**GIT**

1.What is GIT and its significance in SDLC.

GIT is a source code management tool in software development.The Git have repository which has full version tracking capabilities and independent of network access. Developers can push codes into a cloud repository using GIT and others can access it. It is very useful in the case of code merging. If a source code is coded by two or three developers and they can push these codes to a cloud repository and a senior developer can merge these codes to a single piece of source code. It is very useful to connect developers to share their code.

2.What is the difference between GIT and SVN?

**GIT**

* GIT is a distributed version control system
* GIT uses multiple repositories including a centralized repository and server, as well as some local repositories.
* GIT do not require any network to perform git operation
* In GIT we create only .git directory
* GIT have more concepts and commands

**SVN**

* SVN is a centralized version control system.
* SVN does not have a centralized repository or server.
* SVN require Network for running SVN operations
* In SVN we create .svn directory in each folder
* SVN is much easier to learn as compared to GIT.

3. What are the advantages of using GIT?

* Perfomance – GIT provides the best performance when it comes to version control systems. Committing, branching, merging all are optimized for better performance than other systems
* Security – GIT handles security with cryptographic algorithm and it manages versions, files, and directory securely to ensure it is not corrupted.
* Branching model – GIT provides multiple local branches which are independent of each other.
* Staging area – GIT has an intermediate stage between our working directory and our repository called staging area where commits can be formatted and modified before completing the commit.
* Distributed – GIT is distributed so that the repository/complete code base is mirrored onto the developer’s system so that he can work on it only.
* Open source – GIT is an open source Version control system. So it is possible for developers from all over the world to contribute to the software and make it more powerful through features and additional plugins.

4. What is “Staging Area” or “Index” in GIT?

Staging area or Index is an intermediate stage between our working directory and our repository in GIT where commits can be formatted and modified before completing the commit.

5.What is GIT stash?

If we are working on an incomplete part of our project and we want to switch branches, but we don’t want to commit of half-done work. Technically it was not possible but GIT Stashing will allow us to do so. Git Stash command enable us to switch branches without committing the current branch. Generally stash means “store something safely in a hidden place” and it is also same for Stash in GIT. It temporarily saves data without committing.

6.What is the function of git clone?

Git clone is a Git command primarily used to point to an existing repository and make a clone or copy of that repository at in a new directory, at another location. The original repository can be located on the local filesystem or on remote machine. Git clone command copies an existing Git repository into a new directory and it has its own history, manages its own files, and is a completely isolated environment from the original repository

7.How can you create a repository in Git?

Git repository is the virtual storage of our projects where we can save versions of our code and can access it when needed. To create a new repository, we use “git init” command. It will creates a new .git subdirectory in our current working directory. It will also creates a new master branch.

8.What is the purpose of branching in GIT?

Branches are the copies of our main source code. At first when we initialise a repository a branch master is created. By default, every commit that we made will be added in the master branch. But we can make another branch which will have the copy of the source code in the master branch. As a result we can work on the copied source code of newly created branch separately. And those changes will not affect the source code in the master branch. We can do individual commits for each branch.

9.What is the difference between ‘git remote’ and ‘git clone’?

Both are completely different commands. git remoteis used to refer to a remote repository/central repository but git clone is used to clone an existing repository to another directory

10.What is the function of ‘git diff ’ in git?

Git diff command is used to track the difference between the changes made on a file. It takes two input data sets and outputs the changes between them.

11.Explain what the commit message is?

A commit message explains what change we made to our project. It is used to save changes to the a local repository. We can consider commits as a snapshot of our entire repository.

12.Why is it advisable to create an additional commit rather than amending an existing commit?

Git amend internally creates a new commit and replaces the old commit. So there is a chance for eliminating data if we amend existing commit instead of creating a new commit.

13.What is Rebasing

It is a process of combining a sequence of commits to a new base commit. It is much similar to merge command but here it rewrites the commit history.

**Maven Fundamentals**

1.Explain what is Maven? How does it work?

Maven is a very powerful enterprise tool/ build tool which helps in building, documenting and managing a project. It is written in java and it is based on POM (Project Object Model). Maven mainly used for building and managing a java project. Maven simplifies the process of building a project. It helps in downloading dependencies for the project automatically.

2.Explain what is POM and its significance

It is an XML file that has all the project information and configuration details used by maven to build the project. It also includes the plugins used by the maven in the project. While executing a task, maven look for the POM it reads POM and takes the needed configuration information and then runs the task.

3.Explain what a Maven artifact is?

An artifact in maven is a resource generated by a maven project. Each maven project can have exactly one artifact like a JAR,WAR,EAR. The POM file describes how the artifact is build. Each artifact has a group ID (like com.name.package), an artifact ID (a name), and a version string

4.List out the dependency scope in Maven?

Maven dependency scope attribute is used to specify the visibility of a dependency, relative to the different lifecycle phases (build, test, runtime etc). Maven provides six scopes, they are:

* Compile
* Provide
* Runtime
* Test
* System
* import

5.List out what are the build phases in Maven?

Maven has mainly seven phases in its Build Lifecycle**.**

* Prepare - resources - Resource copying can be customized in this phase
* validate - Validates if the project is correct and if all necessary information is available
* compile - Source code compilation is done in this phase
* Test - Tests the compiled source code suitable for testing framework
* Package - This phase creates the JAR/WAR package as mentioned in the packaging in POM.xml
* Install - This phase installs the package in local/remote maven repository
* Deploy - Copies the final package to the remote repository

6.Mention the three build lifecycle of Maven?

* Clean Lifecycle
* Default Life cycle
* Site Lifecycle

7.List out what are the aspects does Maven Manages?

Maven handles the following aspects:

* Build.
* Documentation.
* Reporting.
* Dependencies.
* SCMs.
* Releases.
* Distribution.
* Mailing list.

8.Explain what a Maven Repository is? What are their types?

A repository in Maven holds build artifacts and dependencies of varying types. There are exactly two types of repositories: **local** and **remote**.

* local repository is a directory on the computer where Maven runs. It caches remote downloads and contains temporary build artifacts that you have not yet released.
* remote repositories refer to any other type of repository, accessed by a variety of protocols such as file:// and https://.

9.Explain how you can exclude dependency?

Exclusions are set on a specific dependency in your POM, and are targeted at a specific groupId and artifactId. We can exclude adependency by adding <exclusions> element inside the <dependency> element and provide the groupId and artifactId inside their tags.

10.For POM what are the minimum required elements?

* project root.
* modelVersion - should be set to 4.0. ...
* groupId - the id of the project's group.
* artifactId - the id of the artifact (project)
* version - the version of the artifact under the specified group

**CI/CD**

1.What are the fundamental differences between DevOps & Agile?

**Agile**

* Agile refers to an iterative approach which focuses on collaboration, customer feedback, and small, rapid releases.
* Agile helps to manage complex projects.
* Agile process focusses on constant changes.
* Agile method can be implemented within a range of frameworks like a sprint, safe and scrum.
* Feedback is given by the customer.
* It focuses on functional and non-function readiness

**DevOps**

* DevOps is considered a practice of bringing development and operations teams together.
* DevOps central concept is to manage end-to-end engineering processes.
* DevOps focuses on constant testing and delivery.
* The primary goal of DevOps is to focus on collaboration, so it doesn't have any commonly accepted framework
* Feedback comes from the internal team
* It focuses more on operational and business readiness.

2.What is the need for DevOps?

DevOps is important because it's a software development and operations approach that enables faster development of new products and easier maintenance of existing deployments.

3.What are the advantages of DevOps?

* Faster delivery of features.
* More stable operating environments.
* Improved communication and collaboration.
* More time to innovate rather than fix/maintain.
* Continuous software delivery
* Less complexity to manage
* Higher employee engagement.

4.Explain with a use case where DevOps can be used in industry/ real-life.

Lots of industries are operating DevOps software at present. Consider a simple/local e-commerce website which sells factory-manufactured items.It followed waterfall model so it struggles with many problems like site updates that frequently causes site to go down. It can affect its sale. With the help of DevOps team it can change from waterfall to Agile approach which have a fully automated deployment pipeline and its continuous delivery practices which can result in more than 50 deployments in a day.

5.What are the success factors for Continuous Integration?

Continuous Integration can be defined as Building software and taking it through as many tests as possible with every change. The main factors for continuous integration are:

* Compilation.
* Unit Tests.
* Code Quality Gates.
* Integration Tests.
* Deployment.
* Chain Tests.

6.What are the differences between continuous integration, continuous delivery, and continuous deployment?

**Continuous Integration**

Continuous Integration (CI) is a development practice that requires developers to integrate code into a shared repository several times a day. Each check-in is then verified by an automated build, allowing teams to detect problems early. By integrating regularly, you can detect errors quickly, and locate them more easily

**Continuous Delivery**

Continuous delivery is a practice of automating the entire software release process. The idea is to do Continuous Integration, **plus** automatically prepare and track a release to production. The desired outcome is that anyone with enough privileges to deploy a new release can do so at any time in a few clicks. By eliminating nearly all manual tasks, developers become more productive.

[**Continuous Deployment**](https://semaphoreci.com/community/tutorials/elements-of-a-continuous-deployment-workflow)

[Continuous Deployment](https://semaphoreci.com/community/tutorials/elements-of-a-continuous-deployment-workflow) is a step up from Continuous Delivery in which every change in the source code is deployed to production automatically, without explicit approval from a developer. A developer’s job typically ends at reviewing a pull request from a teammate and merging it to the master branch.

7.What role does the Quality Assurance (QA) team play in DevOps?

Quality Assurance (QA) ties together development and operations and enables them to collaborate to build software and applications. QA plays a strategic role in ensuring that quality is taken up as a responsibility by both Development and Operations. In DevOps, QA detects bugs in the project earlier. So bug fixing become more convenient.

8.Describe an efficient workflow for continuous integration.

For continuous integration, there is a need to have a repository where in the code could be saved, retrieved and maintained. The repository must be good enough to provide the developers with a powerful version controlling system.

Git is one of the version control systems (VCS). It is primarily used for source code management in software development, but it can be used to keep track of changes in any set of files. Let’s consider Centralized work flow provided by GIT.

**Centralized workflow**

This flow uses a central repository to serve as the single point-of-entry for all changes to the project. The default development branch is called master and all changes are committed into this branch. This workflow doesn't require any other branches besides master. Here Developers start by cloning the central repositories in their own local copies of the project. They edit and commit changes locally. Once the changes are tested, the developer "Push" their local master branch to the central repository. But we can say that centralized workflow is more like SVN with some GIT features.

9.What are the best practices for DevOps implementation?

* Agile project Management
* Shift left with CI/CD
* Build with the right tools
* Implement automations
* Monitor the DevOps pipeline and application
* Gather continuous feedback
* Observability

10. How will you approach when a project needs to implement DevOps?

DevOps is a practice that brings development, operations and testing personnel together in cross-functional teams, each of which is responsible for the entire lifecycle of a software product or service. By bringing together collaborative teams across the organization, DevOps creates a stable operating environment for bringing code to market faster, reducing human errors and bugs, enhancing version control and optimizing costs. If we need to implement DevOps in a project, we need to follow these given steps.

* Prepare for cultural shift
* Follow Agile Principles
* Create a continuous Integration, Continuous Delivery platform
* Create a continuous testing environment
* Establish a Continuous deployment system
* Continuously monitor performance
* Make use of Customer feedbacks