**DEVOPS INTRODUCTION**

Devops is a software engineering and culture and practice that aims at unifying software development and software operation.

The main characteristics of the Devops movement is to strongly advocate automation and monitoring at all steps of software construction,

From integration testing releasing to deployment and infrastructure management.

* **What is Devops?**

Devops is not a tool, Devops is culture (or) process.

* It is a process of continuous development, continuous build, continuous test, continuous release the software faster reliable with automation tools.
* It is a combination of development and operations.
* This automation process is happened between the development and IT teams.
* **Why we need Devops?**
* Main culture of Devops is collaboration between teams (development & operations).
* We can achieve the goal fast& reliably.
* To understand Devops culture try to understand the previous models.

Every software company to develop an application or software it has 2 main teams.

1. Development
2. Operations(testing, monitoring, infrastructure)

* **Development**

**- Development team develop the application with some tools.**

Eg: java based website

-front end by UI team (html, css, JavaScript, jQuery,)

-backend by JAVA developers (server side)

-DB admins give access of DATA BASE to developers.

-all people together develop a website.

-unit test the code (developer self-test).

-for another process need to move code into anther environment (testing, QA, etc...) developers not doing this

* **IT TEAM OPERATIONS**

- It team is setup the required environment to development, testing, QA, pre-production, production.

Eg: setup infrastructure for java website application

* Setup the hardware
* Setup the platform (WINDOWS, LINUX, ETC...)
* Setup required applications for developers such JAVA compiler, tomcat, Database, etc...
* They move the code from one environment to another
* They monitor the infrastructure, etc...
* They set the code

**Traditional development lifecycle:**

Waterfall model is there.

**Waterfall model:**

It is a traditional SDLC (software development life cycle).

To develop an application some linear steps are there.

Requirement -- gather information, what purpose documentation &planning

|

Design--setup hardware software

|

|

Implementation -- actual coding

|

Verification - integration

|

Maintained --deploy to customer

-if any update is required you need to design this cycle again.

-no return here

-bugs identic late.

-it takes to months to complete a phase, for whole cycle takes min 2 years.

-for any changes start from 1st stage.

**Agile method:**

-it is a SDLC (software development life cycle) into multiple locations

--1) planning--

| |

| 2) Analysis

5) Testing

| | |

4) build--3) design –

1) Plan for what you need

2) Analysis you’re planning

3) design (actual coding)

4) build (unit testing and build)

5) Testing (integration and testing)

* Here we break the entire application in to small pieces.
* We can develop each piece in one iteration.
* We can complete each iteration with in less time (min.2 weeks)
* These models are only at development stage only
* If the code is move in to another environment the code maybe fair.
* Due to infrastructure, versions of software, coding issues, testing problem, etc...
* Here which team rise the issue is not found.
* Teams are pointing each other for that issues.

- To earlier the product to the customers with prevalent the mistakes the quality assurance is one way.

- But still we have a problem to overcome this DEVOPS culture is very useful.

* **Advantages of Devops:**
* Software building and deployment with fast rate.
* Application maintainence&bug fixing easy.
* Release new features and versions of applications in less time.
* Higher product quality, faster productivity rate, increased customer satisfaction, more revenue and thus more profits.
* Every tool is input of another tool.
* We can get the feedback in this process.

**Tools involved in Devops life-cycle:**

* **Linux**
* **Shell scripting**
* **Git –scm tool**
* **Ant, maven -building tool**
* **Jenkins CI/CD tool**
* **Tomcat - platform to run a java application**
* **Ansible/chef/puppet – configuration management tool**
* **Docker – container management tool for run containers.**
* **Kubernates – a cluster and container management tool orchestration system for automation**
* **Nagios – continuous monitoring**

