15-05-2023

**SOFTWARE DEVELOPMENT LIFE CYCLE (SDLC)**

SDLC stands for Software Development Life Cycle. It is a process used by the software industry to design, develop and test high quality software. It consists of detailed planning on how to develop, maintain and alter, improvise any specific software. The lifecycle defines a methodology for the complete software development process.

SDLC is a 7- step methodology including:

1. **Planning (Requirement Analysis):** Stage of planning performed by the team leads with inputs from sales dept, domain/ industry experts, customer reviews and needs etc. Planning the basic approach for the development of the product and subsequently the study of feasibility, and different laybacks are studied and understood at this stage.
2. **Defining the Requirements:** After the requirement analysis, defining and documentation of the product requirements is done. This is done through the SRS (Software Requirement Specification) document which consists of all the product requirements to be designed and developed through the project’s entire lifecycle.
3. **Designing the Product:** Based on all the specifications in the SRS, more than one design approach is carried out for actually building the product and is documented in the Design Document Specification (DDS). The best design approach is selected based on various parameters like modularity, budget and time constraints, risks and fallacies possible etc. All the external and internal architectural module specifications must be neatly documented in this stage.
4. **Building/ Development of the product:** Programming code is generated in accordance to the DDS. This is the stage of actual development of the product where developers have to closely maintain the building process analogous to the details in the DDS for an accurate and organised end result.
5. **Testing the product:** Testing of the final product is done where the defects, bugs and other logistic problems are reported, tracked, fixed and again retested.
6. **Deployment in the Market:** After successful testing of the product, it is formally launched in the marketplace for actual utilisation by the public. Sometimes, the product may first be released in a limited segment and tested in real business environment (UAT- User acceptance testing) and based on the results of this test, may modify the product and then launch it to the entire section of users.
7. **Maintenance:** In the stage the software is monitored to ensure it continues its functionalities properly. Repairs and upgrades are also performed as and when needed.

**SDLC MODELS:**

1. **Waterfall Model:**

* The entire software development is divided into various discrete phases.
* Linearly ordering of activities, i.e., the output of the previous phase becomes input to the next phase and only after the completion of the prior phase would the next phase start.
* Some verification and validation steps are required to identify the end and beginning of phases.

1. **RAD Model:**

* Rapid Application Development process is a slight improvement to the waterfall model.
* It targets developing software in a short period, based on the concept that a better system can be developed in lesser time by using focus groups to gather system requirements than single entities.
* Business modelling, data modelling, process modelling, application generation, testing and dispatch.

1. **Spiral Model:**

* Risk-driven process model. It is a technique of rapid prototyping and concurrency in design and development activities.
* Each cycle in the spiral begins with identifying the objectives of that cycle.
* The next step in the cycle is to evaluate the different alternatives based on the objectives and constraints.
* The final step is to develop strategies to solve for those particular risks and uncertainties.

1. **V-Model:**

* Here, the steps are planned in parallel. (Verification and validation model)
* So, there are verification phases on the side and the validation phase on the other side. It joins by the coding phase.
* When the requirement is well defined and not ambiguous, we use the V-model.

1. **Incremental Model:**

* It is not a separate model, but a series of waterfall cycles.
* The entire chunk of requirements is initially divided into groups at the start of the cycle and for each individual group, the SDLC is followed.
* The SDLC process is repeated, with each release adding more functionality until all the requirements are met.

1. **Agile Model:**

* Agile model believes that every project needs to be handled differently and that it is subjective and the existing methods needs to be tailored to best suit the project requirements.
* Here, the project is divided into smaller incremental units each designed to deliver specific features of a release.
* Iterative approach is carried out and the working software is delivered after every iteration. (The final build holds all the features required by the end user as a whole)

1. **Big Bang Model:**

* The model focusses on and caters to all the types of requirements as and when they pop up.
* There is extensive coding and building phase with very little or no planning phase.
* Works best for small projects and is ideal where requirements are either unknown or final release specifications aren’t mentioned by the client.

**Advantages of SDLC:**

The primary reason behind adopting a proper SDLC is for the control over development process. Drafting of a clear working plan, management of conflict between participants, budget and time management are other commendable benefits. It also helps improve resource management.