**DevOps**

Waterfall model :

It has distinct goals for each phase of development. Imagine a waterfall on the cliff of a steep mountain. Once the has flowed over the edge of the cliff, it can not turn back.

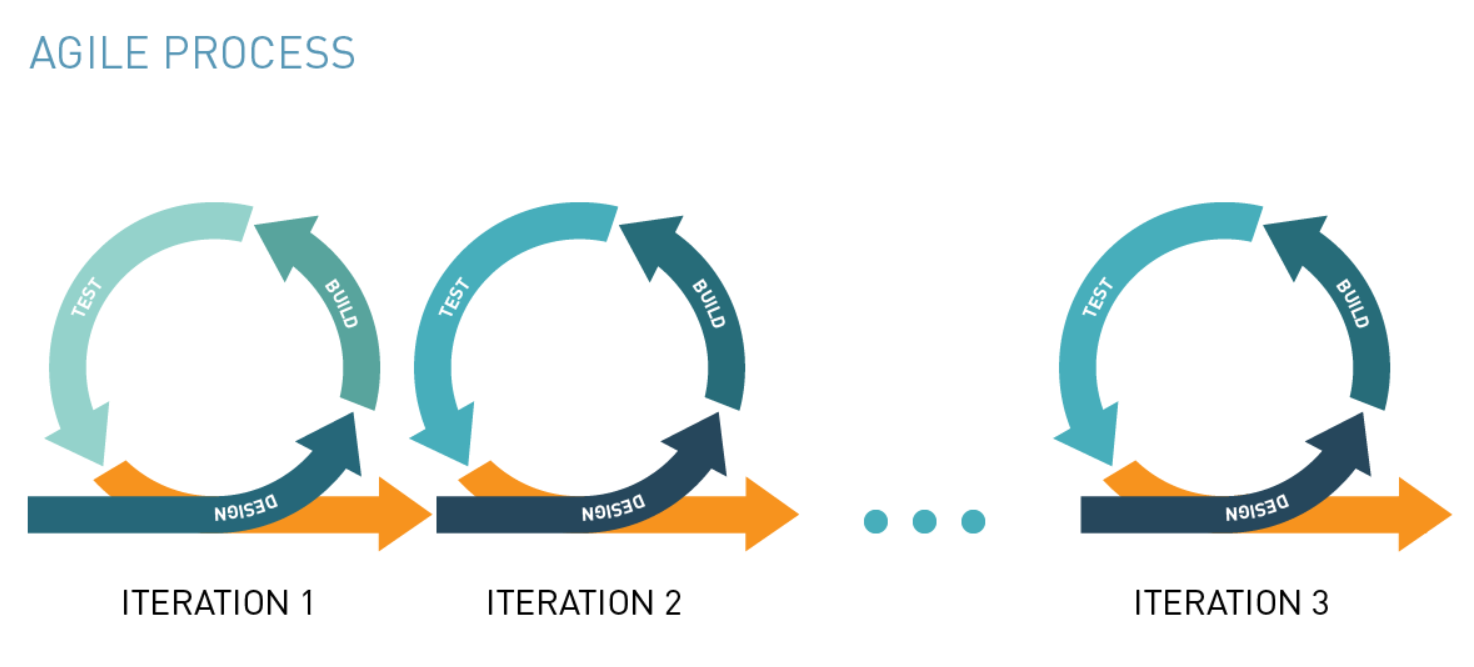
Requirement gathering & analysis 🡺 Design 🡺Implement 🡺Test 🡺 Deploy 🡺 Maintenance

Disadvantages of waterfall model:

* Once the 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.
* No working software is produced until late during the life cycle.
* Not a good model for complex and object oriented projects.
* Not suitable for the projects where requirements are at a moderate to high risk of changing.
* High amount of risk and uncertainty.

Agile methodology :

* In the agile methodology each project is broken up into several iterations.
* All iterations should be of the same time duration (between 2 to 8 weeks)
* At the end of each iteration, a working product should be delivered.
* So agile methodology breaks down software delivery life cycle in to several iterations or sprint involving continues delivery and testing of the software delivery .



Limitations of Agile 🡺

* Codes works in developer’s system but not in production.

Devops :

Devops is a practice of operations and development engineers.

Devops stages:

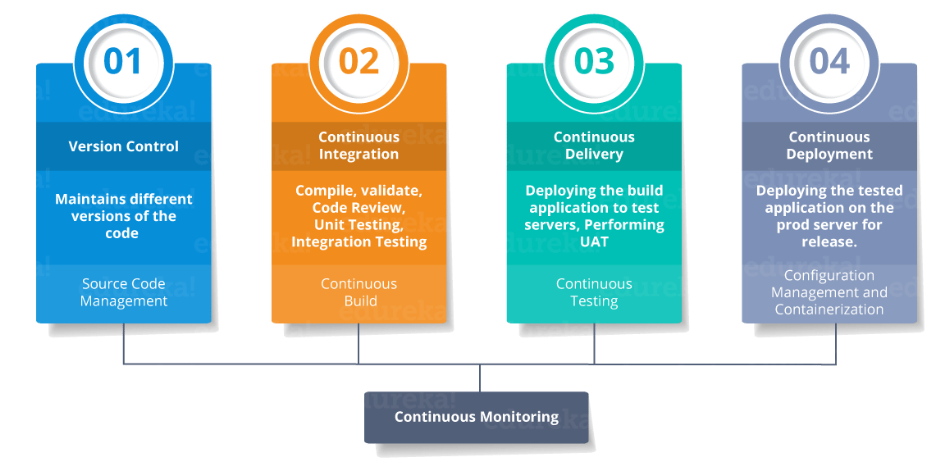
**Version control🡺continues integration 🡺 continues delivery 🡺 continues deployment**

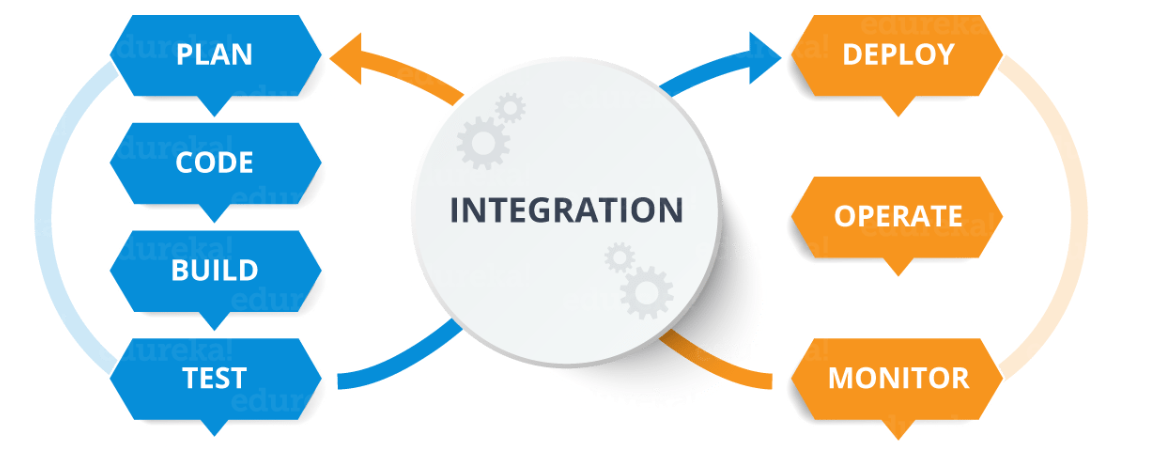
Version control : maintains different versions of the code.

Continues integration : compile, validate, code review, unit testing, integration testing, packaging

Continues delivery: deploying the build application to test servers, performing UAT

Continues deployment: deploying the tested application on the prod server for release.





Plan & code : git, subversion, jira

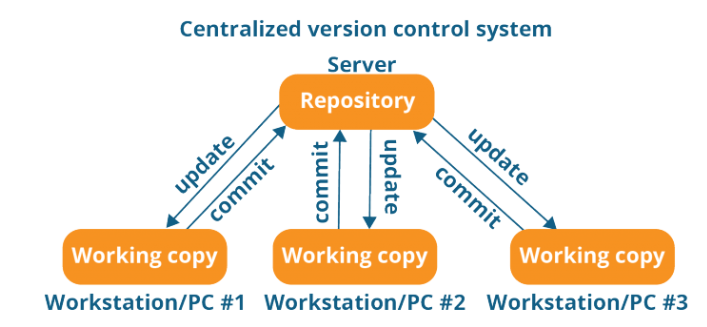
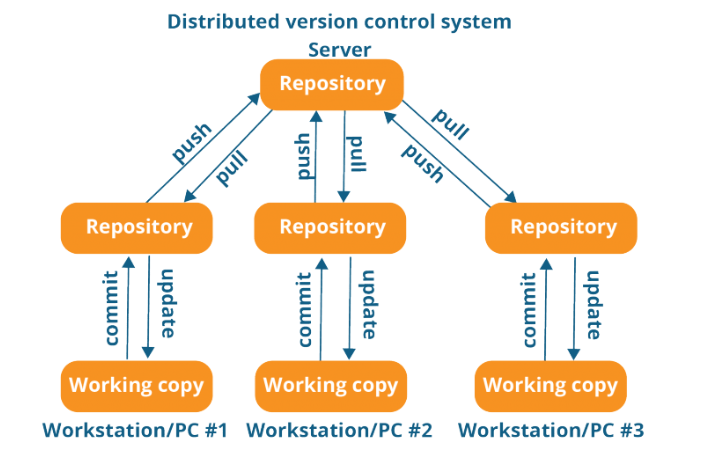
Build : maven

Test : Junit, gradle

Integration : Jenkins

# Source code management : GIT

Version control system is of two types.

* Centralized version control system
* Distributed version control system

Creating repository 🡺

mkdir devops

git init 🡺 initialize git repository

git status 🡺 status of the git repository.

git add a.txt b.txt 🡺 add two files in staging area

git . 🡺 add all files

git commit -m “first commit comment here” 🡺 commit the latest changes to main branch

Syncing Repository 🡺

git remote add origin <https://github.com/nihar/devops> 🡺 adding main repository to your command online

git push origin master 🡺 pushing master branch files to main repository

Clone repository from origin to local 🡺

git clone https://link to be copied from git hub repository

git pull origin master 🡺 getting files from origin to your local system

git clone is used when you copied whole repository. Git pull used when you update the latest commit in the main repository.

Parallel development 🡺

git branch feature1 🡺 creating a branch named feature1

git branch -D feature1 🡺 deleting branch feature1

git checkout feature1 🡺switching to feature1 branch

git checkout master 🡺 switching to master branch

git push origin feature1 🡺 pushing the feature1 branch to remote repository

git log 🡺 display the log

git stash 🡺 it will stop reflecting uncommitted modified branch files into master

git stash pop 🡺 it will back to working mode where you will able to see uncommitted files.

git revert 123bbhyy555commitid 🡺 it will revert to respective commit id

git checkout 123gh67zjjjcommitid 🡺 it will go to respective commit id

git diff 1234commit1 45678commit2 🡺 it will display the difference between the two commits

git diff HEAD . 🡺 difference between latest commit and current file

# Continuous Integration – Jenkins

# 

What is continuous integration : Continuous integration is nothing but integrating all devops cycle tools

Jenkins is a free open source Continuous Integration tool and automation server to monitor continuous integration and delivery. It is written in Java.

It is known as an automated Continuous Delivery tool that helps to build and test the software system with easy integration of changes to the system. Jenkins follows Groovy Scripting.

Also, it enables developers to continuously check in their code and also analyze the post-build actions. The automation testers can use to run their tests as soon as the new code is added or code is modified.

Advantanges of Jenkins :

* Automated build and testing
* Instant feedback to developer
* Low risk and faster delivery
* frequent commits and small feature release

Managing Nodes on Jenkins 🡺

Nodes are nothing but different server, on which jobs will be executed or integration will happen

* Go to manage Jenkins > configure global security > Agents
* Click on Random
* Go to Manage Nodes > New node
* Remote root directory > mention the directory name where you want to keep all your repo
* launch method : jave webstart > save

agents.jar and slave-agent.jnlp

* login to the server that you want to make a node.
* put the agents.jar and slave-agent.jnlp file in the server in the location you have mentioned in the Jenkins setup process for this node.
* go to slave-1 or node-1 or node that you have created in Jenkins, you would get one command shown to be executed in agent slave. Copy that and execute in created note server.

**Pipeline jobs in Jenkins**

**- Delivery pipeline**

**- Jenkins pipeline**

**Delivery pipeline :**

* **you can create multiple freestyle project or any job.**
* **once you create jobs, go to configuration of jobs (each job).**
* **go to build trigger section and check build after other jobs are built.**
* **install delivery pipeline plugin to visualize the pipeline jobs**
* **got to view, create a view > delivery pipeline view> enter view name > got to pipeline component mention the initial job .**

**ex: we have created 3 jobs. job1 job2 job3**

**we can go to job3 and set trigger as build after other jobs are build and mention job2.**

**like wise we can setup for job 2.**

**Jenkins Pipeline** is a combination of plugins that supports integration and implementation of continuous delivery pipelines. A Pipeline is a group of events interlinked with each other in a sequence.

Jenkins File :

* Jenkins pipelines can be defined using a test file called JenkinsFile.
* You can implement pipeline as code using JenkinsFIle and this can be defined by using a domain specific language (DSL).
* in Jenkinsfile you can write the steps needed for running a jenkins pipeline.
* JenkinsFIle can be defined by either webUI or with a Jenkins File.

Types of pipeline syntax :

**There are two types of Jenkins pipeline syntax for defining your JenkinsFile.**

* **Declarative**
* **Scripted**