**Notes:**

1. **September 22nd: Devops video for reference.**

**Evolution of Software Development over the years:**

Before 15 to 20 years in Software, everything seems to be done by manual, where developers used to write codes, build and then deploy.

If we do everything by manual, then there are more chances for human errors and time consuming.

**For ex**: Facebook: If we update our status, it will be available within seconds around the world.

Let us think about Before 20 years, if we update the status in FB, some developer need to commit and deploy it manually. If there are 100 to 1000 servers depending upon the distribution, if the developers will deploy manually it takes lot of time.

But now a day’s status update takes seconds. So, this is happening because of Devops.

**Use Case: How FB uses Devops:**

Devops is all about using automation using agile process.

If we talk about Devops, we need to plan and integrate, build, testing, deliver release.

All these scenarios we need to be integrate in Devops.

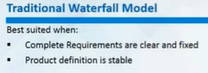
**What is Devops?**

Devops is development + Operations.

For example, we if have a project, from the starting of a project, from 1st day of the project, whoever involved in that project, Business analyst, Business management people, development people, System admin tem people, Operations team ,testing team people and support team will gather in a place and discuss about their plan. This is what is called as Devops.

In software development there are different models like Waterfall and Agile.

In Waterfall, it is step by step process. It takes lot of time. If we take 2 months’ time to write the code, and then the customer comes back and says about the requirements are changed, sometimes even the project also gets cancelled.



So, to overcome these issues in waterfall model, agile approach came into picture.



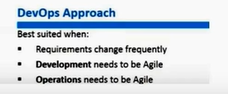
Agile process involved in each project, all the aspects will happen parellally.

They follow Scrum.

SO the same agile methodology we adopted for Devops.

If anyone says that we are using Devops methodology that means they are following agile methodology.

In agile methodology, most of the development happens.



Devops means Development and Operations both will be to be done in agile.

Definition of Devops: Devops is a strategy using development and operations team together work on a particular project from day 1.

Here the requirements of the project will always changes. Development and operations will be agile. Then only we can say it as Devops.

**Devops is a software development approach that involves continuous development continuous testing, continuous integration, Continuous deploy and continuous monitoring of the software throughout its development life cycle.**

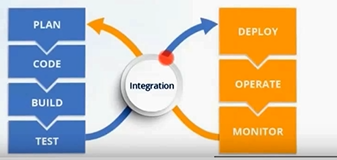
Continuous development: Developers will write their code continuously. They keep on changes to the particular code.

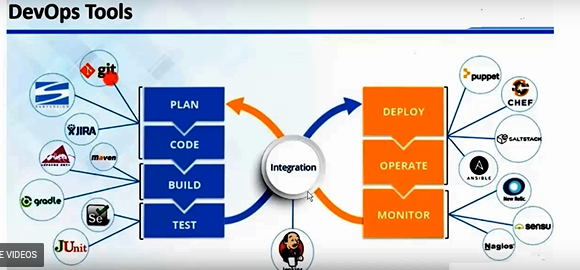
Continuous testing: The moment developers do the changes to their code, it need to be tested.

If this continuous integration and continuous testing doesn’t take place,

If we are following agile, for ex in 2 week time they do lot of codes and lot of changes, if they didn’t tested the particular code on the particular time, after going into the deployment process if they got errors they will never know in which place they are getting these error.

If they do continuous development, if as a developer we write any code and tested it got success… and after some time if we write another code and it got failed, we can easily know where the code failed and where the build failed.





Continuous monitoring: The server or the application should be monitored 24/7. It should be up and running always. We need to have some monitoring tools to achieve that monitoring.

So to Plan and code we need Version control tool. There are so many version controls tools available in the market like CBS, SVN, GIT, Clear case etc. Now market is occupied with GIT and SVN.

Version control tool is used to store your data or code to change or to track those changes.

When our code is ready in Version Control Tool in different files like XML formats which is not an executable files.

Build tool will compile our code it will test the test cases and package our application into an executable format like in windows .exe, .msi; in Linux it will be Jar files.

There are some build tools such as Maven, Gradle.

Testing will be done by testing teams but, sometimes we might do. It depend on the project requirements. Junit, Selenium will automate the testing.

For monitoring we have Nagios, Sensu, and New Relle etc.

There are some configuration management servers.

For example, we have 2 people in the project and almost 1000 servers. And need to install Java on those servers. It may take some hours or days if we do it manually. There is a possibility of mismatch the versions in those servers. Developer A might have installed version Java 1.7 and Developer might install java 1.8. Sometimes we need to give admin access or root access to team leads, we never know what changes they may have done in the server until we go and check it. In those 2 developers, let’s say one on leave he installed some and other new guy came in and wants to know what that guy installed, it will be problem, if we want to understand the installed version/ file .At that time the concept of configuration management came in .

Configuration management is an infrastructure as a code, whatever the infrastructure that we write code it in Python, Ruby, it depends on the tool that we take.

So, we will write in a simple file and execute that particular file in those 1000 servers, it will be a

5 min task. Now all those servers will have the same version. In future if I want to update the version, no need to login to 1000 servers, we can go to that particular file and change the java version and just deploy it. So, if the new person came, it’s easy for him to understand the process. He can just see the code in that particular file and he can easily understand the version that is installed.

If anyone want to do any changes to that particular server, it won’t be applicable to that server,

First we get a notification, we can check that and approve it, and then only the changes that others did will be applicable to that server. So, the control and configuration will be in our hands. So, it is secured and easy to easy to understand the process.

There are Configuration management tools such as Puppet, Chef, Ansible, Salt stack.

For Chef we need to have good understanding of Ruby.

Puppet mostly System admin will use it. We need basic understanding of Puppet.

Ansible is easier if we compare to Puppet or Chef.

But market is more occupied with Puppet.

We have Container tools like Docker

Cloud technologies like AWS.

Why we need a cloud:

For example we are working in a project which has almost 10,000 servers. So, these 10,000 servers need to have a backup servers as well that they call it as DR (Disaster recovery).

To manage these servers individually as 1 will act normal and other will be the backup.

To keep them operative or to manage we need to have a data center. Those are cost effective.

We need to maintain 2 datacenters one in 1 place and other in 1 place. Both need to have power backup 24/7, internet, AC, resources etc. we cannot shutdown those data centers for even a single min.

And we need Red hat Linux, windows machines .After working for 2 years on it and the project got over. We cannot use it for another. Because every server is occupied with the project. And whenever we buy a particular server, we need to pay it full. And to get these Red hat or Linux, it will take a week to get it.

To overcome all these issues, then Amazon came with cloud concept AWS, Pay for what you use.

AWS will give the resources, machines in 2min. They will take care of data centers, maintenance and etc. After we are done with project/ work just terminate it and pay for what you use.

Cloud is nothing but to access that particular application over the internet. There are different cloud operations are available Like Microsoft assure, google cloud, IBM blue mix.

Most of the market is occupied with AWS.

To do all these things we need a bride type of continuation tool called Jenkins.

We can integrate different tools that we use in our project,

For example there is a requirement whenever there is a change to happen in that particular repository in version control tool such as GIT, that particular change, that particular code will be build ,test and build and deploy into the application servers. It will be taken care by Jenkins.

Jenkins will first communicate with GIT and check for that repository changes and if any changes happen, then it will communicate with the build tool and it will build that code, then it talks to application servers, deploy it and sends an email notification like build failed or build successed.

This integration is automated. Before usually everything is done manually.

Now Jenkins will take care of it.