



SOFTWARE DEVELOPMENT LIFE CYCLE

What is SDLC?

The Software Development Life Cycle (SDLC) is a structured process that enables the production of high-quality, low-cost software, in the shortest possible production time.

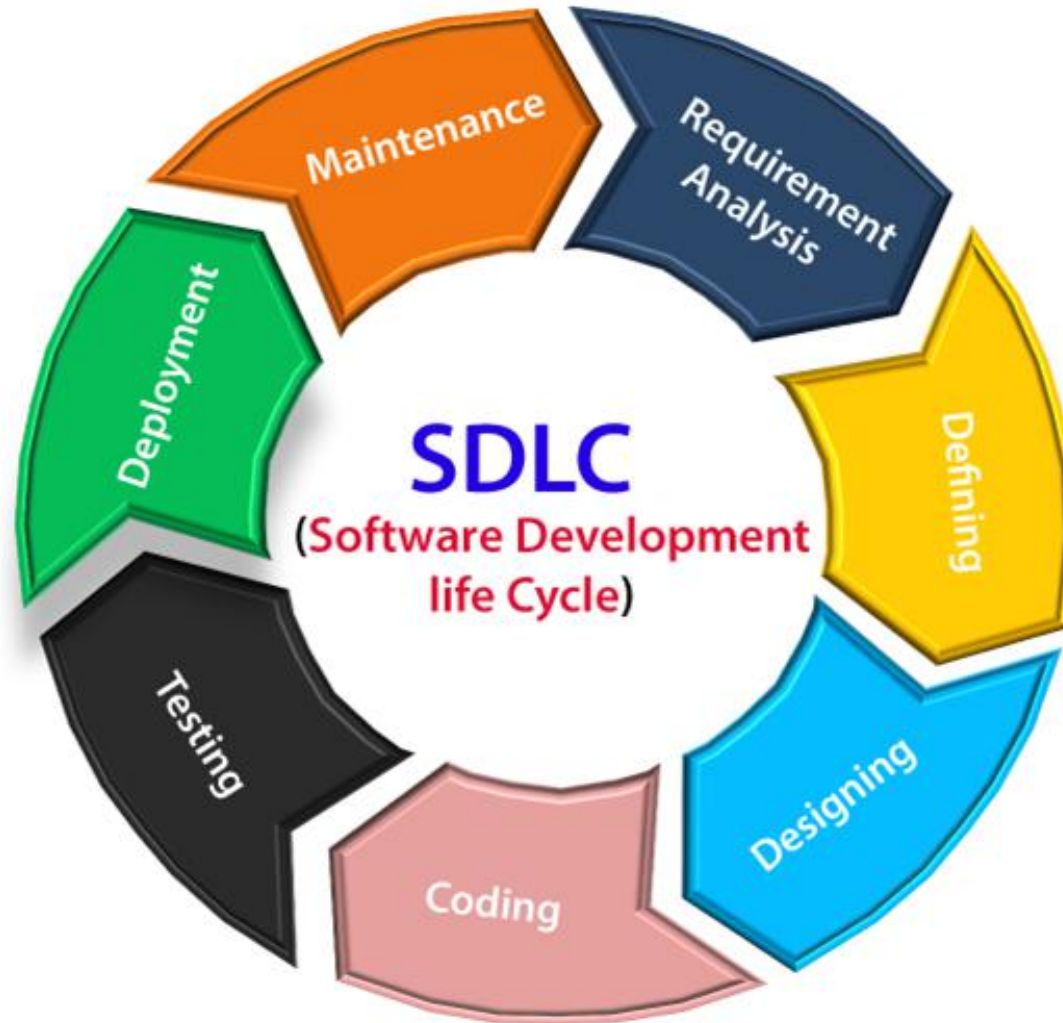
The goal of the SDLC is to produce superior software that meets and exceeds all customer expectations and demands.

The SDLC defines and outlines a detailed plan with stages, or phases, that each encompass their own process and deliverables. Adherence to the SDLC enhances development speed and minimizes project risks and costs associated with alternative methods of production.

Why SDLC and advantages?

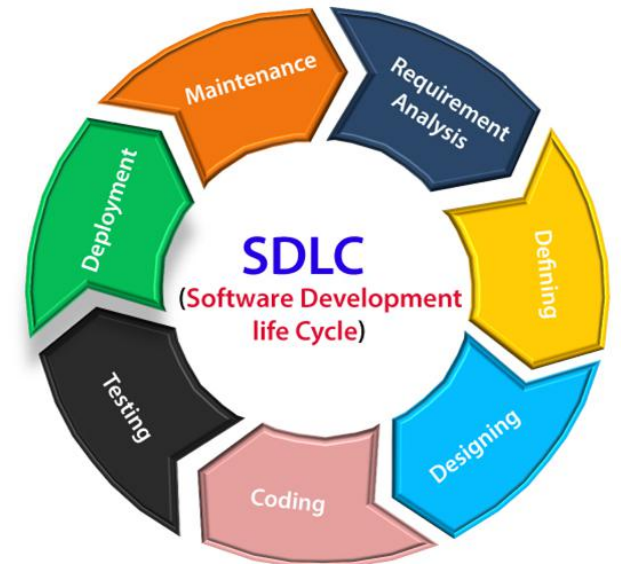
- It provides a standardized framework that defines activities and deliverables
- It aids in project planning, estimating, and scheduling
- It makes project tracking and control easier
- It increases visibility on all aspects of the life cycle to all stakeholders involved in the development process
- It increases the speed of development
- It improves client relations
- It decreases project risks
- It decreases project management expenses and the overall cost of production

SDLC Phases



SDLC Phases

- **Requirement Analysis:**
 - Requirement Gathering is the most important stage in SDLC.
 - The senior members of the team perform it with inputs from all the stakeholders and domain experts or SMEs in the industry
 - Planning for the quality assurance requirements and identifications of the risks associated with the projects is also done at this stage.
 - Business analyst and project organizer do client meeting to gather all the requirements of the clients like what to build, who will be the end-user, what is the purpose of the product, etc.
 - Before creating a product, a core understanding or knowledge of the product is very necessary.
 - Once the requirement analysis is done, the next stage is to surely document the software specifications and get them approved by the project stakeholders.
 - This can be accomplished through the “BRD”- Business Requirement Specification document, which embraces all the product elements to be created and developed during the project life cycle.



SDLC Phases

- **Designing:**

- In this stage, the requirements gathered in the BRD is used as information to obtain the software architecture.
- Moreover, this phase also consists of storyboarding or wireframing software that is needed for functionality.
- Through this, the developers then create either rough working models, or illustrates how the software will work, how it will look, how usage flows will move from screen to screen, and more.
- Result of this stage leads to TDD (Technical Design document)

- **Implementation:**

- In this stage of SDLC, the exact development begins, and the programming is built.
- Developers must follow the coding guidelines defined by their management, and programming tools like compilers, debuggers, etc. are used to generate and implement the code



SDLC Phases

- **Testing:**

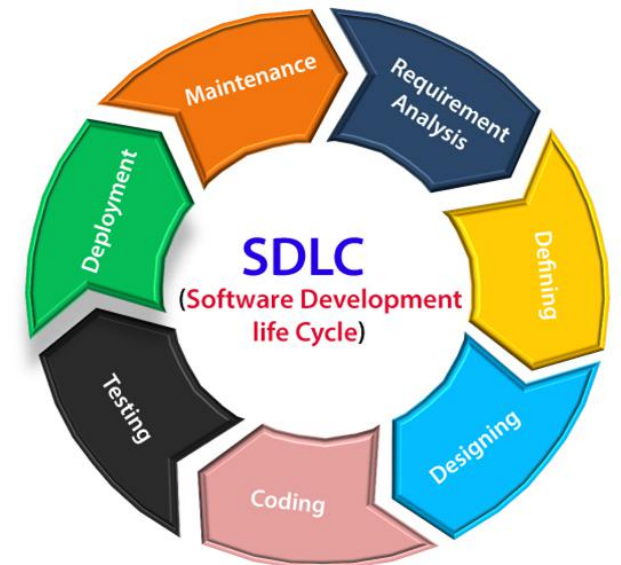
- After the code is generated, it is tested against the specifications to ensure that the products are solving the needs directed and inferred during the requirements stage.
- During this phase, several testing like unit testing, integration testing, system testing (SIT), and approval testing (UAT) are done.

- **Deployment:**

- Once the software is approved, and no bugs or errors are asserted, then it is deployed.
- After the software is deployed, then its maintenance begins.

- **Maintenance:**

- Once the client starts using the developed software, then the real issues start coming up.
- In this stage, the team is required to fix the issues, roll out new features and refine the functionalities as required.
- The method where the care is taken for the finished product is thus known as maintenance.



SDLC models

- There are different software development life cycle models specify and design, which are followed during the software development phase. These models are also called "**Software Development Process Models.**"
- Each process model follows a series of phase unique to its type to ensure success in the step of software development.

Waterfall model

Waterfall model

- Widely trusted and oldest SDLC model.
- In this linear sequential model, the whole process of software development is a step-by-step process from requirement analysis to maintenance
- Systems that have well defined requirements are a good fit for this model
- Little opportunity for customer to preview the system (until it is too late)

Agile Model

- The tasks are divided into small time boxes or time frames to deliver specific features for a release
- Iterative approach is taken and working software build is delivered after each iteration. Each build is incremental in terms of features. The final build holds all the features required by customer.
- Speed up the phases of life cycle
- Usually less formal and reduced scope
- Used for time critical applications
- Depends heavily on customer interaction. So, if customer is not clear, team can be driven in wrong direction

Agile Model

- In the Agile method, the entire project is divided into small incremental builds. All these builds are provided in iterations, and each iteration lasts from one to three weeks (Sprint)

Any agile software phase is characterized in a manner that addresses several key assumptions about the bulk of software projects:

- It is difficult to think in advance which software requirements will persist and which will change. It is equally difficult to predict how user priorities will change as the project proceeds.
- For many types of software, design and development are interleaved. That is, both activities should be performed in tandem so that design models are proven as they are created. It is difficult to think about how much design is necessary before construction is used to test the configuration.
- Analysis, design, development, and testing are not as predictable (from a planning point of view) as we might like.