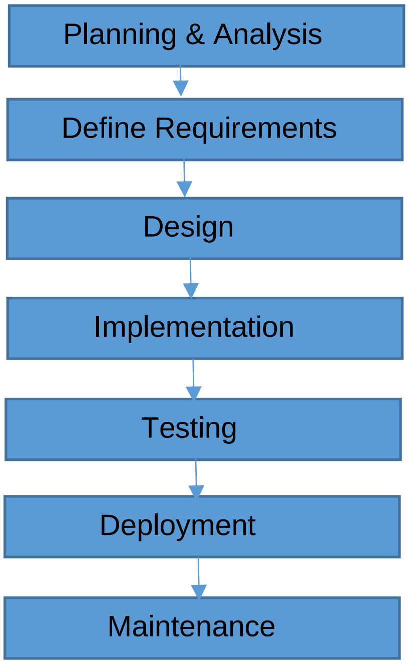
Software Development Life Cycle

- Software Development Life Cycle(SDLC) is a process that is used to develop ,design the software products.
- SDLC defines the whole procedure of software development step by step.



Phases of SDLC:

1)Planning & Analysis:

-> The first phase of the SDLC is the project planning stage where you are gathering business requirements from your client or stakeholders.

-> Requirement analysis is the most important and fundamental stage in SDLC. It is performed by the senior members of the team with inputs from the customer, the sales department, market surveys and domain experts in the industry.

2)Defining requirements:

-> Once the requirement analysis is done the next step is to clearly define and document the product requirements and get them approved from the customer or the market analysts.

->This phase is critical for converting the information gathered during the planning and analysis phase into clear requirements for the development team.

3)Designing:

Software Development Life Cycle

->The original plan and vision are elaborated into a software design document that includes the system design, programming language and platform to use.

->All the important stakeholders and based on various parameters as risk assessment, product robustness, design modularity, budget and time constraints, the best design approach is selected for the product.

4)Implementation:

->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.

->If the design is performed in a detailed and organized manner, code generation can be accomplished without much hassle.

5)Testing:

->However, this stage refers to the testing only stage of the product where product defects are reported, tracked, fixed and retested, until the product reaches the quality standards

6)Deployment:

->Once the product is tested and ready to be deployed it is released formally in the appropriate market. Sometimes product deployment happens in stages as per the business strategy of that organization.

->Then based on the feedback, the product may be released as it is or with suggested enhancements in the targeting market segment.

7)Maintenance:

->The maintenance phase is the final stage of the SDLC .

->In the maintenance stage, users may find bugs and errors that were missed in the earlier testing phase. These bugs need to be fixed for better user experience.

Software development life cycle models

1)Waterfall model

2)Spiral model

3)Iterative model

4)V model

5)Incremental model

6)Agile model

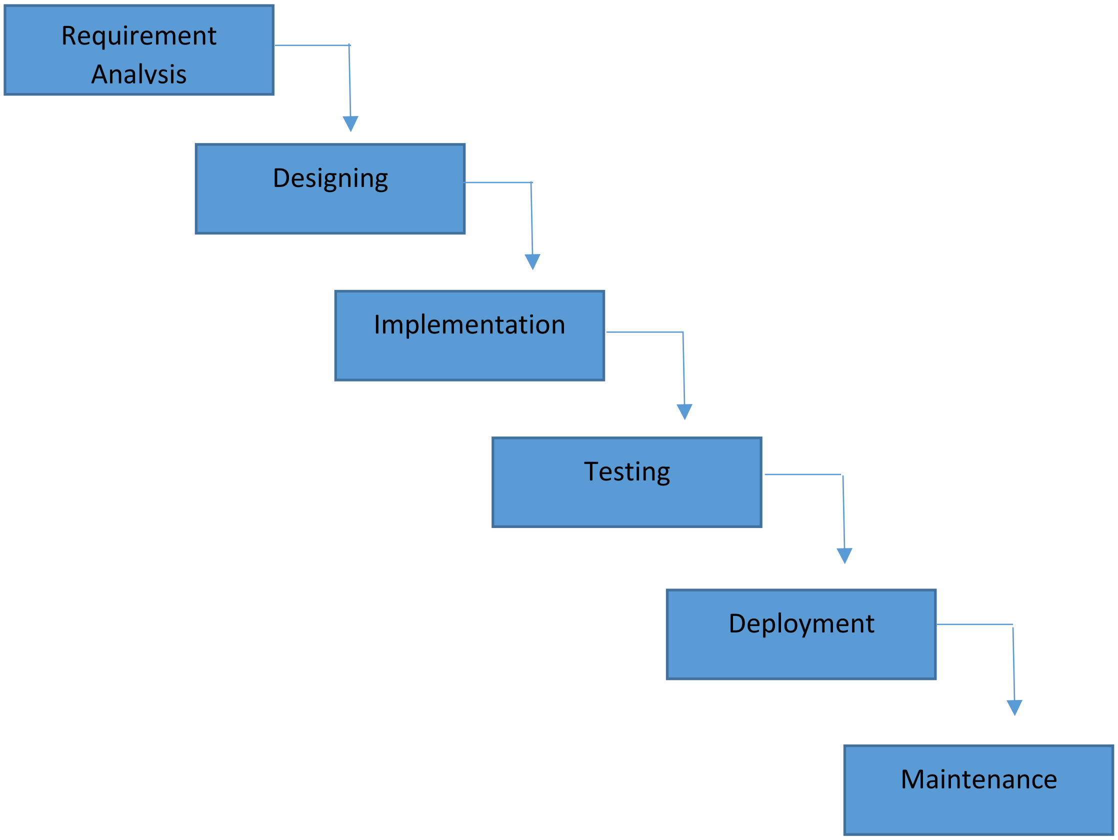
Software Development Life Cycle

Waterfall model:

The Waterfall model is one of the earliest approaches to software development and is often considered a classic model of the Software Development Life Cycle.

->It is a linear and sequential approach where each phase must be completed before the next one begins.

- Sequential phases.
- No overlapping phases.
- Clear documentation.



Requirement Analysis:

- In this initial phase, all possible requirements of the system to be developed are gathered and documented. This phase focuses on understanding what the stakeholders need and defining the software requirements specification (SRS) document.

Software Development Life Cycle

Design:

- This phase involves translating the gathered requirements into a blueprint for the system. It includes both high-level design (HLD) for the system architecture and low-level design (LLD) for the specific components and modules.

Implementation:

- During this phase, the actual source code is written based on the design documents. The development team translates the designs into a working software system, typically starting with small modules and then integrating them into a complete system.

Testing:

- After the coding phase, the individual modules are integrated into a complete system. Testing is performed to ensure that the system meets the requirements specified in the SRS document. Various types of testing, including unit testing, integration testing, system testing, and acceptance testing, are carried out.

Deployment:

- Once the software has been tested and is deemed ready for release, it is deployed to the production environment where it will be used by the end users. This phase involves installation, configuration, and any necessary user training.

Maintenance:

- After deployment, the system enters the maintenance phase where it is monitored and supported to ensure it continues to function correctly. This phase includes fixing any bugs that are discovered, making improvements, and updating the system to adapt to changing user needs or environments.

Advantages: Simple to understand and manage, well-documented stages.

Disadvantages: Inflexible to changes, late detection of issues, high risk if requirements change.

Spiral model:

Spiral model is risk driven software development process model.

It is a combination of waterfall model ,Iterative model.

Software is developed in a series of incremental releases as per each spiral.

Divided into 4 parts:

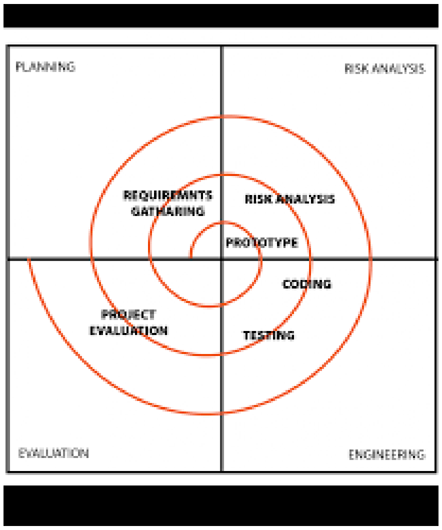
Software Development Life Cycle

1)Planning

2)Risk Analysis

3)Engineering and Execution

4)Customer Evaluation



Planning:

- Communication between customers and project head.
- Collect all the requirement from customers
- Analysis estimated cost and schedule.

Risk Analysis:

- Identification of all potential risks.
- Strategy for solving risks

Engineering and Execution:

- Developer perform actual coding
- Tester perform all testing methods
- Deploy or release product to the customer environment.

Evaluation:

- Take a feedback from customers.
- If customer want any changes,goes to next planning or next spiral iteration.

Advantages: Risk management, flexibility, iterative refinement.
Disadvantages: Complex, requires expertise in risk analysis, can be expensive.

Iterative model:

Software Development Life Cycle

Starts with some requirement and analysis then develop first version of software.

If any changes,new versions are created.Again if any changes,new versions are created called as Iterations.After finalized,Deploy the product.

1)Requirement gathering and analysis

2)Design Phase

3)Coding Phase

4)Testing Phase

5)Review Phase

6)Deployment and Maintenance Phase

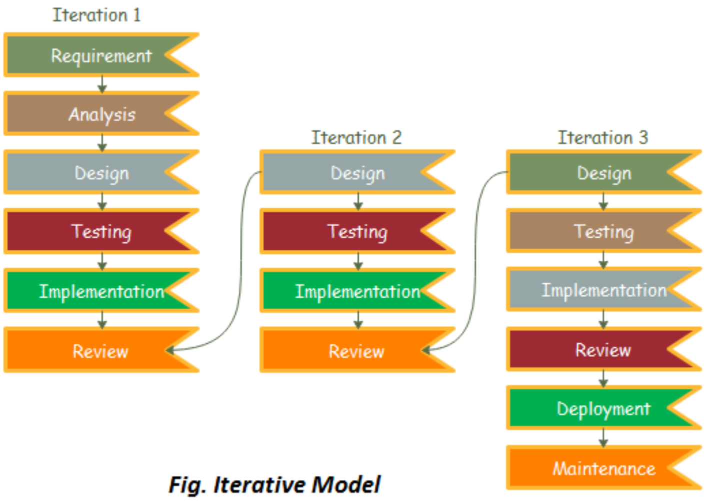


Fig. Iterative Model

Advantages: Flexibility, early detection of issues, continuous improvement, stakeholder engagement.

Disadvantages: Resource intensive, complex management, potential for scope creep.

Incremental model:

Requirements are divided into multiple standalone modules of the software development cycle.

Each module goes through the requirements,design,implementation and testing phases.

The process continues until the complete system achieved.

Software Development Life Cycle

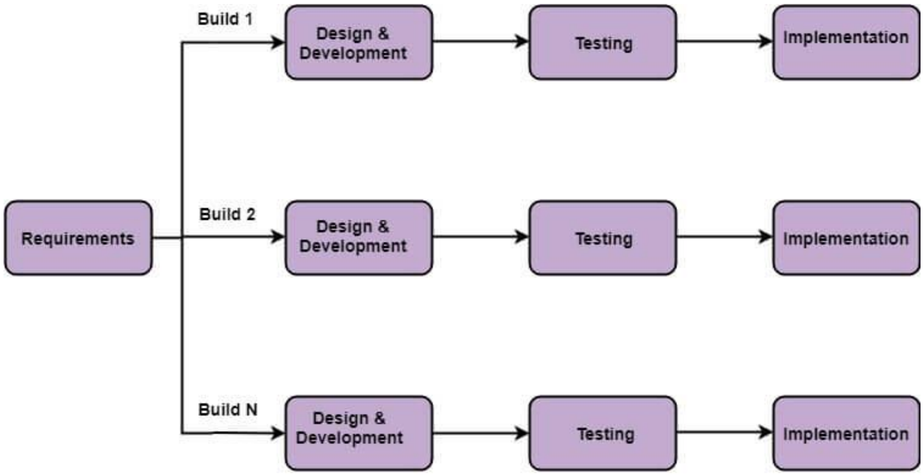


Fig: Incremental Model

Advantages: Early delivery, reduced risk, flexibility, better resource management.

Disadvantages: Integration challenges, overhead costs, incomplete systems until all increments are developed.

V model:

In this each phase of the development life cycle is associated with Testing phase.

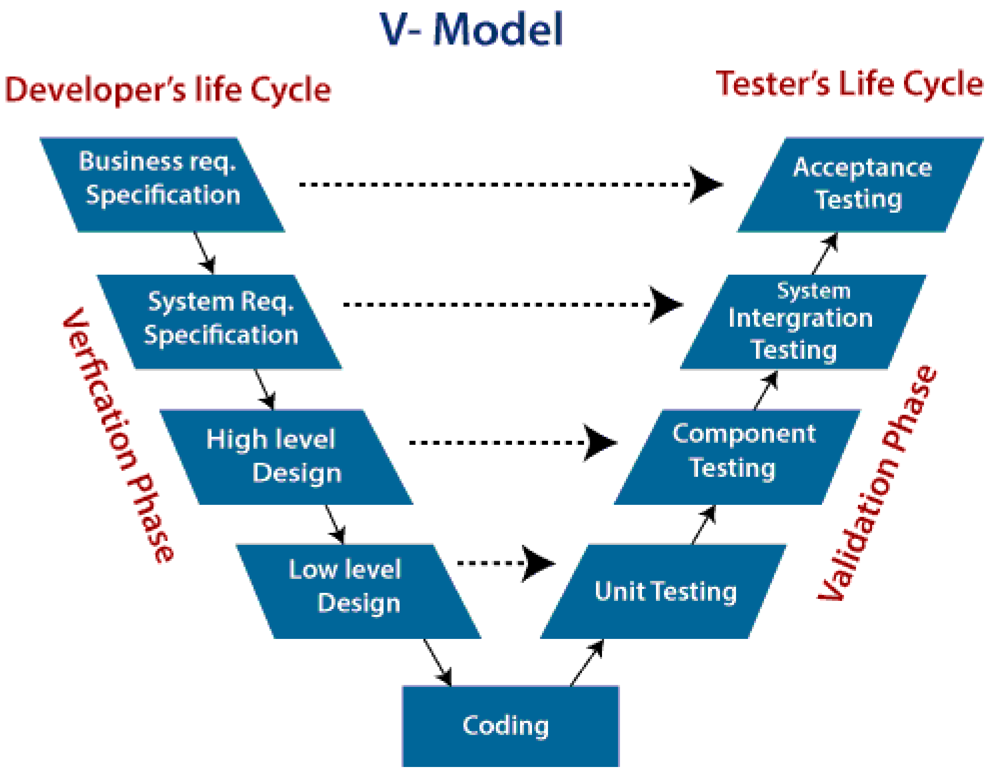
Testing of the software is planned in parallel with the corresponding phase of development.

Similar to waterfall model follow linear process,moves to next phase only after completing its preceding phase.

Advantages: Emphasis on testing, clear milestones, easy to manage.

Disadvantages: Inflexible, expensive, and time-consuming if changes are needed.

Software Development Life Cycle

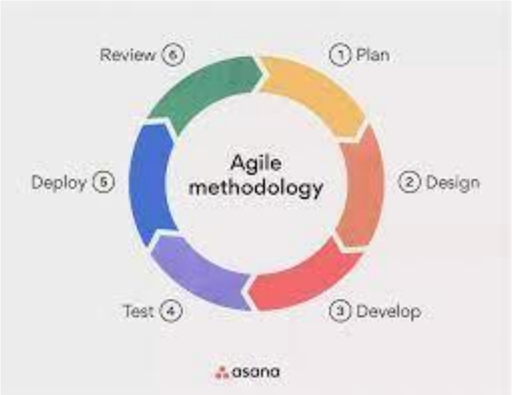


Agile model:

- It is an incremental and iterative process of software development.
- Divides requirement into multiple iterations and provide specific functionality for the release.
- Delivers multiple software requirements.Each iteration are lasts from two or three weeks.

Agile methods break tasks into smaller iterations, or parts do not directly involve long term planning. The division of the entire project into smaller parts helps to minimize the project risk and to reduce the overall project .

Software Development Life Cycle



Product Planning

- This phase involves refining the product backlog, which is a prioritized list of features or user stories.
- Product owners, stakeholders, and the development team collaborate to clarify requirements, add details, and prioritize items for upcoming iterations.

Sprint Planning:

- At the beginning of each sprint, the team conducts a sprint planning meeting to select the items from the product backlog that will be worked on during the sprint.
- The team estimates the effort required for each selected item and defines the tasks necessary to implement them.

Development:

- During the sprint, the development team works on implementing the selected user stories or features.
- Developers collaborate closely with each other, as well as with testers, designers, and other stakeholders as needed.

Daily Stand-up (or Daily Scrum):

- Agile teams typically hold daily stand-up meetings, where team members provide brief updates on their progress, discuss any impediments, and coordinate their work for the day.
- These meetings are time-boxed and help ensure that everyone is aligned and aware of the team's progress.

Continuous Testing:

- Throughout the sprint, developers integrate their code changes frequently, often multiple times per day.

- Automated tests are run continuously to detect and prevent regressions, ensuring that the software remains stable and functional.

Sprint Review:

- At the end of each sprint, the team holds a sprint review meeting to demonstrate the completed work to stakeholders.
- Stakeholders provide feedback on the delivered features, and the team discusses any changes or adjustments needed based on that feedback..

Release (or Deployment):

- Depending on the project's cadence and release strategy, the completed features may be deployed to production at the end of each sprint or accumulated for release at a later time.
- Continuous deployment practices may be employed to automate the release process and deliver new features to users quickly and frequently.

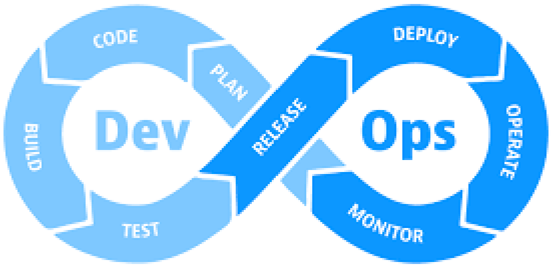
Advantages: Flexibility, customer satisfaction, continuous delivery, adaptability.

Disadvantages: Requires active user involvement, can be chaotic without proper management, hard to predict timeline and budget.

DevOps model

- DevOps is a software development process that emphasises developer and operations team communication. It promotes timely feedback, which expedites the identification of any flaws or problems during the development process. This makes it an ideal strategy for large-scale endeavours.
- It is an alternate methodology for software development that increases speed, efficiency, and output.
- In the past, development and operations jobs were separated,resulting in a knowledge gap between teams and a disconnected workflow these issues manifested mostly as miscommunication and cross-team misunderstandings, frequently causing release delays.

Software Development Life Cycle



Phases of DevOps life cycle:

- **Plan** – The planning phase is exactly planning the project's lifecycle. In contrast to conventional methods to the development lifecycle, this model assumes that each stage will be repeated as necessary. In this manner, the DevOps workflow is planned with the likelihood of future iterations and likely prior versions in mind.
- **Code** – The developers will write the code and prepare it for the next phase during the coding stage. Developers will write code in accordance with the specifications outlined in the planning phase and will ensure that the code is created with the project's operations in mind.
- **Build** – Code will be introduced to the project during the construction phase, and if necessary, the project will be rebuilt to accommodate the new code. This can be accomplished in a variety of ways, although GitHub or a comparable version control site is frequently used.
- **Test** – Throughout the testing phase, teams will do any necessary testing to ensure the project performs as planned. Teams will also test for edge and corner case issues at this stage. An "edge case" is a bug or issue that only manifests during an extreme operating event, whereas a "corner case" occurs when many circumstances are met.
- **Release** – The release phase occurs when the code has been verified as ready for deployment and a last check for production readiness has been performed. The project will subsequently enter the deployment phase if it satisfies all requirements and has been thoroughly inspected for bugs and other problems.
- **Deploy** – In the deploy phase, the project is prepared for the production environment and is operating as planned in that environment. This would be the responsibility of the operations team; in DevOps, it is a shared responsibility. This shared duty pushes team members to collaborate to guarantee a successful deployment.

Software Development Life Cycle

- **Operate** – In the operating phase, teams test the project in a production environment, and end users utilise the product. This crucial stage is by no means the final step. Rather, it informs future development cycles and manages the configuration of the production environment and the implementation of any runtime requirements.
- **Monitor** – During the monitoring phase, product usage, as well as any feedback, issues, or possibilities for improvement, are recognized and documented. This information is then conveyed to the subsequent iteration to aid in the development process. This phase is essential for planning the next iteration and streamlines the pipeline's development process.