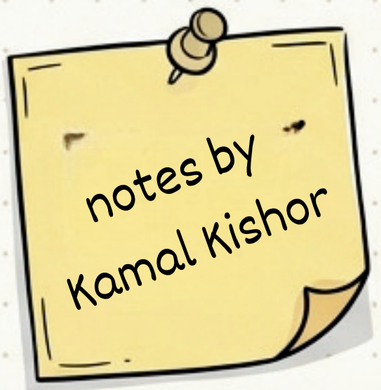
CSA 5009T – System Analysis and Design

(BCA – Detailed Notes)

Roadmap

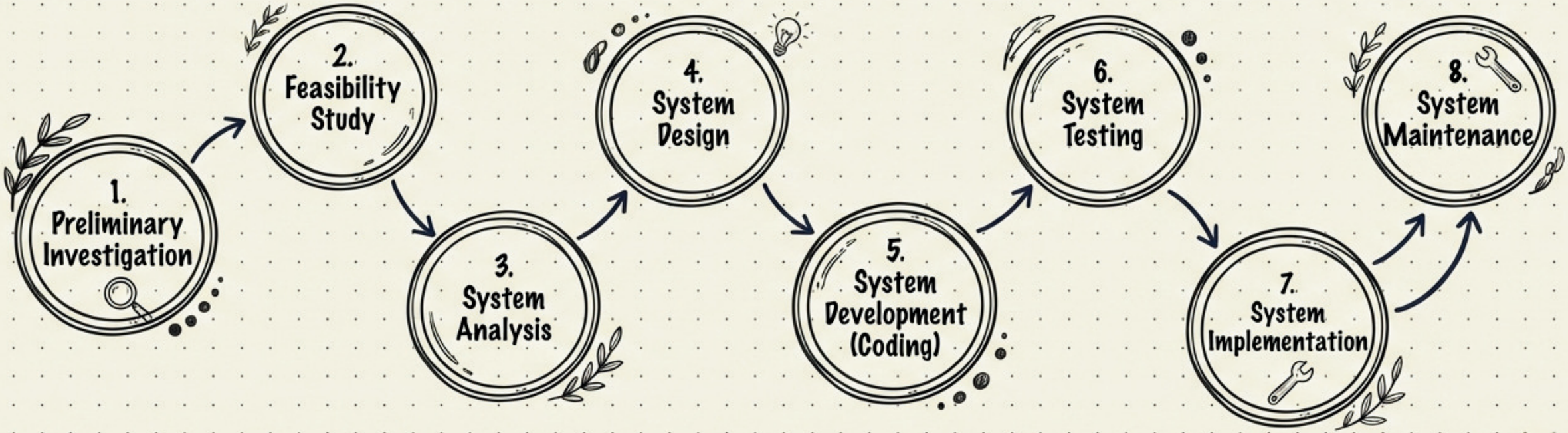
- ☐ System Development Life Cycle
- ☐ Initial Investigation
- ☐ Feasibility Study
- ☐ Cost/Benefit Analysis
- ☐ Tools of Structured Analysis
- ☐ Input/Output & Form Design
- ☐ Documentation Standards
- ☐ System Testing & Quality
- ☐ System Security



1. System Development Life Cycle (SDLC)

Meaning of SDLC

The System Development Life Cycle (SDLC) is a structured process used to develop, maintain, and replace information systems. It ensures that the system meets user requirements, is cost-effective, and works efficiently.



Importance & Initial Investigation

Importance of SDLC

- ✔ Reduces project risk
- ✔ Improves quality of system
- ✔ Proper planning and control
- ✔ Ensures user satisfaction

2. Initial Investigation

Initial investigation is the first step of system development, where the problem is identified and basic requirements are gathered.

Objectives:

- ➢ Identify user needs
- ➢ Understand existing system
- ➢ Determine scope of new system

Fact-Finding Techniques

Part of Initial Investigation

Interviews

Questionnaires

Observation

Document review

3. Feasibility Study

Feasibility study determines whether the proposed system is practical and worth implementing.

Types of Feasibility

1. Technical Feasibility

- Availability of hardware & software
- Technical skills of staff

2. Operational Feasibility

- Will users accept the system?
- Ease of operation

3. Economic Feasibility

- Cost vs benefits analysis
- Return on Investment (ROI)

Feasibility Report

A written document that includes:

Problem definition → Feasibility analysis → Final recommendation

---

4. Cost / Benefit Analysis

It compares the cost of developing the system with the expected benefits.

Types of Costs

- Development cost $
- Operational cost $
- Maintenance cost $

Types of Benefits

- Tangible (increased profit, reduced cost)
- Intangible (better service, customer satisfaction)

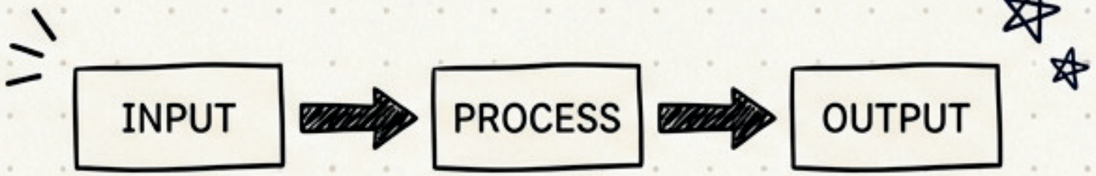
5. Tools of Structured Analysis (1/3)

1. **Logical and Physical Models**
   - Logical Model: What system does
   - Physical Model: How system works
2. **Context Diagram**
   - Highest level DFD
   - Shows system and external entities
3. **Data Flow Diagram (DFD)**
   - Graphical representation of data flow
   - Shows processes, data stores, inputs & outputs
4. **Data Dictionary**
   - Stores definition of data elements
   - Acts as reference for developers

5. Tools of Structured Analysis (2/3)

5. IPO Chart

- Input → Process → Output



6. HIPO Chart

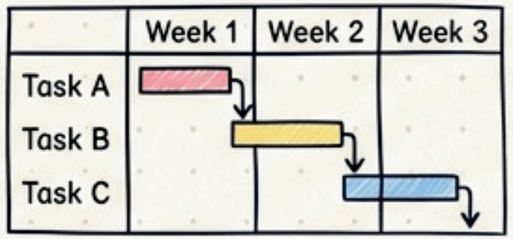
- Hierarchy plus IPO
- Shows system structure

7. Flowcharts

- Graphical representation of logic
- Uses symbols like oval, diamond, rectangle

8. Gantt Chart

- Project scheduling tool
- Shows tasks vs time



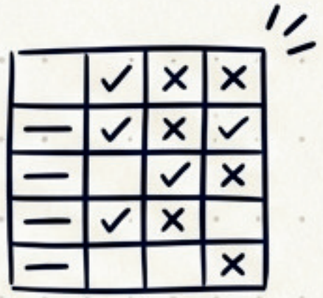
5. Tools of Structured Analysis (3/3)

9. Decision Tree

- Graphical decision-making tool

10. Decision Table

- Tabular representation of decision rules



11. Pseudo Code

- English-like representation of logic

12. Data Validation

- Ensures correctness of input data

6. Input / Output and Form Design

INPUT DESIGN

- Capturing correct data
- Reducing errors
- User-friendly input forms

Form Design Methodologies

- Simple layout
- Clear labels
- ✓ Consistent format

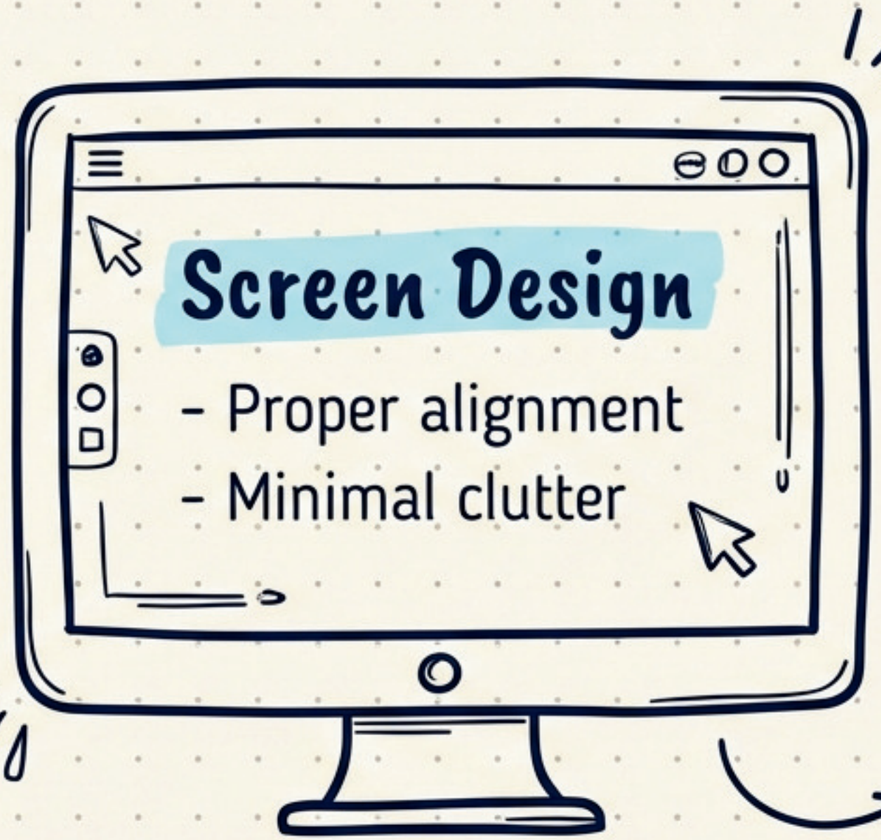
OUTPUT DESIGN

- Producing useful information
- Reports, receipts, summaries

Interface & Layout Design

Menu Design

- Logical grouping
- Easy navigation



Screen Design

- Proper alignment
- Minimal clutter

Layout Consideration

- Readability
- Consistency
- User comfort

7. Management and Documentation Standards

Types of Documentation

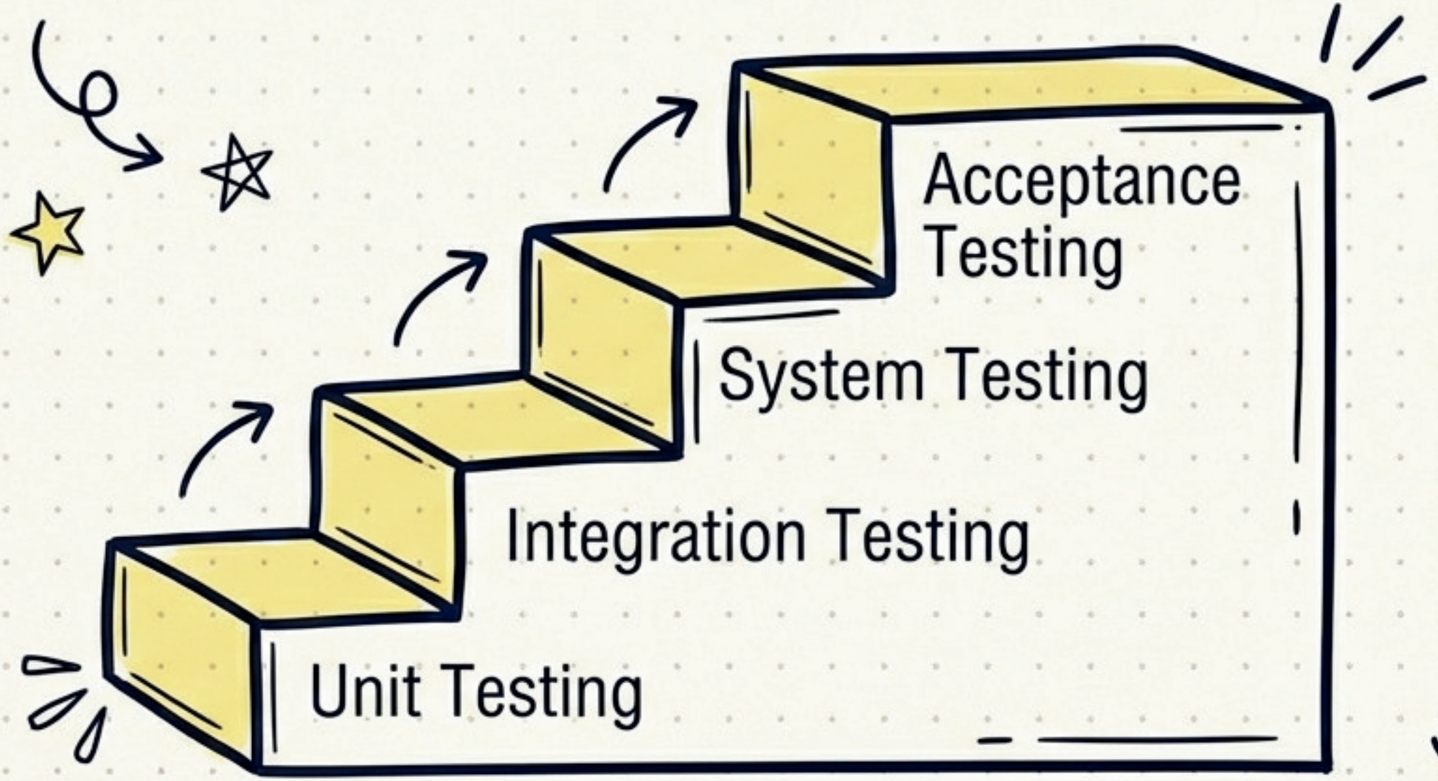
1. **User Manual** - Instructions for end users
2. **System Development Manual** - Overall system description
3. **Programming Manual** - Code logic & structure
4. **Programming Specifications** - Technical details
5. **Operator Manual** - System operation procedures

Importance of Documentation

- ✓ Easy maintenance
- ✓ Training users
- ✓ Future upgrades

8. System Testing and Quality Assurance

System Testing: Ensures system works as per requirements.

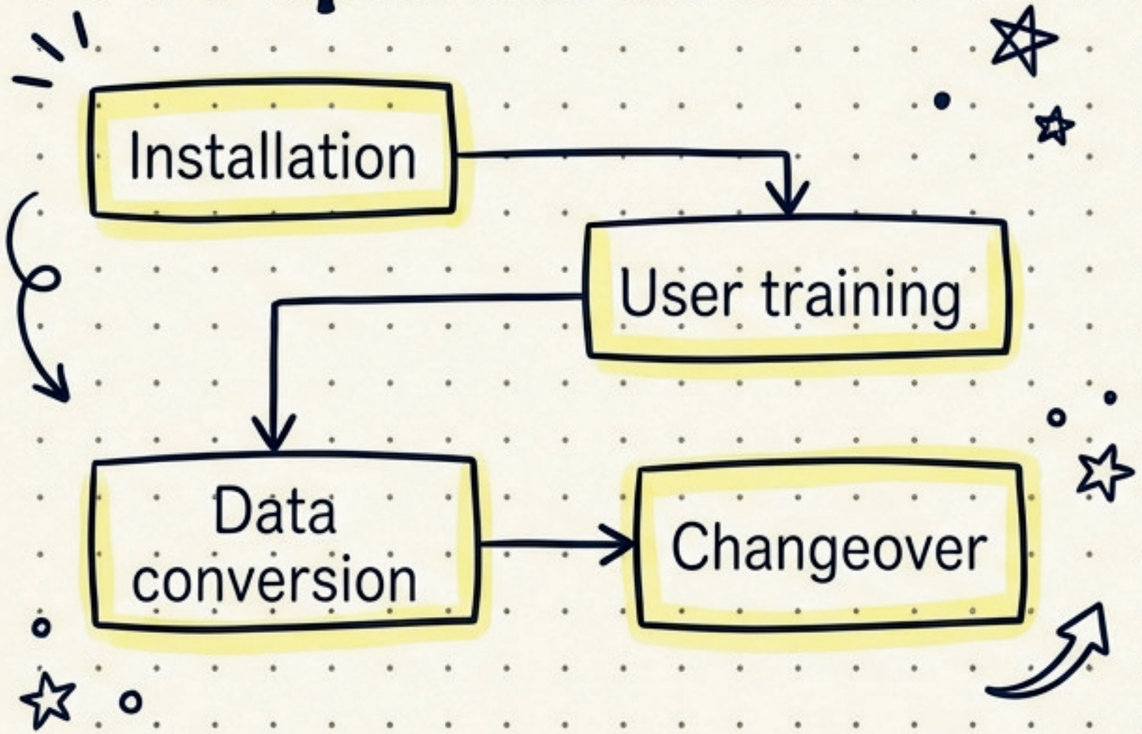


Quality Assurance (QA)

- Ensures software quality
- Prevents defects

Implementation & Maintenance

Steps in System Implementation



Software Maintenance

- Corrective maintenance
- Adaptive maintenance
- Perfective maintenance

9. System Security

Data Security

Protecting data from unauthorized access

Threats

- Virus
- Hacking
- Malware
- Data theft

Risk Analysis

- Identifying threats
- Assessing impact
- Minimizing risks

Disaster Recovery

- Backup strategies
- Recovery plans

Ethics

- User privacy
- Data confidentiality
- Responsible use of technology