★ **What is an information system ?**

An arrangement of Computer Technology and Telecommunication Technology to support and improve day to day operations, problem solving and decision making needs of management and users.

★ **Information systems consist of four main components**

Input (collects data)
Processing (process data)
Output (disseminates data)
Feedback

★ **The resources of a CBIS (Computer Based Information Systems) include**

hardware
software
databases
telecommunications
people
procedures

★ **System Development Environment**

Stakeholders

Any person who has an interest in an information system and its outputs.

There are five types of stakeholders, (Systems User, Systems Owner, Systems Analyst, Systems Designer, Systems Builder)

1.Systems User (client)

A “customer” who will use an information system

Systems users define the business requirements and performance expectations for the system to be built.

2.Systems Owner

Systems owner is an information system’s sponsor and advocate and he owns the final system.

3.Systems Analyst

System analysts are people who understand both business and computing.

Role of system analyst

1. Identify the problem
2. Analyze and understand the problem
3. Identify the solution requirements
4. Identify alternative solutions
5. Design and implement the best solution
6. Evaluate the result

4.Systems Designer

System designers are technical specialists. Translate systems users business requirements and constraints into technical solutions

5.Systems Builder

System builders are technical specialists involved with Constructing,Testing,Delivering the system into operation.

★ Types of Information Systems

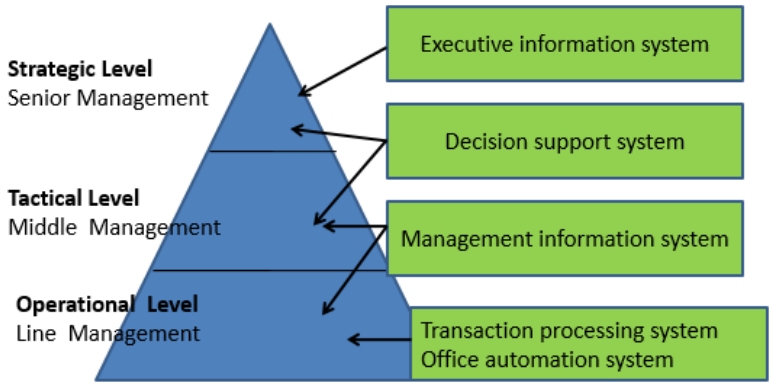
1.Classification by mode of processing

- Batch processing systems
- On-line batch systems
- On-line Real-time systems

2.Classification by System Objectives

- Transaction Processing System (TPS)
- Management Information System (MIS)
- Decision Support System (DSS)
- Executive Information System (EIS)
- Expert Systems (ES)
- Communications and Collaboration Systems
- Office Automation System
- Geographic Information System (GIS)

★ **The use of information systems by management level.**



1. Transaction Processing System (TPS)

   Information Systems that capture and process data about business transactions.
   Used mainly by operational level employees

2. Management Information System (MIS)

   MIS is an information system application that provides for management oriented reporting.

3. Decision Support System (DSS)

   They are interactive systems that assist a decision maker when faced with unstructured or semi structured business problems.

4. Executive Information System (EIS) /Executive Support System(ESS)

   An information system designed for top-level managers.
   They integrates data from all over the organization into graphical indicators and controls

5. Expert Systems (ES)

   An expert system is a programmed decision making information system.

6. Communications and Collaboration Systems

   An is that enables more effective communications between,
   Workers,Partners ,Customers ,Suppliers Enhance their ability to collaborate

7. Office Automation System

   It supports a wide range of business office activities.(activities are
   Work group computing, Work group scheduling, E-mail, Electronic document)

8. Geographic Information System (GIS)

A system designed to capture, store, manipulate, analyze, manage, and present all types of spatial or geographical data.

★ Legacy Systems

Legacy system is an "antiquated" system.

- potentially problematic
- often run on obsolete hardware
- spare parts for such computers become increasingly difficult to obtain
- hard to maintain, improve and expand
- The designers of the system may have left the organization, leaving no one left to explain how it works.

★ Systems Development Life Cycle (SDLC)

It is a logical process by which systems analysts, software engineers, programmers & end users build information systems.

★ Need of a life cycle

- ease the process of building a system.
- build high quality systems (that meets customer expectations, within time and cost estimates)
- maintain and enhance cost effectively
- avoid failures (like unclear objectives, cost overruns)
- work effectively and efficiently

★ Main phases of SDLC

1. Problem Definition (systems Investigation)

This phase identifies and defines a need for the new system.

Terms of Reference (TOR) - Project goals, project bounds & project limits

2. Systems Analysis

It specifies the system users' business requirements, expectations and priorities for a solution to the business problem

3. Systems Design

Produces a design specification for the new system.

Things done during the design phase,

- Identify suitable hardware
- Specify new programs or changes to existing programs
- Specify new database or changes to existing database
- produce detailed procedures that describe how users will use the system

4. Systems Implementation

During the systems implementation phase,

- individual system components are built and tested.
- user interfaces are developed and tried by users.
- a database is built to store data.
- data and tools are used to build the system.

5. Systems Testing

- Individual program modules are tested by their developers.
- Integration testing is done to test whether the modules can be combined.
- It is important to design test cases that test all the conditions that can arise in the system input.

6. Systems Maintenance

In this phase the following problems are corrected.

- Eliminate errors in the system during its lifetime.
- Fix any bugs and problems found by users.
- Tune the system into any variation in its working environment.

★ Major components of system development

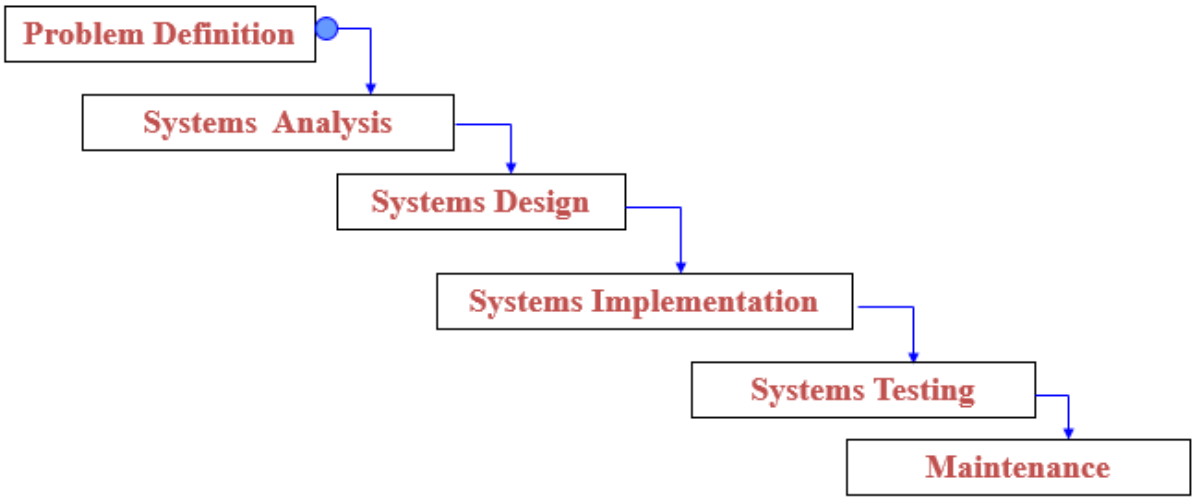
1. Methodology
2. Modeling Methods or Techniques (Data Flow Diagrams, Entity Relationship Diagrams , Structure Charts etc.)
3. Tools

★ Software Process Models

1. Linear or Waterfall Cycle

It is a linear sequential model. It has Separate and distinct phases of specification and development

An approach to system analysis and design.
Completes each phase one after another and only once.



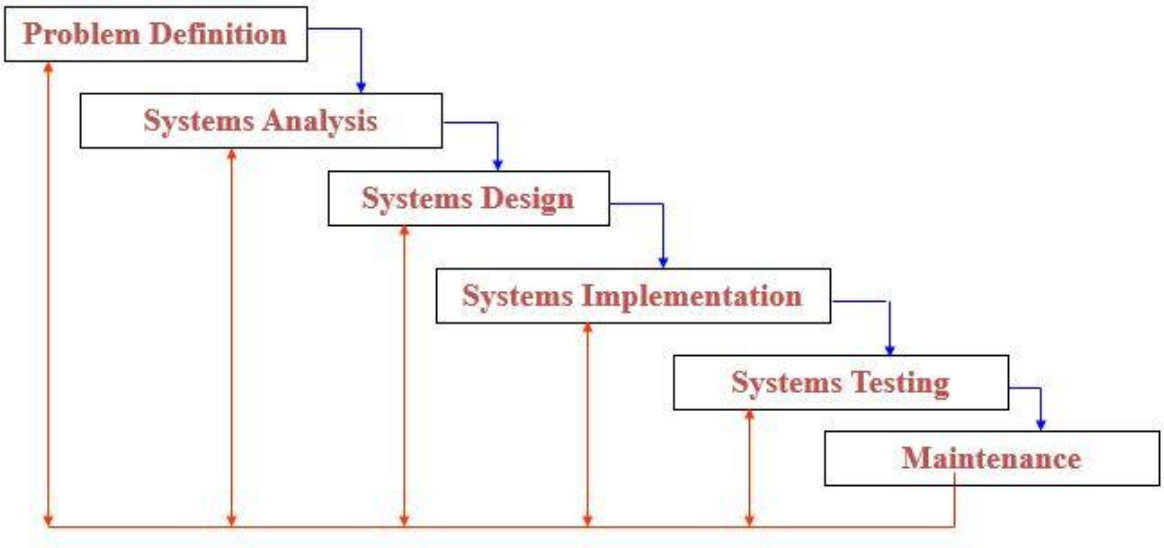
Waterfall Strengths,

- Easy to understand, easy to use
- Good for management control (plan, staff, track)
- Sets requirements stability
- Milestones are well understood
- Provides structure to inexperienced staff
- Works well when quality is more important than cost or schedule

There are several problems with the Waterfall Development Approach.

- Inflexible
- One phase must be completed before the next phase starts
- It has a top-down procedure
- No phase can be repeated
- Time consuming
- It has a rigid design

2. Modified Waterfall development approach



- Uses the same phases as the pure waterfall development approach.
- Progress is more difficult to track.
- Allow some of the stages to overlap, such as the requirements stage and the design stage.
- Overlapping stages make it possible to integrate feedback from the design phase into the requirements.

3. Iterative Development Approach

The iteration process helps to develop a part of the new system and place it into operation as quickly as possible.

Each iteration does some,
analysis
design
construction /implementation

4. Prototyping

Prototyping is exploring ideas before the real implementation to verify proposed solutions.

There are two types of Prototyping techniques,

- Throw-away Prototyping
- Evolutionary Prototyping

1.Throwaway Prototyping

Main objective is to show the user how it may work in the real system

2. Evolutionary Prototyping

The initial prototype is presented to the user. Users provide feedback and suggestions for improvements.

The developer who then presents a more refined version of the prototype. The user once more provides feedback. The process is repeated.

Prototyping applications/tools:

1. InVision
2. Marvel
3. Proto.io
4. Principle
5. Flinto
6. Origami Studio

Structured Evolutionary Prototyping Strengths,

- Customers can “see” the system requirements as they are being gathered
- Developers learn from customers
- A more accurate end product
- Unexpected requirements accommodated
- Allows for flexible design and development
- Interaction with the prototype stimulates awareness of additional needed functionality

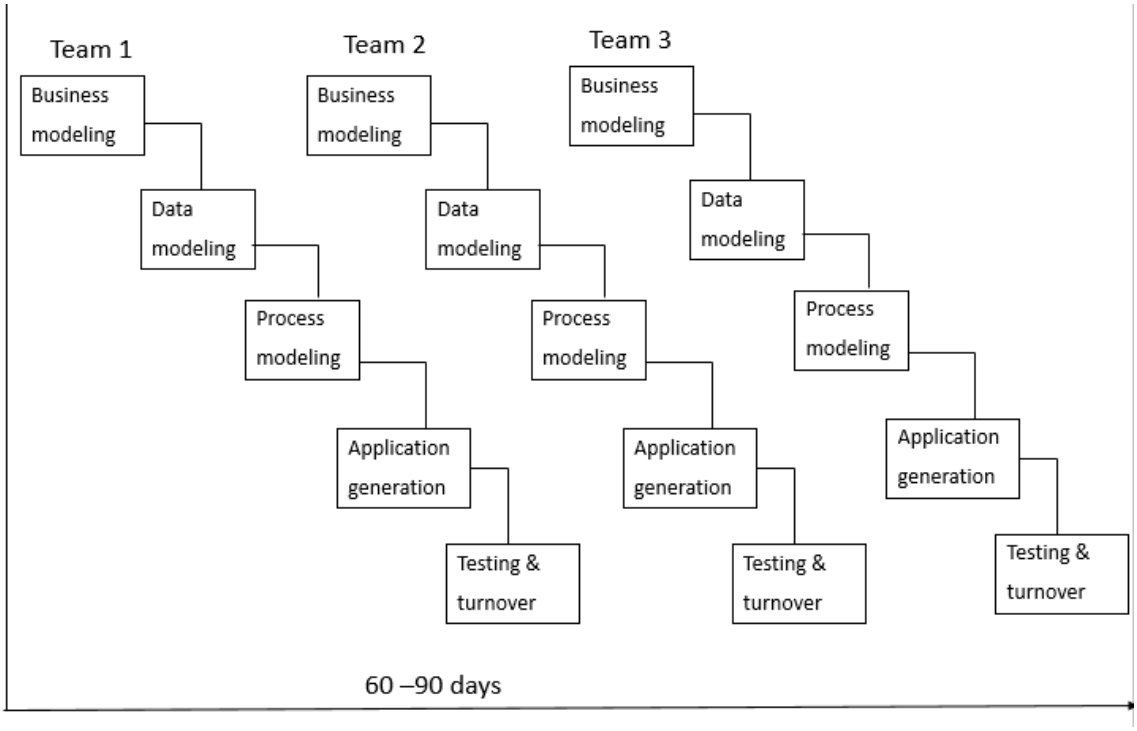
Prototyping Weaknesses,

- Prototyping can lead to false expectations.
- Prototyping can lead to poorly designed systems.
- Overall maintainability may be overlooked
- The customer may want the prototype delivered.
- Process may continue forever (scope creep)
- Continual change tends to corrupt the structure of the prototype system.Maintenance is therefore likely to be difficult and costly.

5. The RAD model

Rapid Application Development (RAD) is an incremental software development process model that emphasizes an extremely short development cycle.

the RAD process enables a development team to create a 'fully functional system' within very short time periods (eg. 60 to 90 days)



- Business modeling - The information flow in a business system considering its functionality.

- Data Modelling - The information flow defined as part of the business modeling phase is refined into a set of data objects that are needed to support the business

- Process Modelling - The data objects defined in the data modeling phase are transformed to achieve the information flow necessary to implement business functions.

- Application generation - RAD assumes the use of 4GL or visual tools to generate the system using reusable components.

- Testing and turnover - New components must be tested and all interfaces must be fully exercised

Advantages of RAD

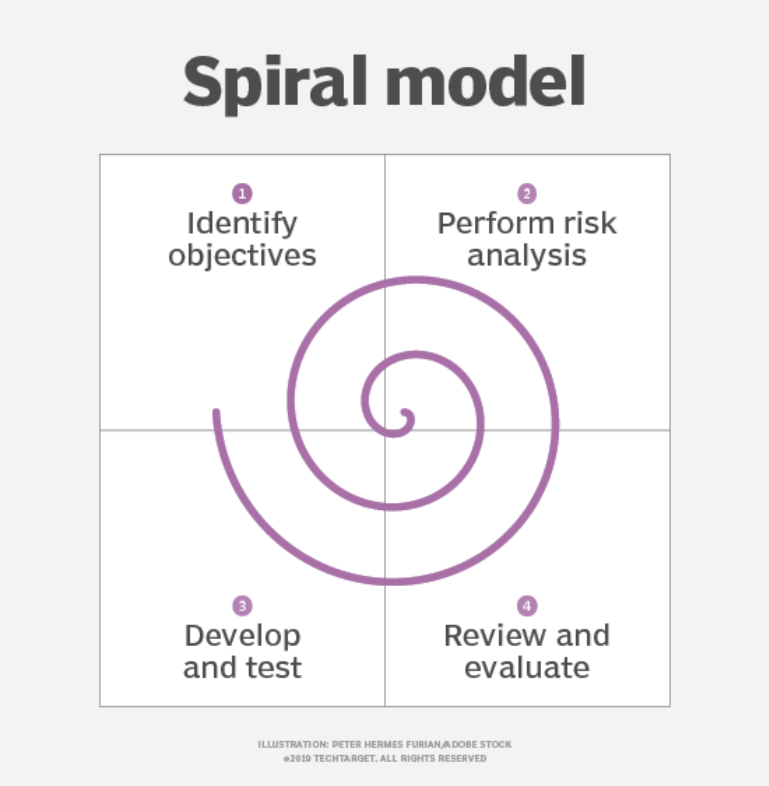
- Can speed up systems development.
- Users are thoroughly involved from the start.
- Improves the process of rewriting legacy applications.

Some problems with the RAD model

- RAD is not applicable when technical risks are high.
- RAD requires sufficient human resources to create right number of RAD teams
- RAD requires developers and customers who are committed to the rapid-fire activities necessary to get a system completed in a much abbreviated time frame.
- If a system cannot be properly modularized, building the components necessary for RAD will be problematic.

6. Spiral model

The spiral model aims at risk reduction by any means in any phase.



Spiral Model Strengths

- The design does not have to be perfect
- Users can be closely tied to all lifecycle steps
- Early and frequent feedback from users
- Users see the system early because of rapid prototyping tools
- Provides early indication of serious risks, without much cost
- Critical high-risk functions are developed first

Spiral Model Weaknesses

- The model is complex
- Risk assessment expertise is required
- Time spent for evaluating risks too large for small or low-risk projects
- May be hard to define objective, verifiable milestones that indicate readiness to proceed through the next iteration

7. V-Shaped SDLC Model

Testing of the product is planned in parallel with a corresponding phase of development

V-Shaped Strengths

- Easy to use
- stages of product development
- Each deliverable must be testable
- Project management can track progress by milestones
- Emphasize planning for verification and validation of the product in early

V-Shaped Weaknesses

- Does not contain risk analysis activities
- Does not handle iterations or phases
- Does not easily handle concurrent events
- Does not easily handle dynamic changes in requirements

8. Agile Development

Agile development is an iterative software-development methodology which teams use in projects.

★ What is a model ?

A model is a simplified representation of a complex system or concept used for understanding and prediction.

★ What is the feasibility study ?

Analysis to assess project viability before investing resources.
Consider ,

- Economic Feasibility
  It is a measure of the cost-effectiveness of a project.

- Technical Feasibility
  It is a measure of the practicality of a technical solution .

- Legal Feasibility
  It is a measure of how well a solution can be implemented within existing legal and contractual obligations.

- Cultural / Political Feasibility
  Cultural feasibility deals with how the end users feel about the proposed system.

- Operational Feasibility
  It is a measure of how well a solution meets the identified system requirements to solve the problem.

- Schedule Feasibility
  A measure of how reasonable a project time table is.

★ System Requirements

Specify what the information system must do, or what property / quality the system must have

1. Functional Requirements

   Specify what the information system must do

2. Non functional Requirements

   Specify a property / quality the system must have

Example : Library system,

Functional - borrowing books, returning process
Non functional - security level, interfaces

★ Requirement Discovery Methods (Fact Finding Techniques)

It is the formal process of using techniques to collect information about systems requirements

Methods,

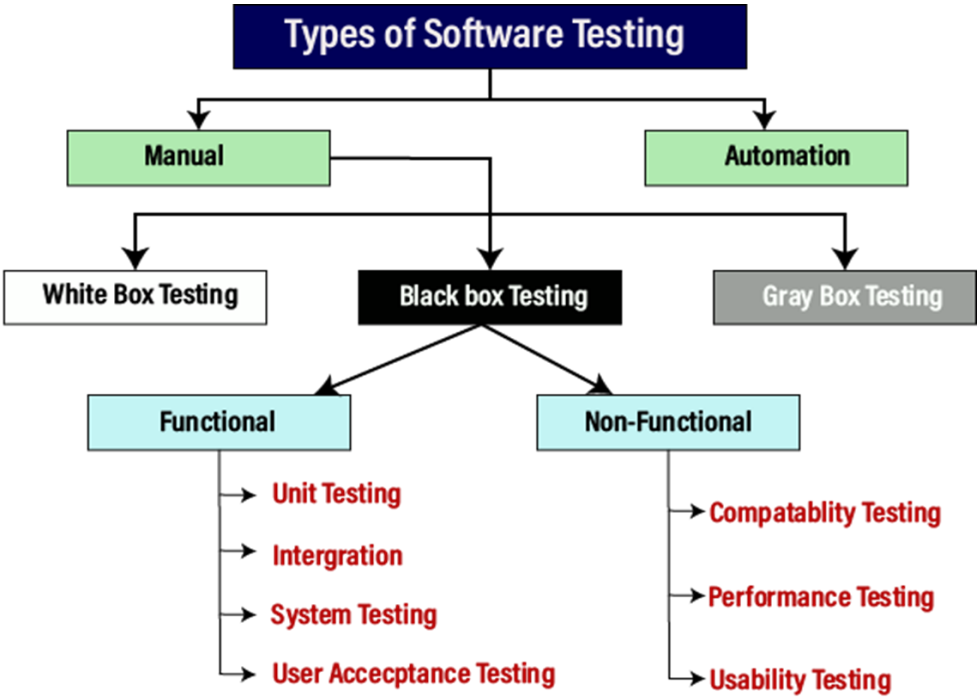
1. Interviewing users and other stakeholders
2. Distributing and collecting questionnaires
3. Reviewing inputs, outputs, and documentation
4. Observing and documenting business procedures
5. Researching vendor solutions
6. Collecting active user comments and suggestions

★ Use Case Diagram Components

1. Actor
2. Use case
3. System boundary
4. Actor to Use case Association
5. Notes
6. Inheritance

★ Software testing

Software testing is an activity conducted in the software development life-cycle to verify that the software is accurate and works according to the requirements.



★ Manual software testing

Manual software testing is when human testers check the quality of a new application without using automation tools or scripting.

1. White box testing

The white box testing is done by Developer,

where they check every line of a code before giving it to the Test Engineer. Since the code is visible for the Developer during the testing

That's why it is also known as White box testing.

2. Black box testing

The black box testing is done by the Test Engineer,

where they can check the functionality of an application or the software according to the customer /client's needs.

In this, the code is not visible while performing the testing

- Unit testing

  Unit testing is a type of software testing that focuses on individual units or components of a software system.

3. Gray box testing

Gray box testing is a combination of white box and Black box testing.

It can be performed by a person who knew both coding and testing.

And if the single person performs white box, as well as black-box testing for the application

★ limitations of software testing

1. Not testing everything
2. Errors that cannot be found
3. Limited time and resources
4. Dependence on test data
5. Cannot be completely sure it's perfect

★ What are the CASE (computer aided software engineering) tools ?

A CASE tool is a product that helps to analyze, model and document business processes.

★ **Function of CASE tools**

- Code Generation Tools
- Testing Tools
- Documentation Generators
- Project Management Tools
- Requirements Management Tools

★ **Verification**

Verification is the process of checking if software is being built correctly according to requirements and standards.

★ **Validation**

Validation is the process of checking if the software meets user needs and functions as intended.