Note Junction

Best Note Provider

Note By: Roshan Bist

UNIT-1

Foundations for Systems Development

⊛. Information System and its types:

System: A system is a collection of components (subsystems) that work together to realize some objective. For example, the library system contains different components to provide knowledge for its members. Every system has three activities or functions. These activities are input, processing and output. The system also includes two additional activities: feedback and control.

Information System (IS): [Imp]

A system that provides information to people in an organization is called information system (IS). Information systems capture and manage data to produce useful information that supports an organization and its employees. Data are the collection of raw facts representing events occuring in organizations. Information is the data shaped in a meaningful form.

The three activities to produce information in an information system are input, processing and output. Input collects raw data within organization or from its external environment. Processing converts those raw data into meaningful information. Output transfers this information to people.

Types of Information Systems: [Imp]

1) **Transaction Processing Systems (TPSs):** These are computerized systems that performs and records the daily routine transactions necessary to conduct business. Some examples include sales order entry, hotel reservation systems, employee record keeping etc. Transaction Processing Systems failure for a few hours can cause big loss. Online transaction processing systems (OLTPs) has direct connection between TPS programs and users, so it is often known as the live system where there is no time lag between data creation and its processing.

iii) Management Information Systems (MISs): These are the systems at the management level of an organization and serve management level functions like planning, controlling and decision-making. Typically these systems use internal data provided by the transaction processing systems. Salary analysis and sales reporting are the examples in which MIS can be used.

iii) Decision Support Systems (DSSs): These systems also serve at the management level of the organization. These systems use internal information from TPS and MIS, and often information from external sources. DSS have more analytical power than other systems. Contract cost analysis is an example in which DSS can be used.

iv) Executive Information Systems (EISs): These systems serve the strategic level of the organization. These systems incorporate data about external events such as new tax laws, but they also draw summarized information from internal MIS and DSS. 5-year operating plan is an example in which EIS can be used.

v) Expert Systems: An expert system is an extension of DSS that captures and reproduces the knowledge and expertise of an expert problem solver and then simulates "actions" of that expert. These systems imitate the logic and reasoning of the experts with their respective fields. Expert systems are implemented with artificial intelligence (AI) technology.

vi) Communication and Collaboration Systems: These systems enable more effective communications between workers, partners, customers, and suppliers to enhance their ability to collaborate. These systems use network technology that allows companies to coordinate with other organizations across great distances. These systems create new efficiencies and new relationships between the organization.

⊛System Analysis and Design:

System analysis and design is a complex, challenging and simulating organizational process that a team of business and systems professionals uses to develop and maintain computer-based information systems. It is an organizational improvement process. Information systems are built and rebuilt for organizational benefits.

An important result of system analysis and design is application software. It is designed to support organizational functions or process. In system analysis and design we use various methodologies, techniques and tools that have been developed, tested and widely used over the years.

Methodologies → Comprehensive, multi-step approaches to systems development.

Techniques → Processes that are followed to ensure that work is well thought-out, complete and comprehensible to others on the project team.

Tools → Computer programs to assist in application of techniques to the analysis and design process.

⊛. Modern Approach to System Analysis and Design:

The growth of computer-based information systems analysis and design methodologies started during the year 1950 to 1960. The researchers argued that software crisis was due to the lack of ~~disp~~ discipline of programmers and some believed that if formal engineering methodologies would be applied to software development, then production of software would become as predictable.

The new technologies and practices which were developed after 1970-1990 were primarily focused on solving the software issues like software crisis. The major elements used were software tools, formal methods, well defined processes that use the methodologies like OOP, CASE tools and Structured Programming approaches. The new century for web application development, wireless PDA's, component based applications were started.

⊛ Developing Information Systems and the Systems Development Life Cycle (SDLC): [Imp]

When developing information systems, most organizations use a standard set of steps called the systems development lifecycle (SDLC) as the common methodology for systems development. SDLC includes phases such as planning, analysis, design, implementation, and maintenance as shown in the figure below:

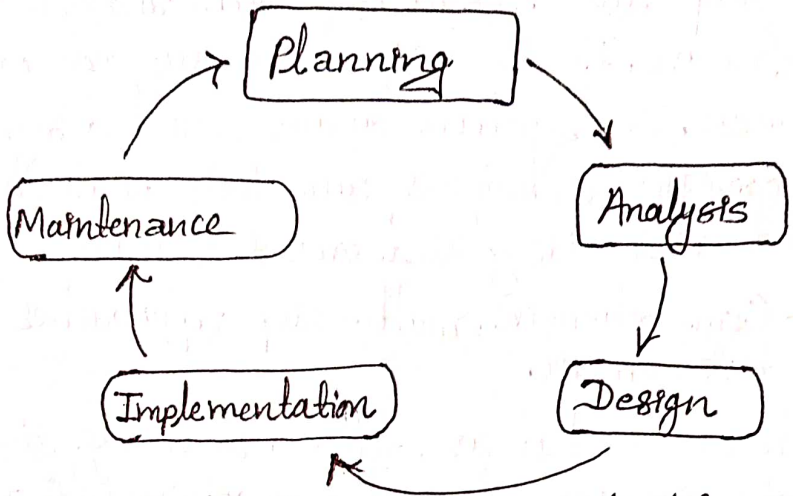


Fig: The systems development life cycle.

Steps/Phases of SDLC

- **Planning** → In this phase, someone identifies the need for a new or enhanced system. These needs are then analyzed, prioritized and arranged. A detailed plan is also developed for conducting the remaining phases.

- **Analysis** → The analysis phase usually requires a careful study of the current system, which continues two sub phases: requirements determination and analysis study. Requirements determination process usually involves a careful study of the current manual and computerized systems that may be replaced or improved within the project. Analysis study process usually involves analysts to study the structural requirements.

- **Design** → In this phase, the description of the recommended solution is converted into logical and then physical system specification. Logical design is independent of any specific hardware or software platform. In Physical design the logical specifications of the system are transformed into technology-specific details.

iv) Implementation → In this phase, the information system is coded, tested, installed, and supported in the organization. During coding programmers write the programs that make up the information system. During testing, programs and the entire system is tested to find and correct errors. During installation, the new system becomes a part of the daily activities of the organization.

v) Maintenance → The final phase of SDLC is called maintenance. In this phase, information system is systematically required and improved. When the system is operating in an organization, users sometimes find problems with how it works and often think of better ways to perform its functions.

Products, Outputs or Deliverables of SDLC Phases:

Phase	Products, Outputs, or Deliverables.
Planning	→ Detailed steps, or work plan for project. → Assignment of team members and other resources. → System justification or business case.
Analysis	→ Description of current system with general recommendation on how to fix, enhance, or replace current system. → Explanation of alternative systems and justification for choosen alternative.
Design	→ Detailed specifications of all system elements. both functional and technical. → Acquisition plan for new technology.
Implementation	→ Code, documentation, training procedures and support capabilities.
Maintenance	→ New versions or releases of software with associated updates to documentation, training and support.

⊛. The Heart of the System Development Process:

→ The heart of system development is analysis–design–implementation.
→ After collecting the system requirements, they are throughly analyzed by experts.
→ After analyzing them properly, the design for implementation is done by keeping a stress on meeting the requirements.
→ As a next step, the system design is implemented with the help of information from previous stages, so that the system meets the expected goals.

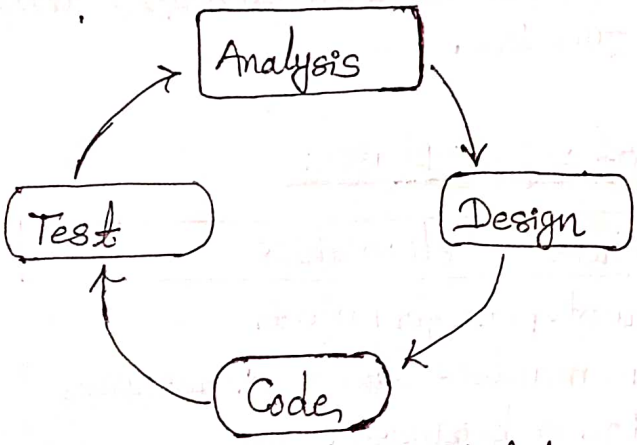


Fig: Analysis–design–code–test loop.

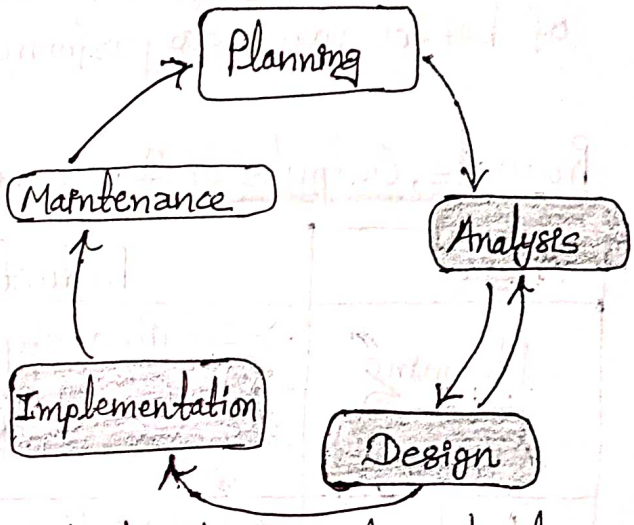
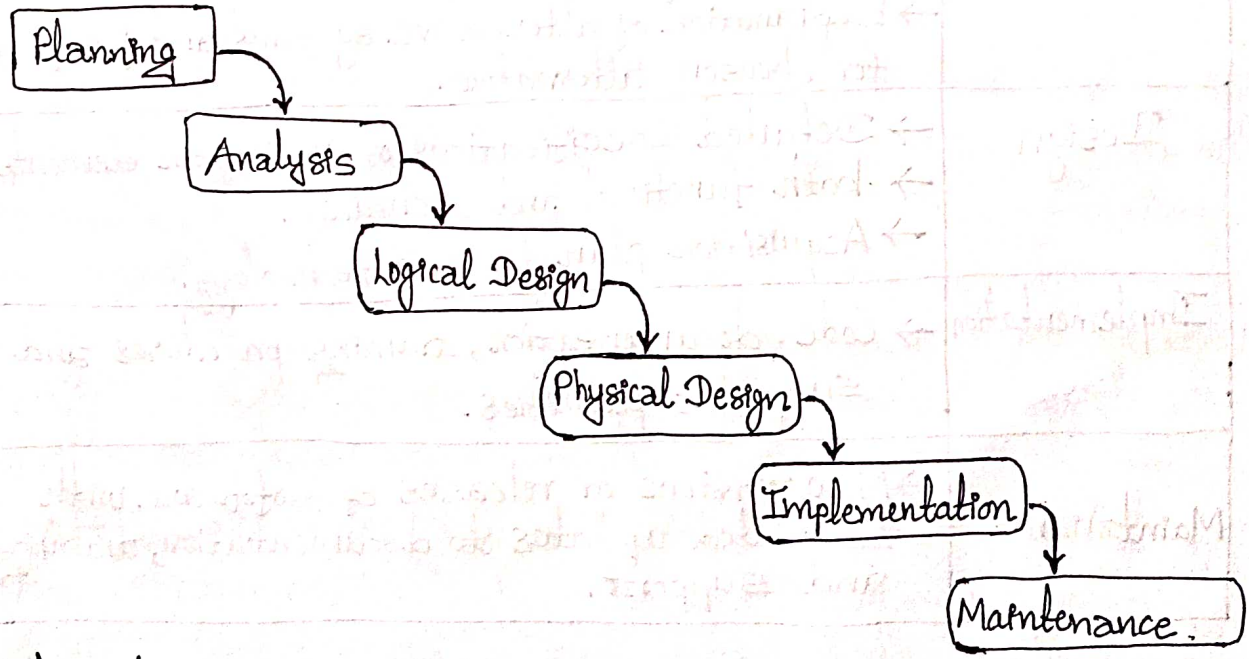


Fig: The heart of system development

⊛. Traditional Waterfall SDLC:



→ One phase begins when another completes, little backtracking and looping.
→ The waterfall model emphasizes a logical progression of steps.

→ The waterfall model is a linear, sequential approach to the system development life cycle (SDLC) that is popular in software engineering and product development.

Waterfall model can be used when:

↳ Requirements are not changing frequently.
↳ Application is not complicated and big.
↳ Project is short.
↳ Requirement is clear.
↳ Environment is stable.
↳ Resources are available and trained.

⊛ CASE Tools: [Impl]

Computer-aided software engineering (CASE) refers to automated software tools used by system analysts to develop information systems. These tools can be used to automate or support activities throughout the systems development process with the objective of increasing productivity and improving the overall quality of systems.

CASE tools are automated software packages that help to automate activities in the SDLC. CASE tools aim to enforce an engineering-type approach to the development of software systems. CASE tools range from simple diagramming tools to very sophisticated programs to document and automate most of the stages in the SDLC.

Types/Examples/Applications

i) Diagram tools → It helps in diagrammatic and graphical representations of the data and system processes. For example Flow Chart Maker tool for making state-of-the-art flowcharts.

ii) Computer Display and Report Generators → It helps in understanding the data requirements and the relationships involved.

iii) Analysis Tools → It focuses on inconsistent, incorrect specifications involved in the diagram and data flow. It helps in collecting requirements.

iv) Central Repository → It provides the single point of storage for data diagrams, reports and documents related to project management.

v) Documentation Generators → It helps in generating user and technical documentation as per standards. For example, Doxygen, DrExplain for documentation.

vi) Code Generators → It aids in the auto generation of code, including definitions, with the help of designs, documents and diagrams.

Components of CASE tools:

i) Upper case: Planning, analysis, and designing of different stages of the software development life cycle can be performed using upper case.

ii) Lower case: Implementating, testing, and maintenance can be performed using lower case.

iii) Integrated tools: All the stages of the software development life cycle right from the gathering of requirements for testing and documentation can be performed using integrated tools.

iv) Central Repository: It provides the single point of storage for data diagrams, reports and documents related to project management.

⑧. Other Approaches:-

1. Prototyping → Prototyping is a form of rapid application development (RAD). Instead of spending a lot of time producing very detailed specifications, the developers find out only generally what the users want. The developers do not develop the complete system all at once. Instead they quickly create a prototype, which either contains portions of the system or is a small-scale working model of the entire system. After reviewing the prototype with the users, the developers refine and extend it. This process is continued until the final specifications.

[Imp]

Phases:

i) Requirements Gathering → Requirements of the system are defined in detail. The user is interviewed in order to know requirements.

ii) **Quick Design:** When requirements are known, quick design for the system is created. It is not a detailed design and includes only the important aspects of the system.

iii) **Build Prototype:** Information gathered from quick design is modified to form the first prototype, which represents the working model of the required system.

iv) **User Evaluation:** The proposed system is presented to the user to recognize its strengths and weakness such as what is to be added or removed.

v) **Refining Prototype:** If the user is not satisfied with the prototype, current prototype is refined with the addition information provided by the user.

vi) **Engineer Product:** Once the requirements are completely met, the user accepts the final prototype. The final system is followed by the routine maintenance on regular basis for preventing large-scale failures and minimizing downtime.

Merits:

→ Flexibility in design.
→ New requirements can be easily accommodated.
→ Missing functionalities can be easily figured out.
→ Errors can be detected much earlier saving a lot of effort and cost.
→ The developed prototype can be reused by the developer for more complicated projects in future.

Demerits:

→ There are no parallel deliverables.
→ It is time consuming if customer asks for changes in prototype.
→ This methodology may increase the system complexity.
→ The invested effort in the prepration of prototypes may be too much if not properly monitored.
→ Customer may get confused in the prototypes and real systems.

2. Spiral Approach:

The spiral development model is a risk-driven process model generator that is used to guide multi-stakeholder concurrent engineering of software intensive systems. In this diagrammatic representation, it looks like a spiral with many loops. The exact number of loops of the spiral is unknown and can vary from project to project. Each loop of the spiral is called a phase of the software development process. Each phase is divided into four quadrants as shown in the figure.

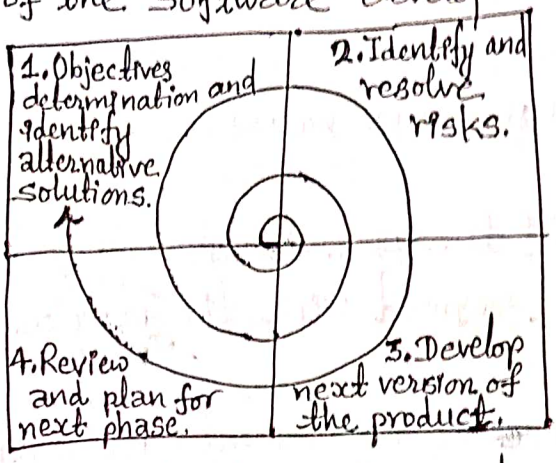


Fig: Different phases of spiral model.

Phases:

i) **Objectives determination and identify alternative solutions:**
Requirements are gathered from the customers and the objectives are identified and analyzed. Then alternative solutions possible for the phase are proposed in this quadrant.

ii) **Identify and resolve risks** → The risks associated with the solution are identified and the risks are resolved using the best possible strategy.

iii) **Develop next version of the Product** → During this third quadrant, the identified features are developed and verified through testing.

iv) **Review and plan for the next phase** → In this fourth quadrant, the customers evaluate developed version of the software.

Merits:

→ High risk analysis, hence avoidance of risk is enhanced.
→ Good for large projects.
→ Strong approval and documentation control.
→ Additional functional can be added at a later date.
→ Software is produced early in the software life cycle.

Demerits:

→ Can be a costly model to use.
→ Risk analysis requires highly specific expertise.
→ Projects sucess is highly dependent on the risk analysis phase.
→ Doesn't work well for smaller projects.

3. Rapid Applications Development Approach:

Rapid application development (RAD) is an object-oriented approach to systems development that includes a method of development as well as software tools. RAD is trying to meet rapidly changing business requirements more closely. Some developers are looking at RAD as a helpful approach in new ecommerce.

Phases:

i) **Requirements Planning Phase** → In this phase, users and analysts meet to identify objectives of the application or system. It may involve users from different levels of the organization. We may be working with the Chief Information Officer (CIO) as well as strategic planners.

ii) **RAD Design Workshop** → The RAD design workshop phase is a design-and-refine phase that can best be characterized as a workshop. During RAD design workshop, users respond to actual working prototypes and analysts refine designed modules based on user responses.

iii) **Implementation phase** → Analysts work with users to design the business or non-technical aspects of the system. As soon as these aspects are agreed on and the systems are built and refined.

Merits:

→ Changing requirements can be accomodated.
→ Progress can be measured.
→ Reduce development time.
→ Increases reusablity of components.
→ Productivity with fewer people in short time.

Demerits:

→ Dependency on technically strong team members for identifying business requirements.
→ Requires highly skilled developers.
→ High dependency on modeling skills.
→ Inapplicable to cheaper projects.

4. Agile Development Approach: It was mainly intended for helping developers build a project which can adopt to transforming requests quickly. So, it was developed to make easy and rapid project achievement. There are activities and behaviours that shape the way development team members and customers act during the development of an agile project.

Phases:

i) Exploration → During this we practice estimating the time needed for variety of tasks. This is all about the curious attitude toward the work environment, its problems, technologies, and people.

ii) Planning → In this phase we and customers agree on a date anywhere from two months to half a year from the current date to deliver solutions to their business problems.

iii) Iterations to the first release → In this phase we will sketch out the entire architecture of the system, even though it is just outline or skeletal form.

iv) Productionizing → The product is released in this phase, but may be improved by adding other features.

v) Maintenance → Once, the system has released, it needs to be kept running smoothly.

Merits:
→ Fewer errors
→ Reduces total development time of the whole project.
→ Easy to change or add requirement if needed.

Demerits:
→ Due to lack of formal documents, it creates different confusions.
→ Due to absence of proper documentation, maintenance of the developed project can become a problem.

⊛. Project Management and its Phases:- [Imp]

Project management is a controlled process of initating, planning, executing, monitoring, controlling and closing project. Project Management is an important aspect of the development of information systems. The focus of project management is to ensure that systems development projects meet customer expectations and are delivered within budget and time constraints.

Phases:

i) Initiating → During this phase, the project is conceptualized and feasibility is determined. Defining the project goal, defining the project scope, identifying the project manager, identifying potential risks etc. are the activities performed during this process.

ii) Planning → The project manager will create a blueprint to guide the entire project. This blueprint will map out the project's scope, resources required to create deliverables, estimated time, execution plan etc.

iii) Executing → The project manager will oversee the work on the project while maintaining good relationships with all team members and keeping the entire project on time and on budget.

iv) Monitoring and control → During this process project managers will closely measure the progress of the project to ensure it is developing properly. Monitoring and controlling is closely related to project planning. Monitoring will detect any necessary corrective action in the project to keep project on track.

v) Closing → During this phase, the project manager will close contracts with suppliers, external vendors, consultants and other third-party providers. All documentation will be archived and a final project report will be produced.

⊛ Managing the Information System Project:

Shaping a project: A successful project must be completed on time, within budget, and deliver a quality product that satisfies users and meets requirements. Project management techniques can be used throughout the SDLC. There is always a blance between constraints, and interactive elements such as project cost, scope and time.

Project Triangle: For each project, it must be decided, what is important, because the work cannot be good and fast and cheap. When it comes to project management, things are not quite so simple. The challenge is to find the optimal balance among the three factors cost, scope and time. Any chance in one of the factor will affect other two factors.

Project Manager: The project manager is a system analyst with a diverse set of skills — management, leadership, technical, conflict management, and customer relationship. The project manager is responsible for initiating, planning, executing, and closing down the project. Following are the common activities and skills of a project manager.

Activity	Description	Skill
Leadership	Influencing the activities of others through use of intelligence, personality and abilities	Communication Assigning activities Monitoring Progress.
Management	Getting projects completed using resources.	Defining and sequencing activities Monitoring Outcomes.
Technical problem solving	Designing and sequencing activities to achieve goals.	Defining and sequencing activities designing solutions to problems
Conflict Management	Assures that conflict is too high or low.	Problem solving Goal setting
Customer Relations	Interacting with customers to meet requirements.	Site prepration Contact point for customers.

⊛. Representing and Scheduling Project Plans:

1. Gantt Charts:

Gantt chart is a graphical representation of a project that shows each task activity as a horizontal bar which is propotional to its time for completion. Gantt Chart is the most commonly used method for graphical reports. Gantt charts are often useful for simple projects or subparts of a larger project. It is also useful for monitoring the progress of activities. Gantt Charts do not show how tasks must be ordered but simply show when it should end.

2. Network Diagrams:

Network diagram is a graphical representation of a project that shows how each activity relates to others in the project. The project manager can track each element of project and report progress to stakeholders with the help of network diagrams. Network diagram shows the ordering of activities by connecting a task to its predecessor and successor tasks.

Gantt Charts vs. Network diagrams:

	Gantt Charts	Network Diagrams.
Definition	Gantt chart is a graphical representation that shows each task activity with their corresponding start and finish dates.	Network Digram is a schematic display of logical relationships among project activities.
Tasks	Visual representation of duration of tasks.	Visual representation of dependencies between tasks.
Depicts	Time overlap between task.	Tasks can be done in parallel.
Slack time	Visually shows slack time.	Shows slack time by using rectangles filled by data.
Category	Bar chart	Flow chart

3. Representing Project Plans:-

Project scheduling and management requires time, costs and resources to be controlled. Resources are any person, group of people, or material used in accomplishing an activity. Network diagramming is a critical path scheduling technique used for controlling resources. A major strength of Network diagramming is its ability to represent how completion times vary for activities. Because of this, it is more often used than Gantt charts to manage projects such as information systems development where variability in the duration of activities is the norm.

4. Calculating Expected time durations using PERT:

PERT (program evaluation review technique) is a technique that uses optimistic, pessimistic, and realistic time estimates to calculate the expected time for a particular task. This technique helps us to obtain a better time estimate when we are uncertain as to how much time a task will require to be completed.

$$ET = \frac{O + 4r + p}{6}$$

where,
ET = expected time completion for an activity.
O = optimistic time completion for an activity.
r = realistic time completion for an activity.
P = pessimistic time completion for an activity.

5. Using Project Management Software:-

A wide variety of automated project management tools are available to help to manage a development project. New versions of these tools are continuously being developed and released by software vendors. Most of the available tools have a set of common features that include the ability to define and order tasks, assign resources to tasks, and easily modify tasks and resources. Project management tools are available to wide variety of systems. Most programs offer features such as PERT/CPM, Gantt charts, resource scheduling and cost tracking. Asana, Wrike, Trello, Jira, ClickUp etc are some top project management tools.

⊛. Why do we need system analysis and design?

Ans:- We need system analysis and design for following reasons:

→ It helps to design systems where subsystems may have conflicting objectives.

→ It helps in understanding of complex structures.

→ It gives an advantage of understanding and comparing the subsystems functions with complete system.

→ It helps to achieve inter compatibility and unity of the sub systems.

⊛ Why do we need information system? [Imp]

Ans:- We need information system for following reasons:

→ To capture and manage data to produce useful information that supports an organization and its employees.

→ To gain maximum benifits from company's information system.

→ To increase effectiveness and accuracy.

→ To make better decisions based on data present in the system.

→ For keeping records of a company for future use.

⊛. What is forward, reverse and round trip engineering? Explain.

Ans: Forward and reverse engineering are the two distinct ways to develop system models which are provided by today's CASE tools. Forward engineering requires the system analyst to draw system models, either from scratch or from templates. The resulting models are subsequently transformed into program code.

Reverse engineering on the other hand, allows a CASE tool to read existing program code and transform that code into a representative system model that can be edited and refined by the systems analyst.

CASE tools that allow for bi-directional forward and reverse engineering are said to provide for "round-trip engineering".

⊛. Define system analyst and explain skills needed to be a good system analyst.

Ans:- The system analyst is an agent of change and innovation having the primary responsibility of system analysis and design.

Skills of System Analyst:

→ Strong analytical skills.
→ Teamwork skills
→ Interpersonal skills.
→ Written and verbal communication skills.
→ Flexibility
→ Adaptability
→ Initiative
→ Ability to gather and analyze information.