

# SDLC

**Lecture # 4.5**  
**15, 16, 18 Feb**

Rubab Jaffar  
[rubab.jaffar@nu.edu.pk](mailto:rubab.jaffar@nu.edu.pk)

# Introduction to Software Engineering

## SE-110



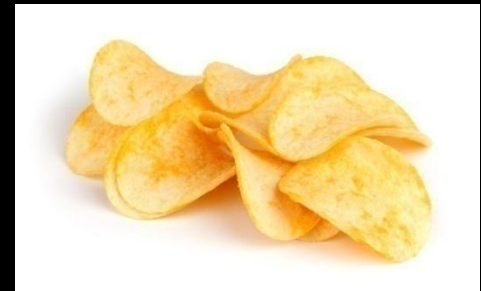
# Today's Outline

- Process and Product
- Software engineering Framework
- What is system?
- SDLC
- SDLC Phases

# Process

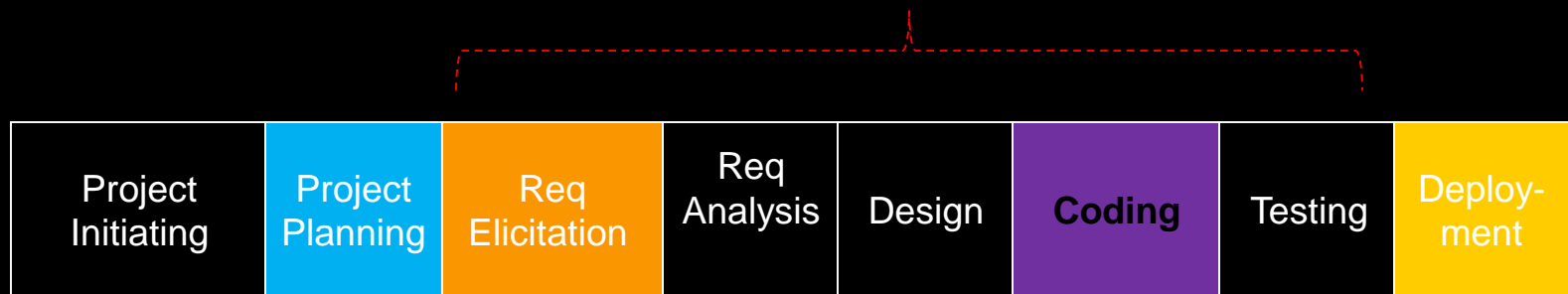
**Process:** A particular method, generally involving a number of steps.

Process for making *potato chips*



# Software Process

- **Software Process**: A set of steps, along with ordering constraints on execution, to produce software with desired outcome.
  - Many types of **activities** performed by **different teams**
  - Software process is comprising of many component processes



# Process

What we should do, to develop a  
*good quality* software ?

What are the attributes of a *good process* ?  
Visible, Repeatable, Measureable

Can you give examples of *Technical* and  
*Managerial*  
problems in software development process ?

Project fails due to **Managerial** problems

# Product

- **Products** are outcomes of executing a **process** for a project.
- Does Process quality and Product quality has any relation ?
- Software development life cycle (SDLC), is a structure imposed on the development of a software product.
- Software Engineering says if you follow the process the output is predicable and repeatable no matter who does that.
- Software Engineering focuses on **process**. Mature processes will help achieve **project objectives** of high Quality Product.

# Software Engineering Framework

# Software Engineering Framework

- What is **framework** and why we need framework?
  - Framework means; set of rules to be followed.
- What are **those rules**? Those rules have been **adopted** by organizations that produce good results.
- Experts convert those rules into a framework to be used by every organization with respect to their needs.
- **Example:** Framework for Requirements Development



# Software Engineering Framework

What

Definition

How

Development

Change

Maintenance

Umbrella / parallel activities

- 
- Quality Assurance
  - Configuration Mg'mt
  - Project Monitoring
  - Measurement

## Definition Phase

- *Definition* phase focuses on *what* (is required).
- During definition, SW-development-Team and user attempts to identify the following questions:
  - What is **need** (or problem)?
  - What *features* are required?
  - What interfaces are to be established?
  - Any budget or technical **constraints** ?
  - What is *success criteria* ?.

## Development Phase

- *Development* phase focuses on the **how**.
- During development, the SW-development-Team attempts to define how:
  - How database would be designed
  - How software *architectures* would be designed
  - How the design will translate into programming language
  - How *testing* will be performed

## Development Phase

- Methods applied during development phase, *will vary* (depending on the SDLC) but the three steps will occur in some form:
- *Design*: Design translate the requirements into some graphical or tabular representations.
- *Coding*: Design is then translated into programming language.
- *Testing*: The executable code must be tested to uncover errors.

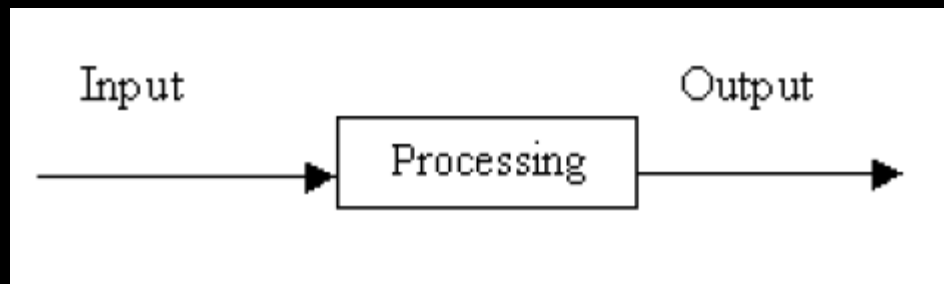
# Maintenance Phase

- Maintenance phase focuses on changes that associated with
  - Error Correction (Corrective)
  - Platform Adaptations required (Adaptive)
  - Enhancement due to change (Perfective)
  - The work carried out order to avoid any breakdown or malfunction (Preventive)

# SDLC

# System

- A collection of components that work together to realize some objective forms a system.
- Basically there are three major components in every system namely input, processing and output.



# System

A System can be a Application program or it can be an Information System.

- Computer App: is an application program (app for short) is a computer program designed to perform a **group of coordinated functions**, tasks, or activities for the benefit of the user.
- Information System: is software that helps you organize and analyze data. This makes it possible to answer questions and solve problems relevant to the mission of an organization.

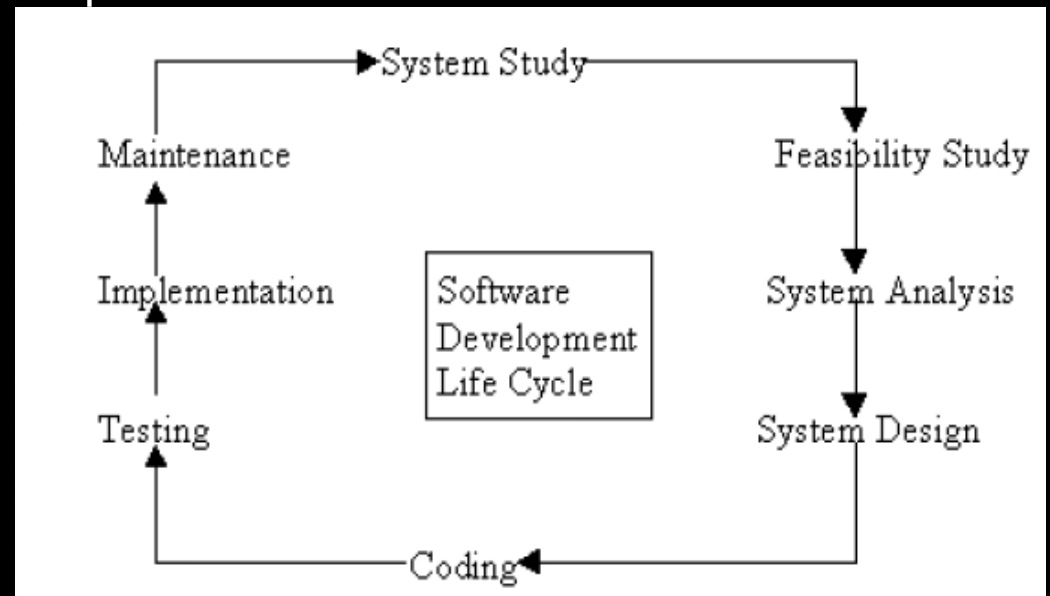




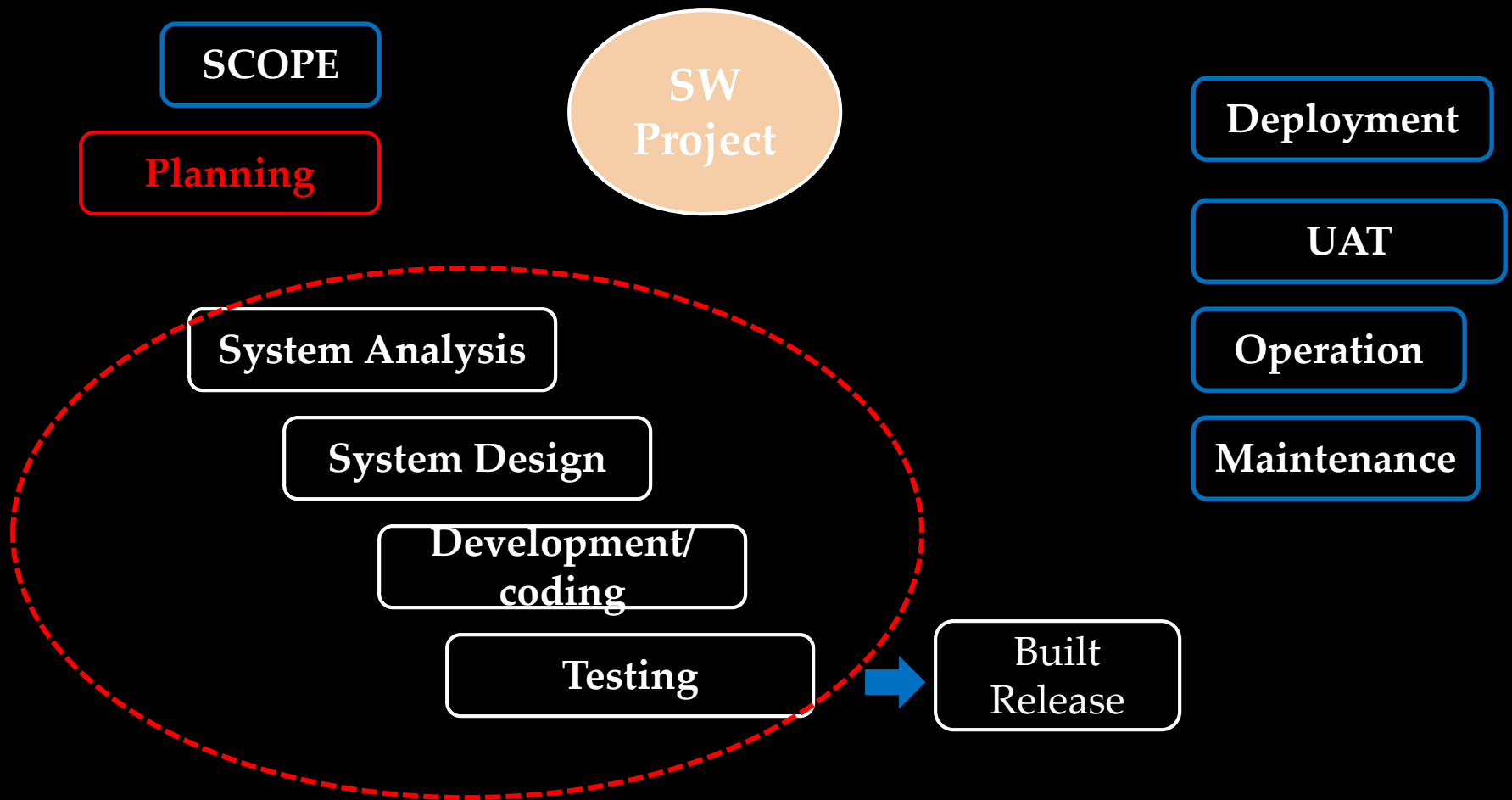
# SDLC

- Software Development Life Cycle (SDLC) is a process used by the software industry to design, develop and test high quality software.
- It consists of a detailed plan describing how to develop and maintain software.
- SDLC consists of many activities/ phases.
- Following are the major phases of SDLC.

- System study
- Feasibility study
- System analysis
- System design
- Coding
- Testing
- Built release
- Maintenance

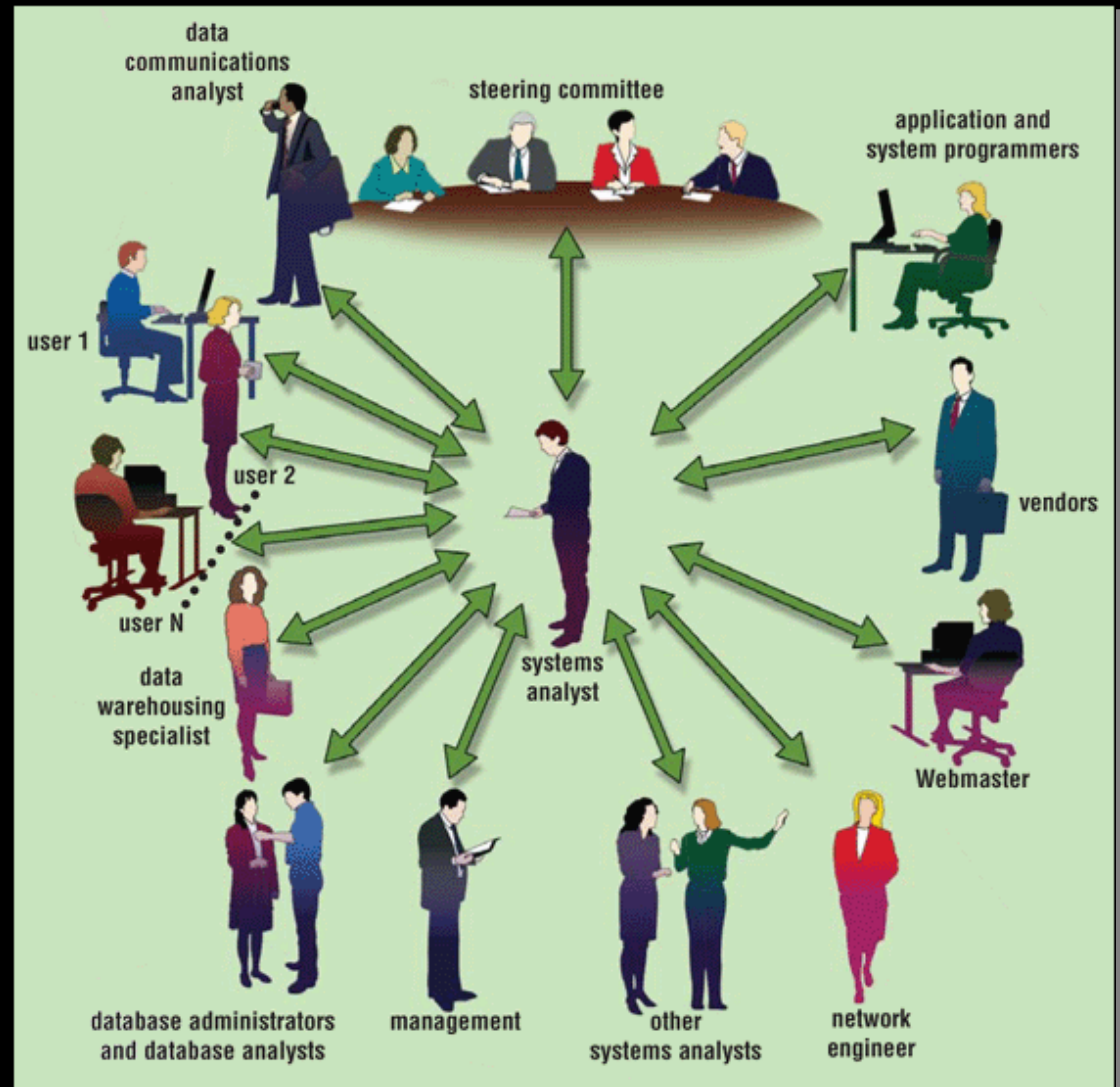


# Software Development Life Cycle



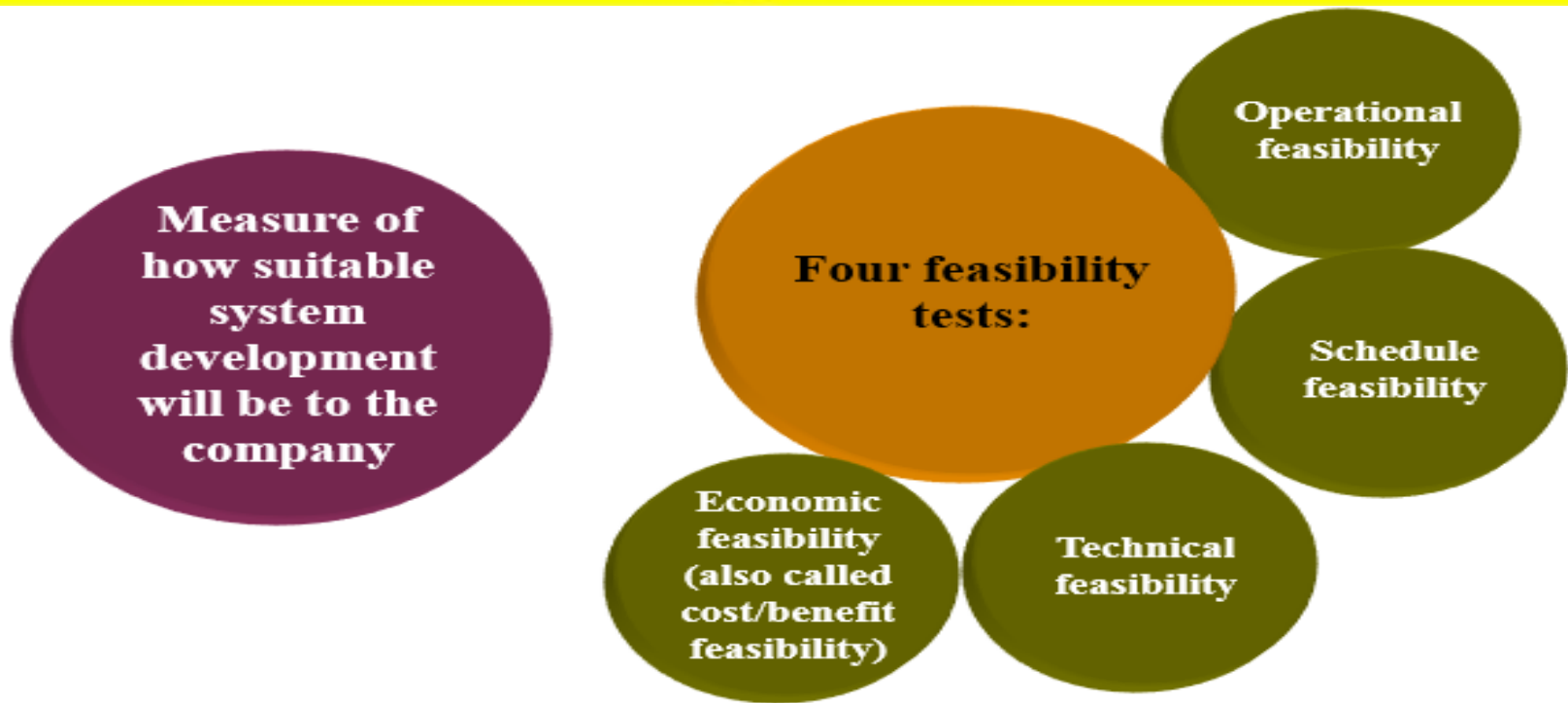
# The System Development Life Cycle

Who participates in the system development life cycle?



# Feasibility Study

***An analysis of a proposed project to determine whether it is feasible and should go ahead.***





# Stage 2: Feasibility Study

- A feasibility study precedes the decision to begin a project. It is an assessment of the practicality of a proposed system.
- A feasibility study leads to a decision: go or no-go.
- Output: FSR

# Five common factors (TELOS)

1. **Technology and system feasibility**
2. **Economic feasibility**
3. **Legal feasibility**
4. **Operational feasibility**
5. **Schedule feasibility**

# 1. Technical Feasibility

- This assessment is based on an outline design of system requirements, to determine whether the company has the technical expertise to handle completion of the project.
  - Is the project possible with current technology?
  - What technical risk is there?
  - Availability of the technology:
    - Is it available locally?
    - Can it be obtained?
    - Will it be compatible with other systems?



# 2. Economic Feasibility

- Economic analysis is the most frequently used method for evaluating the effectiveness of a new system. More commonly known as cost/benefit analysis, the procedure is to determine the benefits and savings that are expected from a candidate system and compare them with costs. If benefits outweigh costs, then the.
  - What are the development and operational costs?
  - Are the benefits worth the costs?

# 3. Legal Feasibility

- Determines whether the proposed system conflicts with legal requirements, e.g. a data processing system must comply with the local Data Protection Acts.

# 4. Operational Feasibility

- Is a measure of how well a proposed system solves the problems, and takes advantages of the opportunities identified during scope definition and how it satisfies the requirements identified in the requirements analysis phase of system development.
  - If the system is developed, will it be used?
  - Human and social issues...
    - Potential labour objections?
    - Manager resistance?
    - Organizational conflicts and policies?
    - Social acceptability?

# 5. Schedule Feasibility

- A project will fail if it takes too long to be completed before it is useful. Typically this means estimating how long the system will take to develop, and if it can be completed in a given time .
- Some projects are initiated with specific deadlines. You need to determine whether the deadlines are mandatory or desirable.
  - Is it possible to build a solution in time to be useful?
    - What are the consequences of delay?
    - Any constraints on the schedule?
    - Can these constraints be met?

# Other feasibility factors

- **Market and real estate feasibility**
- **Resource feasibility**
- **Cultural feasibility**

# Stage 3: System Analysis

- Forms the **basis of agreement** between user and developer. System analysis establish the system's services, constraints and goals by consultation with users.
- It is the study of **specifications, operations and relationships** with in the system and outside the system.
- Specifies what not how. (Hard task)
- Define the **boundary** of the new system keeping in view the problems and the new requirement.
- **Output:** is the Software Requirements Specification (SRS) document.

# Stage 4: System Design

- A major step in moving from *problem* to *solution*.
- Based on system analysis, the new system must be designed.
- *Two main tasks*
  - *General design:* (preliminary design) components and connectors that should be there in the system
  - *Detailed design:* (*Detailed Design*) logic of modules

*Output: SDS(system Design Specification)*

# Tools and Techniques for Designing

**1. Flow Chart**

**2. Data Flow Diagram**

**3. Data Dictionary**

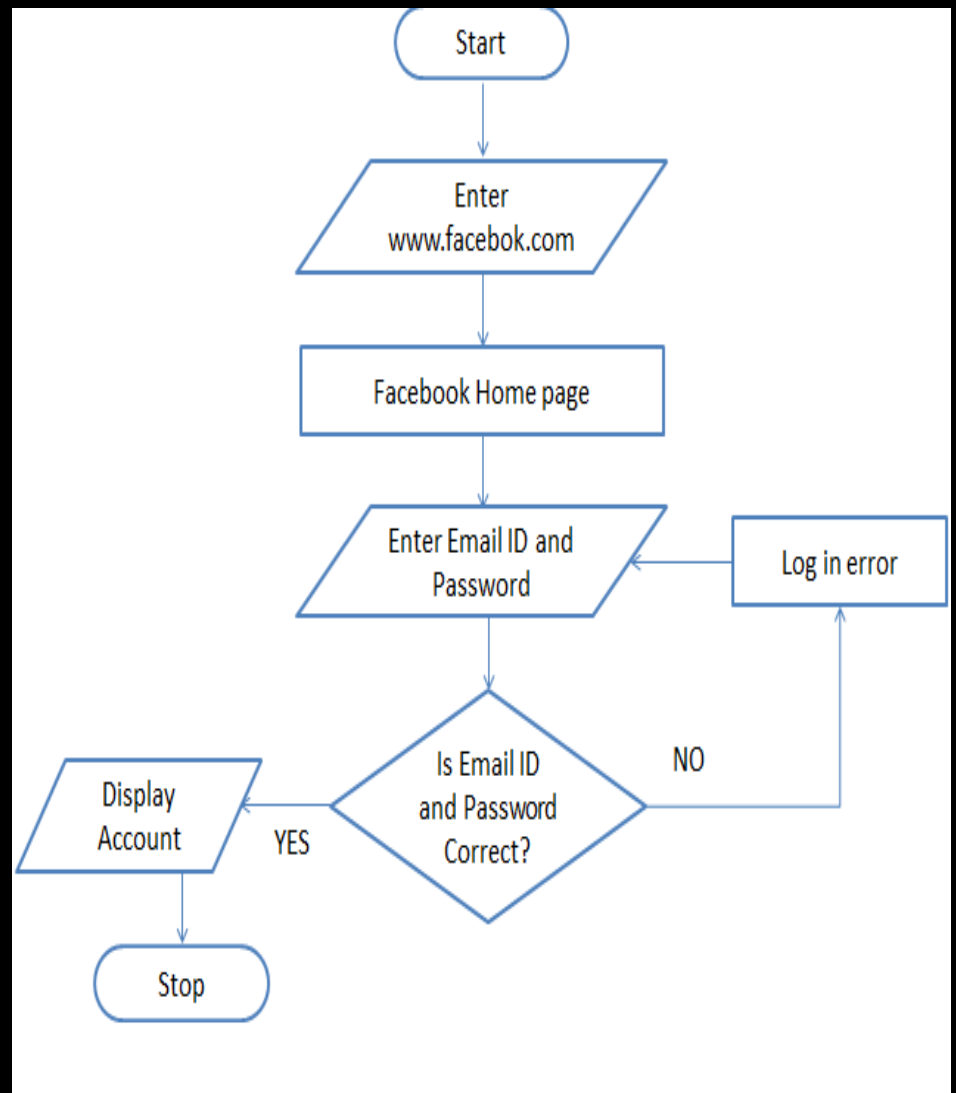
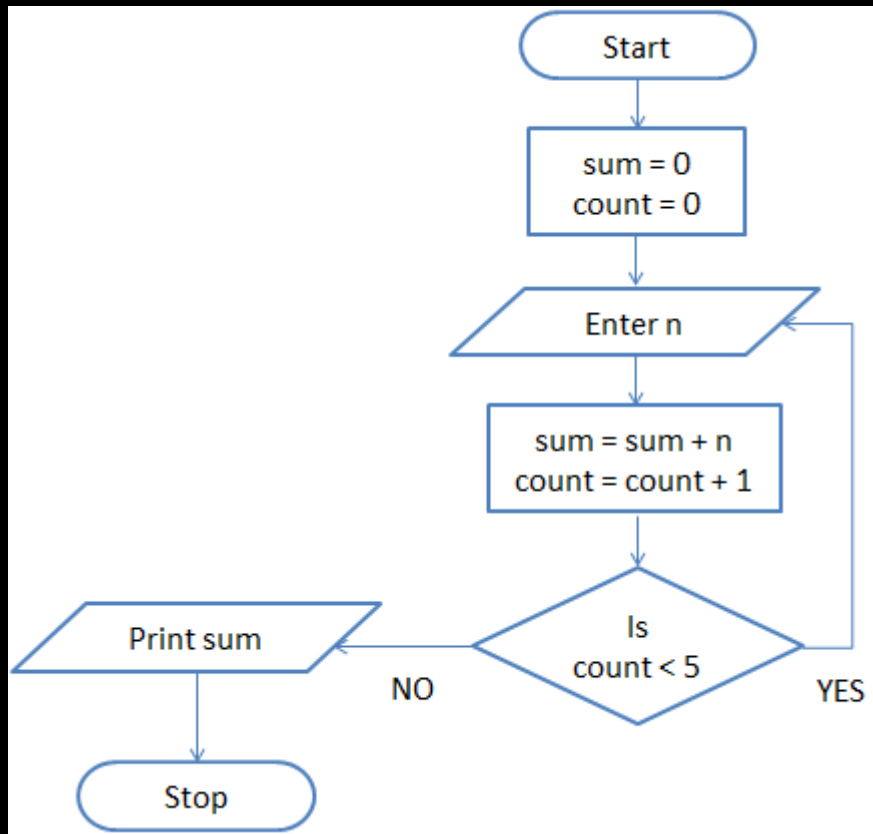
**4. Structured English**

**5. Decision Tables**



# Flow Chart

- Draw a flowchart to find the sum of 5 numbers
- Draw a flowchart to log in to face book account



# Decision Tables

- 1) List all actions that can be associated with a specific procedure (or module)
- 2) List all conditions (or decisions made) during execution of the procedure

List of Conditions	Combination Of Conditions
List of Actions	The Corresponding Set of Actions

Rules						
Conditions	1	2	3	4	5	6
Regular customer	T	T				
Silver customer			T	T		
Gold customer					T	T
Special discount	F	T	F	T	F	T
Actions						
No discount	✓					
Apply 8 percent discount			✓	✓		
Apply 15 percent discount					✓	✓
Apply additional x percent discount		✓		✓		✓

# Decision Tables

## Activity Task:

For the SafeHome problem, assume that the system is connected to the network. Write a decision table based on the following facts;-

The homeowner is supposed to get an E-Mail if and only if noise level goes beyond a level. If the temperature goes beyond a level not only homeowner will be getting an E-Mail, but also alarm has to be switched on along with a telephone call to a local police station. Same thing goes by for the fact when pressure goes beyond certain level.

# Stage 5: Coding

- Converts design into **code** in specific language
- **Goal:** Implement the design with simple and easy to understand code
- programs must be **modular** in nature. This helps in fast development, maintenance and future changes, if required.
- Coding phase affects both **testing** and **maintenance**.
  - Well written code reduces testing and maintenance effort.
- **Output:** is source-code.

# Stage 6: Testing

- **Defects** are introduced in each phase
  - Must be found and removed to achieve high quality
- Software testing is a process of analyzing software for the purpose of finding bugs.
- Using test data, following test runs are carried out
  - **Unit test**: performed by the respective developers on the individual units of source code to ensure that the individual parts are correct in terms of requirements and functionality.
  - **System test**: done after unit test. System testing tests the system as a whole. Actual output of the system is matched with the expected outputs. Errors are identified and fixed.
  - **User acceptance testing (UAT)** – determines if the system satisfies the business requirements
- **Outputs:** are
  - Test plans/results
  - Final tested (reliable) code

# Stage 7: Built Release

- After UAT, deployment phase begins.
- Final phase of SDLC, puts the product into production
- All programs of system are loaded onto the user's computer.
- Then training of user starts including
  - how to execute the package
  - how to enter the data
  - how to process data

# Built Release Strategies

- **Parallel run:** computerized & manual systems are executed in parallel.
- Advantages of Parallel run:
  - Manual results comparison with the computerized one.
  - Failure of the computerized system at the early stage, does not affect the working of the organization.
- **Pilot run:** New system is installed in parts. Some part of the new system is installed first and executed successfully for considerable time period.
- **Advantages:**
  - When results are found satisfactory then only other parts are implemented.
  - This strategy builds the confidence and the errors are traced easily.



# Stage 8: Maintenance

- Maintenance phase focuses on **changes** that associated with
  - Error Correction
  - Platform Adaptations required
  - Enhancement due to change
  - Re-engineering
- Maintenance is required to:
  - eliminate errors in the system during its working life
  - tune the system to any variations in its working environment.
- System Review: is necessary from time to time for:
  - knowing the full capabilities of the system
  - knowing the required changes or the additional requirements
  - studying the performance
- Major change during the review:
  - If a major change to a system is needed, a new project may have to be set up to carry out the change.
  - New project will then proceed through all above life cycle phases.



That is all