PAPER XXII(M404)
MANAGEMENT INFORMATION SYSTEM
UNIT - III

System development life cycle (SDLC) : models, tools used for development

Prepared By:

AFTAB

Guest Faculty

Dept. of Management

MMHAPU

Software/System Development Life Cycle

- Software life cycle models describe phases of the software cycle and the order in which those phases are executed.
- There are following six phases in every Software development life cycle model:
  1. Requirement gathering and analysis
  2. Design
  3. Implementation or coding
  4. Testing
  5. Deployment
  6. Maintenance

1) Requirement gathering and analysis:

- Business requirements are gathered in this phase. This phase is the main focus of the project managers and stake holders.
- Meetings with managers and users are held in order to determine the requirements .
- After requirement gathering these requirements are analyzed for their validity and the possibility of incorporating the requirements in the system to be development is also studied.
- Finally, a Requirement Specification document is created which serves the purpose of guideline for the next phase of the model.

2) Design:

- In this phase the system and software design is prepared from the requirement specifications which were studied in the first phase.
- System Design helps in specifying hardware and system requirements and also helps in defining overall system architecture.

3) Implementation / Coding:

- On receiving system design documents, the work is divided in modules/units and actual coding is started.
- In this phase the code is produced so it is the main focus for the developer.
- This is the longest phase of the software development life cycle.

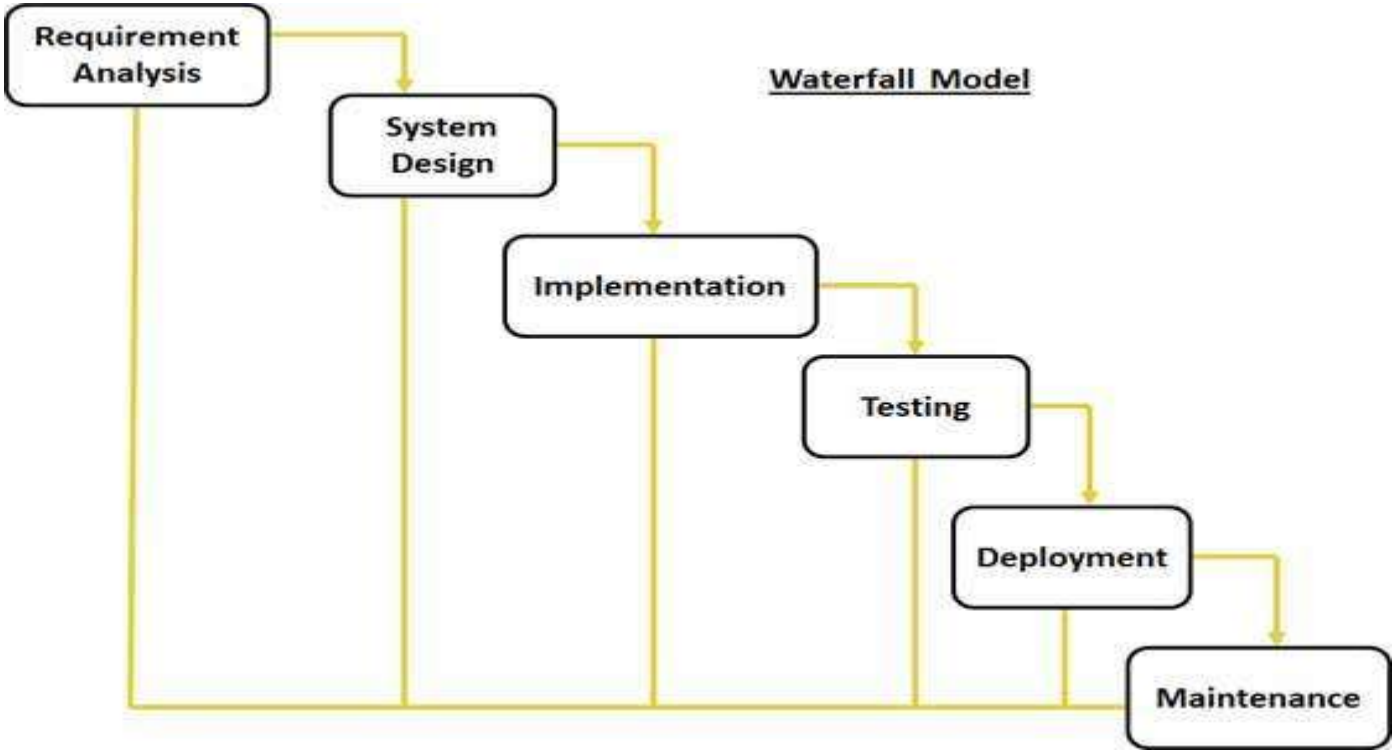
SDLC Models:

There are various software development life cycle models defined and designed which are followed during software development process.

- Waterfall Model
- Iterative Model
- Incremental Model
- Spiral Model

Waterfall Model

- a linear-sequential life cycle model.
- In a waterfall model, each phase must be completed before the next phase can begin and there is no overlapping in the phases.

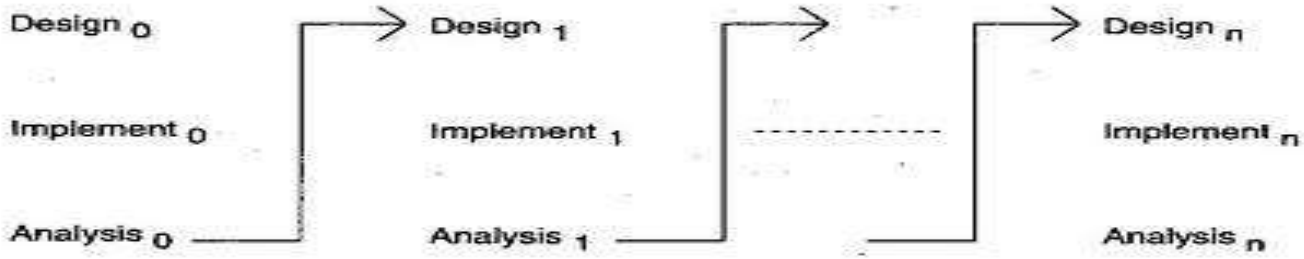


- **Requirement Gathering and analysis:** All possible requirements of the system to be developed are captured in this phase and documented in a requirement specification doc.
- **System Design:** The requirement specifications from first phase are studied in this phase and system design is prepared. System Design helps in specifying hardware and system requirements and also helps in defining overall system architecture.
- **Implementation:** On receiving system design documents, the work is divided in modules/units and actual coding is started.

- **Testing:** All the units developed in the implementation phase are integrated into a system after testing of each unit. Post integration the entire system is tested for any faults and failures.
- **Deployment of system:** Once the functional and non functional testing is done, the product is deployed in the customer environment or released into the market.
- **Maintenance:** There are some issues which come up in the client environment. To fix those issues patches are released. Also to enhance the product some better versions are released. Maintenance is done to deliver these changes in the customer environment.

Iterative Model/Prototyping:

- It is the process of quickly putting together a working model (a prototype) in order to test various aspects of a design, illustrate ideas or features and gather early user feedback.
- Development begins by specifying and implementing just part of the Information system, which can then be reviewed in order to identify further requirements.
- This process is then repeated, producing a new version of the information system for each cycle of the model.

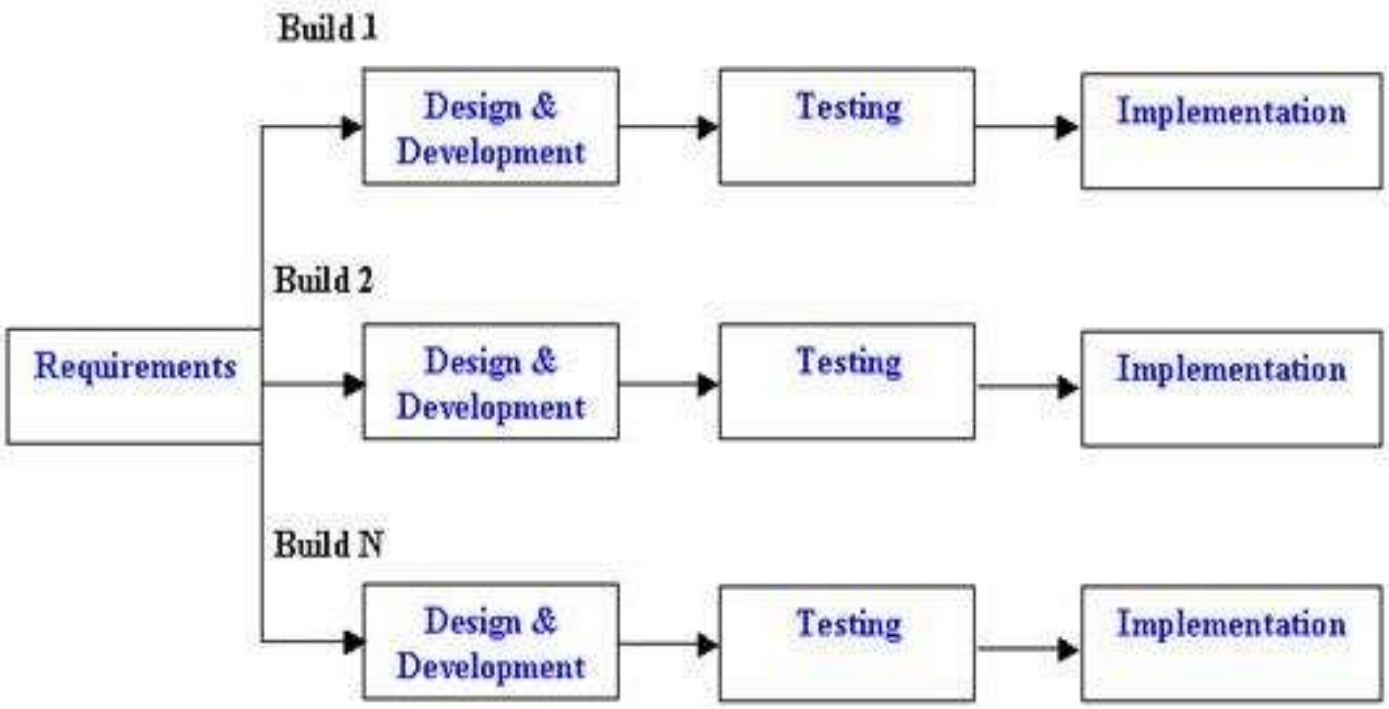


Incremental Model:

- In incremental model the whole requirement is divided into various builds. A step by step module is prepared.

keep on adding the pieces until it's complete.

FIG. INCREMENTAL MODEL



Incremental Life Cycle Model

Advantages & Disadvantages of Prototyping

Advantages

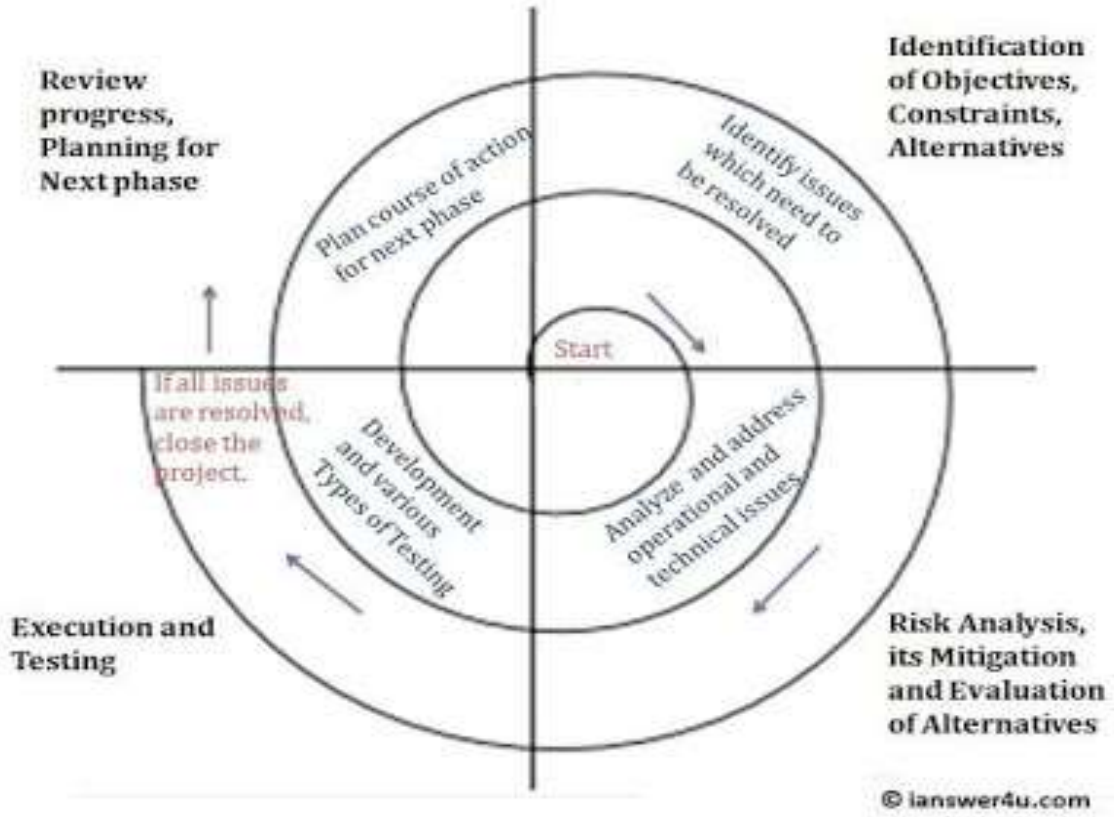
- Users can try the system and provide constructive feedback during development
- An operational prototype can be produced in weeks
- Users become more positive about implementing the system as they see a solution emerging that will meet their needs
- Prototyping enables early detection of errors

Disadvantages

- Each iteration builds on the previous iteration and further refines the solution. This makes it difficult to reject the initial solution as inappropriate and start over.
- Formal end-of-phase reviews do not occur. Thus, its is very difficult to contain the scope of the prototype.
- System documentation is often absent or incomplete, since the primary focus is on development of the prototype
- System backup and recovery, performance, and security issues can be overlooked

Spiral Model:

- It is generally implemented in high risk projects. It was first proposed by Boehm. In Spiral model we can arrange all the activities in the form of a spiral.



Each loop has four sections or quadrants :

1. ***To determine the objectives, alternatives and constraints***. We try to understand the product objectives, alternatives in design and constraints imposed because of cost, technology, schedule, etc.
2. ***Risk analysis and evaluation of alternatives***. Here we try to find which other approaches can be implemented in order to fulfill the identified constraints. Operational and technical issues are addressed here. Risk mitigation is in focus in this phase. And evaluation of all these factors determines future action.
3. ***Execution of that phase of development***. In this phase we develop the planned product. Testing is also done. In order to do development, waterfall or incremental approach can be implemented.
4. ***Planning the next phase***. Here we review the progress and judge it considering all parameters. Issues which need to be resolved are identified in this phase and necessary steps are taken.

RAD Model (Rapid Application Development)

- Developed by IBM in 1980
- User requirement is necessary from requirement phase to delivery of the product.
- User's expectations are fulfilled

Steps:

1. Requirement planning
2. User description
3. Construction phase
4. Cut over phase

- *Requirement planning*: Active involvement of users for understanding the project is the only issue.
- *User description*: Joint teams of developers and users are constituted to prepare, understand and review the requirements
- *Construction phase*: Here detailed design, coding and testing are conducted but by user involvement.
- *Cut over phase*: Here installation of software is studied and user training is conducted.

Tools used in System Development:

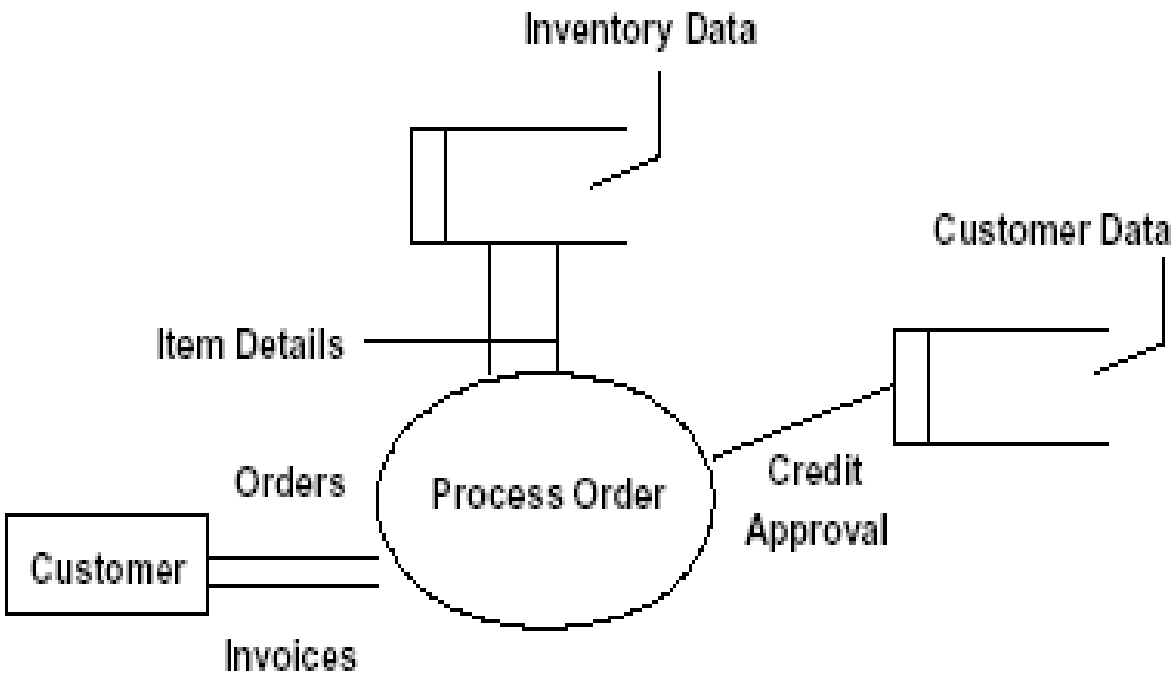
- Data Flow Diagram (DFD)
- Flowchart
- Decision Tables

Data Flow Diagram

- It is a graphical representation of the logical flow of data. It is also known as a bubble chart.
- It decomposes the requirement specifications down to the lowest level of details.
- A DFD consists of a series of bubbles joined by lines representing data flow in the system.
- Four main symbols

Symbol	Name	Description
Library	Process	Circles are used to represent processes. Processes are actions taking place to transform inputs to outputs. In a context diagram, a system is represented by a single, labelled circle. In a data flow diagram, multiple circles represent multiple processes within the system.
Payment	Data Flow	Curved lines represent data flows between processes, data stores and external entities. Data flows should be named to identify the piece of data.
Student	External Entity	Boxes are used to represent external entities. These are any item, person or organisation sitting outside the systems that provides data to the system or receives data from the system.
The following symbol is only used in data flow diagrams.		
User Accounts	Data Store	An open-ended rectangle is used to represent a data store. Data stores include electronic or non computer-based stores of data. They should be named with a logical name.

Example of Data Flow Diagram :



NOTE : Using Physical and Logical Data Flow Diagrams :

Flowchart

- A flowchart is a pictorial representation of an algorithm.
- Programmers often use it as a program planning tool for visually organizing a sequence of steps necessary to solve a problem using a computer.
- An algorithm is first represented as a flowchart and the flowchart is then expressed in a programming language to prepare a computer program.

Example

Write an algorithm and draw a flowchart to convert the length in feet to centimeter.

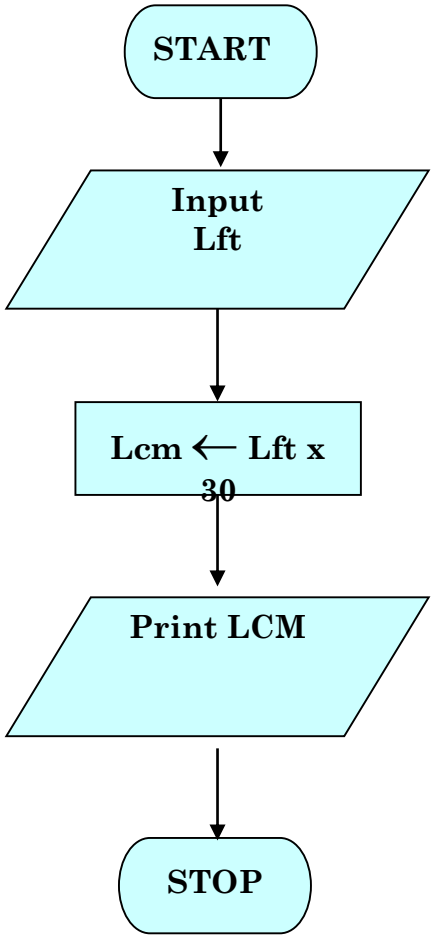
Pseudocode:

- *Input the length in feet (Lft)*
- *Calculate the length in cm (Lcm) by multiplying LFT with 30*
- *Print length in cm (LCM)*

Flowchart Example:

Algorithm

- Step 1: Start
- Step 2: Input Lft
- Step 3: Lcm ← Lft x 30
- Step 4: Print Lcm
- Step 5: Stop



Advantage/Disadvantage of Flow Chart

Advantages:

- Better communication
- Efficient coding
- Systematic debugging
- Concentrate on the logic
- Easy to detect errors

Disadvantages:

- They are time consuming
- Change or modification requires a completely new flowchart
- Redrawing a flowchart is a tedious task

Decision table

- A matrix representation of the logic of a decision
- Specifies the possible conditions and the resulting actions
- Best used for complicated decision logic
- It consists of rows and columns divided into 4 separate quadrants.

Conditions	Condition Table
Actions	Action Table

Example of Decision Table

	Rules							
Conditions	1	2	3	4	5	6	7	8
C1. Infant passengers (age: < 2)	Y	Y						
C2. Youth passengers (age: 2 to 16)			Y	Y				
C3. Frequent flyers					Y	Y		
C4. Domestic flights	Y							
C5. International flighers		Y						Y
C6. Early reservation				Y		Y	Y	
C7. Off-season traveling								Y
Actions	1	2	3	4	5	6	7	8
A1. Offer 10% discounts			X				X	
A2. Offer 15% discounts						X		X
A3. Offer 20% discounts				X	X			
A4. Offer 70% discounts		X						
A5. Offer 80% discounts	X							

ADVANTAGES AND DISADVANTAGES

Advantages :-

- They are easier to draw.
- They provide a compact representation of the decision making process. A small table can be replace several pages of flow chart.
- Decision table can be changed according to the situation.
- It is also much easier to understand a particular path down one column than through several pages of the flow chart.
- Decision tables are based suited for calculating discount, commissions or inventory control procedures.

Disadvantages :-

- Decision table can not express the complete sequence of operations to solve a problem, It may be difficult for a programmer to translate a decision table directly into a computer program.
- When there are two many alternatives, decision table cannot list them all.
- Decision table does not depict the flow of logic for the solution to a given problem.