Software Development Life Cycle

SDLC

Agenda:

- Introduction
- SDLC Overview
- SDLC Phases
- SDLC Flow
- SDLC Model
- Waterfall Model
- Agile Model
- Conclusion

Introduction

Introduction

The SDLC is a *framework* that describes the activities performed at each stage of a software development project.

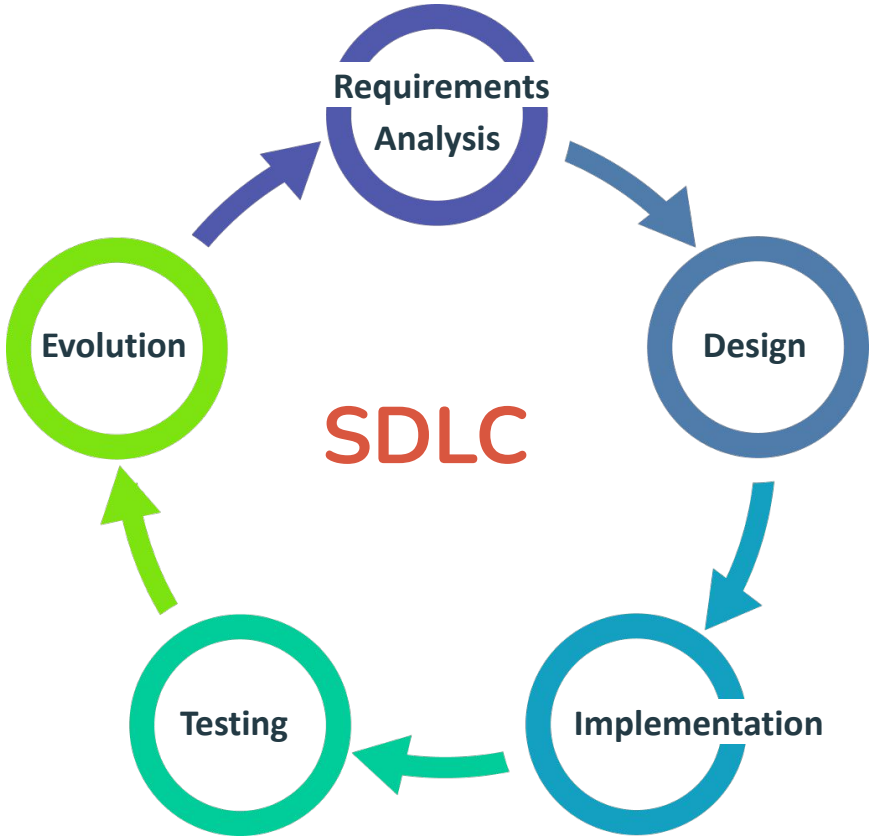
SDLC process is used by the software industry to design, develop and test high quality software. It aims to produce *the quality software that meets or exceeds customer expectations, reaches completion within time and budget.*

ISO/IEC 12207 is an international standard for software life-cycle processes. It aims to be the standard that defines all the tasks required for developing and maintaining software.

Software Engineering Process Technology Company, (SEPT) is a firm specializing in meeting the *software process standards information* needs of the professional community, particularly concerning ISO/IEC 12207.

SDLC phases

1. Planning and Requirements Analysis
2. Defining Requirements
3. Designing the Software
4. Building or Developing the Software
5. Testing the Software
6. Deployment and Maintenance



Planning & Requirement Analysis

Requirement analysis is the most important and fundamental stage in SDLC.

It is performed by the senior members of the team with inputs from all the *stakeholders* and *domain experts* or *SME*s in the industry.

Planning for the *quality assurance requirements* and *identification of the risks associated with the project* is also done at this stage. SDLC by Manohar Prasad

Requirements Analysis

- Business Requirements
- Stakeholder Requirements
- Solution Requirements
  - Functional Requirements
  - Non-functional Requirements
- Transition Requirements

Defining Requirements

Once the requirement analysis is done the next step is to clearly define and document the software requirements and get them approved from the project stakeholders.

This is done through ‘SRS’ – **Software Requirement Specification** document which consists of all the product requirements to be designed and developed during the project life cycle.

Defining Requirements

- Enterprise Analysis
- Business Analysis Planning & Monitoring
- Elicitation
- Requirements Analysis
- Requirements Management & Communication
- Solution Assessment & Validation

Designing the Software

Based on the requirements specified in SRS, usually more than one design approach for the product architecture is proposed and documented in a DDS - **Design Document Specification**.

This DDS is reviewed by all the stakeholders and based on various parameters as risk assessment, design modularity , budget and time constraints , the best design approach is selected for the software.

Developing the Software

In this stage of SDLC the actual development starts and the product is built. The programming code is generated as per DDS during this stage.

Developers have to follow the coding guidelines defined by their organization and programming tools like compilers, interpreters, debuggers etc are used to generate and implement the code.

Testing the Software

This stage is usually a subset of all the stages as in the modern SDLC models, the testing activities are mostly involved in all the stages of SDLC.

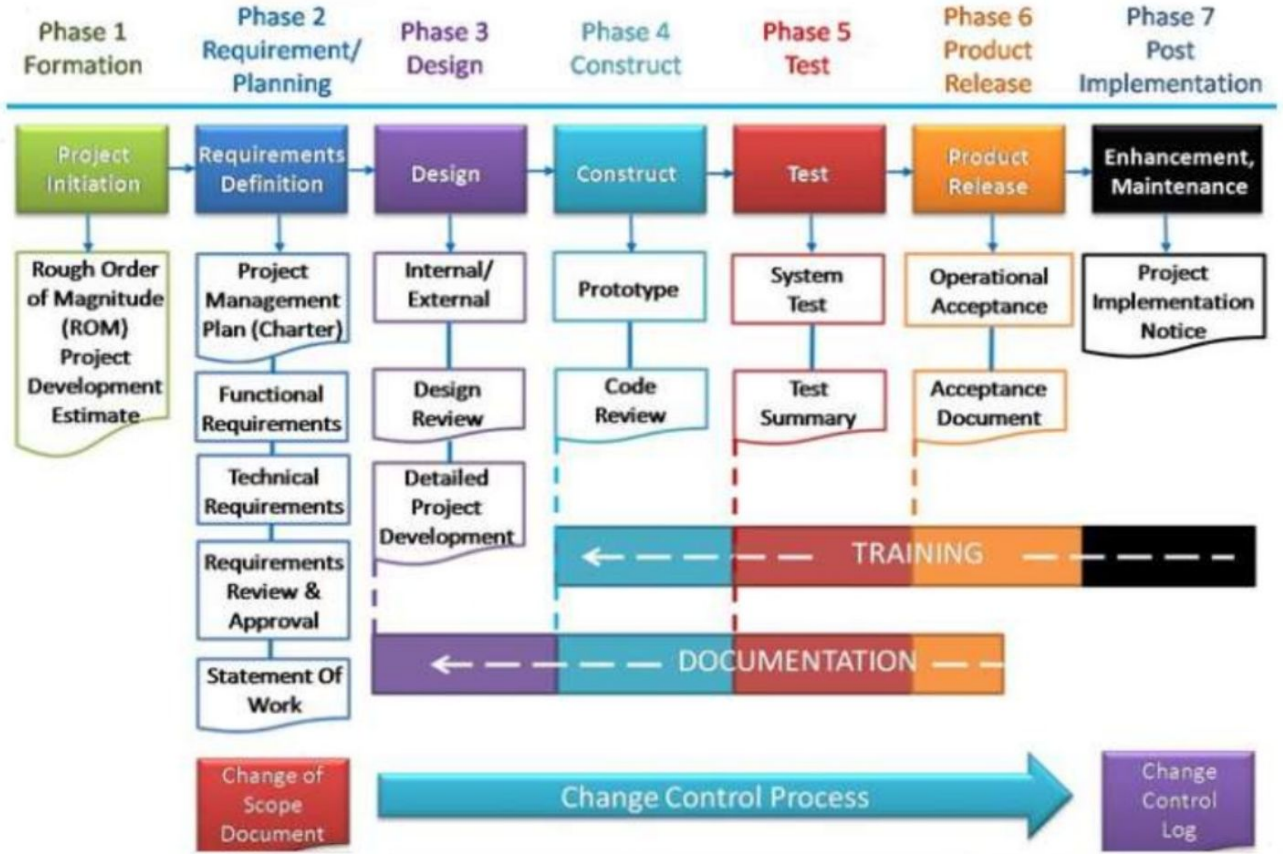
However this stage refers to the testing only that stage of the software where defects are reported, tracked, fixed and retested, until the software reaches the quality standards defined in the SRS.

Deployment and Maintenance

Once the software is tested and no bugs or errors are reported then it is deployed.

Then based on the feedback, the software may be released as it is or with suggested enhancements in the target segment.

After the software is deployed then its maintenance starts.


Phase 1
Formation
Phase 2
Requirement/
Planning
Phase 3
Design
Phase 4
Construct
Phase 5
Test
Phase 6
Product
Release
Phase 7
Post
Implementation
Project Initiation
Requirements Definition
Design
Construct
Test
Product Release
Enhancement, Maintenance
Rough Order of Magnitude (ROM) Project Development Estimate
Project Management Plan (Charter)
Functional Requirements
Technical Requirements
Requirements Review & Approval
Statement Of Work
Internal/ External
Design Review
Detailed Project Development
Prototype
Code Review
System Test
Test Summary
Operational Acceptance
Acceptance Document
Project Implementation Notice
TRAINING
DOCUMENTATION
Change of Scope Document
Change Control Process
Change Control Log

SDLC Models

To help understand and implement the SDLC phases various SDLC models have been created by software development experts, universities, and standards organizations.

Reasons for Using SDLC Models

- Provides the base for project planning, estimating & scheduling.
- Provides framework for standard set of terminologies, activities & deliverables.
- Provides mechanism for project tracking & control.
- Increases visibility of project progress to all stakeholders.

Advantages of Choosing an Appropriate SDLC

- Increased development speed
- Increased product quality
- Improved tracking & control
- Improved client relations
- Decreased project risk
- Decreased project management overhead

SDLC Models

- *Waterfall Model*
- Iterative Model
- Spiral Model
- *Agile Model*
- V – Model
- Big Bang Model

Conclusion

- Project Management – Lead to an endeavor.
- Planning is a map, a guide, especially for a team. (Relatively simple and helpful techniques)
- SDLC is mostly about people, process, time management & communication.
- Risks are inevitable, planning helps to avoid stupid ones (Experience counts).
- Assessing the scope of work, timing, risks and resources will lead to the project success.

Q&A