SDLC

SDLC

-------------------------------------------------------------------------------------------------

WATERFALL MODEL:

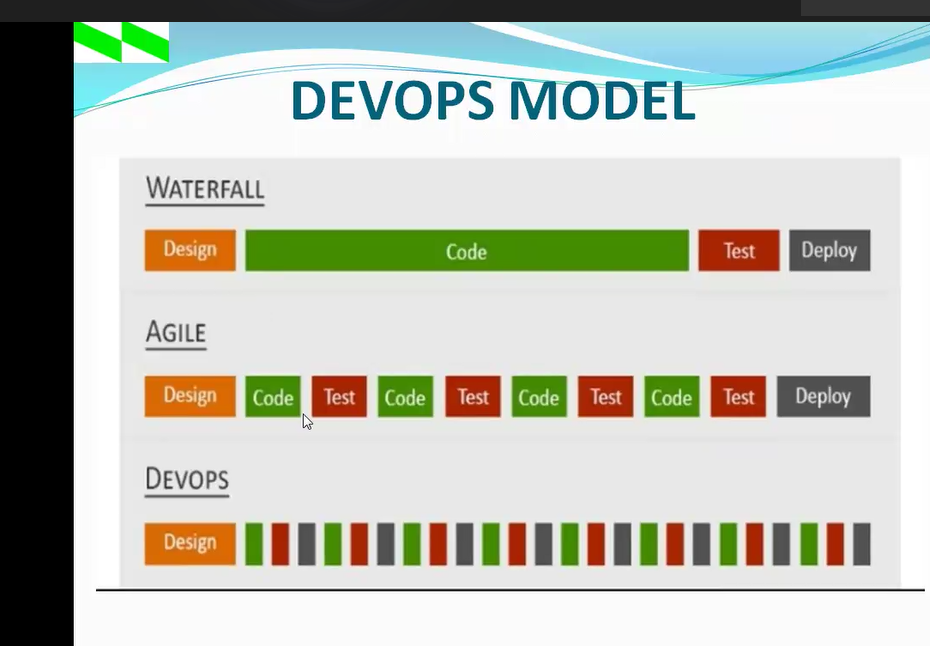
requirements->design->implementation->testing->deployment->maintenance

AGILE:

at last deployment happens

--------------------

devops:development+testers+operations together at same cycle



Git/big bucket(version contraol system)- maintains the original source code.

Devops- used to perform automated operations.

Source: GitHub

Cicd(continuous integration, continuous delivery)-Jenkin tool to pull the code and integrate

Apache maven- build tool (used for java)-gradel,ant are also some tools for building the code

Refactor tool -nexus& jfrog these stores executable codes.

Checking quality of code- sonar qube

To monitor the application, we use: Nagios, data dog, dynamics (monitoring tool)

To work the code in all platform then make the application as image this tool called containerization tool(this is docker)

We can make application as a image using docker and run application as a containers

I am running my img as a container, if I stop the container the app is not available sometimes and sometimes available , here docker volume is the concept, we make the mounting point even after del available the containers

Creating same user in 100 servers? Run the script , we need all 100 servers info to create(server name,domain tool)

We can install,uninstall, del user,create user with the help of configuration mgmt. tool ( ansible,chef,operate), with this we can perform automation for all servers

**Kubernetes(**managing the containers)(container orchestration tool performs deployment, load balancing, roll out ,roll back,scaleout, scalein operations) – all the servers are interconnected , if I send a req to one server that req will reach an one of the server

**Devops**

deployment is done frequently

Change volume

Deployment tym

Failed deployment rate

Tym to detection

Lead tym

Availability

Cycle tym

Mttr(mean tym to recovery)

**CREATE GITHUB ACCOUNT**

**Watch at 2:50:00**

Go to githb

Go to repository

+symbol->new repository(watch at 2:58:00

Git log – log info about commit operations

Get log - - oneline // command

Get branch -m main // tomove to the main branch

Pushing the code into github:

Make connectivity b/w remote to local

Get remote add origin link of new githubrepository created

Git remote -v

Git push -u origin main// push the code

Change the name: git rename oldname newname

**Pushing the code using tokens,ssh two ways.**

**1]tokens:**

Using hhtps generate tokens

Click on profile

settings

Developer setting

Personal access tokens

Tokens classic

Generate new token

Give token name

->we can use any tym pf tokens like fine grained, classic

Generate token

Copy token and paste somewhere because only one tym available

**Watch at 3:19:00**

**2]SSH**

We should deberate ssh key

Ssh-keygen

cd.ssh/

copy the public key

cat id\_rsa.pub

click on profile

watch at 3:20:00

settings

ssh and gpp keys

new ssh key

give name

copy key in cmd and paste in key section

now we need to push the code

create one more repository

mkdir demo(new directory)

**pushing code**

cd demo/

ls -a

git init

ls -a

vi sample

//write code//

Cat sample

git add .//adds all the files

git add sample// only specific file

git commit -m “commit in sample”

git branch -m main

git remote add origin

select ssh in github and copy path

git remote add origin addpayh

git remote -v

// ssh paswd is not req but in https req

//if changing the mach after sometym we cannot push the code using the same key , we need to generate the new key again

// ssh more secure and userfriendly when compared to hhetps(tokens)