Devops is the automation of the software industry

What is sdlc in Devops

The Software Development Life Cycle (SDLC) is a structured process that enables the production of high-quality, low-cost software, in the shortest possible production time. The goal of the SDLC is to produce superior software that meets and exceeds all customer expectations and demands.

SDLC- Software development life cycle

Requirement gathering (understanding what the client’s needs)

Feasibility study -Feasibility analysis is the process by which feasibility is measured. The Feasibility analysis in a project that is feasible at one point in time may become infeasible at a later point in time.

(Realizing whether the requirements can be met at current IT situation)

Design (Documenting the module based structure of the software applications)

Coding (Write code for actually developing the designed modules)

Testing (Doing analysis of the developed modules to find out if it complies to the actual design)

Deployment (Putting the prepared module on the production servers)

Maintenance (Keeping track of performance, checking for any potential issues and resolving them)

Example

Google pay :

Requirement gathering: UPI Based payment through QR code scanning, sending the money through phone numbers registered on Gpay

Feasibility study:

Design: User registration module, QR code scanner module, Money transfer module

Coding: Developing the modules and putting them together

Testing: QA-ing the application by checking each of the module functionality

Deployment: Movement to Google play store and on the Google servers.

Maintenance: Bug fixes (Play store Review/Google threads)

Waterfall Model of software development: (Time constraint, Effort constraint)

* Entire requirement
* - Entire feasibility
* - Entire design
* - Entire coding
* - Entire testing
* -complete Deployment
* - Maintenances

Agile Model of software development: (We have to do In single line.) : Agile SDLC model is a combination of iterative and incremental process models with focus on process adaptability and customer satisfaction by rapid delivery of working software product. Agile Methods break the product into small incremental builds.

Requirement -> feasibility -> Design -> coding -> Testing -> Deployment -> Maintenances

Agile has the particular problem

And that is Maintenances Problem

There is dead line in Maintenances

Agile is small work is okay

Development in operations (Dev0ps)

Dev0ps flow :

Developers upload their in a repository

Testers write the script ,which is run on the code that has been uploaded

If everything goes well , have a check on the environment in production and see if it matches to the expectation.

If yes , deploy that on production ; if no , prepare the env , then go for deployment

Monitor the stuffs (Application/Servers)

Dev0ps flow (jargons):

Developers PUSH their code in GITHUB

JENKINS checks the GIT STATUS periodically (1 Min) and PULLS

the newly pushed code

JENKINS BUILDS the code (complied) using MAVEN.

SELENIUM test scripts are configured on JENKINS and MAVENS takes those scripts and runs a test on the compiled code

JENKINS fires a deployment on DOCKER (single server deployment) / SWARM (clusterised deployment ) / KUBERNETES (clusterised deployment)

NAGIOS is used for monitoring the server / application

Any Maintenance job to be performed using TERRAFORM /ANSIBLE.

2 – 4 continuous integration (CI)

5 – Continuous Deployment (CD)

6 – Continuous Monitoring (CM)

7 – Configuration Management (CM)