Day 1

Software Development
&
Project Planning

Workshop By:

Dr. Vishal Bharvesh

Introduction to Projects

- A project is a structured task with a specific goal, timeline, and output.
- In Computer Science, it means developing software solutions.

Major vs Minor Projects

- Minor: Short duration, basic model
- Major: Long duration, complete system with full implementation

Importance of Projects

- Builds practical and real-world knowledge.
- Improves problem-solving and logical thinking.
- Develops technical and programming skills.
- Enhances research and analytical ability.
- Increases confidence in handling real systems.
- Helps in placements and job interviews.
- Creates a strong portfolio / resume value.
- Encourages innovation and creativity.
- Improves teamwork and communication skills.
- Provides hands-on experience of SDLC.

Types of Projects

- Web Applications
- Mobile Apps
- AI/ML Systems
- ERP Systems
- Desktop Applications

Project Selection

- Identify a real-world problem
- Check feasibility (time, cost, skills)
- Ensure innovation or uniqueness
- Availability of data and resources
- Match with your interest and career goals
- Define clear scope
- Possibility of future enhancement
- Avoid copy-paste or repeated projects

Problem Statement

- A problem statement clearly defines the issue your project will solve
- It explains:
  - Current situation
  - Existing problem
  - Impact of the problem
- Should be:
  - Clear
  - Specific
  - Measurable
  - Realistic

Project Objectives

- Define what the project aims to achieve
- Should be:
  - Clear
  - Specific
  - Measurable
  - Achievable
  - Time-bound
- Types of Objectives:
  - Primary Objective
  - Secondary Objectives
  - Must align with problem statement
  - Guide the entire development process

Project Scope

- Defines boundaries.
- What is included and excluded.

Project Scope defines the clear boundaries of a project by specifying what is included and what is excluded.

It outlines the features, functionalities, and deliverables that the system will provide, along with its limitations.

A well-defined scope helps in avoiding confusion, scope creep, and unnecessary expansion of work. It ensures that the project remains focused, manageable, and achievable within the given time and resources.

Project scope also helps in setting clear expectations for stakeholders and team members. By defining scope early, students can plan development effectively and ensure successful and timely completion of their project.

Feasibility Study

- Technical, Economic, Time feasibility

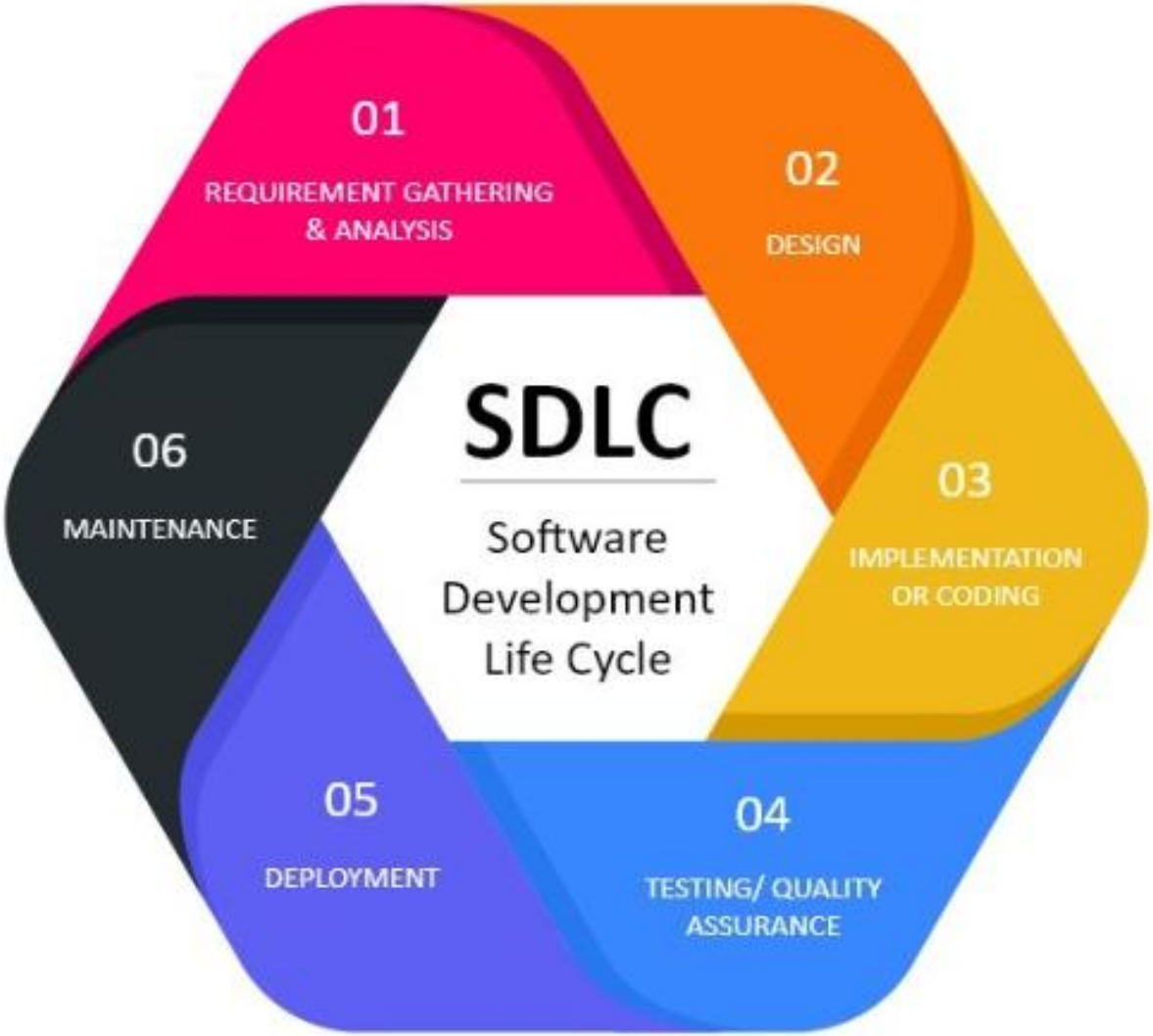
The purpose of a **Feasibility Study** is to evaluate whether a project idea is practical, achievable, and worth implementing before starting development.

It helps in identifying potential challenges, required resources, and possible risks at an early stage.

- ✓ By analyzing **technical feasibility**, we check if the necessary tools and skills are available.
- ✓ Through **economic feasibility**, we ensure the project is cost-effective.
- ✓ With **time feasibility**, we confirm that the project can be completed within the given deadline.

Overall, it prevents failure, saves time and effort, and ensures that the selected project is realistic and successfully executable.

SDLC Overview



Planning Phase

- Define goals
- Identify resources
- Create timeline

The planning phase answers:

- What to build?
- Why to build?
- How to start?
- What resources are needed?

Requirement Analysis

In this phase, requirements are collected from users, clients, or stakeholders through discussions, interviews, surveys, or observation.

The goal is to identify both functional requirements (what the system should perform, such as login, data processing, reports) and non-functional requirements (performance, security, usability, reliability).

All the gathered requirements are then analyzed, refined, and documented in a structured format called the Software Requirement Specification (SRS). This document acts as a blueprint for the entire project.

Design Phase

In this phase, the team converts requirements into a detailed blueprint that defines how the system will be built. It includes two levels:

1. **High-Level Design (HLD):**
   Describes the overall system architecture, modules, data flow, and technology stack.
2. **Low-Level Design (LLD):**
   Provides detailed design of each module, including algorithms, database structure, and logic.

- The design phase also covers:
  - Database design (tables, relationships)
  - User Interface (UI) design (screens, layouts)
  - System architecture (how components interact)

- In simple terms, this phase answers:
  - How will the system work?
  - How will different parts connect?

Development Phase

In this phase, developers start building the software based on the design documents (HLD and LLD). Each module of the system is developed step by step using selected programming languages, frameworks, and tools.

The work is usually divided into smaller parts (modules), and developers follow a modular approach to make the system easy to manage and maintain. During development, coding standards, naming conventions, and proper documentation are followed to ensure quality.

Developers also perform basic testing (unit testing) to check whether each module is working correctly before integration.

This phase is important because:

- It converts design into a working system
- It requires strong logical and programming skills
- It directly impacts system performance and quality

Testing Phase

Bug fixing and quality assurance

The **Testing Phase** of the **Software Development Life Cycle (SDLC)** ensures that the developed system is accurate, reliable, and free from errors.

In this phase, the software is carefully evaluated using different testing methods such as unit testing, integration testing, and system testing.

The goal is to identify bugs, verify functionality, and ensure the system meets all specified requirements. Test cases are executed to check performance, security, and usability.

Any defects found are reported and fixed before deployment. This phase is crucial because it improves software quality, enhances user satisfaction, and reduces the risk of failure in real-world usage.

Deployment & Maintenance

The Deployment and Maintenance Phase of the Software Development Life Cycle (SDLC) is the final stage where the developed system is delivered and continuously improved.

In the deployment phase, the software is installed on a live server or user environment, making it accessible to end users.

Proper configuration, data setup, and user training may also be conducted. After deployment, the maintenance phase begins, where the system is monitored for performance, errors, and user feedback.

Necessary updates, bug fixes, and enhancements are made over time. This phase ensures the software remains efficient, secure, and relevant to changing user needs.

Planning Tools

- Flowcharts
  - A flowchart is a graphical representation of a process using symbols and arrows to show the step-by-step flow of logic.
- Pseudocode
  - Pseudocode is a way of writing program logic in simple English-like statements without following strict programming syntax.
- Gantt Chart
  - A Gantt Chart is a visual timeline used to plan and track project tasks over a specific period.

Common Mistakes

Common mistakes in **Software Development & Project Planning** include having no clear problem definition, which leads to poor requirements, unclear objectives, and weak system design.

Without a well-defined problem, the entire development process becomes directionless. Poor time management is another critical issue, where improper planning, lack of scheduling, and missed milestones delay development, testing, and delivery. It affects overall project quality and team coordination.

Additionally, copying projects without understanding reduces practical learning, innovation, and problem-solving ability. These mistakes result in inefficient systems, poor performance, and difficulty during evaluation, and must be avoided for successful project execution.

Summary

- Project basics
- Topic selection
- SDLC
- Planning tools

End of Day 1