Day1-3/8/21(SDLC)

SDLC stands for software development life cycle. It consists of various phases which we perform to develop software. We use SDLC to perform a systematic approach to save time , cost and satisfy the requirements of customers. In general a SDLC model consists of 5-7 phases. That phases are as follows :

- Planning and Requirements Analysis
- Defining Requirement
- Designing
- Development
- Testing
- Deployment
- Maintenance

Let's have a deep understanding what these phases :

Planning and Requirements Analysis :- Planning is one of the most important and necessary Step in SDLC. It is done by taking inputs from all the stakeholders and domain experts in the company. Quality assurance requirements and identification of the risks associated with the project were also done in this phase.Business analyst and Project organizer set up a meeting with the client to gather all the data e.g. if a customer wants an application that concerns money transitions. In this method the requirement has to be precise like what kind of operations will be done , how it will be done , in which currency it will be done and so on.
Once the required function is done, an analysis is complete with auditing the feasibility of the growth of a product. In case of any ambiguity, a signal is set up for further discussion.

Defining Requirements :- Once the requirement is understood, the SRS (Software Requirement Specification) document is created. After that this SRS document is sent to project stakeholders to get their approvals.The SRS document contains all the product requirements to be constructed and developed during the project life cycle.

Designing :- In this phase we bring down all the knowledge , analysis and design of the software project. It is a product of the above two phases , Input from the customer and requirements gathering were used to design the product. We design the layout of the product by defining cohesin, coupling, DFD’s ,ER-diagram, Technologies used etc.

Development :- In this phase of SDLC, the actual development begins, and the programming is built. The implementation of design begins concerning writing code. Developers have to follow the coding guidelines described by their management and programming tools like compilers, interpreters, debuggers, etc. are used to develop and implement the code.

Testing :- After the code is generated, it is tested against the requirements to make sure that the products are solving the needs addressed and gathered during the requirements stage. There are two types of testing

- Functional

- Non Functional

Functional:- Various Functional testing methods are as follow:-

- Unit testing
- Component testing
- API testing
- UI testing
- White-box testing
- Black-box testing

Non Functional :- Various non functional testing methods are as follow:-

- Availability testing
- Performance testing
- Compatibility testing
- Localization testing
- Volume testing
- Scalability testing
- Usability testing

Deployment:- Once the software is certified, and no bugs or errors are stated, then it is deployed.Then based on the assessment, the software may be released as it is or with suggested enhancement in the object segment.

Maintenance:- Once when the client starts using the developed systems, then the real issues come up and requirements to be solved from time to time.
This procedure where the care is taken for the developed product is known as maintenance.

Agile SDLC model

Agile SDLC model is a combination of iterative and incremental process models with focus on process adaptability and customer satisfaction by rapid delivery of working software product. Agile Methods break the product into small incremental builds. These builds are provided in iterations. Each iteration typically lasts from about one to three weeks. Every iteration involves cross functional teams working simultaneously on various areas like −

- Planning
- Requirements Analysis
- Design
- Coding
- Unit Testing and
- Acceptance Testing.

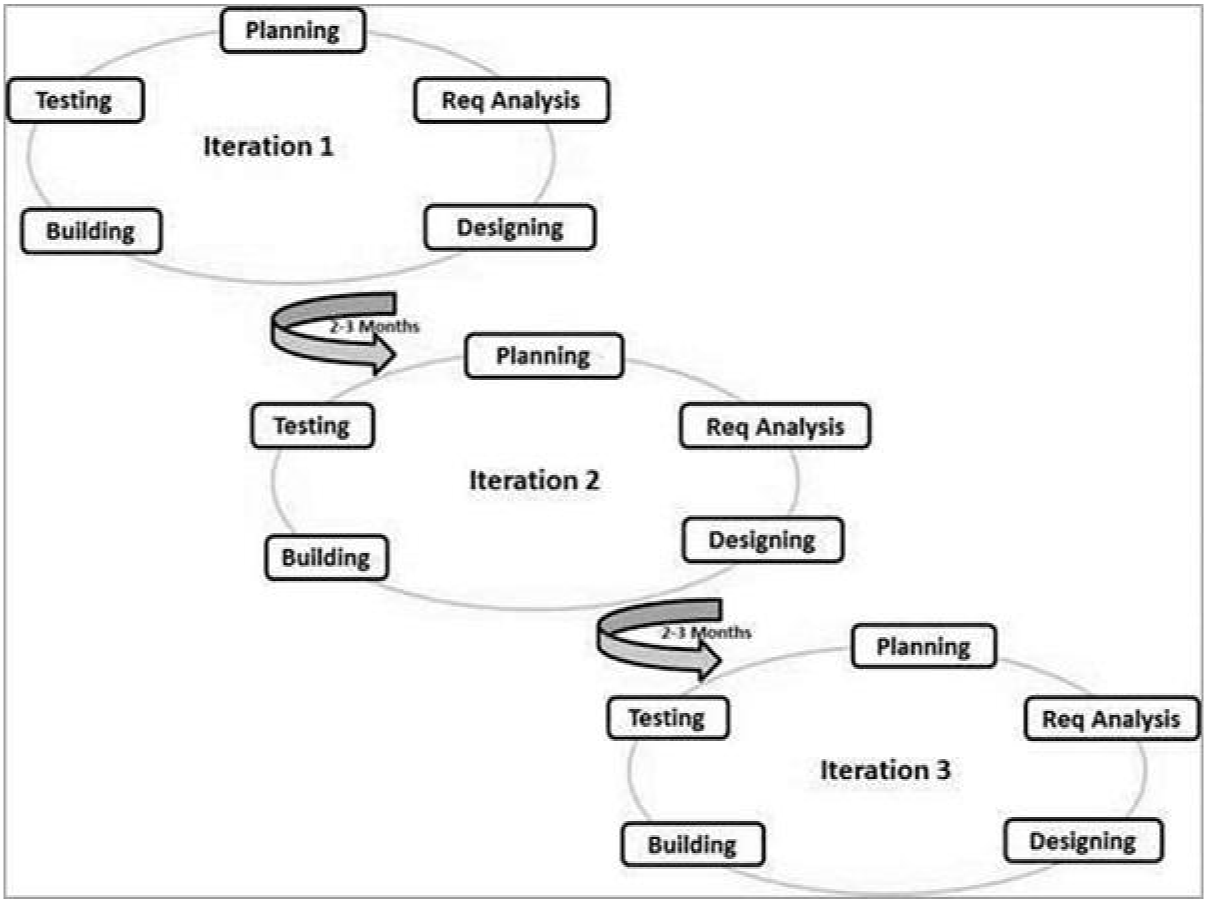
At the end of the iteration, a working product is displayed to the customer and important stakeholders.

What is Agile?

Agile model believes that every project needs to be handled differently and the existing methods need to be tailored to best suit the project requirements. In Agile, the tasks are divided to time boxes (small 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 the customer.

Here is a graphical illustration of the Agile Model −



Following are the Agile Manifesto principles −

- **Individuals and interactions** − In Agile development, self-organization and motivation are important, as are interactions like co-location and pair programming.

- **Working software** − Demo working software is considered the best means of communication with the customers to understand their requirements, instead of just depending on documentation.
- **Customer collaboration** − As the requirements cannot be gathered completely in the beginning of the project due to various factors, continuous customer interaction is very important to get proper product requirements.
- **Responding to change** − Agile Development is focused on quick responses to change and continuous development.

Agile Model - Pros and Cons

Agile methods are being widely used in the world recently.

The main Pros of Agile model are:

- Is a very realistic approach to software development.
- Promotes teamwork and cross training.
- Functionality can be developed rapidly and demonstrated.
- Resource requirements are minimum.
- Suitable for fixed or changing requirements
- Delivers early partial working solutions.
- Good model for environments that change steadily.
- Little or no planning required.
- Easy to manage.
- Gives flexibility to developers.

The main Cons of Agile model are:

- Not suitable for handling complex dependencies.
- More risk of sustainability, maintainability and extensibility.
- Strict delivery management dictates the scope, functionality to be delivered, and adjustments to meet the deadlines.
- Depends heavily on customer interaction, so if the customer is not clear, the team can be driven in the wrong direction.
- There is a very high individual dependency, since there is minimum documentation generated.
- Transfer of technology to new team members may be quite challenging due to lack of documentation.

Server

A server is a computer that serves information to other computers through the internet. A server is used to collect and send information. When we enter a url in the browser our computer communicates with the server hosting the website and pulls the data on our system.

Containerization

Containers encapsulate an application as a single executable package of software that bundles application code together with all of the related configuration files, libraries, and dependencies required for it to run. Containerized applications are “isolated” in that they do not bundle in a copy of the operating system. Instead, an open source runtime engine (such as the Docker runtime engine) is installed on the host’s operating system and becomes the conduit for containers to share an operating system with other containers on the same computing system.

Kubernetes

Kubernetes is a portable, extensible, open-source platform for managing containerized workloads and services, that facilitates both declarative configuration and automation. It has a large, rapidly growing ecosystem. Kubernetes services, support, and tools are widely available.