*Enterprise to the Power of Digital™*

# DevOps On Java Platform

**Mahendra Pachpor**

birlasoft

CK BIRLA GROUP

# Session Objectives

* What is DevOps
* Need Of DevOps
* DevOps Goal
* DevOps WorkFlow
* Introduction to Continuous Integration
* Advantages of Continuous Integration
* Implementation of Continuous Integration
* DevOps Tools
* Set-up Git in your system
* Using Git as DevOps Tool

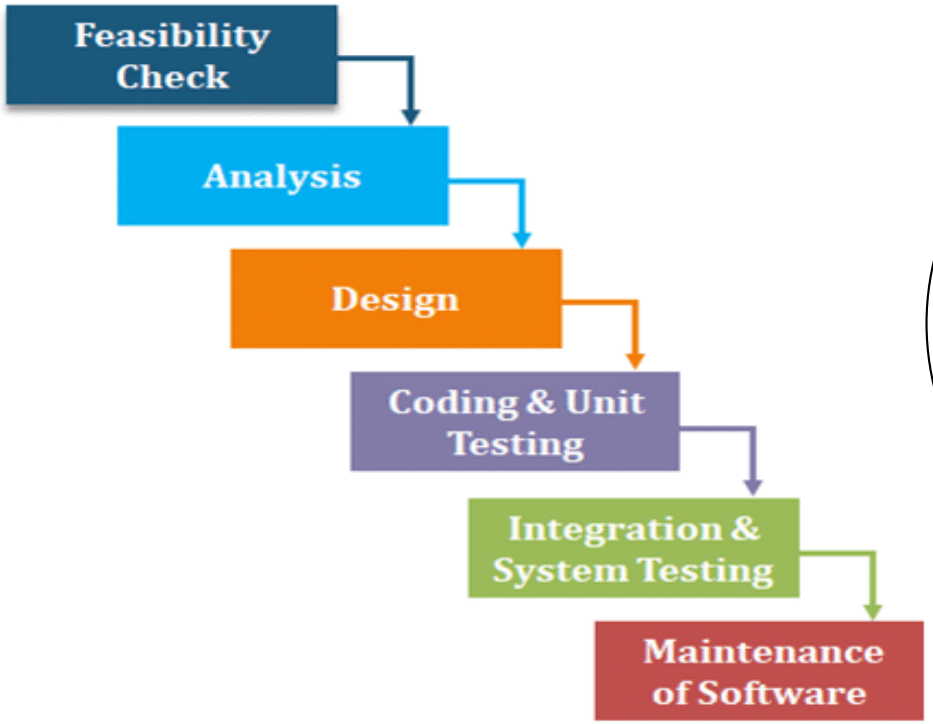
# What is DevOps

**DevOps** is a set of practices that combines software development (Dev) and information-technology operations (Ops) which aims to shorten the systems development life cycle and provide continuous delivery with high software quality

# Need of DevOps

- Waterfall Model :

"Reliable SDLC approach used for Software Development process in a sequential flow"




- This method was easy and simple to use
- Easy to manage due to the rigid model
- Each phase had review process
- One at a time process phases, due to which these phases do not overlap each other
- Reliable for the small projects

# Need of DevOps

- While your application is in testing stage, it gets really difficult to go back and make changes to any issues those were developed in **previous steps** due to miscommunication or the lack of knowledge
- Risky process as it is difficult to diagnose and provide the feedback
- Its main focus was to help the internal teams to work efficiently on user/clients feedback between the process
- Delays in testing process, because this method insists the teams to wait until process is in previous stage

solution to this is **Agile** method came into existence

# Need of DevOps

## Agile Software Development

"Agile refers to an incremental approach like Waterfall model"

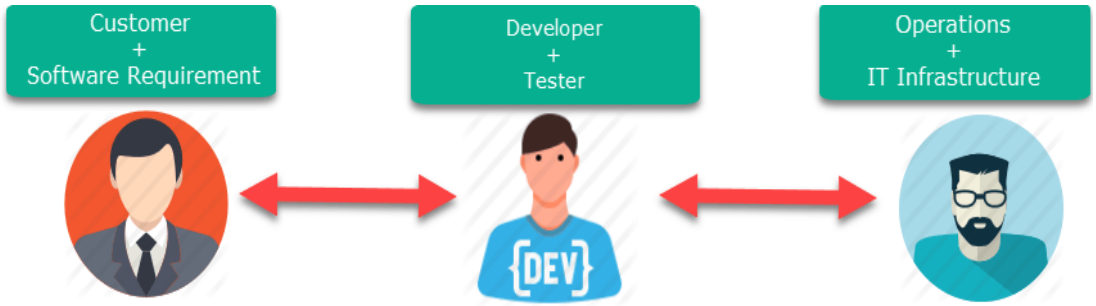
- With the iterative approach
- focusing on customers feedback
- small rapid changes as well as releases

"breaks the product into smaller divisions and finally integrates them for the testing process"

"Requires more time and commitment"
testers, developers and customers must interact with each other constantly and should agree to each other's decisions in order to get the task done,

# Need of DevOps

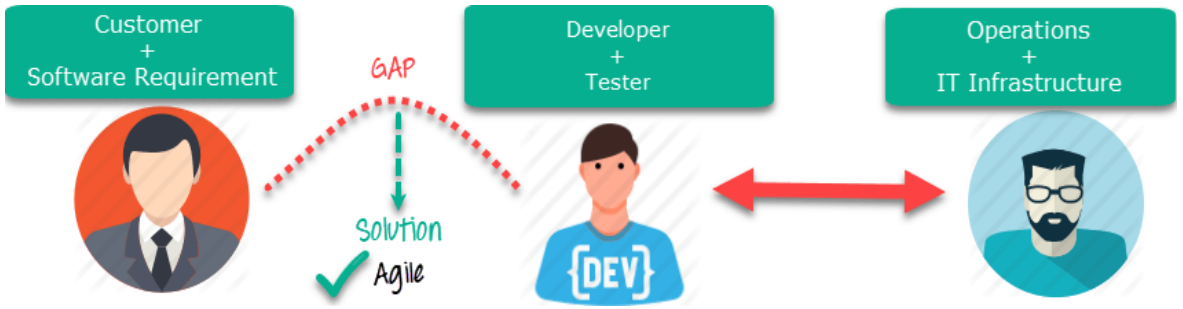
**Waterfall Method** there was a gap between Customer Software Requirements and the Developers,





Developers Team and Operations Team Work Together

**Agile Method** there was a gap between Developers and Operations Team




overcome this gap between Developers and Operations team **DevOps** was introduced

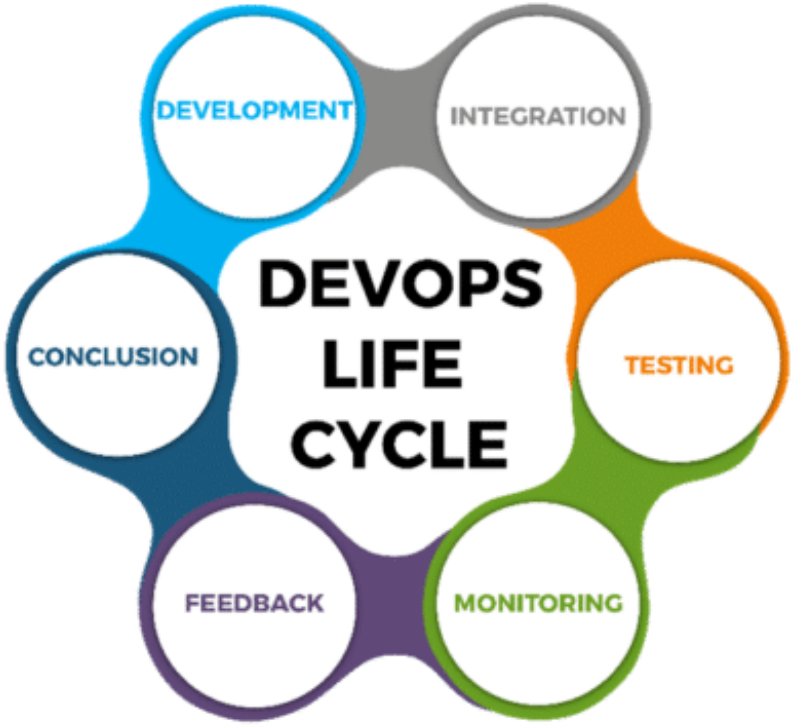
# DevOps

| |
|---|
| It brings Developers and Operations team together |
| Its primary goal is the Automation in order to increase the efficiency while deployment |
| DevOps focuses on Development, testing and implementation with equal importance |
| It overcomes the gap between Developers and Operations Team |

DevOps basically emphasis on bringing all the Development + Operations + infrastructures teams together

# DevOps Goal

- Continuous Development
- Continuous Integration
- Continuous Testing
- Continuous Monitoring
- Virtualization and Containerization

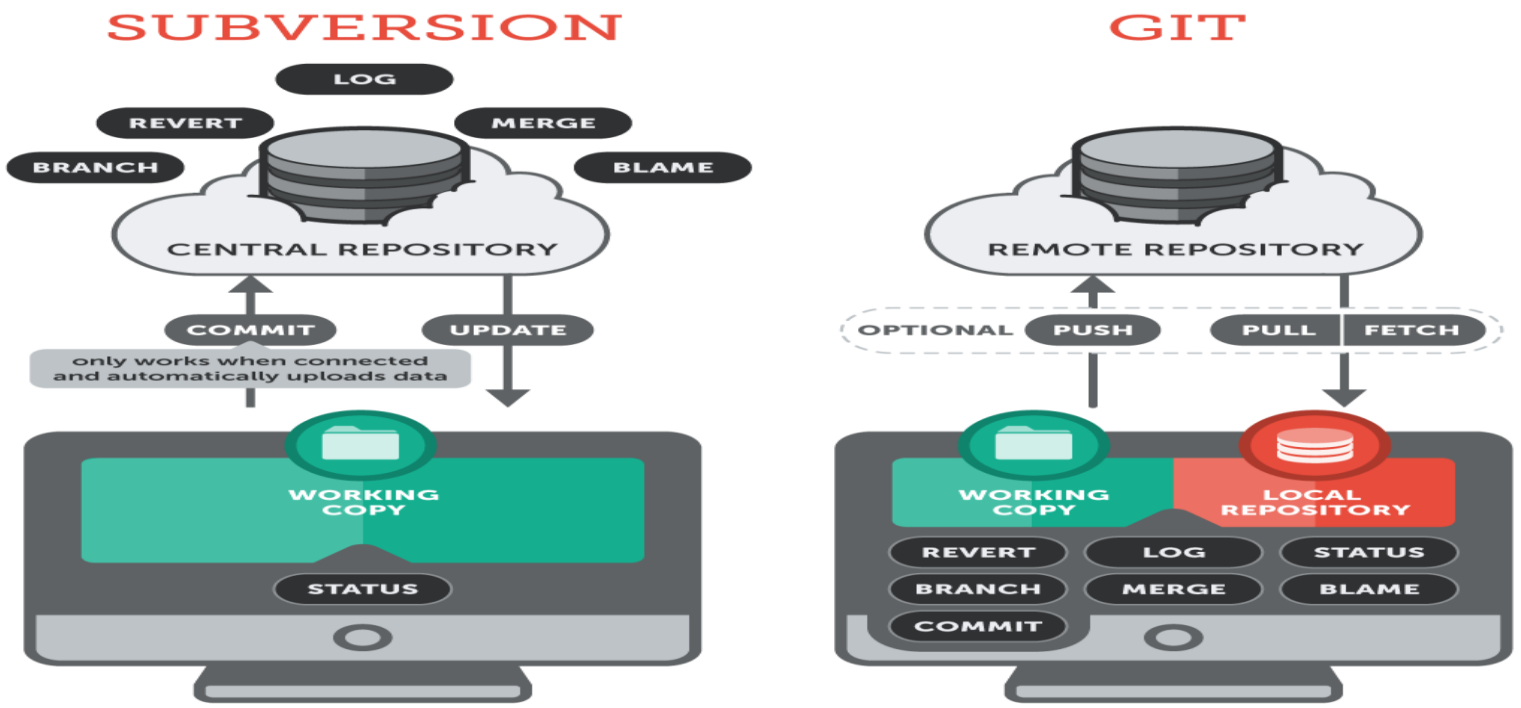


# Continuous Development

In this stage software is getting developed continuously.

Tools used:

As we Code & Build in this stage, we can use GIT to maintain the different versions of the code



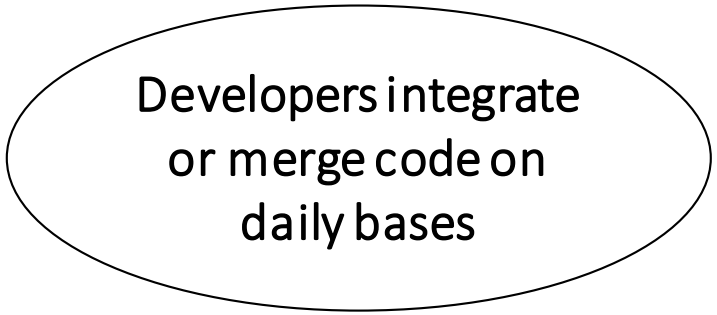

Developers integrate or merge code on daily bases

# Hosting  Source Project(s)

- GitLab
- Bitbucket
- Launchpad
- Sourceforge
- Phabricator
- GitBucket

# Continuous Integration

- Continuous Integration is a development practice that requires developers to integrate or merge code into a shared repository continuously
- The code is verified by the automated build which allows teams to identify problems before it's too late



# Integration Process


1.Maintain a code repository.

2.Make your **build** self-testing.

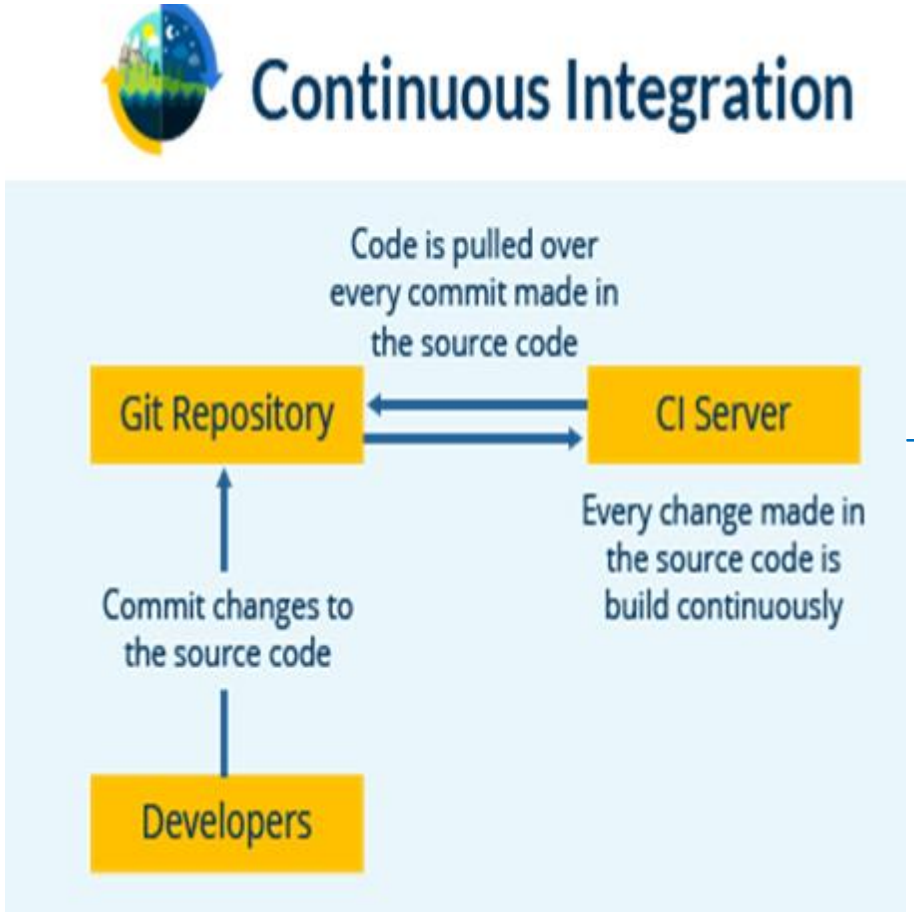
3.Daily commits to the baseline by everyone on the team.

4.Every commit (to the baseline) should be built.

5.Keep your **builds** fast.

6.Clone the production environment and test there.

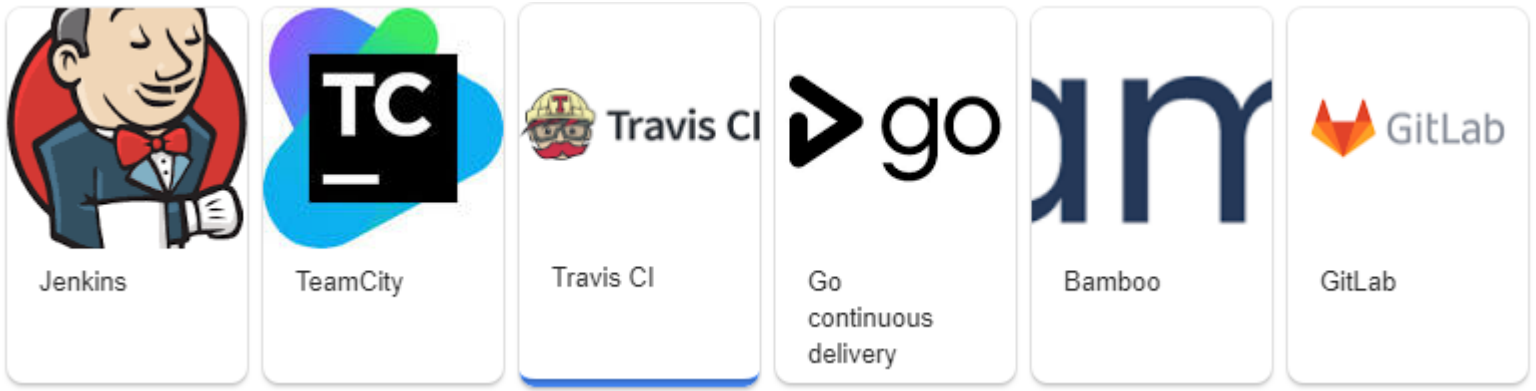
# Continuous Integration Process



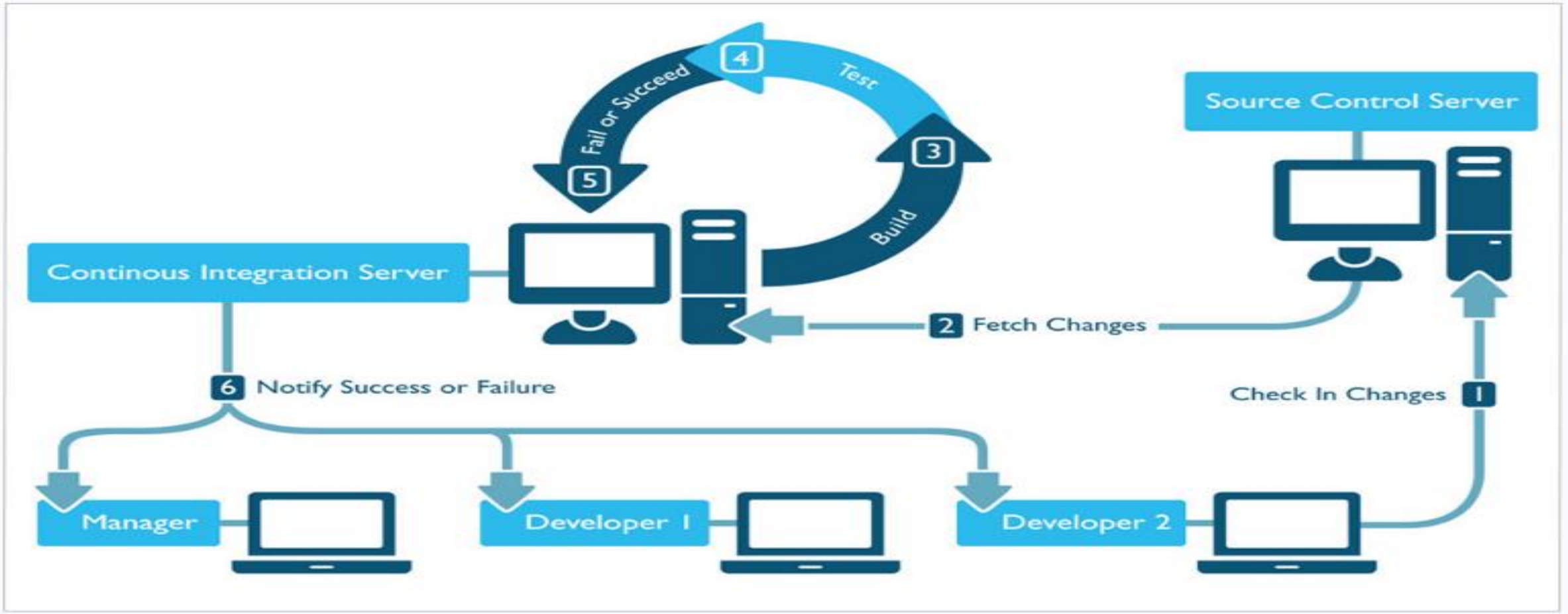

1. Maintain a code repository.
2. Automate your **build**.
3. Make your **build** self-testing.
4. Daily commits to the baseline by everyone on the team.
5. Every commit (to the baseline) should be built.
6. Keep your **builds** fast.
7. Clone the production environment and test the app.

# Continuous Integration tools

- Jenkins.
- TeamCity.
- Go CD.
- Bamboo.
- GitLab CI.
- CircleCI.
- Codeship.

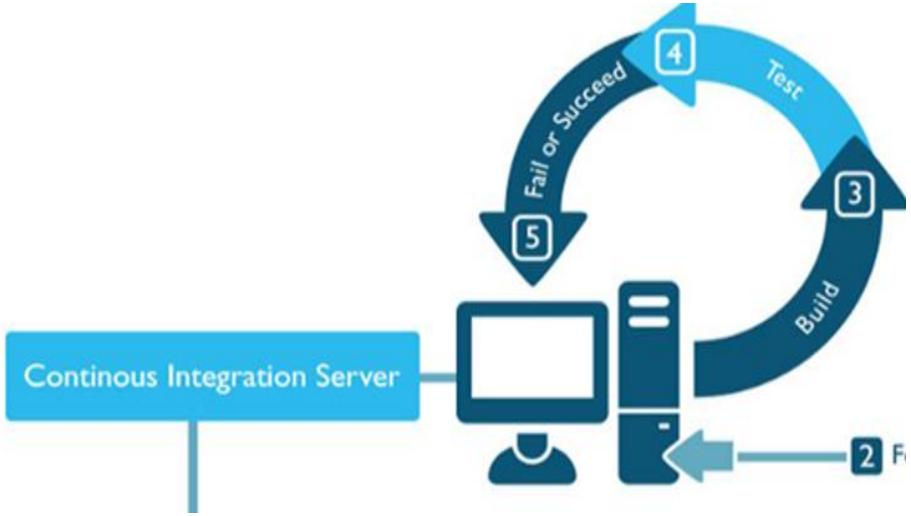


# Continuous Integration & Testing Process



# Benefits of Continuous Integration

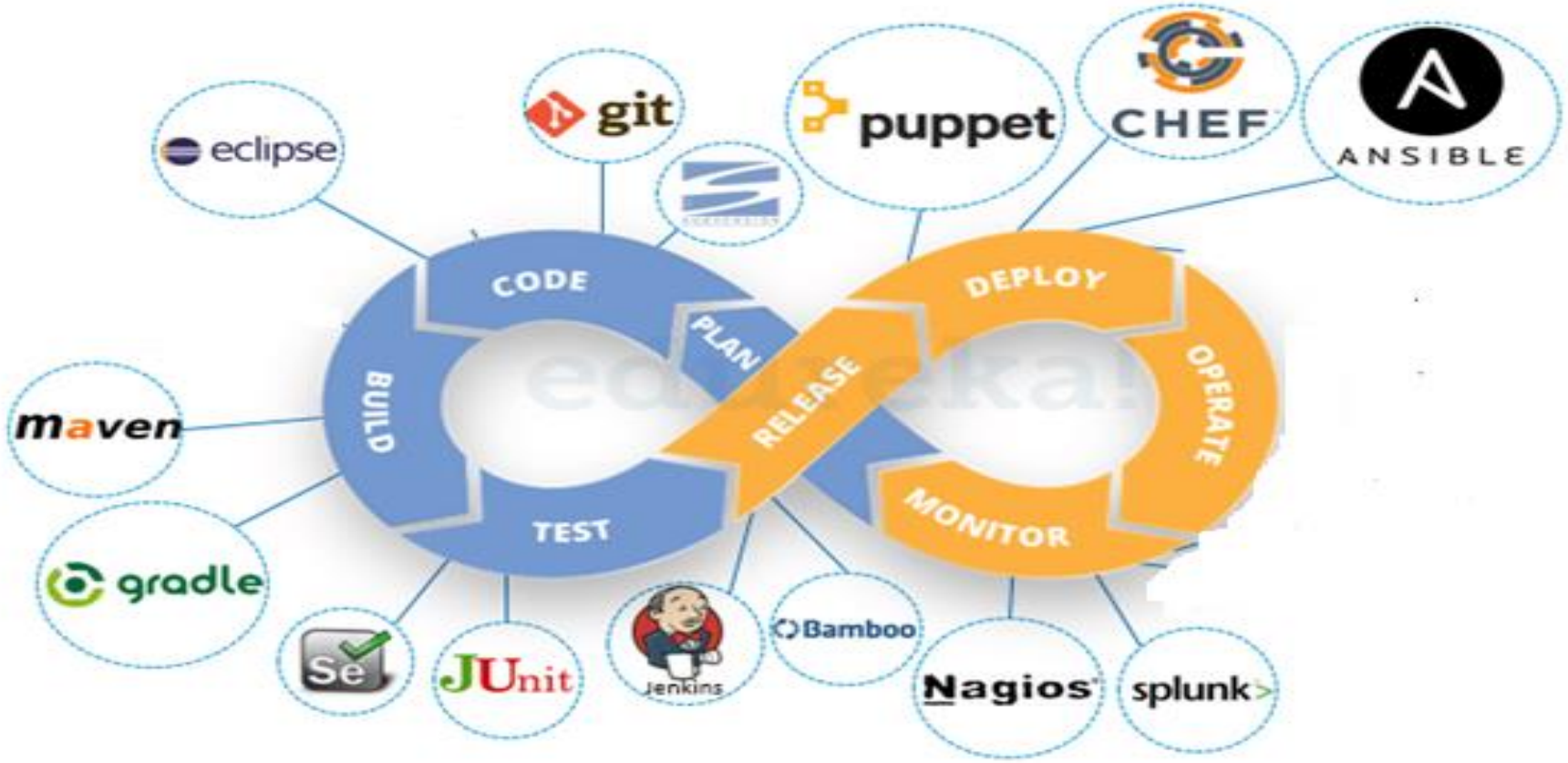
- Early Bug Detection
- Reduces Bug Count
- Automating the Process
- Cost-Effective Process



# Continuous Integration, Continuous Delivery, and Continuous Deployment

- **Continuous integration:** is the beginning of the cycle where builds are created and tested.
- **Continuous delivery:** cycle starts from the beginning of CI and ends with the “release” of the verified artifacts – packages, VM or container images – published onto a server.
- **Continuous deployment :** ensures that qualified releases are automatically deployed to production

# DevOps Tools



# Questions ?

birlasoft

# Thank You

www.birlasoft.com

birlasoft

CK BIRLA GROUP