1). What is SDLC

Software development life cycle- it’s a structure to follow during developing software lifecycle

Defining SDLC in detail the process starts with planning, implementation, testing, , documentation deployment and maintenance of on-going software and support.

In this there are series of steps or phases that we follow to design software as per the requirement to deliverable software; there is a model for development and lifecycle management of an application or software.

The SDLC process may vary as per companies, industries or as per organization but standards such as ISO/IEC 12207 is determined, ISO/IEC full form stands as international software engineering standard that defines software engineering process that associates with its lifecycle process from conception to its deliverable product. There are modes for development, acquisition and configuration of software systems.

There are phases that begin with requirements collection /gathering; Analysis, Design, Implementation, Testing, Deployment and Maintenance support.

2). What is software testing?

Testing is the process of evaluating a system or its components with intent to find whether we got the results as per specific requirement or not. This activity results in expected and actual differences between the requirements.

Testing is a process that we execute on software to identify any gaps, any errors or missing requirement or any bugs in the software.

It’s a process consisting of all life cycle for developing a software both in a static and in dynamic way, were static testing relating after designing phase and dynamic testing relates to after coding phase. The work starts to give life to a software that designs in the steps require like planning, implementing, designing architectural structure, then it comes to coding phase where design in technically coded, formatted and then deliverable project is made.

If I break this definition, it starts with Process part where there are different steps to follow then it goes to lifecycle activities where SDLC came in contact with where testing starts right from the first step of SDLC, where functional nonfunctional requirements checks then it comes test basis, test cases to check for software fulfilling the requirement. Testing goes into parts such as static and dynamic testing that helps in maximizing the quality of software, helps in presenting the best possible way for software fulfilling deliverable.

3). What is agile methodology?

Agile method is a combination of Iterative and incremental methodology, the process we take in a part and after finishing specific requirement or adaptability the end result of product given to customer as per requirement in the deliverable of working software.

There are builds we focus on agile, small parts or small components that focus on one task at a time then merge it in incremental builds. These builds are process like iteration, repetition.

There are teams simultaneously working in iterative build for giving fast-paced product to beat today’s fastest growing industries and the customer requirement are changing also, so customer will be able to see working software and do the change which is adaptable in this method.

Iteration teams working to build software in planning, requirement, design, Coding, unit testing and accepting testing that works as cross functional teams that results in displayed working software for customer and for important stake holder.

Agile model believes that every project has different requirements, their nature of changing capacity is different so every project should be handled differently by tailoring the existing methods to best suit a project as per requirement. The best part of agile is working in small time boxes that divide the tasks into small frames to deliver specific features for a working software release.

For working software Iteration approach is followed to build working software. Each build goes as a process of incremental detailing features resulting in all featuring holding to build for deliver product.

4). Explain Phases of the waterfall model?

Water fall model is a classic method in which each phase completes their task then only other phase will get started for developing software. For example: if requirement gathering is finished its task, then only analysis phase will work on this project.

This model as some condition like requirement has to be fixed or frozen of customer.

The phase of developing software is likewise the other model. The most important phase is gathering information. Requirement gathering by the customer has to be correct, unambiguous, and fixed.

Second phase is the analysis phase that what functionality requires no matter how it will get accomplished. In this phase analysis done what is the requirement as per gathering information phase. From that documentation is build of a software. Implementation is done in this phase of what and how it will be done. Product definition is stable in this phase.

Third phase is the designing phase, it will create an architectural structure for a software based on the analysis phase. Technology is understood which is limited not in dynamic part, to get clearance for another phase.

Fourth will be coding phase, developer will do the coding on the designing phase, requirement is fixed so as per that phase will work for working software.

Five will be Testing phase, for tester there will be limited task, because every information is static then work will get minimal so this phase there will be no exhaustive testing.

Then deployment part, whatever the last phase of testing will go for this phase, the deliverable software and working software will be handed over to the customer.

5). Explain working methodology of agile model and also write pros and cons?

The Agile method best suits today's time where customers will get working software faster and can change as per requirement also, so this method is based on flexibility and adaptability.

Pros

* Realistic approach for software development.
* Functionality developed fast and customers can see the demonstration early.
* Good model for changing nature of requirement, for example environment base and also recommended for fixed requirement.
* Promoting cross functional training along with teamwork.
* Resources require minimal as per build execution.
* Easy to manage build, where little planning will work.
* For developers this method is very flexible.

Cons

* For complex dependencies this model is not suitable.
* Agile nature of iterative faces more risk of sustainability, maintainable.
* The agile nature of incremental faces more risk of extensibility.
* Overall planning is a must which has to be discussed in between agile leader and PM practices otherwise this model will not work properly.
* Delivery management strictly detects the scope, the functionality as per requirement and delivers the project faster to manage the deadlines.
* Model is dependable on customer interaction, if customer is not clear then team will drive in wrong direction for completing task.
* As documentation is very less this model heavily depends on team for generating working product as per requirement.
* Challenging this while transferring this to new team members due to lack of documentation.