.

Prototyping Tools

Software prototype is simulated version of the intended software product. Prototype provides initial look and feel of the product and simulates few aspect of actual product.

Prototyping CASE tools essentially come with graphical libraries. They can create hardware independent user interfaces and design. These tools help us to build rapid prototypes based on existing information. In addition, they provide simulation of software prototype. For example, Serena prototype composer, Mockup Builder.

Web Development Tools

These tools assist in designing web pages with all allied elements like forms, text, script, graphic and so on. Web tools also provide live preview of what is being developed and how will it look after completion. For example, Fontello, Adobe Edge Inspect, Foundation 3, Brackets.

Quality Assurance Tools

Quality assurance in a software organization is monitoring the engineering process and methods adopted to develop the software product in order to ensure conformance of quality as per organization standards. QA tools consist of configuration and change control tools and software testing tools. For example, SoapTest, AppsWatch, JMeter.

Maintenance Tools

Software maintenance includes modifications in the software product after it is delivered. Automatic logging and error reporting techniques, automatic error ticket generation and root cause Analysis are few CASE tools, which help software organization in maintenance phase of SDLC. For example, Bugzilla for defect tracking, HP Quality Center.

OR

16. Explain detail about various methods of presenting information.

The output generally refers to the result and information that are generated by the system. One of the most important features of an information system from the point of view of users is the output it produces. If the output is of poor quality, the whole system is in peril because the users will then avoid using it.

Output Media:
Output from a computer system can take a variety of forms. The systems analyst has to determine the most appropriate medium for the output. The most common media are

1. Printed Output:

The device used for printed output may be line printer, dot matrix printer, laser printer or plotter.

2. Visual Output:

With the increasing emphasis on the design of interactive management information systems, the CRT unit is becoming a widely used form of output. The most common use of CRT unit is inquiry whether hard copy is not required.

3. Turnaround Document Output:

In order to reduce the input workload at a later date, turnaround documents in the form of punched cards are widely used. The applications include Credit Card Billing and Employee Time Cards. Additionally OCR (Optical Character Recognition) forms can be prepared as output which at a later date serves as input to the computer system.

4. Secondary Storage Output:

This generally includes magnetic disk, magnetic drum, and magnetic tape.

5. Microfilm or Microfiche Output:

Microfilms are photographically reduced documents on films. The information can subsequently be inspected by using a viewer which projects on to screen.

6. Audio Response Output:

A newer form of output is the audio response unit. The unit is capable of providing on-line inquiry into the systems where output is restricted to short messages. This system consists of message handling unit.

17. Explain various input validation method in detail.

Input Design

Input design involves capturing of data as well as inputting it to the computer. According input design consists of data capturing and data validation.

Input Design Objectives

1. Input Design is the process of converting a user oriented description of the input into a computer based system. This design is important to avoid errors in the data input process and show the correct direction to the management for getting correct information from the computerized system.

2. It is achieved by creating user friendly screens for the data entry to handle large volume of data. The goal of designing input is to make data entry easier and to be free from errors. The data entry screen is designed in such a way that all the data manipulates can be performed. It also provides record viewing facilities.

3. Then the data is entered it will check for its validity. Data can be entered with the help of screens propriate messages are provided as when needed so that the user will not be in mail of instant. Thus the objective of input design is to create an input layout that is easy to follow.

Data Capture:

There can be no information system without "Data". Data are the facts which describe events and entities. Data are communicated by various types of symbols such as letters of the alphabets, numbers, speech patterns, dots and dashes, hand signals, pictures and so on. The processed data with specific purpose are called information. Data obtained in general are not suitable for directly feeding into the computer.

Data Validation:

The objective of a data validation system is to detect errors at the earliest possible stage before costly activities are performed on invalid data. Some data validation is done by way of manual verification in data capture stage itself.

Validation Checks: There are various categories of checks which can be applied to data during a validation run.

a. **Field Checks:** Includes the followings:

1**. Limit check:** May be applied to each field (data item) of a record to ensure that its contents lie within predefined size.

2. **Picture check:** May be applied to each field to detect entry of incorrect characters in the field.

3. **Valid code check**: To validate input against predefined transaction codes. These predefined codes may either be embedded in the programs or stored in files.
4. **Check Digit**: It is used to detect transposition errors when recording "key" fields.
5. **Arithmetic Check**: are used to ensure the validity of the results by performing arithmetic operations in different ways.
6. **Cross Check:** may be applied to verity fields appearing in different files to verify that result fully.

b**. Transaction checks**: include the following:

1**. Sequence checks**: are applied to detect any missing transaction.
2. **Formal completeness**: are used to check the presence and position of all fields in a transaction.
3. **Redundant** data checks: are employed to check the validity of codes with reference to description.
4. **Combination checks:** may be applied on various fields of a file.
5. **Probability checks**: are sued to avoid unnecessary rejection of data.
6. **Pass words**: may be exercised to check entry of data by unauthorized persons in on line system.
7. **Checks:** may be incorporated to ensure that transaction pertains to the current period.
8**. Batch total**: Can be used to ensure that transaction have been transcribed correctly.
9. **Hash total:** A control total i.e. the sum of values in a particular field or record area of a file, to ensure that transactions have been transmitted currently.

OR

18. Explain various dialog strategies in detail.

Dialog Representation

To represent dialogs, we need formal techniques that serves two purposes −

- It helps in understanding the proposed design in a better way.
- It helps in analyzing dialogs to identify usability issues. E.g., Questions such as "does the design actually support undo?" can be answered.

Introduction to Formalism

There are many formalism techniques that we can use to signify dialogs. In this chapter, we will discuss on three of these formalism techniques, which are −

- The state transition networks (STN)
- The state charts
- The classical Petri nets

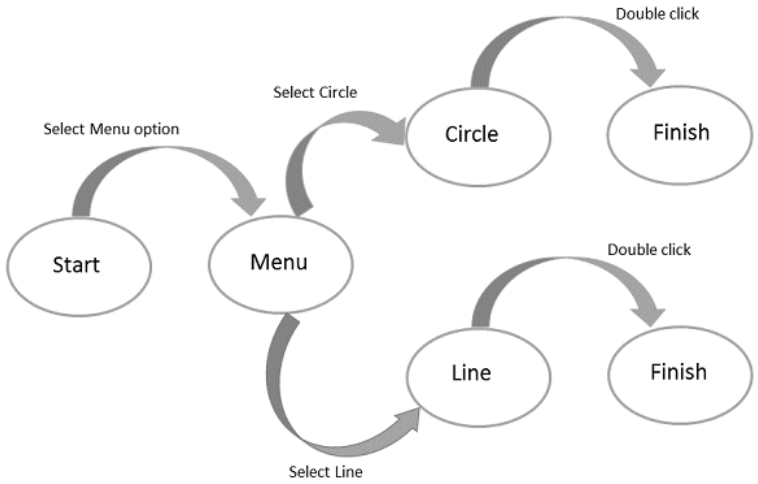
State Transition Network (STN)

STNs are the most spontaneous, which knows that a dialog fundamentally denotes to a progression from one state of the system to the next.

The syntax of an STN consists of the following two entities −

- **Circles** − A circle refers to a state of the system, which is branded by giving a name to the state.
- **Arcs** − The circles are connected with arcs that refers to the action/event resulting in the transition from the state where the arc initiates, to the state where it ends.

STN Diagram



State Charts

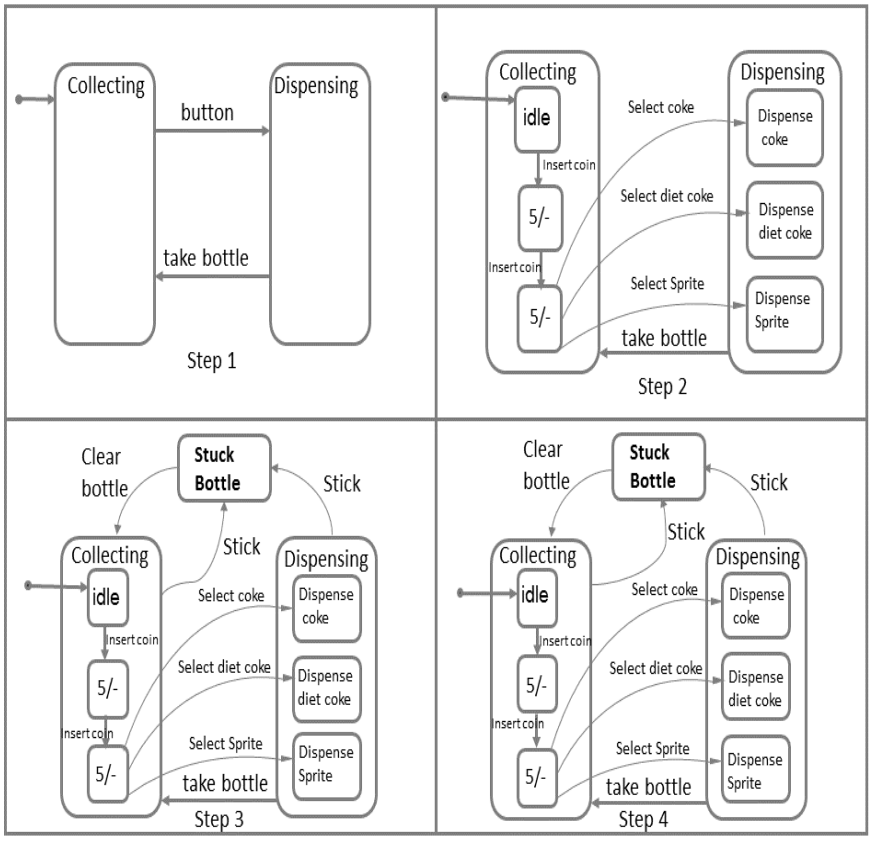
State Charts represent complex reactive systems that extends Finite State Machines (FSM), handle concurrency, and adds memory to FSM. It also simplifies complex system representations. State Charts has the following states −

- Active state − The present state of the underlying FSM.
- Basic states − These are individual states and are not composed of other states.
- Super states − These states are composed of other states.

Illustration

For each basic state b, the super state containing b is called the ancestor state. A super state is called OR super state if exactly one of its sub states is active, whenever it is active.

Let us see the State Chart Construction of a machine that dispense bottles on inserting coins.



The above diagram explains the entire procedure of a bottle dispensing machine. On pressing the button after inserting coin, the machine will toggle between bottle filling and dispensing modes. When a required request bottle is available, it dispense the bottle. In the background, another procedure runs where any stuck bottle will be cleared. The 'H' symbol in Step 4, indicates that a procedure is added to History for future access.

Petri Nets

Petri Net is a simple model of active behavior, which has four behavior elements such as − places, transitions, arcs and tokens. Petri Nets provide a graphical explanation for easy understanding.

- Place − This element is used to symbolize passive elements of the reactive system. A place is represented by a circle.
- Transition − This element is used to symbolize active elements of the reactive system. Transitions are represented by squares/rectangles.
- Arc − This element is used to represent causal relations. Arc is represented by arrows.

Token − This element is subject to change. Tokens are represented by small filled circles.

19. What are the various conversion methods in system implementation? Explain in detail.

Conversion

It is a process of migrating from the old system to the new one. It provides understandable and structured approach to improve the communication between management and project team.

Conversion Plan

It contains description of all the activities that must occur during implementation of the new system and put it into operation. It anticipates possible problems and solutions to deal with them.

It includes the following activities −

- Name all files for conversions.
- Identifying the data requirements to develop new files during conversion.
- Listing all the new documents and procedures that are required.
- Identifying the controls to be used in each activity.
- Identifying the responsibility of person for each activity.
- Verifying conversion schedules.

Conversion Methods

The four methods of conversion are −

- Parallel Conversion
- Direct Cutover Conversion
- Pilot Approach
- Phase-In Method

Method	Description	Advantages	Disadvantages
Parallel Conversion	Old and new systems are used simultaneously.	Provides fallback when new system fails. Offers greatest security and ultimately testing of new system.	Causes cost overruns. New system may not get fair trail.
Direct Cutover Conversion	New system is implemented and old system is replaced completely.	Forces users to make new system work Immediate benefit from new methods and control.	No fall back if problems arise with new system Requires most careful planning

Pilot Approach	Supports phased approach that gradually implement system across all users	Allows training and installation without unnecessary use of resources. Avoid large contingencies from risk management.	A long term phasein causes a problem of whether conversion goes well or not.
Phase-In Method	Working version of system implemented in one part of organization based on feedback, it is installed throughout the organization all alone or stage by stage.	Provides experience and line test before implementation When preferred new system involves new technology or drastic changes in performance.	Gives impression that old system is erroneous and it is not reliable.

File Conversion

It is a process of converting one file format into another. For example, file in WordPerfect format can be converted into Microsoft Word.

For successful conversion, a conversion plan is required, which includes −

- Knowledge of the target system and understanding of the present system
- Teamwork
- Automated methods, testing and parallel operations
- Continuous support for correcting problems
- Updating systems/user documentation, etc

Many popular applications support opening and saving to other file formats of the same type. For example, Microsoft Word can open and save files in many other word processing formats.

OR

20. What do you mean by post implementation review? Explain various methods.

Post-Implementation Evaluation Review (PIER)

PIER is a tool or standard approach for evaluating the outcome of the project and determine whether the project is producing the expected benefits to the processes, products or services. It enables the user to verify that the project or system has achieved its desired outcome within specified time period and planned cost.

PIER ensures that the project has met its goals by evaluating the development and management processes of the project.

Objectives of PIER

The objectives of having a PIER are as follows −

- To determine the success of a project against the projected costs, benefits, and timelines.
- To identify the opportunities to add additional value to the project.
- To determine strengths and weaknesses of the project for future reference and appropriate action.
- To make recommendations on the future of the project by refining cost estimating techniques.

The following staff members should be included in the review process −

- Project team and Management
- User staff
- Strategic Management Staff
- External users

PIR PROCESS

The PIR process comprises three stages:

Stage1. Post-issuance date implementation monitoring
Stage 2. Post-effective date evaluation of costs and benefits
Stage 3. Summary of research and reporting.

The commencement of a PIR and opportunities to participate in PIR activities will be announced publicly to stakeholders. The following is a summary of each stage.

Stage 1: Post-Issuance Date Implementation Monitoring

This stage begins after issuance of the standard and continues until at least three years after the latest effective date of the standard. During the post-issuance date implementation monitoring period (and before the effective date), at a macro level the Board will:

1. Actively monitor practice as stakeholders prepare for initial implementation of the standard.
2. Develop and disseminate implementation guidance and educational material.
3. Communicate and perform outreach with stakeholder organizations, including outreach with the academic community (leveraging advisory councils, committees, and resource groups where applicable) to generate interest in research activities associated with the standard that are the subject of the PIR process.

After the effective date of the final standard, the Board will perform an archival review of financial reports and survey financial statement preparers (with the assistance of the academic community when needed).

Stage 2: Post-Effective Date Evaluation of Costs and Benefits

This stage begins after the effective date of the final standard and continues for approximately three to five years.
Stage 2 activities consist of:

1. Understanding the costs that an entity incurred in applying the standard as well as the costs that investors and other users incurred in analyzing and interpreting the information that the standard provides
2. Understanding the benefits of the standard to investors and other users as well as to entities
3. Monitoring the ongoing application of the standard.

In this stage, the Board may consider sponsoring academic research to assess certain aspects of the standard to provide information to the Board on the effectiveness of the standard.

Stage 3: Summary of Research and Reporting

Following the completion of Stages 1 and 2, the Board summarizes their research in a final report. The final report is a culmination of previous reports and describes the activities conducted and actions taken to address any identified issues. In addition, a memorandum summarizing the PIR findings is prepared and discussed by the Board at a public meeting.

PART – C
(Compulsary)

21. What is the role of preliminary investigation in application development? Explain various ways to perform Preliminary investigation.

- Introduction.
- System Development Life cycle (SDLC)
- Preliminary investigation(feasibility study)
- Way to perform Preliminary investigation
- Conclusion.

Introduction:

This case study is about System Development Life cycle and about role of preliminary investigation in application development and various ways to perform Preliminary investigation.

System Development Life Cycle

An effective System Development Life Cycle (SDLC) should result in a high quality system that meets customer expectations, reaches completion within time and cost evaluations, and works effectively and efficiently in the current and planned Information Technology infrastructure.

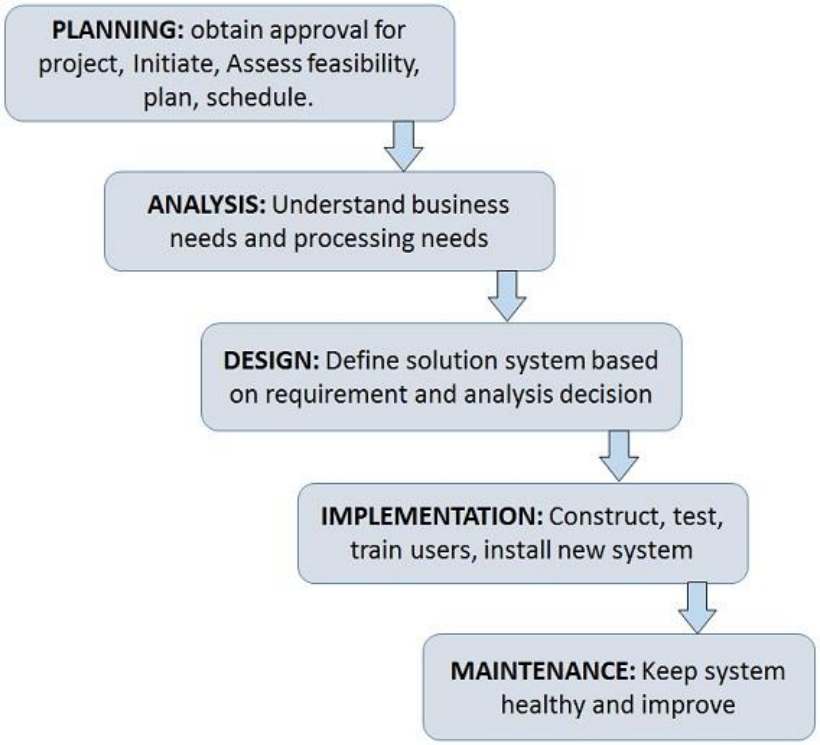
System Development Life Cycle (SDLC) is a conceptual model which includes policies and procedures for developing or altering systems throughout their life cycles.

SDLC is used by analysts to develop an information system. SDLC includes the following activities −

- requirements
- design
- implementation
- testing
- deployment
- operations
- maintenance

Phases of SDLC

Systems Development Life Cycle is a systematic approach which explicitly breaks down the work into phases that are required to implement either new or modified Information System.



Feasibility Study or Planning

- Define the problem and scope of existing system.
- Overview the new system and determine its objectives.
- Confirm project feasibility and produce the project Schedule.
- During this phase, threats, constraints, integration and security of system are also considered.
- A feasibility report for the entire project is created at the end of this phase.

Analysis and Specification

- Gather, analyze, and validate the information.
- Define the requirements and prototypes for new system.
- Evaluate the alternatives and prioritize the requirements.
- Examine the information needs of end-user and enhances the system goal.
- A Software Requirement Specification (SRS) document, which specifies the software, hardware, functional, and network requirements of the system is prepared at the end of this phase.

System Design

- Includes the design of application, network, databases, user interfaces, and system interfaces.
- Transform the SRS document into logical structure, which contains detailed and complete set of specifications that can be implemented in a programming language.
- Create a contingency, training, maintenance, and operation plan.
- Review the proposed design. Ensure that the final design must meet the requirements stated in SRS document.
- Finally, prepare a design document which will be used during next phases.

Implementation

- Implement the design into source code through coding.
- Combine all the modules together into training environment that detects errors and defects.
- A test report which contains errors is prepared through test plan that includes test related tasks such as test case generation, testing criteria, and resource allocation for testing.
- Integrate the information system into its environment and install the new system.

Maintenance/Support

- Include all the activities such as phone support or physical on-site support for users that is required once the system is installing.
- Implement the changes that software might undergo over a period of time, or implement any new requirements after the software is deployed at the customer location.
- It also includes handling the residual errors and resolve any issues that may exist in the system even after the testing phase.
- Maintenance and support may be needed for a longer time for large systems and for a short time for smaller systems.

Feasibility Study

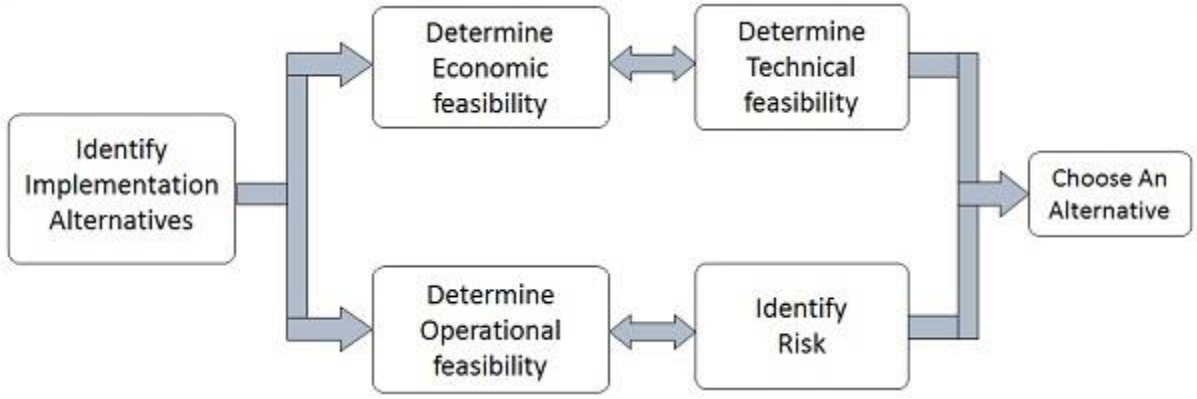
Feasibility Study can be considered as preliminary investigation that helps the management to take decision about whether study of system should be feasible for development or not.

- It identifies the possibility of improving an existing system, developing a new system, and produce refined estimates for further development of system.
- It is used to obtain the outline of the problem and decide whether feasible or appropriate solution exists or not.
- The main objective of a feasibility study is to acquire problem scope instead of solving the problem.
- The output of a feasibility study is a formal system proposal act as decision document which includes the complete nature and scope of the proposed system.

Steps Involved in Feasibility Analysis

The following steps are to be followed while performing feasibility analysis −

- Form a project team and appoint a project leader.
- Develop system flowcharts.
- Identify the deficiencies of current system and set goals.
- Enumerate the alternative solution or potential candidate system to meet goals.
- Determine the feasibility of each alternative such as technical feasibility, operational feasibility, etc.
- Weight the performance and cost effectiveness of each candidate system.
- Rank the other alternatives and select the best candidate system.
- Prepare a system proposal of final project directive to management for approval.



Types of Feasibilities

Economic Feasibility

- It is evaluating the effectiveness of candidate system by using cost/benefit analysis method.
- It demonstrates the net benefit from the candidate system in terms of benefits and costs to the organization.
- The main aim of Economic Feasibility Analysis (EFS) is to estimate the economic requirements of candidate system before investments funds are committed to proposal.
- It prefers the alternative which will maximize the net worth of organization by earliest and highest return of funds along with lowest level of risk involved in developing the candidate system.

Technical Feasibility

- It investigates the technical feasibility of each implementation alternative.

- It analyzes and determines whether the solution can be supported by existing technology or not.
- The analyst determines whether current technical resources be upgraded or added it that fulfill the new requirements.
- It ensures that the candidate system provides appropriate responses to what extent it can support the technical enhancement.

Operational Feasibility

- It determines whether the system is operating effectively once it is developed and implemented.
- It ensures that the management should support the proposed system and its working feasible in the current organizational environment.
- It analyzes whether the users will be affected and they accept the modified or new business methods that affect the possible system benefits.
- It also ensures that the computer resources and network architecture of candidate system are workable.

Behavioral Feasibility

- It evaluates and estimates the user attitude or behavior towards the development of new system.
- It helps in determining if the system requires special effort to educate, retrain, transfer, and changes in employee's job status on new ways of conducting business.

Schedule Feasibility

- It ensures that the project should be completed within given time constraint or schedule.
- It also verifies and validates whether the deadlines of project are reasonable or not.

Conclusion:

The above mention method and techniques are used to conduct a preliminary investigation.