1. **What is waterfall methodology?**

* The waterfall model is software development process in a liner sequential flow.
* Means that any phase on the development process begins only if the previous phase is complete.

**Steps**:

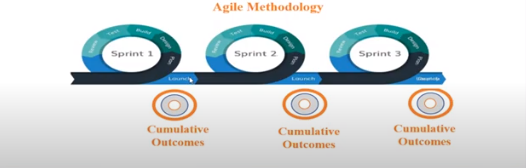
Gathering requirements🡺Design🡺Development🡺Integration&Testing🡺Deployment🡺Maintenance.

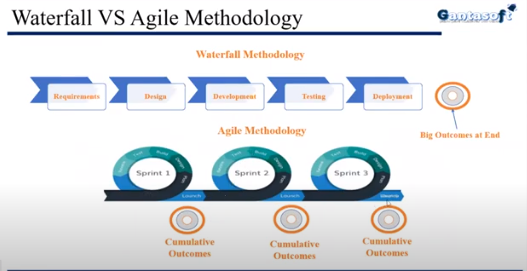
1. **When to use Waterfall Methodology?**

* Requirements are not changing frequently.
* Application is not complicated.
* Project is short.
* Requirement is clear.
* Environment is stable.
* Technology and tools used are not dynamic and is stable.

1. **What is agile methodology?**

* Agile methodology is a practice that promotes continues integration of development and testing throughout the software development lifecycle of the project.
* Both development and testing activities are concurrent unlike the waterfall methodology.





1. What is DevOps?

* DevOps is collaboration between development and operation teams which enables continues delivery of applications and services to our end users.



1. **Benefits of DevOps?**

* Dev + Ops i.e. collaboration between developers and operations.
* Errors can be fixed at initial stage.
* Reduce the implementation time of ne services from hours to minutes.
* Easy deployment.
* Increase the customer satisfaction.
* A satisfied customer always mean more business and that’s the reason Organization’s need it.

1. **DevOps diagram with various DevOps Tools?**

* **Plan**: Planning purpose we are using the **Jira** and Valley.
* **Code**: For this we are using **Git**, SVN (Subversion) (old Tool) & TFS (Team foundation Server) (Microsoft) – Majorly we are using Git because easy to push and pull.
* **Build**: For this we are using **Maven**(Most popular), Gradle(Most popular) and Ant(Very old one)
* **Test**: For this we are using JUnit and SE – we are not maintaining and testing team taken care.
* **Release**: For this we are using bamboo, team city and **Jenkins** (CI/CD tool) – it is continues integration, deployment and delivery and open source and plugin based.
* **Monitor**: For this we are using **Splunk** and Nginx.
* **Operations**: For this we are using **kubernetes** and docker.
* **Deployment**: For this we are using **Ansible** (Most power full because its agent less) and chef.



1. **Version Control System(VCS):**

* VCS allows you to track the history of a collection of a files.
* It supports creating different versions of a collection.
* Each version captures a snapshot of the files at a certain point in time and the VCS allows you to switch between these versions.
* Different types of VCS are:
* **GIT**
* SVN
* TFS

1. **Build tools**:

* Build tools are programs that automate the creation of executable applications from source code (Ex: .apk for android app).
* There are different types of builds tools available:
* **MAVIN**
* GRADLE
* ANT

1. **Artifactory tools**:

* It is used to store the artifacts (binaries).
* There are different types of artifactory tools available:
* **Nexus**
* Jfrog artifactory

1. **Code quality tools**:

* It detects the bugs and vulnerabilities with showing the bugs as grade system.
* It shows the code coverage percentage too.
* There are different types of code quality tools available:
* **Sonarqube**.
* PMD

1. **CI/CD(Continues Integration/Continues Delivery) Tools**:

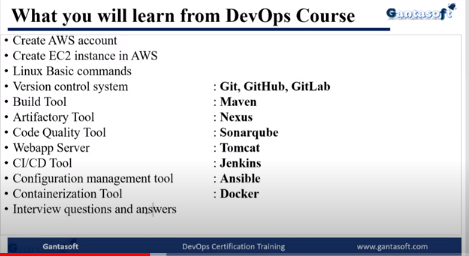
* A CI/CD is the backbone of the modern DevOps environment.
* It bridges the gap between development and operations teams by automating the building, testing and deployment of applications.
* There are different types of CI/CD tools available:
* **Jenkins**
* Bamboo
* Team city

1. **Configuration Management tools**:

* It enables changes and deployment to be faster, repeatable, scalable, predictable and able to maintain the desired state, which brings controlled assets into an expected state.
* There are different types of configuration management tools available:
* **Ansible**
* Chef
* Puppet

1. Containerization tools:

* It is allows developers to create and deploy applications faster and more securely.
* With traditional methods, code is developed in a specific computing environment which, when transfer to a new location, often results in bugs and errors.
* There are different types of containerization tools available:
* **Docker**
* Rocket



**Extra to learn as below**:

1. AWS
2. Terraform---------> IAC (Infrastructure as a code) – Mohamed – Udemy
3. Python scripting ------------> Narendra(Udemy) – Telugu web guru – YouTube
4. Kubernetes