

21/09/2020

DevOps Assignment - 1

(i) What are 4Ps in the Software Engineering?

→ The 4Ps in Software Engineering are:-

- i) Product
- ii) Process
- iii) People
- iv) Project

i) Product :-

It is the deliverable of the project. The project manager should define the product scope to ensure a successful outcome as well as technical hurdles that may encounter.

ii) Process :-

The second P of software engineering is process. Project managers and team members should have a methodology and plan that outlines their approach. Without a clearly defined process, team members will not know what to do and when to carry out project activities.

iii) People :-

Identifying the roles people play in almost any given project is the first step to a successful project. People are the primary resource on every project, and a well-managed team can greatly increase the chances for success.

i) Project:-

→ The fourth and final P of software engineering is Project. This is where the project manager's roles and responsibilities come into play. He or she must guide team members to achieve the project goals and objectives.

Q2) Define Software Engineering:-

- Software engineering is an engineering branch associated with development of software product using well-defined scientific principles, methods and procedures. The outcome of Software engineering is an efficient and reliable software product.
- Software engineering is the systematic application of engineering approaches to the development of software.
- Software engineering is the practical application of scientific knowledge to the creative design and building of computer programs. It also includes associated documentation needed for developing, operating and maintaining them.

Q3) What are the important phases in the Waterfall SDLC model?

→ The waterfall model was the first process model to be introduced. It is very simple to understand and use. In a waterfall model, each phase must be completed before the next phase can begin and there is no overlapping in the phases.

→ There are ~~six~~ six important phases in the waterfall SDLC model:-

- i) Requirements
- ii) System Design
- iii) Implementation
- iv) Integration and Testing
- v) Deployment of System
- vi) Maintenance

i) Requirements:-

→ The first phase involves understanding what needs to design and what is its function, purpose, etc. Here, the specifications of the input and output or the final product are studied and marked.

ii) System Design:-

→ The requirement specifications from the first phase are studied in this phase and system design is prepared. System design helps in specifying hardware and system requirements and also helps in defining the overall system architecture.

iii) Implementation :-

- > With inputs from system design, the system is first developed in small programs called units, which are integrated into the next phase. Each unit is developed and tested for its functionality which is referred to as unit testing.

iv) Integration and Testing :-

- > All the units developed in the implementation phase are integrated into a system after testing of each unit. The software designed, needs to go through constant software testing to find out if there are any flaws or errors.

v) Deployment of System :-

- > Once, the functional and non-functional testing is done, the product is deployed in the customer environment or released into the market.

vi) Maintenance :-

- > This step occurs after installation, and involves making modifications to the system or an individual component to alter attributes and improve performance.

Q4) What are advantages and disadvantages of the Waterfall Model?

Advantages:-

1. The advantage of waterfall development is that it allows for departmentalization and control.
2. The waterfall model progresses through easily understandable and explainable phases and thus it is easy to use.
3. It is easy to manage due to the rigidity of the model - each phase has specific deliverables and a review process.
4. In this model, phases are processed and completed one at a time and they do not overlap. The waterfall model works well for smaller projects where requirements are very well understood.

Disadvantages:-

1. It is difficult to estimate time and cost for each phase of the development process.
2. Once an application is in the testing stage, it is very difficult to go back and change something that was not well-thought-out in the concept stage.

3. Not a good model for complex and object-oriented projects.

4. Not suitable for the projects where requirements are at a moderate to high risk of changing.

Q5) What do you mean by Scrum?

→ Scrum is a subset of Agile. It is a lightweight process framework for agile development, and the most widely-used one.

→ A "process framework" is a particular set of practices that must be followed in order for a process to be consistent with the framework.

→ For example, the Scrum process framework requires the use of development cycles called sprints, the XP framework requires pair programming and so forth.

→ "Lightweight" means that the overhead of the process is kept as small as possible to maximize the amount of productive time available for getting useful work done.

Q6) What are roles and responsibilities of the Scrum Master and Product Owner?

Scrum Master:-

- The Scrum master is the keeper of the process. The scrum master is responsible for making the process run smoothly, for removing obstacles that impact productivity and for organizing and facilitating the critical meetings.
- Teach the Product Owner how to maximize return on investment(ROI) and meet his/her objectives through Scrum.
- Improve the productivity of the development team in any way possible.

Product Owner:-

- The product owner is the keeper of the requirements. The Product Owner provides the "single source of truth" for the Team regarding requirements and their planned order of implementation.

- Product Owner buffers the Team from feature and bug-fix requests that come from many sources, and is the single point of contact for all questions about product requirements.
- Product Owner works closely with the team to define the user-facing and technical requirements, to document the requirements as needed and to determine the order of their implementation.

(Q7) What is pair programming?

- Pair programming is where two developers work using only one machine. Each one has keyboard and a mouse.
- One programmer acts as a driver who codes while the other will serve as observer who will check the code being written, proofread and spellcheck it.
- These roles can be switched at any time: the driver will then become the observer and vice-versa.
- Pair programming helps the programmer learn from each other while coming up with programs and applications with better quality code and fewer bugs.

Q8) What are the phases in the DevOps?

→ DevOps brings together people, processes and products, automating software delivery to provide continuous value to your users.

1. Plan:-

> The plan stage covers everything that happens before developers start writing code, requirements and feedback are gathered from stakeholders and customers and used to build a product road map to guide future development.

2. Code:-

> In this phase, the developers in the team that are developing the project starts coding the modules that are assigned to them.

3. Build:-

> The build phase is where DevOps really kicks in. Once a developer has finished a task, they commit their code to a shared code repository.

4. Test:-

> Once a build succeeds, it is automatically deployed to the staging environment for deeper, out-of-band testing.

5. Release:-

- The release phase is a milestone in a DevOps pipeline - it's the point at which we say a build is ready for deployment into the production environment. By this stage, each code change has passed a series of manual and automated tests, and the operations team can be confident that breaking issues and regressions are unlikely.

6. Deploy:-

- Finally, a build is ready for the big time and it is released into production. There are several tools and processes that can automate the release process to make releases reliable with no outage window.

7. Operate:-

- The new release is now live and being used by the customers. The operation teams is now hard at work, making sure that everything is running smoothly.

8. Monitor:-

- The final phase of the DevOps cycle is to monitor the environment. This builds on the customer feedback provided in the Operate phase by collecting data and providing analytics on customer behaviour, performance, errors and more.