



A Quest Global Company

SOFTWARE DEVELOPMENT LIFE CYCLE(SDLC) AND MODELS

G.SANGAMESH

SOFTWARE DEVELOPMENT LIFE CYCLE(SDLC)



- ❖ SDLC stands for Software Development Life Cycle and is also referred to as the Application Development life-cycle.
- ❖ SDLC is a systematic process for building software that ensures the quality and correctness of the software built.
- ❖ SDLC process aims to produce high-quality software that meets customer expectations. The system development should be complete in the pre-defined time frame and cost.





- ❖ SDLC consists of a detailed plan which explains how to plan, build, and maintain specific software.
- ❖ Every phase of the SDLC life Cycle has its own process and deliverables that feed into the next phase.

- **7 Phases Of SDLC:-**

- 1.Requirement Analysis
- 2.Feasibility Study
- 3.Design
- 4.Coding
- 5.Testing
- 6.Installation
- 7.Maintenance

1.REQUIREMENT ANALYSIS



- ❖ Business Analyst will be involved in this phase.
- ❖ In this phase Business Analyst will go to the customer place and collect the requirements into business language and converted into software language and explain into Project Manager , Developers as well as Test Engineers.
- ❖ Business Analyst act like a bridge between customers and software company.



2.FEASIBILITY STUDY



- Once after requirement collection is done then we go for feasibility.
- In this phase software company will decide whether we must take up the project or not .if they take up the project, they will check they do have sufficient resource, sufficient lab setup ,sufficient technology is there or not .if they have all of these, they will decide they get profit or not and then take up the project in feasibility study.

3.DESIGN



- Once after feasibility study is done then we will go for Design stage.
- It is done by architect or senior architect.
- **In design we have 2 Types**
 - 1.High level design
 - 2.Low level Design



-
- 1.High level Design:-

- If we consider any application designing the main modules of the application is called as High-level design.
- High level design is also known as external design.

- 2.Low level Design:-

- If we consider any application designing each and every module in depth of the application is called as Low-level design.
- Low level design is also known as internal design.

4.CODING



- Once after Design stage is completed then we go for coding stage.
- It is done by Senior Developer , Junior Developer.
- The actual development phase is where the development team members divide the project into software modules and turn the software requirement into code that makes the product.
- This SDLC phase can take quite a lot of time and specialized development tools_It's important to have a set timeline and milestones so the software developers understand the expectations and you can keep track of the progress in this stage

5.TESTING



- Once after coding is completed then we go for Testing stage.
- Developers will take the project and give to the testing team where in all test engineers will start testing the software in order find out the defects According to customer specification .
- If test engineers find out the defects they will communicate the defect to the developing team and This Process will continue until the defects are give to the testing team.



6.INSTALLATION



- ❖ Once after testing is completed then we go for installation stage.
- ❖ In this stage IT engineers from the company will go to the customer place and setup the environment ,install the software and give the feature demo to the customers about the application to the customers about how the application is developed .This process is known as Installation

7.MAINTENANCE



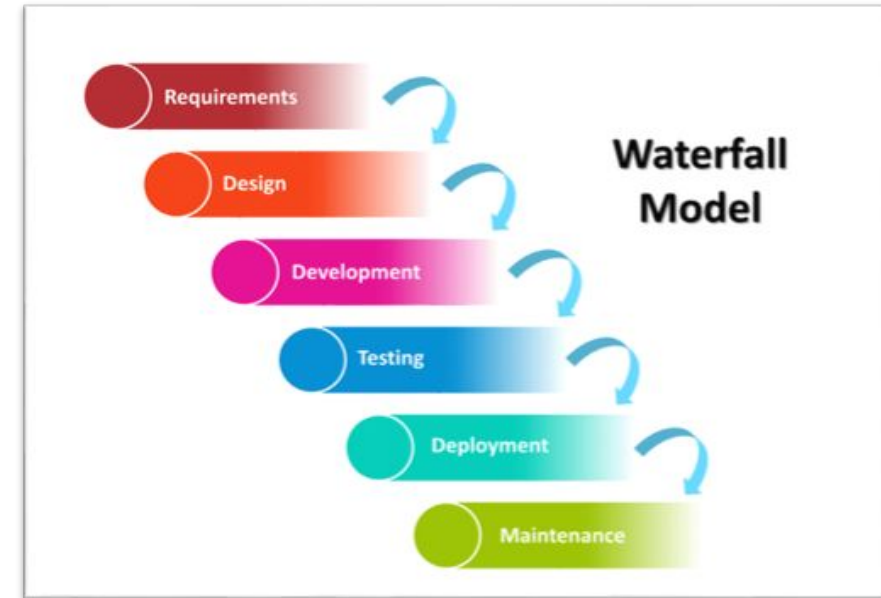
- ❖ Once the software is given to the customers ,they will use the software to run his business during this time .if customers find any defect company will fix the defect without charging customer for certain period of time ,after the period if customer find any defect, then company will fix the defect while taking charges this stage is called as Maintenance



WATERFALL MODEL



- It is a step-by-step process or standard procedure to develop the software or application.
- In this model once after feasibility study is done company will freeze the requirements so that back tracking is not possible that is we cannot change the requirements. so this model is called as waterfall model.
- If we change the requirements the corresponding design will get altered.
- If design is altered the respective coding will be altered.
- If coding is altered, it will introduce lots of defects in the testing phase due to which there will be delay in releasing the software to the customer.



PHASES OF WATERFALL MODEL



1. **Requirements analysis and specification phase:** The aim of this phase is to understand the exact requirements of the customer and to document them properly. Both the customer and the software developer work together so as to document all the functions, performance, and interfacing requirement of the software.
2. **Design Phase:** This phase aims to transform the requirements gathered in the SRS into a suitable form which permits further coding in a programming language.
3. **Implementation and unit testing:** During this phase, design is implemented. If the SDD is complete, the implementation or coding phase proceeds smoothly, because all the information needed by software developers is contained in the SDD.
4. **Integration and System Testing:** This phase is highly crucial as the quality of the end product is determined by the effectiveness of the testing carried out. The better output will lead to satisfied customers, lower maintenance costs, and accurate results.
5. **Operation and maintenance phase:** Maintenance is the task performed by every user once the software has been delivered to the customer, installed, and operational.

DISADVANTAGES



- In this model, the risk factor is higher, so this model is not suitable for more significant and complex projects.
- This model cannot accept the changes in requirements during development.
- It becomes tough to go back to the phase. For example, if the application has now shifted to the coding phase, and there is a change in requirement, It becomes tough to go back and change it.
- Since the testing done at a later stage, it does not allow identifying the challenges and risks in the earlier phase, so the risk reduction strategy is difficult to prepare.



- Agile methodology is a project management framework that breaks projects down into several dynamic phases, commonly known as sprints. In this article, get a high-level overview of Agile project management, plus a few common frameworks to choose the right one for your team.

Key Principles and Values

- **Individuals and interactions** over processes and tools.
- **Working software** over comprehensive documentation.
- **Customer collaboration** over contract negotiation.
- **Responding to change** over following a plan.
- **Agile Product Management**
 - In simple terms, Agile Product Management is a concept and approach that focuses on speed while taking the demands of customers into account throughout the product development life cycle.
- **Scrum**
 - [Scrum](#) is a common Agile methodology for small teams and also involves sprints. The team is led by a Scrum master whose main job is to clear all obstacles for others executing the day-to-day work.



- **Scrum master**:-A Scrum Master is a team leader who supports a Scrum team by guiding them through the Scrum process. They help the team stay organized, remove any roadblocks, and promote collaboration, ensuring that everyone follows Agile principles for effective project delivery.
- **Scrum meeting**:-Development Team (Scrum Master and Product Owner may attend but do not participate actively). Each team member answers three questions:
 - 1) What did you do yesterday?
 - 2) What did you do today?
 - 3) What are you going to do tomorrow?
- **Sprint planning**:-**Purpose**: To define the goals and tasks for the upcoming sprint.
 - **Participants**: Scrum Master, Product Owner, Development Team.
 - **Activities**: The team discusses what can be accomplished in the sprint, selects items from the product backlog, and breaks them down into actionable tasks.

