**Software Development Life Cycle**

* SDLC is a process that consists of a series of planned activities to develop or alter the Software Products.
* Why SDLC:

1. Process used by software industry to design, develop and test high quality software.
2. Aims to provide high quality software that meets or exceeds customer expectations, reaches completion with times and cost estimates.
3. Consists of a detailed plan describing how to develop, maintain, replace and alter or enhance specific software.
4. Basically it defines a methodology for improving the quality of software and overall development process.

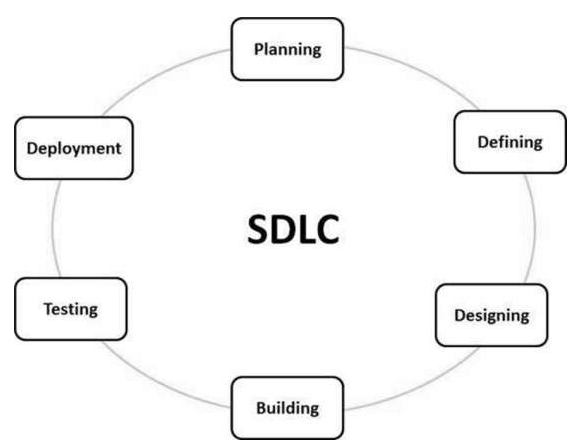
* SDLC Models:

Following are the most important and popular SDLC models followed in the industry –

1. Waterfall Model
2. Iterative Model
3. Spiral Model
4. V-Model
5. Big Bang Model

Other related methodologies are Agile Model, RAD Model, Rapid Application Development and Prototyping Models.

* SDLC graphical representation:



1. Stage 1 : Planning and Requirement Analysis

This 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. This is then used to plan the basic approach and to conduct project feasibility study in the economical, operational and technical areas.

Planning for the QA requirements and the identification of the risks associated with the project are also done in planning stage.

1. Defining requirements:

Once planning is done, next step is to clearly define and document the product requirements and get them approved from the customer or market analysts. This is done through **SRS(Software Requirement Specification)** which consists of all the product requirements to be designed and developed during the life cycle.

1. Designing the Product Architecture:

SRS is the reference for product architects to come out with the best architecture for the product to be developed. Usually more than one design approach for the product architecture is proposed and documented in the DDS – Design Document Specification.

Overall design both internal, external and third party modules even the minutest details are defined in this.

1. Building or developing the product:

Actual development starts and the product is built. All the coding logic is written at this step.

1. Testing the Product:

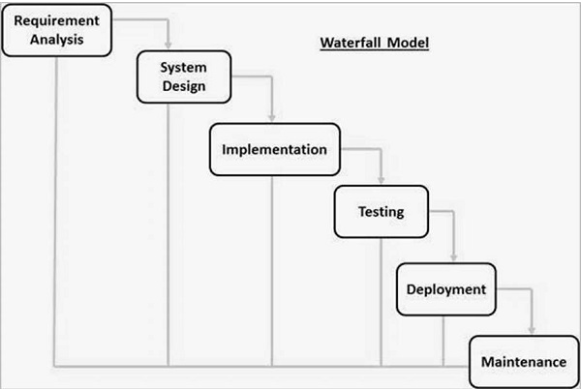
It is usually a subset of all stages as testing activities are performed at almost all steps. However, in this stage, product defects are tracked, fixed and retested until code meets standards as mentioned in SRS.

1. Deployment in the Market and Maintenance

Once the product is tested and ready to be deployed, it is released formally in the appropriate market. Sometimes deployment happens in stages as per business requirement. The product may be released in a limited segment and tested in the real business environment (UAT – Unit Acceptance Testing)

* SDLC Models:

1. Waterfall Model:
   * Linear-sequential life cycle model.
   * Each phase must be completed before the next phase can begin and there is no overlapping in the phases.
   * Outcome of one phase acts as an input of next phase sequentially.



* + Requirement Gathering and Analysis:

All requirements captured in this phase and documented in a requirement specification document.

* + System Design:

Requirements from previous phase are studied and system design is prepared. This helps in defining hardware and software requirements and also in defining overall system architecture.

* + Implementation:

With inputs from system design, the system is first developed in small programs called units, which are integrated in next phase. Each unit is developed and tested for its functionality which is referred to as unit testing.

* + Integration and Testing:

All units are integrated into a system after testing each unit. Post integration, entire system is tested for any fault or failures.

* + Deployment of System:

Once functional and non-functional testing is completed, it is deployed to market.

* + Maintenance:

To fix issues showing up in client environment, patches are released. Also to enhance the product some better versions are released. Maintenance is done to deliver these changes in the customer environment.

All these phases are cascaded to each other in which progress is seen as flowing steadily downwards (like a waterfall).

* + Waterfall Application:
    - When requirements are very well documented, clear and fixed.
    - Product definition is stable.
    - Technology is understood and is not dynamic.
    - Ample resources are available to support the product.
    - Project is short.
  + Advantages:
    - Simple and easy.
    - Easy to manage because of its rigidity.
    - Phases are processed and completed one at a time.
    - Works well for smaller projects.
    - Clearly defined stages.
    - Well understood milestones.
    - Easy to arrange tasks.
    - Process and results are well understood.
  + Disadvantages:
    - No working software is prepared until late.
    - High amounts of risk and uncertainty.
    - Not a good model for complex and object-oriented projects.
    - Poor model for long and ongoing projects.
    - Not good if requirements of project are moderate to high risk of changing.
    - Difficult to track progress.
    - Integration is done at the end, which doesn’t allow identifying any technological or business bottleneck or challenges early.