Unit 1 **System Development Fundamentals** **9 Hrs.**

a. The Systems Development Environment

Introduction, Modern Approach of System Analysis and Design, Information System and its Type, Developing Information Systems and the Systems Development Life Cycle, The Heart of the Systems Development Process, The Traditional Waterfall SDLC, Approaches for Improving Development, CASE Tools, Rapid Application Development, Service-Oriented Architecture, Agile Methodologies, eXtreme Programming, Object- Oriented Analysis and Design

b. The Origins of Software

Introduction, System Acquisition, Reuse

c. Managing the Information Systems Project

Introduction, Managing Information Systems Project, Representing and Scheduling Project Plans, Using Project Management Software

Compiled By : Er. Omkar Nath Gupta

SYSTEM ANALYSIS AND DESIGN(SAD)

Chapter 1

System Development Fundamentals

Some Terminologies :

Data: Data are the raw facts about people, places, events, and things that are of importance in an organization.

Information: Information is the data that has been processed or reorganized into a more meaningful form for someone.

Knowledge: Knowledge is the data and information that is further refined based on the facts, truths, beliefs, judgments, experiences, and expertise of the recipient.

System: A system is a collection of components (subsystems) that work together to realize some objective. A system takes input, performs processing of data to give some desired outputs. Basically, there are three major components in every system. I.e. Input, Processing and Output. Example**:**

Computer systems: A set of devices that input, process, store, and output data.

IT systems: A system that may include computers and their related resources.

Business systems: A system made up of methods, procedures, and routines.

- Every system has three activities or functions. These activities are **input, processing and output**.

Input: It involves capturing and assembling elements that enter the system to be processed. Inputs to the system are anything to be captured by the system from its environment. For example, raw materials.

Processing: It involves transformation processes that convert input to output. For example, a manufacturing process.

Output: It involves transferring elements that have been produced by a transformation process to their ultimate destinations. Outputs are the things produced by the system and sent into its environment. For example, finished products.

- The system also includes other activities. These activities **include feedback, control, Environment Boundaries /Interface**

Feedback: It is data about the performance of a system. It is the idea of monitoring the current system output and comparing it to the system goal. Any variation from the goal are then fed back in to the system and used to adjust it to ensure that it meets its goal.

Control: It involves monitoring and evaluating feedback to determine whether a system is moving toward the achievement of its goals. The control function then makes necessary adjustments to a system's input and processing components to ensure that it produces proper output.

Environment : The environment is the "super system" within which an organization operates. It is the source of external elements that strike on the system.

Boundaries and Interface: A system should be defined by its boundaries. Boundaries are the limits that identify its components, processes, and interrelationship when it interfaces with another system.

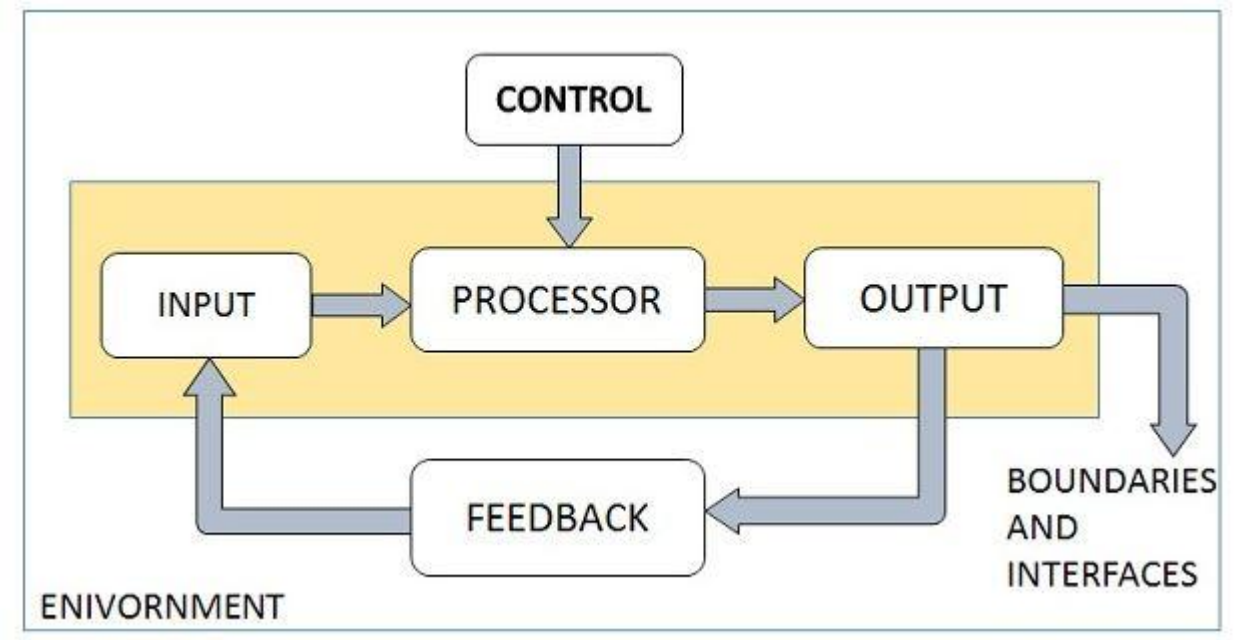


Fig: Basic Elements of system

Characteristics of a system

Structure: The way the parts of a system are organized

Function: What the system does

Behavior: How the system behaves

Interconnectivity: How the parts of a system interact with each other

Emergence: The idea that the system's behavior is a result of its parts interacting with each other and the environment

Goal: The purpose of the system.

System Analysis :

- System analysis is a process that studies a system to identify its goals and create systems to achieve them efficiently.
- It's a problem-solving technique that involves breaking down a system into its parts to understand how they work together.

What does system analysis do?

Identify goals: Determine what the system is trying to achieve

Analyze components: Understand how each part of the system works and interacts

Create solutions: Design systems and procedures to achieve the system's goals

Make decisions: Use the information gathered to help decision makers choose the best course of action

System design:

System design is the process of defining the structure and components of a system to meet specific goals. It's an interdisciplinary engineering activity that involves creating a blueprint for building a system.

What does system design involve?

Defining components: Identifying the individual pieces that make up the system, such as databases, user interfaces, and application servers

Defining interfaces: Determining how the components will interact with each other

Defining data flow: Determining how data will flow between the components

Defining architecture: Deciding on the overall structure of the system, such as whether to use a client-server, micro services, or monolithic architecture

Making trade-offs: Balancing the needs of different parts of the system to achieve the desired functionality

Information System

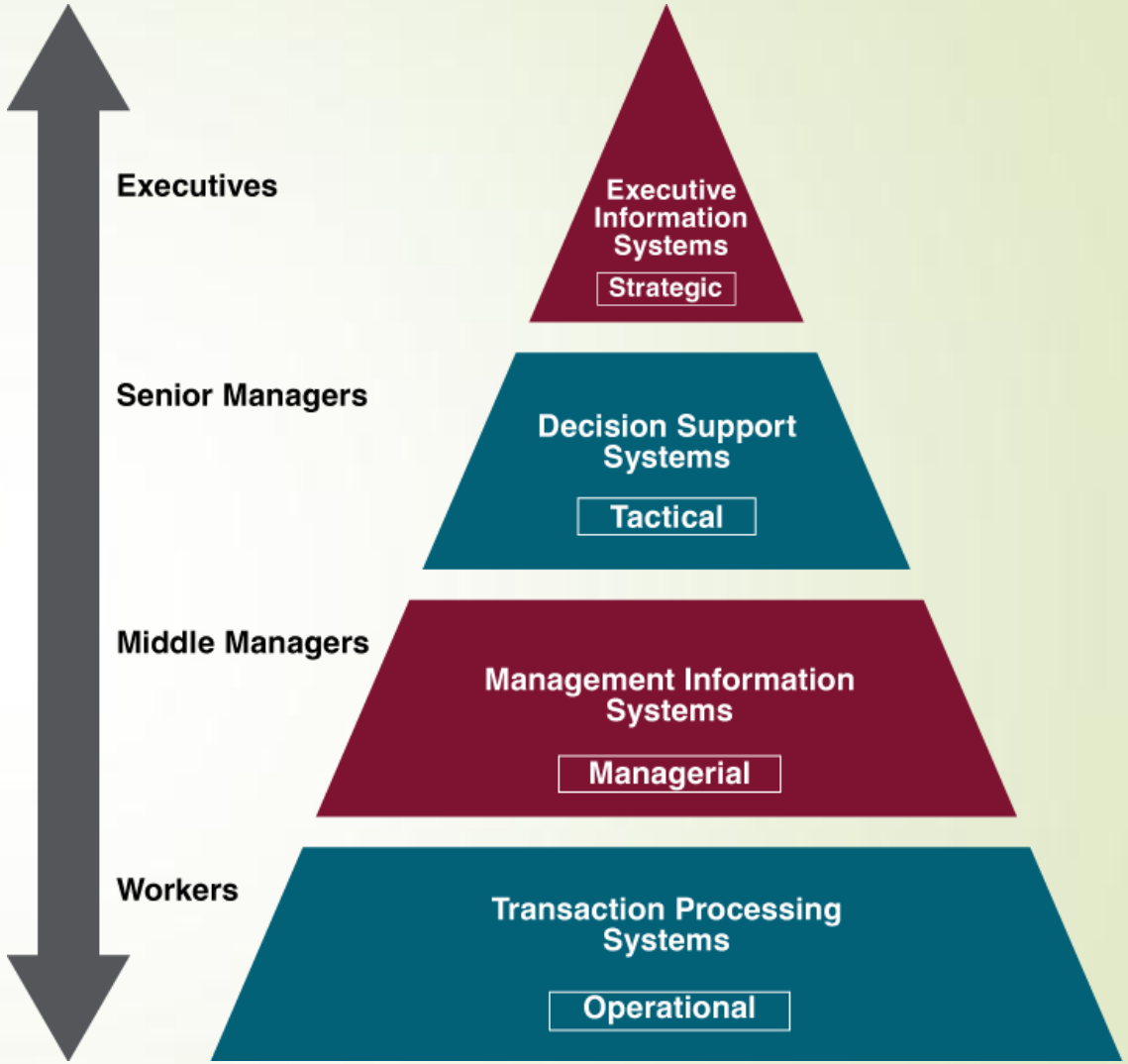
- A system that provides information to people in an organization is called information system (IS).
- An information system is an arrangement of people, data, processes, interfaces, networks, and technology that interact for the purpose of supporting and improving both day-to-day operations in a business (sometimes called data processing), as well as supporting the problem solving and decision making needs of management (sometimes called information services).
- Information systems in organizations capture and manage data to produce useful information that supports an organization and its employees, customers, suppliers and partners. So, many organizations consider information system to be the essential one.
- Information systems produce information by using data about significant people, places, and things from within the organization and/or from the external environment to make decisions, control operations, analyze problems, and create new products or services

The three activities to produce information in an information system are **input, processing, and output** .

- **Input** captures or collects raw data from within the organization or from its external environment for processing.
- **Processing** converts these raw data into the meaningful information.
- **Output** transfers this information to the people who will use it or to the activities for which it will be used.
- Information systems also require **feedback,** which is used to monitor the current information system output and compare it to the system goal.

Types of Information System(IS)

1. Office automation Systems
2. Transaction processing Systems
3. Management information Systems
4. Decision support Systems
5. Expert Systems
6. Executive Support Systems



Office Automation Systems (OAS)

- Office automation system is a collection of software and hardware products that increase productivity within the office setting.
- Office automation systems are among the newest and most rapidly expanding computer based information systems. They are being developed with the hopes and expectations that they will increase the efficiency and productivity of office workers, typists, secretaries, administrative assistants, staff professionals, managers and the like. Many organizations have taken the First step toward automating their offices.
- Often this step involves the use of word processing equipment to facilitate the typing, storing, revising and printing of textual materials. Another development is a computer based communications system such as electronic mail which allows people to communicate in an electronic mode through computer terminals.

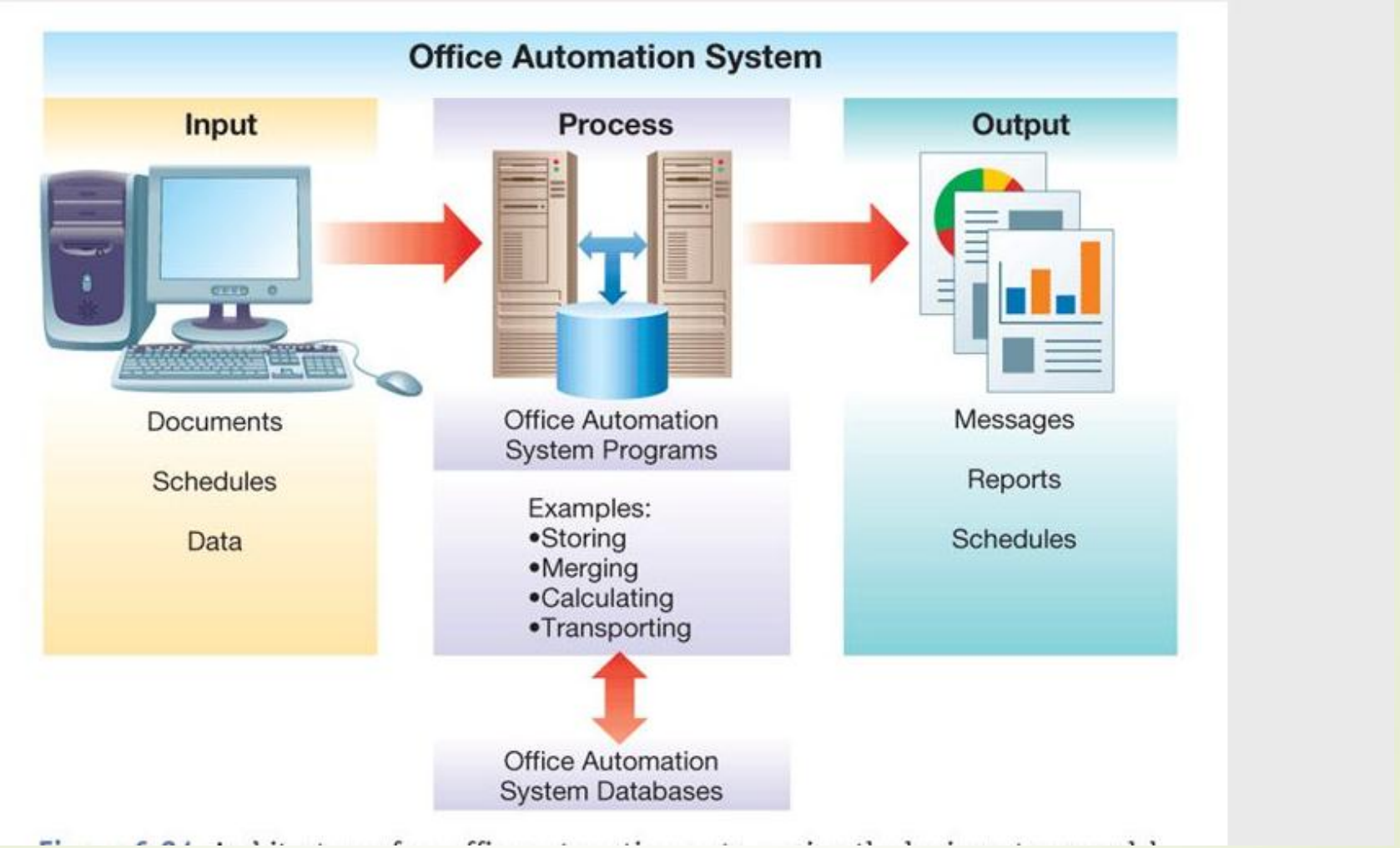


Fig: Office automation system

Transaction Processing System (TPS)

- Transaction Processing System are information system that processes data resulting from the occurrences of business transactions.
- Their objectives are to provide transaction in order to update records and generate reports i.e. to perform store keeping function.
- The TPS receives raw data from internal and external sources and prepares this data for storage in a database. In fact, all the company's key data is stored in a single huge database that becomes the company's central information resource. The database management system tracks the data and allows users to query the database for the information they need. The database can be updated in two ways:
- **Batch processing** is where data is collected over some time period and processed together. Batch processing uses computer resources very efficiently and is well-suited to applications such as payroll processing that require periodic rather than continuous processing.
- **Online processing** keeps the company's data current. When you make an airline reservation, the information is entered into the airline's information system, and you quickly receive confirmation, typically through an e-mail. Online processing is more expensive than batch processing, so companies must weigh the cost versus the benefit. For example, a factory that operates around the clock may use real-time processing for inventory and other time-sensitive requirements but process accounting data in batches overnight.

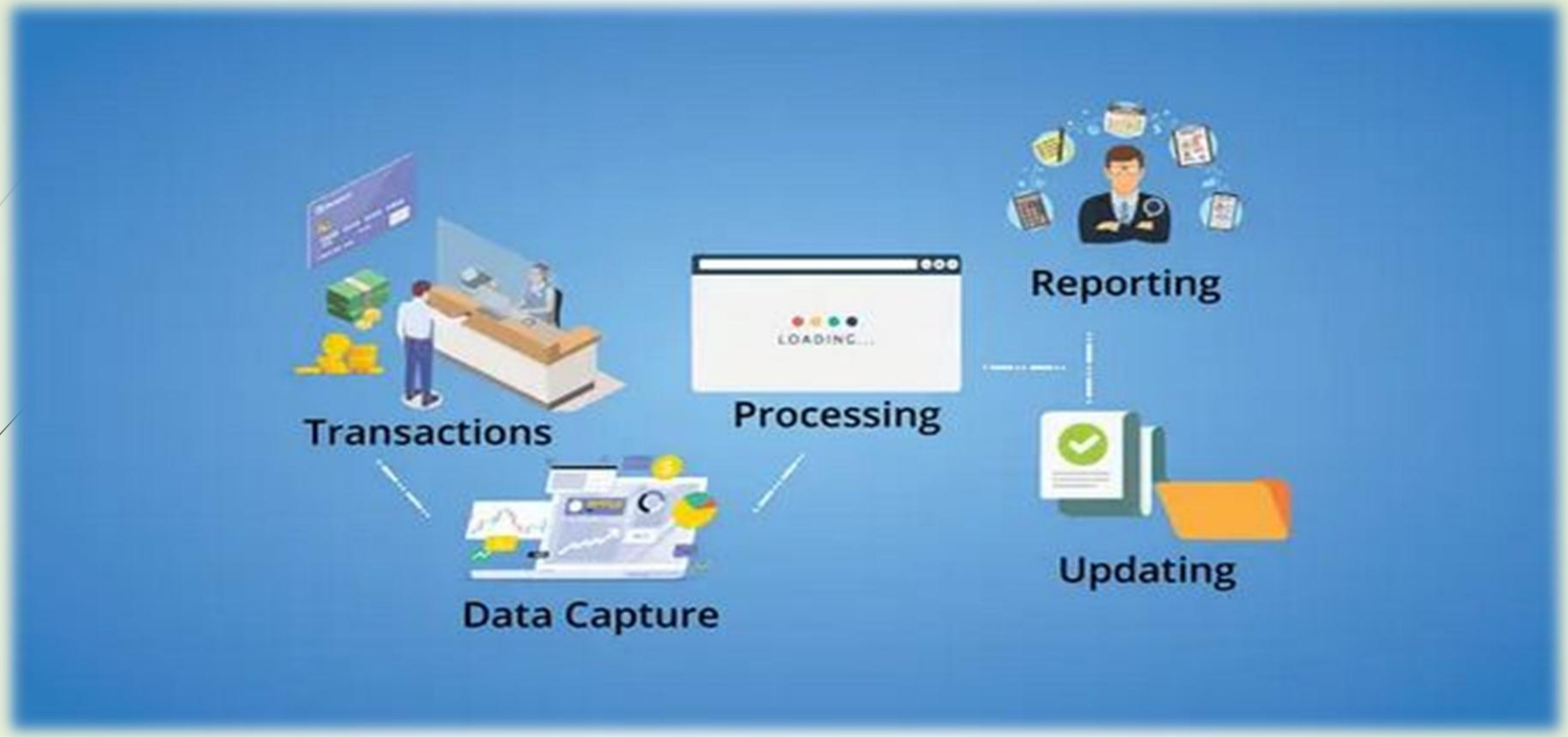


Fig: Transaction Processing System

Management Information System (MIS)

- Management Information System is designed to take relatively raw data available through a Transaction Processing System and convert them into a summarized and aggregated form for the manager, usually in a report format. It reports tending to be used by middle management and operational supervisors.
- Management Information Systems (MIS) are used by tactical managers to monitor the organization's current performance status. The output from a transaction processing system is used as input to a management information system.
- Many different types of report are produced in MIS. Some of the reports are a summary report, on-demand report, ad-hoc reports, etc.
- The MIS system analyzes the input with routine algorithms i.e. aggregate, compare and summarizes the results to produced reports that tactical managers use to monitor, control and predict future performance.
- For example, input from a point of sale system can be used to analyze trends of products that are performing well and those that are not performing well. This information can be used to make future inventory orders i.e. increasing orders for well-performing products and reduce the orders of products that are not performing well.

- MIS systems provide the information needed to make the structured decision and based on the experience of the tactical managers, they make judgment calls i.e. predict how much of goods or inventory should be ordered for the second quarter based on the sales of the first quarter
- Examples of management information systems include;

Sales management systems - they get input from the point of sale system

Budgeting systems - gives an overview of how much money is spent within the organization for the short and long terms.

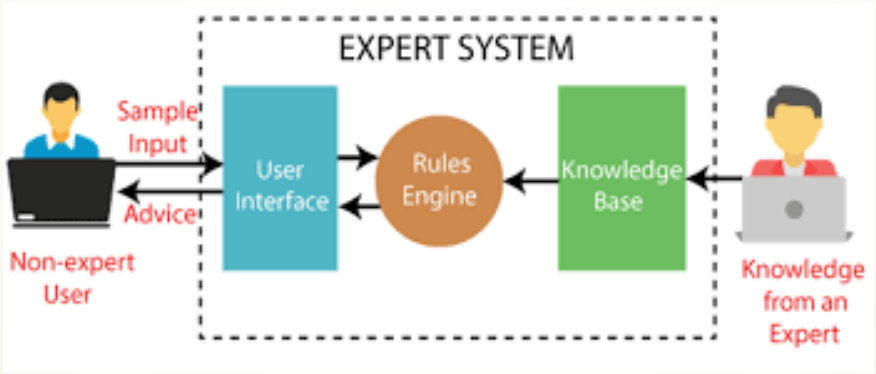
Human resource management system - overall welfare of the employees, staff turnover, etc.

Decision Support Systems (DSS)

- DSS are special-purpose information systems designed to support managerial-level employees in organizational decision making.
- DSS is an organized collection of people, procedures, databases, and devices used to support problem-specific decision making.
- These systems assist higher management to make long term decisions. These type of systems handle unstructured or semi structured decisions.
- A decision is considered unstructured if there are no clear procedures for making the decision and if not all the factors to be considered in the decision can be readily identified in advance.
- DSS is tool that aids in the process of decision-making but it cannot take decisions. A manager in addition to the information gained by DSS relies on his experience and intuition.
- **Example:** *Bank loan management systems – it is used to verify the credit of the loan applicant and predict the likelihood of the loan being recovered.*

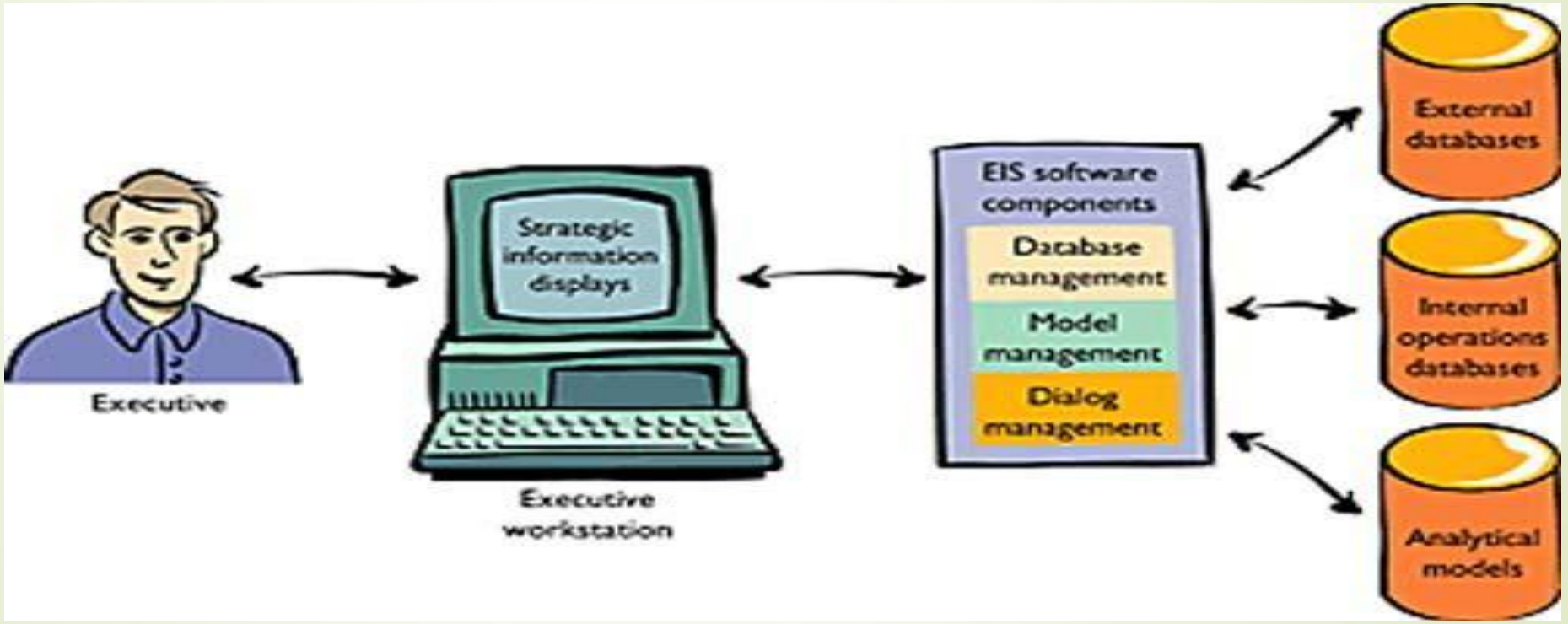
Experts System

- Experts systems include expertise in order to aid managers in diagnosing problems or in problem-solving. These systems are based on the principles of artificial intelligence research.
- Experts Systems is a knowledge-based information system. It uses its knowledge about a specify area to act as an expert consultant to users.
- Knowledge base and software modules are the components of an expert system. These modules perform inference on the knowledge and offer answers to a user's question.



Executive Management System(EIS)

- Senior managers of an organization use the EIS. Therefore, it must be easy to use so that executives can use it without any assistance.
- EIS can do trend analysis, exception reporting and have drill-down capabilities.
- The results are usually presented in a graphical form tailored to the executive's information needs.



System analysis and design (SAD)

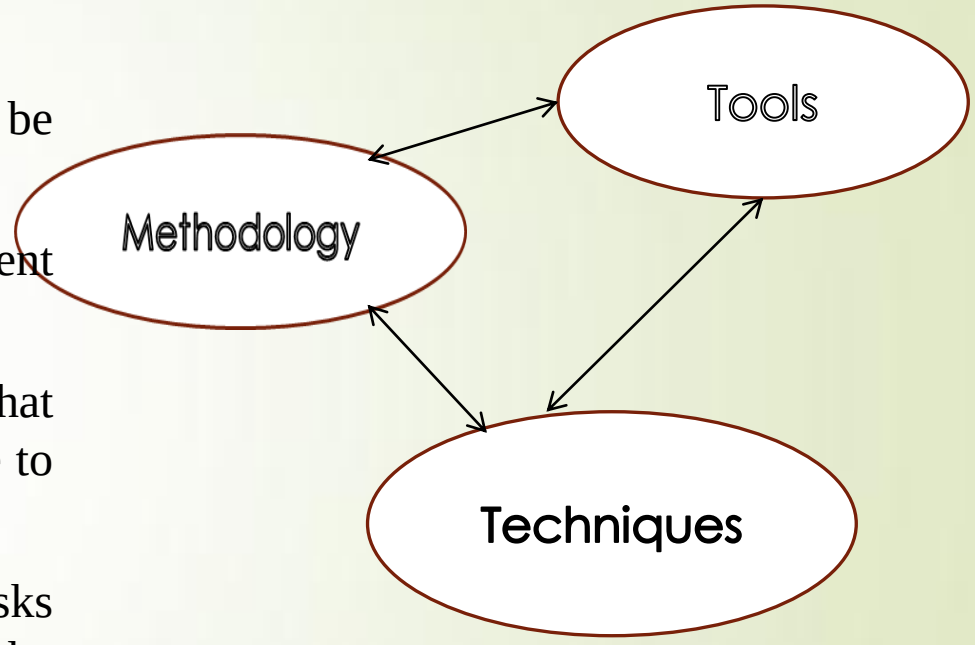
- System analysis and design is a complex, challenging, and simulating organizational process that a team of business and systems professionals uses to develop and maintain computer-based information systems.
- It is an organizational improvement process. Information systems are built and rebuilt for organizational benefits.
- An important (but not the only) result of system analysis and design is application software i.e. software designed to support organizational functions or processes such as inventory management, payroll, or mark-sheet analysis.
- In systems analysis and design, various methodologies, techniques are used and tools that have been developed, tested, and widely used over the years to assist people during system analysis and design.

Methodologies are comprehensive, multistep approaches to systems development that will guide your work and influence the quality of your final product: the information system.

- Methodologies use a standard set of steps.
- A methodology adopted by an organization will be consistent with its general management style.
- Most methodologies incorporate several development techniques.

Techniques are particular processes that will help to ensure that your work is well thought-out, complete, and comprehensible to other on the project team.

- Techniques also provide support for a wide range of tasks like conducting interviews, planning and managing the activities in a system development project, diagramming the system's logic, and designing the reports that the system will generate.



- **Tools** are typically computer programs that make it easy to use and benefit from the techniques and to faithfully follow the guidelines of the overall development methodology.
- To be effective, both techniques and tools must be consistent with an organizations system development methodology. These make easy for system developers to conduct the steps in methodology.

Importance of Systems Analysis and Design

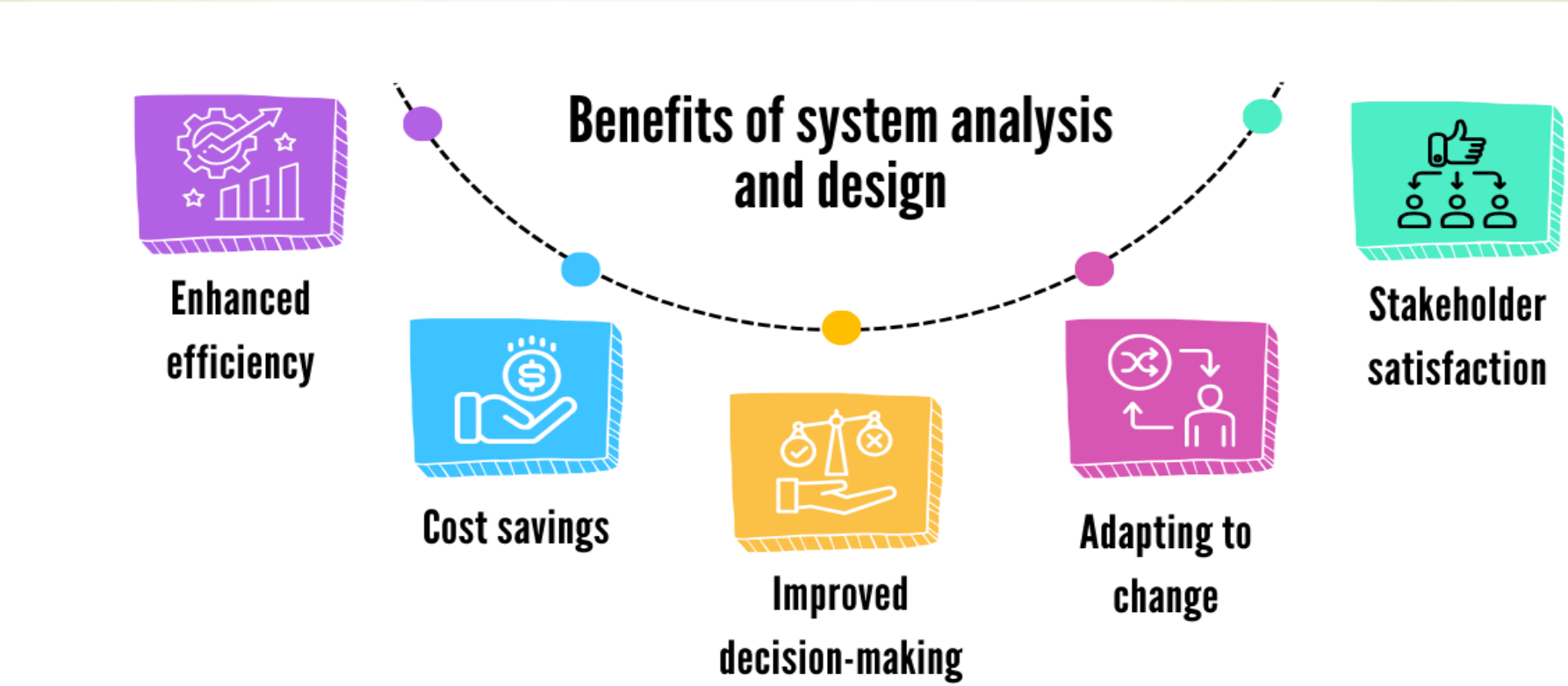


Fig:Importnace of System analysis & Design

Importance of Systems Analysis and Design

- **Enhanced efficiency:** Identifying inefficiencies and streamlining processes within a system to optimize resource utilization and reduce waste.
- **Improved communication:** Fostering better understanding of system requirements by facilitating dialogue between stakeholders and developers.
- **Cost reduction:** Proactive identification of potential problems early on to prevent costly rework and system failures.
- **Data-driven decision making:** Providing insights through data analysis to make informed decisions regarding system improvements.
- **Risk mitigation:** Recognizing potential risks and vulnerabilities within a system to proactively address them.
- **User-centric design:** Ensuring the new system aligns with user needs and expectations for better usability.
- **Adaptability to change:** Creating flexible systems that can easily adapt to future business requirements.
- **Quality improvement:** Delivering a high-quality system by addressing issues systematically throughout the development process.

System owners

- System owners are the information system's sponsors and chief advocates.
- They are usually responsible for funding the project of development, operate, and maintain the information system.
- They are interested with-how much will the system cost? And how much value or what benefit will the system return to the business?
- Every information system has one or more system owners. They usually come from the ranks of managers to supervisors.

System Users

- These are the people who use or are affected by the information system on a regular basis.
- They are concerned with the system's functionality related with their jobs and the system's ease of learning and use.
- A system user may capture, validate, enter, respond, store and exchange data and information.
- System users are also called clients. To know business requirements, discussions with most users need to be kept.

System Designers

- These are technology specialists who translate system users' business requirements and constraints into technical solutions.
- These are interested in information technology choices and the design of systems within the constraints of the chosen technology.
- They design the computer database, inputs, outputs, screens, networks, and programs that will meet the system users' requirements. These designs guide the construction of the final system.

System Builders

- These are also technology specialists who construct information systems and components based on the design specifications generated by the system designer.

Systems Analysts

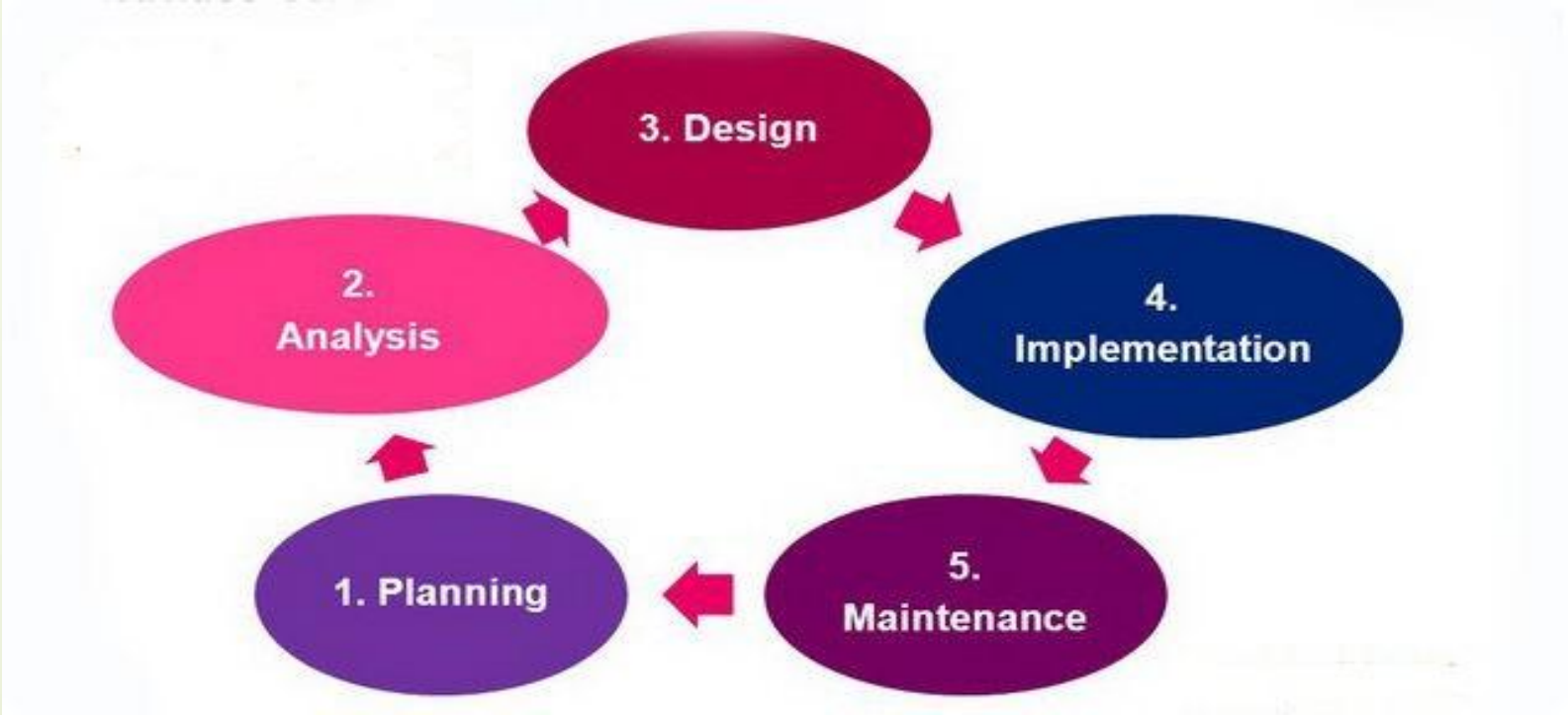
- Although, many people in organizations are responsible for systems analysis and design, in most organizations the systems analyst has the primary responsibility.
- The primary role of a systems analyst is to study the problems and needs of an organization in order to determine how people, methods and information technology can best be combined to bring about improvements in the organization.
- System analysts identify and validate problems and needs and ensure that the technical solution fulfills these problems and needs.
- Systems analysts study the system and identify and validate its problems and needs for system owners and users and ensure that the technical solution fulfills the business needs.
- The systems analyst performs systems analysis and design.

Project Managers

- To build a good information system and applications all the stakeholders must work together as a team. Teams require leadership.
- For this reason, usually one or more of these stakeholders takes on the role of project manager to ensure that systems are developed on time, within budget and acceptable quality.
- So, project manager is responsible for planning, monitoring, and controlling projects with respect to schedule, budget, deliverables, customer satisfaction, technical standards and system quality.

System Development Life Cycle (SDLC)

- The **System Development Life Cycle (SDLC)** provides a well-structured framework that gives an idea, of how to build a system. It consists of steps as follows – Plan, Analyze, Design, Implement and Maintain.



Planning

- The Planning phase sets the foundation for the entire SDLC. This stage involves identifying the system's objectives, defining the scope, setting timelines, and allocating necessary resources. Effective planning ensures that the development process aligns with the organization's goals, guiding the project in a clear and structured direction.

Analysis

- In the Analysis phase, the focus is on understanding and documenting the system's requirements. This involves gathering input from stakeholders, reviewing current processes, and identifying the system's needs. The data collected forms the basis for developing a system that addresses both user expectations and organizational challenges.

Design

- The Design phase translates the requirements gathered during Analysis into a detailed technical blueprint. This includes designing the system's architecture, database models, user interfaces, and defining system components. The outcome of this phase provides the technical structure needed to guide the upcoming development and implementation activities.

Implementation

- The Implementation phase involves deploying the developed system into a live environment. Key activities include system installation, migrating data, training users, and configuring infrastructure. This phase requires thorough planning to ensure a smooth transition from the existing system to the new one with minimal disruptions.

Maintenance

- Maintenance is an ongoing phase where the system is monitored, maintained, and updated as needed. This includes bug fixes, performance enhancements, security patches, and responding to user feedback. Proper maintenance ensures the system remains efficient, secure, and adaptable to future business needs.

The heart of system development

- The heart of system development is analysis-design-Implementation.
- After collecting the system requirements, they are thoroughly 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.

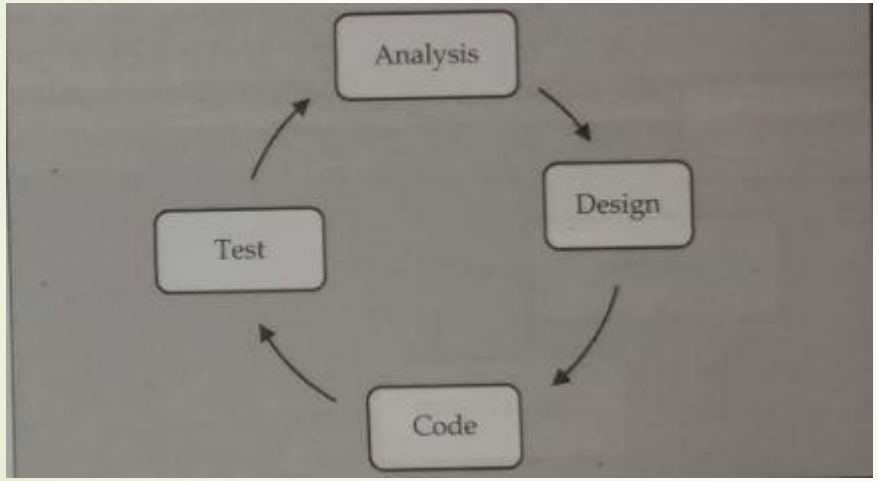


Fig: the analysis -design -code -test loop

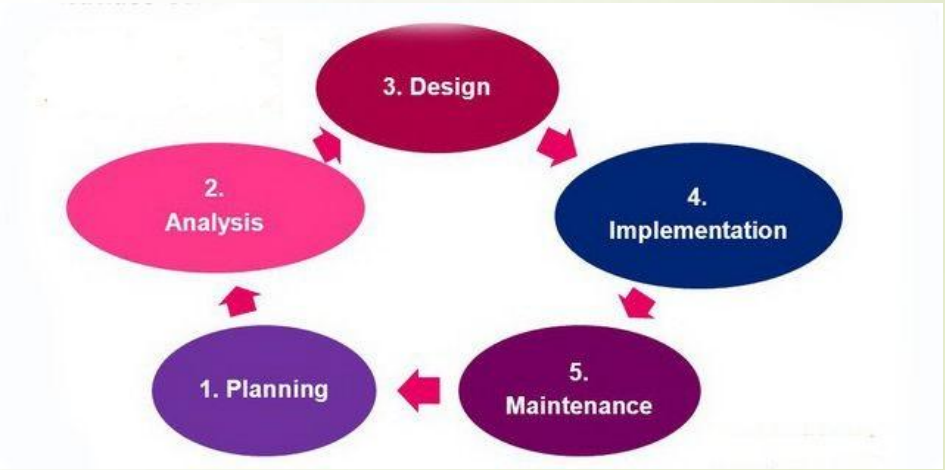
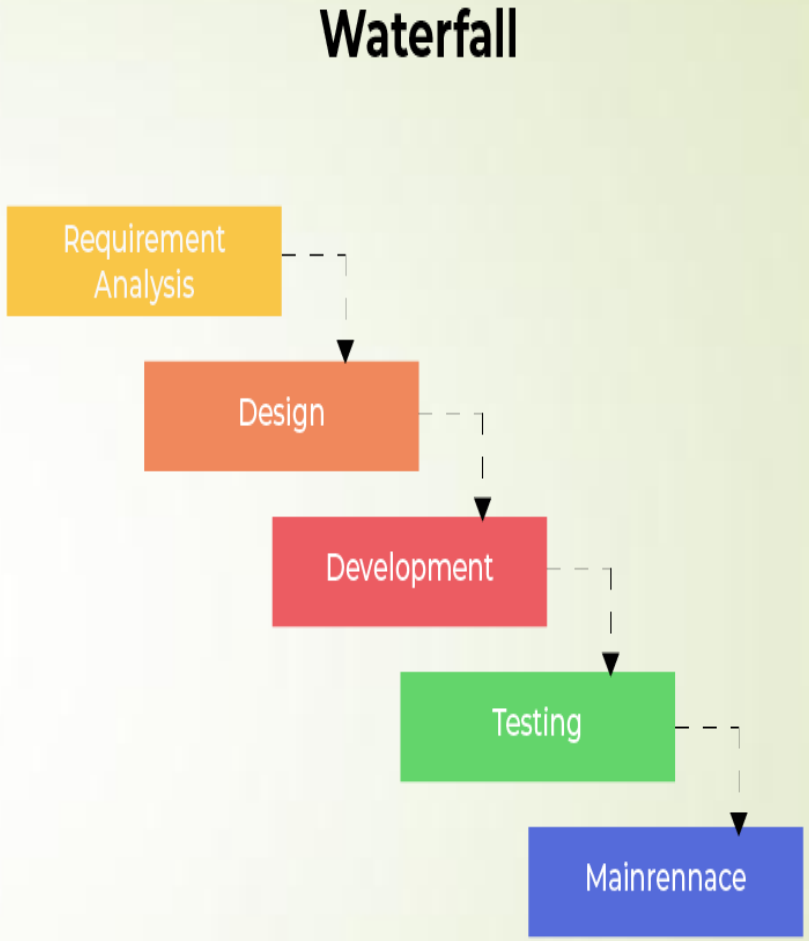


Fig:The heart of system development

Traditional Waterfall Model

- The Waterfall Model is a classical software development methodology.
- It was first introduced by Winston W. Royce in 1970.
- It is a linear and sequential approach to software development that consists of several phases.
- It must be completed in a specific order.
- This classical waterfall model is simple and idealistic.
- It was once very popular. Today, it is not that popularly used. However, it is important because most other types of software development life cycle models are a derivative of this.



- **Assignment:** Advantages and disadvantages of waterfall model.

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:

- 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 overall quality of systems .
- CASE tools are automated software packages that help to automate activities in the SDLC. CASE tools am 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:

- **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 .
- **Computer Display and Report Generators:** It helps in understanding the data requirements and the relationships involved.
- **Analysis Tools :** It focuses on inconsistent, incorrect specifications involved in the diagram and data flow. It helps in collecting requirements.
- **Central Repository** : It provides the provides the single point of storage for data diagrams, reports and documents related to project management.
- **Documentation Generators** :It helps in generating user and technical documentation as per standards. For example, Doxygen, DrExplan for documentation.
- **Code Generators :** It aids in the auto generation of code, including definitions, with the help of designs, documents and diagrams .

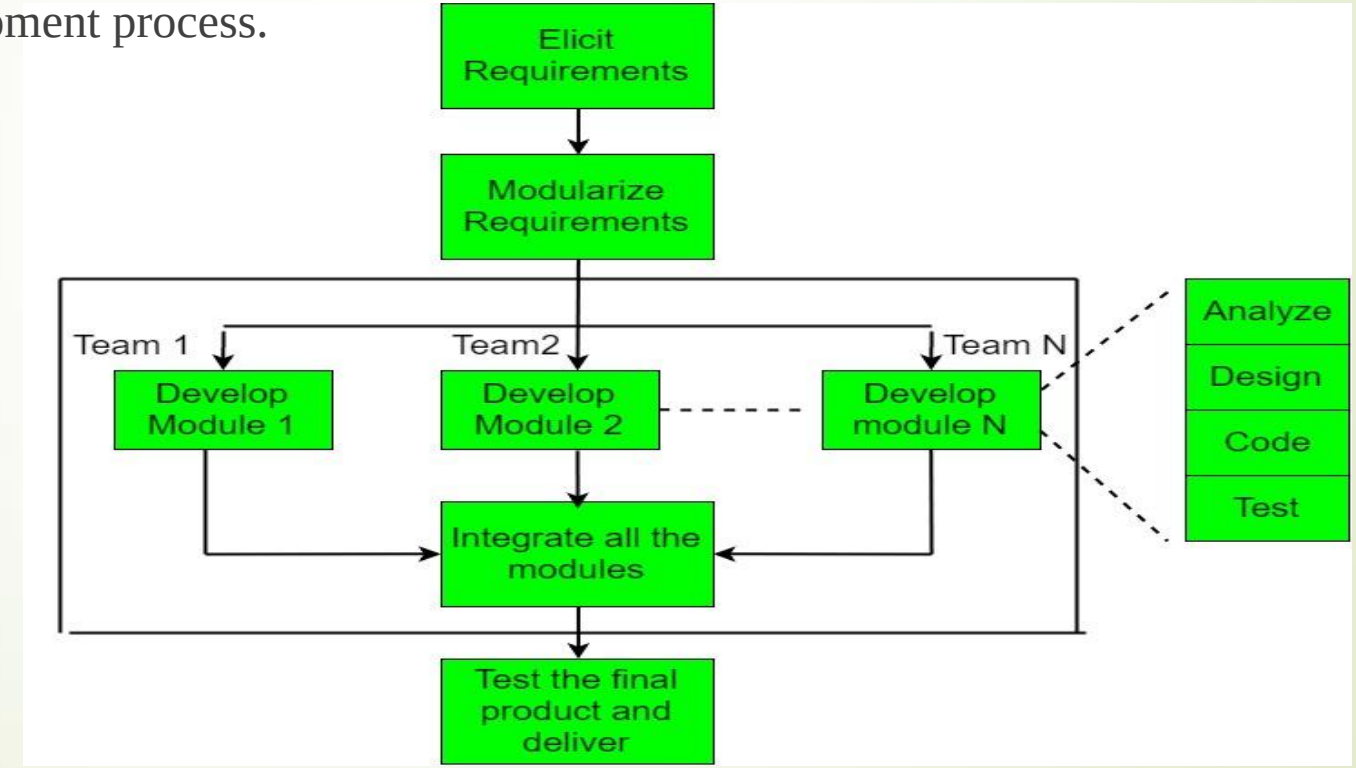
Components of CASE tools:

- **Upper case:** Planning, analysts, and designing of different stages of the software development life cycle can be performed using upper case.
- **Lower case:** Implementation, testing, and maintenance can be performed using lower case.
- **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,
- **Central Repository: It** provides the single point of storage for data diagrams, reports and documents related to project management.

Other approaches:

Rapid Application Development Model (RAD)

- The RAD model or Rapid Application Development model is a type of software development methodology that emphasizes quick and iterative release cycles, primarily focusing on delivering working software in shorter timelines. Unlike traditional models such as the Waterfall model, RAD is designed to be more flexible and responsive to user feedback and changing requirements throughout the development process.



- The critical feature of this model is the use of powerful development tools and techniques. A software project can be implemented using this model if the project can be broken down into small modules wherein each module can be assigned independently to separate teams.
- The use of powerful developer tools such as JAVA, C++, Visual BASIC, XML, etc. is also an integral part of the projects. This model consists of 4 basic phases:
  - **Requirements Planning** – This involves the use of various techniques used in requirements elicitation like brainstorming, task analysis, form analysis, user scenarios, FAST (Facilitated Application Development Technique), etc. It also consists of the entire structured plan describing the critical data, methods to obtain it, and then processing it to form a final refined model.
  - **User Description** – This phase consists of taking user feedback and building the prototype using developer tools. In other words, it includes re-examination and validation of the data collected in the first phase. The dataset attributes are also identified and elucidated in this phase.
  - **Construction** – In this phase, refinement of the prototype and delivery takes place. It includes the actual use of powerful automated tools to transform processes and data models into the final working product. All the required modifications and enhancements are to be done in this phase.
  - **Cutover** – All the interfaces between the independent modules developed by separate teams have to be tested properly. The use of powerfully automated tools and subparts makes testing easier. This is followed by acceptance testing by the user.

Advantages of Rapid Application Development Model (RAD)

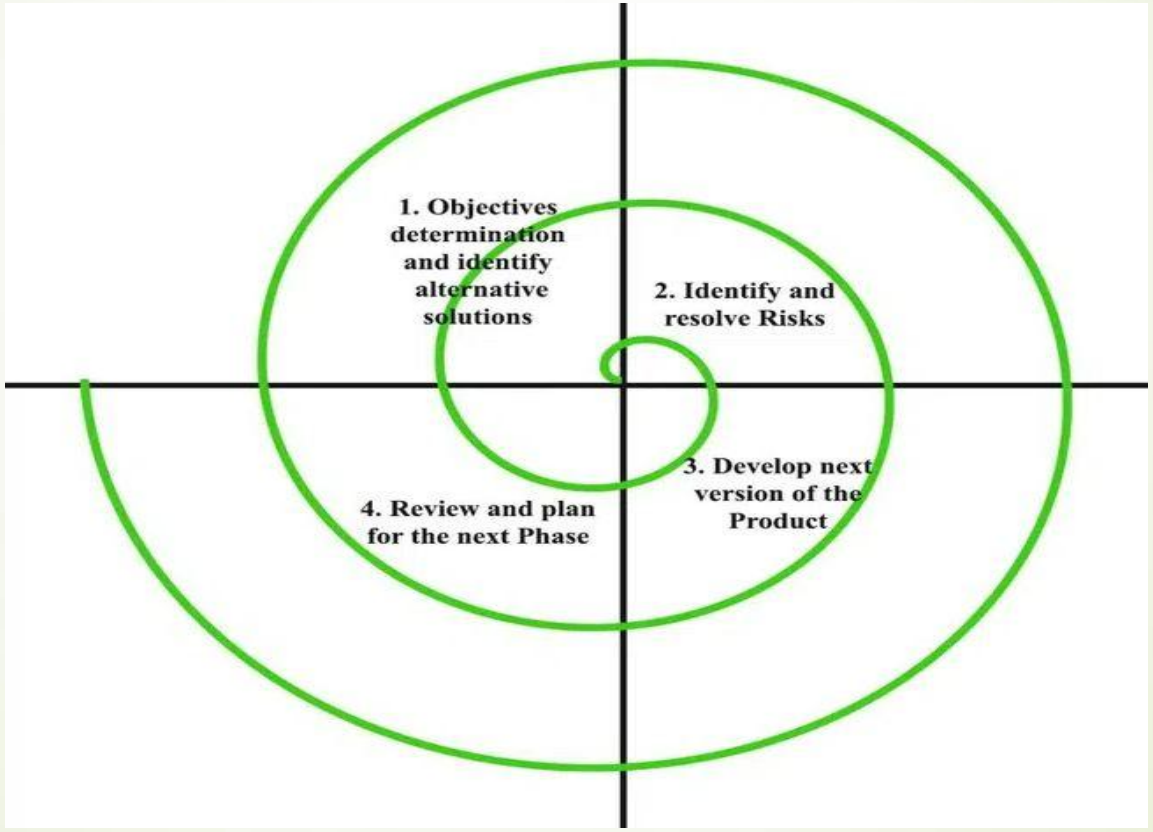
- The use of reusable components helps to reduce the cycle time of the project.
- Feedback from the customer is available at the initial stages.
- Reduced costs as fewer developers are required.
- The use of powerful development tools results in better quality products in comparatively shorter periods.
- The progress and development of the project can be measured through the various stages.
- It is easier to accommodate changing requirements due to the short iteration time spans.
- Productivity may be quickly boosted with a lower number of employees.

Disadvantages of Rapid application development model (RAD)

- The use of powerful and efficient tools requires highly skilled professionals.
- The absence of reusable components can lead to the failure of the project.
- The team leader must work closely with the developers and customers to close the project on time.
- The systems which cannot be modularized suitably cannot use this model.
- Customer involvement is required throughout the life cycle.
- It is not meant for small-scale projects as in such cases, the cost of using automated tools and techniques may exceed the entire budget of the project.
- Not every application can be used with RAD.

Spiral Model

- The Spiral Model is a **Software Development Life Cycle (SDLC)** model that provides a systematic and iterative approach to software development. In its diagrammatic representation, 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.



- The exact number of phases needed to develop the product can be varied by the project manager depending upon the project risks.
- As the project manager dynamically determines the number of phases, the project manager has an important role in developing a product using the spiral model.
- It is based on the idea of a spiral, with each iteration of the spiral representing a complete software development cycle, from requirements gathering and analysis to design, implementation, testing, and maintenance.

Phases of the Spiral Model

- The Spiral Model is a risk-driven model, meaning that the focus is on managing risk through multiple iterations of the software development process. It consists of the following phases:
- **Objectives Defined: I**n first phase of the spiral model we clarify what the project aims to achieve, including functional and non-functional requirements.
- **Risk Analysis:** In the risk analysis phase, the risks associated with the project are identified and evaluated.
- **Engineering:** In the engineering phase, the software is developed based on the requirements gathered in the previous iteration.
- **Evaluation:** In the evaluation phase, the software is evaluated to determine if it meets the customer's requirements and if it is of high quality.
- **Planning:** The next iteration of the spiral begins with a new planning phase, based on the results of the evaluation.

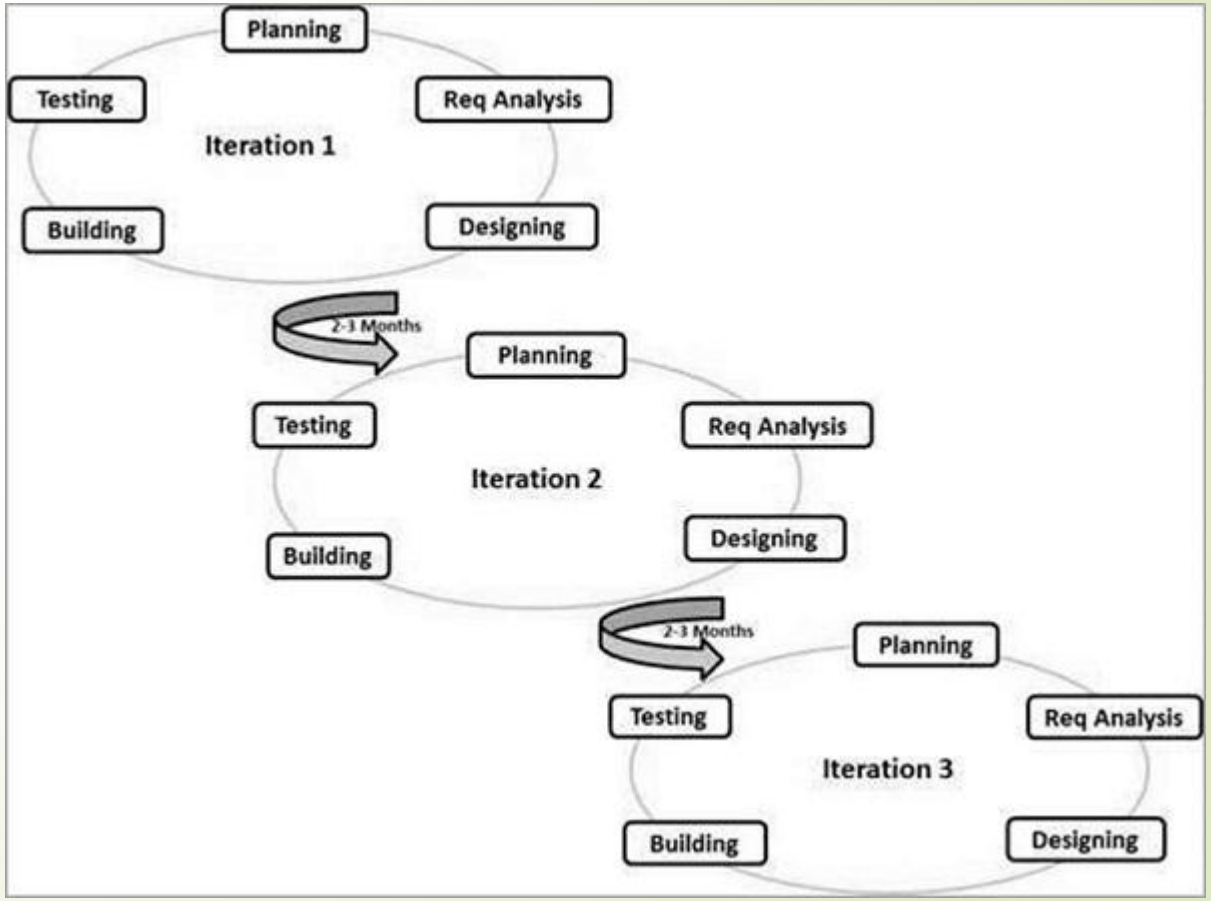
The Spiral Model is often used for complex and large software development projects, as it allows for a more flexible and adaptable approach to software development. It is also well-suited to projects with significant uncertainty or high levels of risk.

Agile Model

- **The Agile Model** was primarily designed to help a project adapt quickly to change requests. So, the main aim of the Agile model is to facilitate quick project completion. To accomplish this task, agility is required. Agility is achieved by fitting the process to the project and removing activities that may not be essential for a specific project. Also, anything that is a waste of time and effort is avoided. The Agile Model refers to a group of development processes.

ASSIGNMENT:

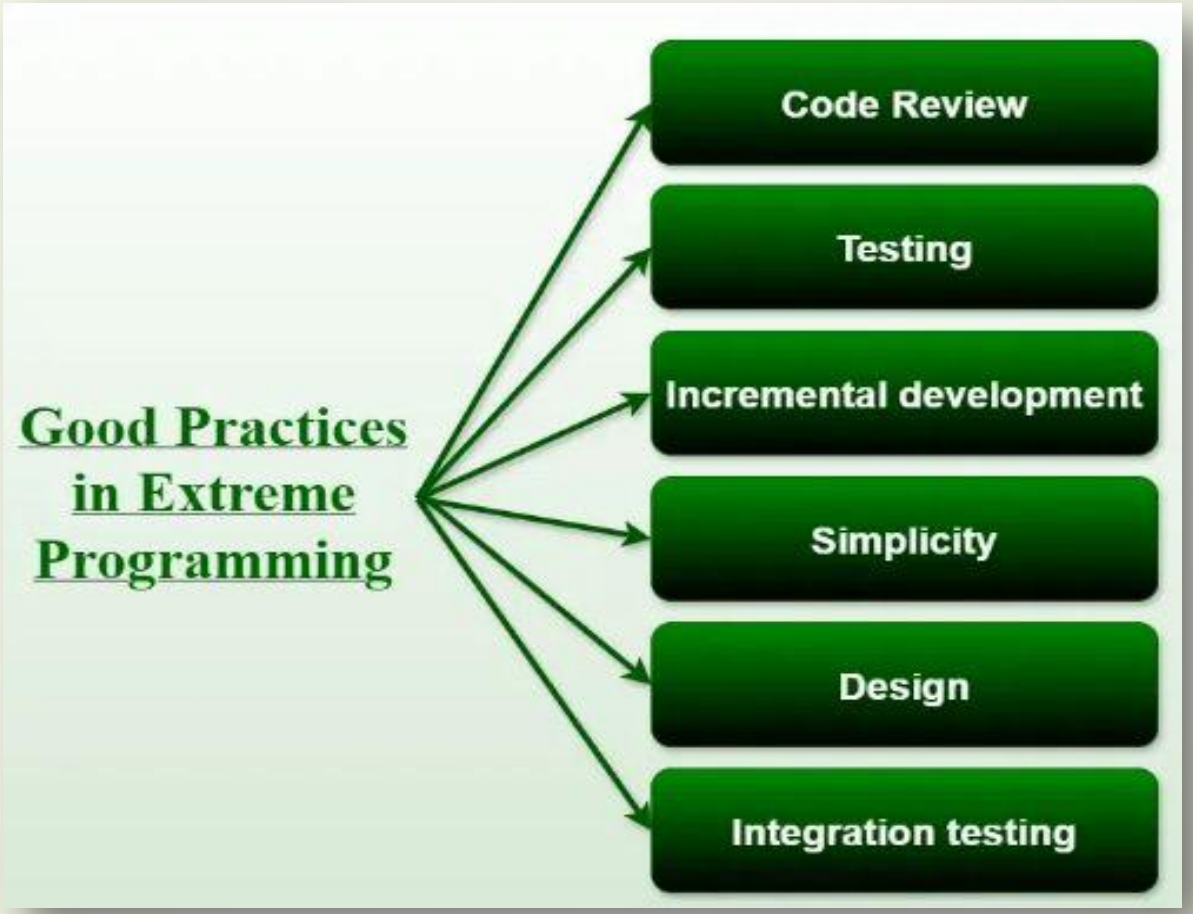
- Explanation
- Importance
- Advantages
- disadvantages



Extreme Programming (XP)

- Extreme programming (XP) is one of the most important software development frameworks of Agile models. It is used to improve software quality and responsiveness to customer requirements.
- Extreme Programming is one of the Agile software development methodologies.
- XP emphasizes a close working relationship between the development team, the customer, and stakeholders, with an emphasis on rapid, iterative development and deployment.
- An XP project starts with user stories which are short descriptions of what scenarios the customers and users would like the system to support.

Good Practices in Extreme Programming

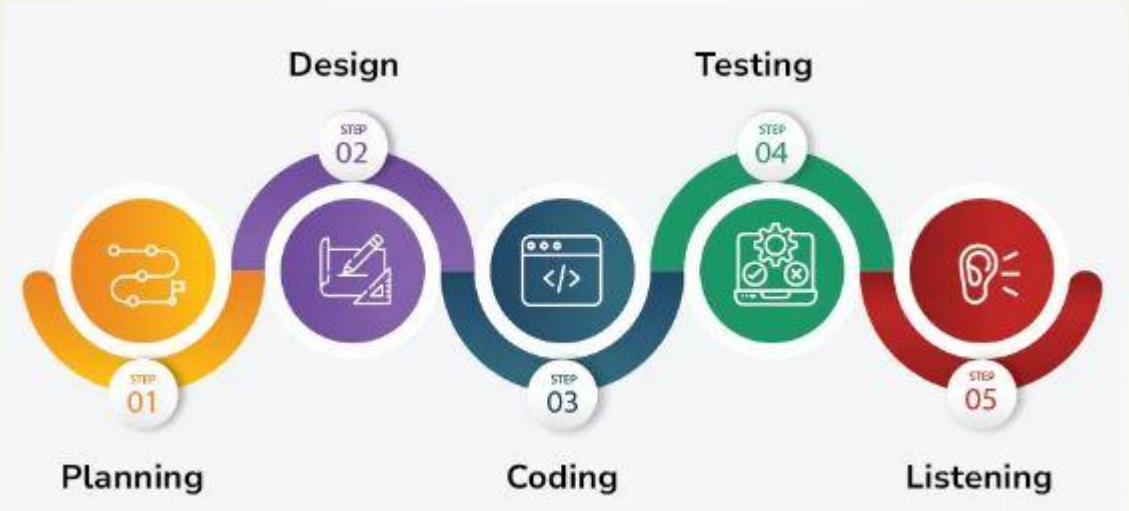


Some of the good practices that have been recognized in the extreme programming model and suggested to maximize their use are given below:

- **Code Review:** Code review detects and corrects errors efficiently. It suggests pair programming as coding and reviewing of written code carried out by a pair of programmers who switch their work between them every hour.
- **Testing:** Testing code helps to remove errors and improves its reliability. XP suggests test-driven development (TDD) to continually write and execute test cases. In the TDD approach, test cases are written even before any code is written.
- **Incremental development:** Incremental development is very good because customer feedback is gained and based on this development team comes up with new increments every few days after each iteration.
- **Simplicity:** Simplicity makes it easier to develop good-quality code as well as to test and debug it.
- **Design:** Good quality design is important to develop good quality software. So, everybody should design daily.
- **Integration testing:** Integration Testing helps to identify bugs at the interfaces of different functionalities. Extreme programming suggests that the developers should achieve continuous integration by building and performing integration testing several times a day.

Life Cycle of Extreme Programming (XP)

The Extreme Programming Life Cycle consist of five phases:



Planning: The first stage of Extreme Programming is planning. During this phase, clients define their needs in concise descriptions known as user stories. The team calculates the effort required for each story and schedules releases according to priority and effort.

Design: The team creates only the essential design needed for current user stories, using a common analogy or story to help everyone understand the overall system architecture and keep the design straightforward and clear.

Coding: Extreme Programming (XP) promotes pair programming i.e**.** we developers work together at one workstation, enhancing code quality and knowledge sharing. They write tests before coding to ensure functionality from the start (TDD), and frequently integrate their code into a shared repository with automated tests to catch issues early.

Testing: Extreme Programming (XP) gives more importance to testing that consist of both unit tests and acceptance test**.** Unit tests, which are automated, check if specific features work correctly. Acceptance tests, conducted by customers, ensure that the overall system meets initial requirements. This continuous testing ensures the software's quality and alignment with customer needs.

Listening: In the listening phase regular feedback from customers to ensure the product meets their needs and to adapt to any changes.

Applications of Extreme Programming (XP)

Some of the projects that are suitable to develop using the XP model are given below:

- **Small projects**
- **Projects involving new technology or Research projects**
- **Web development projects**
- **Collaborative projects**
- **Projects with tight deadlines**
- **Projects with rapidly changing requirements**
- **Projects where quality is a high priority**

Object-Oriented Analysis & Design(OOAD)

- OOAD is based on the concepts of object-oriented programming (OOP) and is an organized and systematic approach to designing and developing software systems.
- It is a software engineering paradigm that integrates two distinct but closely related processes: Object-Oriented Analysis (OOA) and Object-Oriented Design (OOD).

Important Aspects of OOAD

Below are some important aspects of OOAD:

- **Object-Oriented Programming:** In this the real-world items are represented/mapped as software objects with attributes and methods that relate to their actions.
- **Design Patterns:** Design patterns are used by OOAD to help developers in building software systems that are more efficient and maintainable.
- **UML Diagrams:** UML diagrams are used in OOAD to represent the different components and interactions of a software system.
- **Use Cases:** OOAD uses use cases to help developers understand the requirements of a system and to design software systems that meet those requirements.

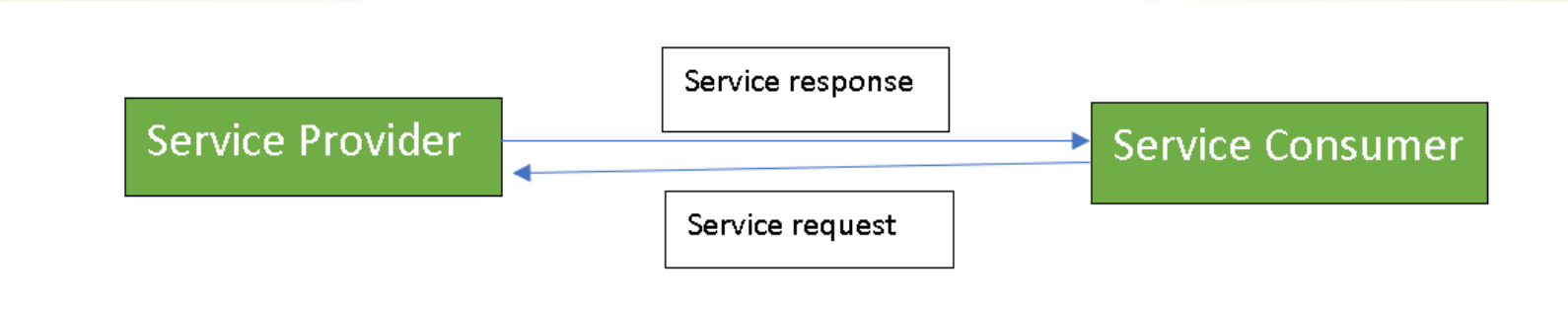
Service-oriented architecture

- Service-oriented architecture (SOA) is a method of software development that uses software components called services to create business applications.
- Each service provides a business capability, and services can also communicate with each other across platforms and languages.
- Developers use SOA to reuse services in different systems or combine several independent services to perform complex tasks.
- SOA is an architectural approach in which applications make use of services available in the network.
- In this architecture, services are provided to form applications, through a network call over the internet.
- It uses common communication standards to speed up and streamline the service integrations in applications. Each service in SOA is a complete business function in itself.
- The services are published in such a way that it makes it easy for the developers to assemble their apps using those services.

There are two major roles within Service-oriented Architecture:

Service provider: The service provider is the maintainer of the service and the organization that makes available one or more services for others to use. To advertise services, the provider can publish them in a registry, together with a service contract that specifies the nature of the service, how to use it, the requirements for the service, and the fees charged.

Service consumer: The service consumer can locate the service metadata in the registry and develop the required client components to bind and use the service.



System Acquisition

Outsourcing: If one organization develops or runs a computer application for another organization, that practice is called outsourcing.

Advantages of Outsourcing:

Cost Reduction: Accessing lower labor costs in different regions can significantly cut operational expenses.

Access to Expertise: Outsourcing allows companies to tap into specialized skills and knowledge that might not be readily available internally.

Increased Efficiency: By delegating tasks to specialized providers, companies can focus on core competencies and improve overall productivity.

Scalability: Easily scale up or down operations based on demand by adjusting outsourced services.

Faster Time to Market: Utilize external expertise to quickly launch products or services

Competitive Advantage: Gain a competitive edge by accessing superior quality or cost-effective services .

Disadvantages of Outsourcing:

Loss of Control: Reduced ability to directly manage outsourced functions, potentially impacting quality and decision-making

Communication Challenges: Language barriers and time zone differences can create communication difficulties with external providers

Quality Issues: Potential for inconsistent quality if the outsourced provider does not meet expectations

Data Security Risks: Sharing sensitive information with a third party can raise concerns about data breaches and confidentiality

Hidden Costs: Unexpected fees or additional costs associated with managing outsourced contracts

Sources of Software

The sources of software may grouped into six major categories:

i. Information technology services firms
ii. Packaged software producers,
iii. Enterprise-wide solutions
iv. Cloud computing vendors
v. Open-source software
vi. In-house developers

Information Technology Services Firms

- If a company needs an information system but does not have the expertise or the personnel to develop the system in-house, and a suitable off-the-shelf system is not available, the company will likely consult an information technology services firm.
- IT services firms help companies develop custom information systems for internal use, or they develop, host, and run applications for customers, or they provide other services.
- These firms employ people with expertise in the development of information systems. Their consultants may also have expertise in a given business area.

Packaged Software Producers

- The growth of the software industry has been phenomenal since its beginnings in the mid-1960s.
- Some of the largest computer companies in the world are companies that produce software exclusively.
- A good example is Microsoft, probably the best-known software company in the world. Almost 87 percent of Microsoft's revenue comes from its software sales, mostly for its Windows operating systems and its personal productivity software, the Microsoft Office Suite.

- The packaged software development industry serves many market segments.
- Their software offerings range from general, broad-based packages, such as productivity tools, to very narrow, niche packages, such as software to help manage a day care center.
- Software companies develop software to run on many different computer platforms, from microcomputers to large mainframes.
- The companies range in size from just a few people to thousands of employees.

Enterprise Solutions Software

- Many firms have chosen complete software solutions, called enterprise solutions or enterprise resource planning (ERP) systems, to support their operations and business processes.
- These ERP software solutions consist of a series of integrated modules.
- Each module supports an individual, traditional business function, such as accounting, distribution, manufacturing, or human resources.
- The traditional approach would use different systems in different functional areas of the business, such as a billing system in accounting and an inventory system in the warehouse.

- Using enterprise software solutions, a firm can integrate all parts of a business process in a unified information system.
- All aspects of a single transaction occur seamlessly within a single information system, rather than as a series of disjointed, separate systems focused on business functional areas.
- The benefits of the enterprise solutions approach include a single repository of data for all aspects of a business process and the flexibility of the modules.
- A single repository ensures more consistent and accurate data, as well as less maintenance.
- The modules are flexible because additional modules can be added as needed once the basic system is in place.
- Added modules are immediately integrated into the existing system.
- However, there are disadvantages to enterprise solutions software. The systems are very complex, so implementation can take a long time to complete.

Cloud Computing

- Cloud computing is the provision of computing resources, including applications, over the Internet, so customers do not have to invest in the computing infrastructure needed to run and maintain the resources.
- Another method for organizations to obtain applications is to rent them or license them from third-party providers who run the applications at remote sites.
- Users have access to the applications through the Internet or through virtual private networks.
- The application provider buys, installs, maintains, and upgrades the applications. Users pay on a per-use basis or they license the software, typically month to month.
- A well-known example of cloud computing is Google Apps, where users can share and create documents, spreadsheets, and presentations.

Open-Source Software

- Open-source software is different because it is freely available, not just the final product but the source code itself.
- It is also different because it is developed by a community of interested people instead of by employees of a particular company.
- Open-source software performs the same functions as commercial software, such as operating systems, e-mail, database systems, web browsers, and so on.
- Some of the most well-known and popular open-source software names are Linux, an operating system; mySQL, a database system; and Firefox, a web browser.
- Open source is developed and maintained by communities of people, and sometimes these communities can be very large.

In-House Development

- In-house development has become a progressively smaller piece of all systems development work that takes place in and for organizations.
- In-house development can lead to a larger maintenance burden than other development methods, such as packaged applications.

TABLE 2-2 Comparison of Six Different Sources of Software Components

Producers	When to Go to This Type of Organization for Software	Internal Staffing Requirements
IT services firms	When task requires custom support and system can't be built internally or system needs to be sourced	Internal staff may be needed, depending on application
Packaged software producers	When supported task is generic	Some IS and user staff to define requirements and evaluate packages
Enterprise-wide solutions vendors	For complete systems that cross functional boundaries	Some internal staff necessary but mostly need consultants
Cloud computing	For instant access to an application; when supported task is generic	Few; frees up staff for other IT work
Open source software	When supported task is generic but cost is an issue	Some IS and user staff to define requirements and evaluate packages
In-house developers	When resources and staff are available and system must be built from scratch	Internal staff necessary though staff size may vary

The most common criteria include the following:

- Cost
- Functionality
- Vendor support
- Viability of vendor
- Flexibility
- Documentation
- Response time
- Ease of installation

Reuse

- The use of previously written software resources, especially objects and components, in new applications.
- Reuse is the use of previously written software resources in new applications.
- So many bits and pieces of applications are relatively generic across applications, it seems intuitive that great savings can be achieved in many areas if those generic bits and pieces do not have to be written anew each time they are needed.
- Reuse should increase programmer productivity because being able to use existing software for some functions means they can perform more work in the same amount of time.
- Reuse should also decrease development time, minimizing schedule overruns.
- Because existing pieces of software have already been tested, reusing them should also result in higher-quality software with lower defect rates, decreasing maintenance costs.

The advantages of Reuse

- Increase software productivity
- Shorten software development time
- Improve software system interoperability
- Develop software with fewer people
- Move personnel more easily from project to project
- Reduce software development and maintenance costs
- Produce more standardized software
- Produce better quality software and provide a powerful competitive advantage

MANAGING THE INFORMATION SYSTEMS PROJECT

- Project management is an important aspect of the development of information systems and a critical skill for a systems analyst.
- The focus of project management is to ensure that systems development projects meet customer expectations and are delivered within budget and time constraints.
- The project manager is a systems analyst with a diverse set of skills—management, leadership, technical, conflict management, and customer relationship—who is responsible for initiating, planning, executing, and closing down a project.
- In some organizations, the project manager is a very experienced systems analyst, whereas in others, both junior and senior analysts are expected to take on this role, managing parts of a project or actively supporting a more senior colleague who assumes the project manager role.
- Creating and implementing successful projects requires managing the resources, activities, and tasks needed to complete the information systems project.
- A project is a planned undertaking of a series of related activities to reach an objective that has a beginning and an end.
- Project management is a controlled process of initiating, planning, executing, and closing down a project.

A project manager juggles numerous activities

Common activities of a Project Manager

Activity	Description
Leadership	Influencing the activities of others toward the attainment of a common goal through the use of intelligence, personality, and abilities
Management	Getting projects completed through the effective utilization of resources
Customer relations	Working closely with customers to ensure that project deliverables meet expectations
Technical problem solving	Designing and sequencing activities to attain project goals
Conflict management	Managing conflict within a project team to assure that conflict is not too high
Team management	Managing the project team for effective team performance
Risk and change management	Identifying, assessing, and managing the risks and day-to-day changes that occur during a project

Four phases of project management

1. Initiating the project
2. Planning the project
3. Executing the project
4. Closing down the project

Initiating the project

- The first phase of the project management process in which activities are performed to assess the size, scope, and complexity of the project and to establish procedures to support later project activities.
- During project initiation, the project manager performs several activities to assess the size, scope, and complexity of the project and to establish procedures to support subsequent activities.
- Depending on the project, some initiation activities may be unnecessary and some may be very involved.
- The types of activities you will perform when initiating a project are summarized as follows:

i. Establishing the project initiation team.
ii. Establishing a relationship with the customer
iii. Establishing the project initiation plan
iv. Establishing Management Procedures
v. Establishing the Project Management Environment and Project Workbook
vi. Developing the Project Charter

PLANNING THE PROJECT

- The second phase of the project management process that focuses on defining clear, discrete activities and the work needed to complete each activity within a single project.
- The next step in the project management process is project planning.
- Research has found a positive relationship between effective project planning and positive project outcomes.
- Project planning involves defining clear, discrete activities and the work needed to complete each activity within a single project.
- It often requires you to make numerous assumptions about the availability of resources such as hardware, software, and personnel.

Activities During Project Planning

i. Describing Project Scope, Alternatives, and Feasibility

ii. Dividing the Project into Manageable Tasks

iii. Estimating Resources and Creating a Resource Plan

iv. Developing a Preliminary Schedule

v. Developing a Communication Plan

vi. Determining Project Standards and Procedures

vii. Identifying and Assessing Risk

viii. Creating a Preliminary Budget

ix. Developing a Project Scope Statement

x. Setting a Baseline Project Plan

EXECUTING THE PROJECT

- The third phase of the project management process in which the plans created in the prior phases (project initiation and planning) are put into action.
- Project execution puts the Baseline Project Plan into action.
- Within the context of the SDLC, project execution occurs primarily
- during the analysis, design, and implementation phases.

Activities During Project Execution

1. Executing the Baseline Project Plan
2. Monitoring Project Progress against the Baseline Project Plan
3. Managing Changes to the Baseline Project Plan
4. Maintaining the Project Workbook
5. Communicating the Project Status

CLOSING DOWN THE PROJECT

- The final phase of the project management process that focuses on bringing a project to an end.
- The focus of project closedown is to bring the project to an end.
- Projects can conclude with a natural or unnatural termination. A natural termination occurs when the requirements of the project have been met—the project has been completed and is a success.
- An unnatural termination occurs when the project is stopped before completion.

Activities During Project Closing

1. Closing down the project.
2. Conducting post project reviews.
3. Closing the customer contract.

REPRESENTING AND SCHEDULING PROJECT PLANS

- A project manager has a wide variety of techniques available for depicting and documenting project plans.
- These planning documents can take the form of graphical or textual reports, although graphical reports have become most popular for depicting project plans.
- The most commonly used methods are Gantt charts and network diagrams. Because Gantt charts do not (typically) show how tasks must be ordered (precedence) but simply show when a task should begin and when it should end, they are often more useful for depicting relatively simple projects or subparts of a larger project, showing the activities of a single worker, or monitoring the progress of activities compared to scheduled completion dates.
- A network diagram shows the ordering of activities by connecting a task to its predecessor and successor tasks.
- Sometimes a network diagram is preferable; other times a Gantt chart more easily shows certain aspects of a project.

Fig: Gantt Chart

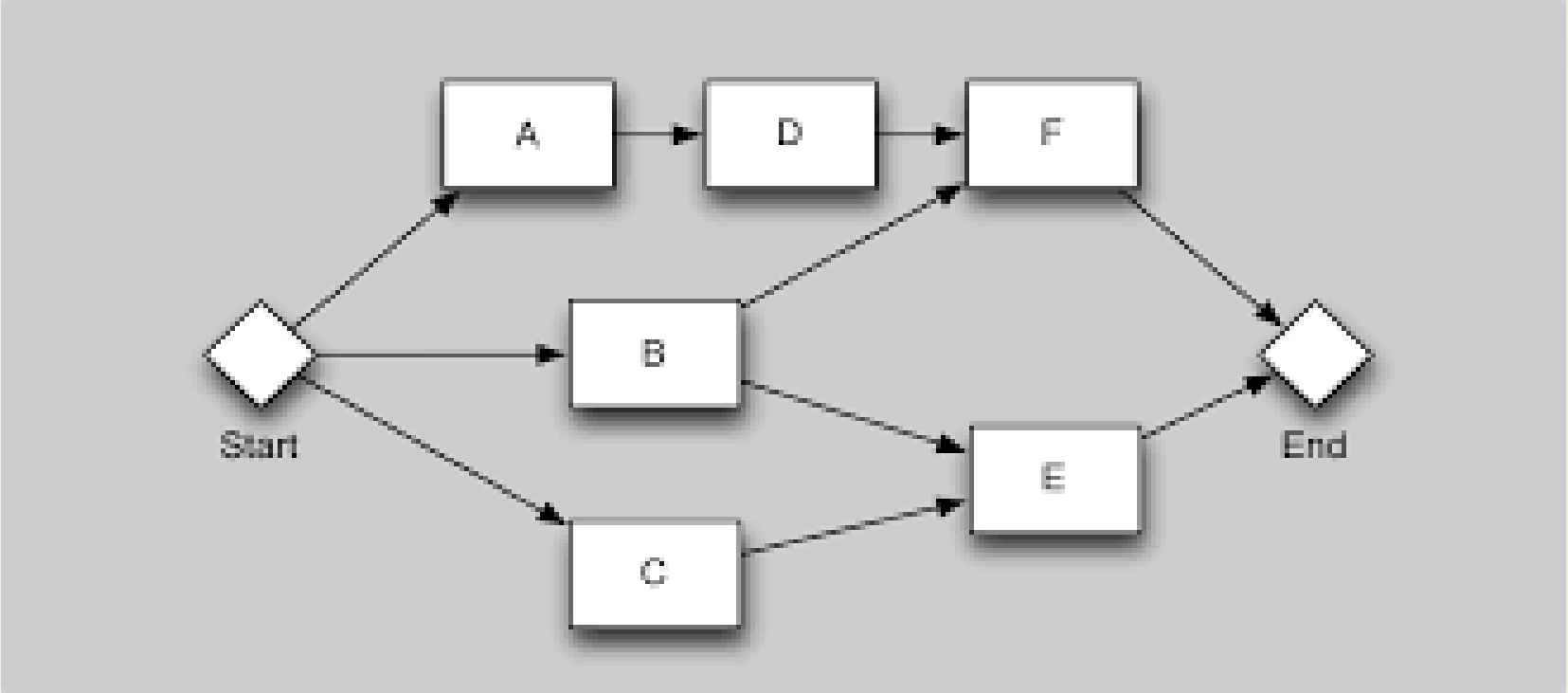


Fig: Network Daigram

Assignment

PERT (Program Evaluation Review Technique)

Using Project Management Software

- Using project management software" means utilizing a digital platform to plan, organize, and track the progress of a project, including assigning tasks to team members, setting deadlines, monitoring progress, and managing resources, all within a centralized system to facilitate collaboration and ensure project completion on time and within budget.
- People also refer to project management software as Task Management Software or Project Portfolio Management (PPM).
- The three major pillars of project management are planning, tracking, and collaboration.
- Example:

Smartsheet : Spreadsheet-like interface for flexible project planning

Monday.com: Highly customizable with various visual boards and workflows