[55958- Aman Goswami, Documentation-1 Date-12 Dec 2025]

Software Development Life Cycle

SDLC – Software Development Lifecycle

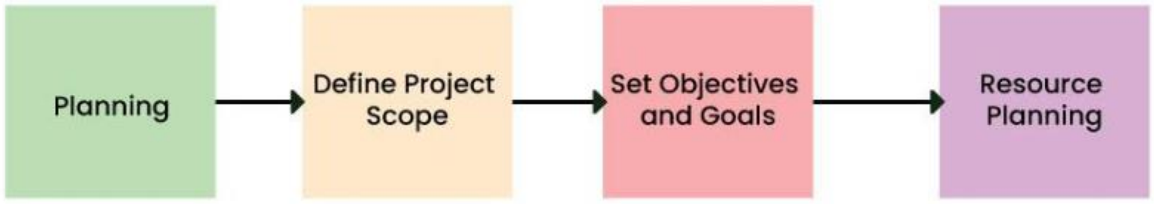
The Software Development Lifecycle (SDLC) is a structured process that development teams follow to build high-quality software. It ensures that every stage—from understanding the requirements to deploying and maintaining the application—is executed in a planned and organised manner.

1. Requirement Document

This is the starting point of any project.
Here, the Product Owner (PO) or client shares:

- Detailed requirements in text form
- Wireframes or rough sketches of the screens
- A high-level abstract explaining what the product should achieve

Goal: Understand "What needs to be built?" clearly before development begins.



2. Technical Document

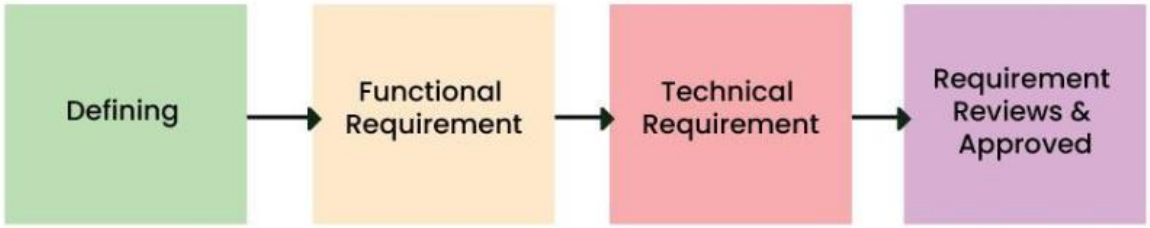
Once the requirements are finalised, the Technical Lead (TL) or Architect prepares a detailed technical design document.

This includes:

- Number of classes and their responsibilities
- Interfaces, inheritance structure
- Application modules and their interactions
- Components and internal architecture

The document usually goes through **multiple reviews** to ensure the design is scalable, maintainable, and aligned with requirements.

Goal: Create a strong blueprint of "How will it be built?"



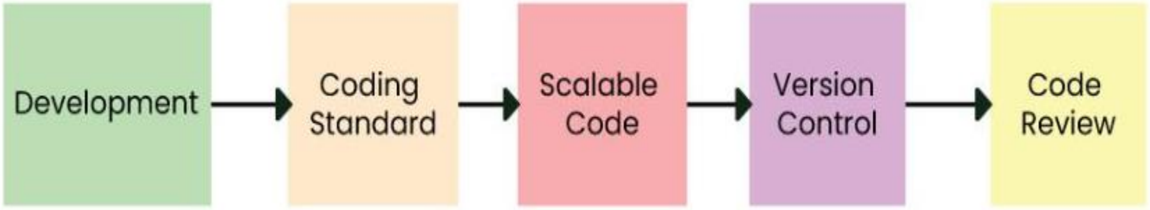
3. Coding Phase

After the design is approved, developers (SSEs, SEs) begin writing the actual code.

Key activities include:

- Building individual features and modules
- Writing clean, testable, and optimised code
- Following coding standards and best practices
- Performing peer reviews to maintain code quality

Goal: Convert design into a working software product.



4. Testing Phase

Once development is completed, the code is moved to the **QA Server**.

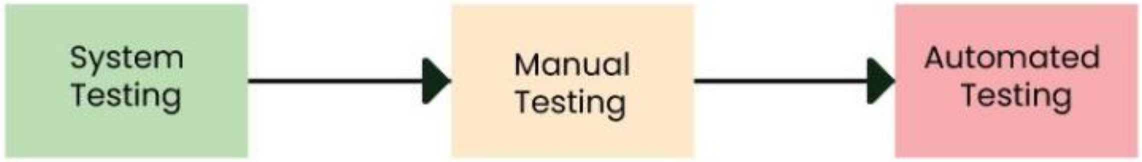
QA Engineers perform:

- Functional testing
- Integration testing

- Usability testing

Whenever they find issues, they **raise bugs** so developers can fix them.

Goal: Ensure the application works correctly and meets requirements.



5. Staging Environment

After QA approves the build, it moves to the **Staging Environment**.

Here, more advanced testing is done:

- Regression testing (to ensure old features still work)
- Load testing (to check performance under heavy traffic)
- Automation test cases execution

This environment is very close to production and helps identify any last-minute issues.

6. UAT Environment (User Acceptance Testing)

In this phase, the client or end users test the application themselves.

They verify:

- Whether the system behaves as expected
- If it solves their real business problems
- Any improvements they want before release

Only after UAT approval, the product can move to the next stage.

7. PPE — Pre-Production Environment

This is a mirror copy of the live production system.

Its purpose:

- To perform a final round of checks

- Validate deployment scripts
- Ensure no technical gaps remain

It is the last safety check before going live.

8. PE — Production Environment (Go Live)

The final approved version is deployed to the **Production Environment**, where real users can access the system.

This is known as the **Go Live** stage.

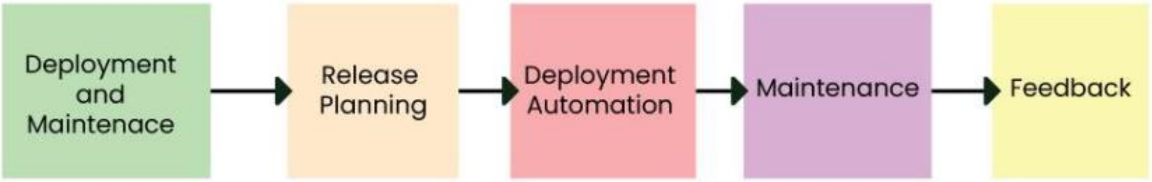
Goal: Make the application available to actual users.

9. Bug Fixing & Maintenance

Even after the software goes live, the team continuously maintains it by:

- Fixing reported issues
- Releasing patches and performance improvements
- Adding new features based on user feedback

Maintenance ensures that the software remains stable, secure, and up to date.


Deployment and Maintenace
Release Planning
Deployment Automation
Maintenance
Feedback