Boss Bhandari
Telecom NMS(Java)
bossbhandari@gmail.com

ASSIGNMENT-1

Problem Statement: SDLC Overview - Create a one-page infographic that outlines the SDLC phases (Requirements, Design, Implementation, Testing, Deployment), highlighting the importance of each phase and how they interconnect.

The software development lifecycle (SDLC) is the cost-effective and time-efficient process that development teams use to design and build high-quality software. The goal of SDLC is to minimize project risks through forward planning so that software meets customer expectations during production and beyond. This methodology outlines a series of steps that divide the software development process into tasks you can assign, complete, and measure.

SDLC- Software Development Lifecycle is an efficient and effective process that is used by development teams to design and creating a software or project demanded by client. It is checked by the team that the whole process would require less cost and time to be completed. The aim is to plan the project making in such a way that it meets the client's requirements up to the mark. This methodology outlines a series of steps that divide the software development process into tasks you can assign, complete, and measure.
Given below are the various common phase of SDLC used by almost every management team in companies:

1. **Requirements:** This phase deals with understanding and collecting the business requirements from the client to develop a product that meets their expectations. It involves a series of meetings between the business analyst and the project manager to understand the nitty-gritty of the project – the client's needs, who the end-user is, what changes are required in the existing product, if any, and such. It is ensured that all requirements are clearly understood and there's no scope for any ambiguity.
2. **Design:** In this phase, team of engineers determines requirements and identify the best solutions to create the software product. For example, they may consider integrating pre-existing modules, make technology choices, and identify development tools. They will look at how to best integrate the new software into any existing IT infrastructure the organization may have.
3. **Implementation:** In the implementation phase, the software team works on the design document created earlier to translate the statements to source code. It is ensured that all software components are completed in this stage.
4. **Testion**: The development team combines automation and manual testing to check the software for bugs. Quality analysis includes testing the software for errors and checking if it meets customer requirements. Because many teams immediately test the code they write, the testing phase often runs parallel to the development phase.

5. **Deployment:** When teams develop software, they code and test on a different copy of the software than the one that the users have access to. The software that customers use is called *production*, while other copies are said to be in the *build environment*, or testing environment. Having separate build and production environments ensures that customers can continue to use the software even while it is being changed or upgraded. The deployment phase includes several tasks to move the latest build copy to the production environment, such as packaging, environment configuration, and installation.

These phases are interconnected because each phase builds upon the outputs of the previous phase and feeds into the next phase. For example, the requirements identified during the planning and analysis phases inform the design and implementation phases. Similarly, the design informs the implementation, which in turn leads to testing, deployment, and maintenance. Changes made in one phase may require revisions in subsequent phases, highlighting the iterative and interconnected nature of the SDLC.

ASSIGNMENT-2

Problem Statement: Develop a case study analyzing the implementation of SDLC phases in a real-world engineering project. Evaluate how Requirement Gathering, Design, Implementation, Testing, Deployment, and Maintenance contribute to project outcomes.

Answer:

Case Study: Let's consider a case of a startup online learning management company who just want to have a website where a user/student can

1. Register with their email id, and create personal password
2. Select courses according to their respective required skills
3. Explore through the whole website including differents walls
4. Pay to enroll in his selected course through some payment gateway mode
5. Start learning and submit assignments given at the end of each course module

The company gives ask for a software service based company to make such web portal. Now the software companies team need to make the plan according to the SDLC of how the things are going to be managed and the product is to be given before the given deadline. For this the team need to plan the following phases to reach the required results and arrange various internal meetings to decide how things are to be done. The Software Development Life Cycle phases by which the ideal work is to be done are as given below:

1. **Requirement Gathering**: The requirements of the clients are taken firstly and then the team need to design the documents on that demands and how it needs to be fulfilled. The demands by the client and the final product should be the same and the client must become satisfied receiving that. This condition is to be checked for sure that the website is, for example, good and easy to access by the user or not or the payment is made easily or not, etc.
2. **Design**: before actually starting the works the whole process needs to be pre planned such that no issue in steps would get created further. The design of the process, teams are to be made, which work will be assigned to which team, what will the sprint duration to give the incomplete but still workable project to the client, the time duration and the strategy to complete that project in that duration are to be pre planned in the internal meeting. Various teams are made to work on different functions like to create secure payment gateway, backend development, frontend development, login page, etc.and explanation will be given to the respective members of that team about how the results will be achieved and what are their individual roles in this project.
3. **Implementation**: the designed things are to be implemented in this phase. Whole process is to be carried out following the plans made earlier. Whole different teams works together cooperatively to complete the project in their given duration.4. **Testing**: after or during the project being made various internal and external testings are to be done to check that if the product made works according the respective demands of client or not. All the functionalities of the web portal are checked and tested if any error occur it is sent back back to the developers for the correction best possible.
5. **Deployment**: Finally after all the tests and developments the final product is given to the LMS company to use and wait for them to give their feedback on the website.

6. **Maintenance**: Even after the project is completed it is ensured by the service based company that in future there should be no errors or problems created in the product and if that happens they would maintain the product in the best possible manner whenever needed.

ASSIGNMENT - 3

Problem Statement: Research and compare SDLC models suitable for engineering projects. Present findings on Waterfall, Agile, Spiral, and V-Model approaches, emphasizing their advantages, disadvantages, and applicability in different engineering contexts.

Solution:

● **WATERFALL MODEL:**

- Waterfall model is a sequential or linear SDLC process used in developing softwares in which the workflow of the process seems to be falling in downwards direction just like a real waterfall through various phases of SDLC.
- It is also known as Linear-Sequential life cycle model
- Every step or stage is needed to be completed separately at right time and no steps can be jumped off or ignored.
- In this process documentation is done at every step
- The steps involved in process are explained in documents with simplicity such that it will be easier for people to understand whats happening in each step.
- Testing is also done on each and every step.

● **Advantages of Waterfall model:**

- It is simple and easier to understand and use
- Rigid but easier to manage
- Each phase has some deliveries to be made and a review process
- One step has to be completed to start another which results in clarity of processes
- Best suitable for small projects
- all stages can easily be defined and simpler to understand
- Each phase proceeds in strict and easy to understand orders.
- It has well understandable milestones.

● **Disadvantages of Waterfall model:**

-
It is very difficult to go back and change anything in previous steps

-
Customer will only have accessed to the final product at last after the submission by the developers.

-
This model is not suitable for large and complex projects

-
It is not easy to follow the sequential linearly flow in the process

-
Member in the teams which does not have the required skills on which the work is going on currently will be seated idle which is not good

● **Applicability of Waterfall model:** Being a safe, secure and simple model it finds its application in small level projects which even require less number of team members andless complex results. As it works in streamline flow, we can easily have the access to check that on which stage what happened and if any error is found at a certain step or stage then it can be prevented in future easily as it generally have less complex steps to be performed. Thus it find its application in construction level projects, basic manufacturing and infrastructure development projects, etc.

● **AGILE MODEL:**

-
This model mainly focuses on incrementation and iteration in the process

-
Rapid delivery to the consumer is the main feature

-
It breaks down the software in to small components called builds

-
These builds are given a number of iterations

-
Each iterations are also called sprints. These sprints can lasts from 1 to 3 weeks

-
Every iterations can involves various phases like planning, requitement analysis, design, coding, unit testing, and acceptance testing

-
After the end of each iteration a working product is been given to consumer

● **Advantages of Agile model:**

-
The team are able to detect any errors and problems earlier and can prevent the further failure

-

A high quality software is achieved as compared to other models

- Since whole team works together and the working model model is given to the client after every sprint duration it results in improvement in control over the project of the teams as well as the clients
- Dependency over individuals are reduced as whole group of team is involved in achieving the results
- It also increases flexibility
- 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-
Minimal rules, documentation easily employed
- Enables concurrent development and delivery within an overall planned context
- Little or no planning required
- Easy to manage
- Gives flexibility to developers

● **Disadvantages of Agile model:**

- It is not suitable for all the projects or processes
- There is a risk of overlapping on process activities
- Continuous changes and addition or removal of features in process may lead to indifferent final product which the team estimated before starting working on that
- It depends on customer’s feedback after every sprint which makes it slight non

flexible and the results may be delayed.

● **Applicability of Agile Model:**

-

Agile model is the most applied model in solving a problem or making a finalized product in software engineering. For the projects continuously evolving with and conditions we find this model as most suitable. It enables continuous communication in between the team and the client which makes the product more better than what was estimated in the beginning. It find its application in software development, R&D products, etc.

● **V MODEL:**

-

Also called Validation or Verification model

-

The project is splitted into various steps

-

After each step testing is done simultaneously

-

Development of each step is directly proportional and connected to the testing-
Next phase can only be started after the development phase is completed and testing is also performed

● **Advantages of V Model:**

-

Used for simpler project where there is not requirement of complex processes to be done

-

Easily understandable

-

Tracking the processes is easy

-

Testing on every stage helps in preventions form coming big problems

-

Communication is easier

-

Consists of cleared and structured process

● **Disadvantages of V model:**

-

Not suitable for big and complex projects

-

Does not support iteration in processes

-

Consumes huge amount of time as testing and documentation are done at every level

-

If the project needs complex solution we can’t opt for this

-

Not suitable for ever changing projects

● **Applicabilities:**

-

This model is best for the projects requires testing at every particular level. Thus it find its application in medical devices, critical safety systems, etc.

● **SPIRAL MODEL:**

-

It works on systematic as well as iterative approach

-

Looks like a spiral in diagramatic way

-

Each spiral runs over the loops of phases which can varies from project to project- Exact number of loops or phases are generally canton fixed and only be decided by the team differently

● **Advantages of Spiral model:**

-

Risk handling is done on every phase thus good for risk projects

-

Suitable for large and complex projects

-

Client can have a watch over the steps of project making before the finishing

-

Good communication as review and feedback is allowed

-

Even change requests are followed

● **Disadvantages of Spiral model:**

-

It is more complex than other models

-

It is expensive as well as difficult to understand from the user point of view

-

More time consuming as it consists of many evaluations and reviews

-

Time estimation is difficult on initial stages

● **Applicabilties:**

-

It is suitable for projects having risks and complex requirements and which needs frequent validation and verification. Thus it finds its application in aerospace and defence systems and various large scale software systems.