

DevOps

JN 30

NUTS

Agenda

- What's DevOps
- Agile **Vs.** DevOps
- DevOps team & DevOps Engineer
- Principles of DevOps
- DevOps cycle (Phases & Tools)
- DevOps Objectives
- Demo

- A product or a tool
- A separate team
- Combining Development & Operations teams

X Tools

X Automation

X Teams

ISN'T
ONLY

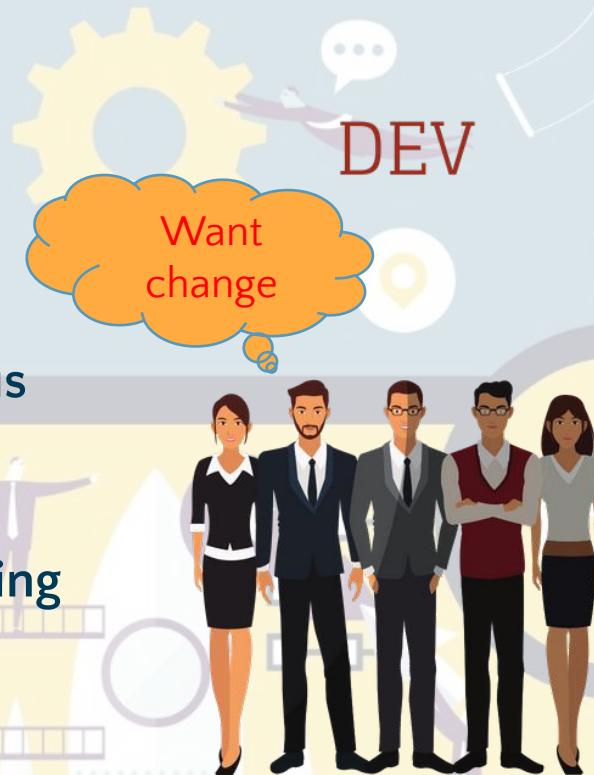
IS!

- A culture that you can adopt for continuous improvement.
- A way to bring your Developers Team and Operations Team on the same page, allowing them to work together with ease.

✓ Communication

✓ One Objective

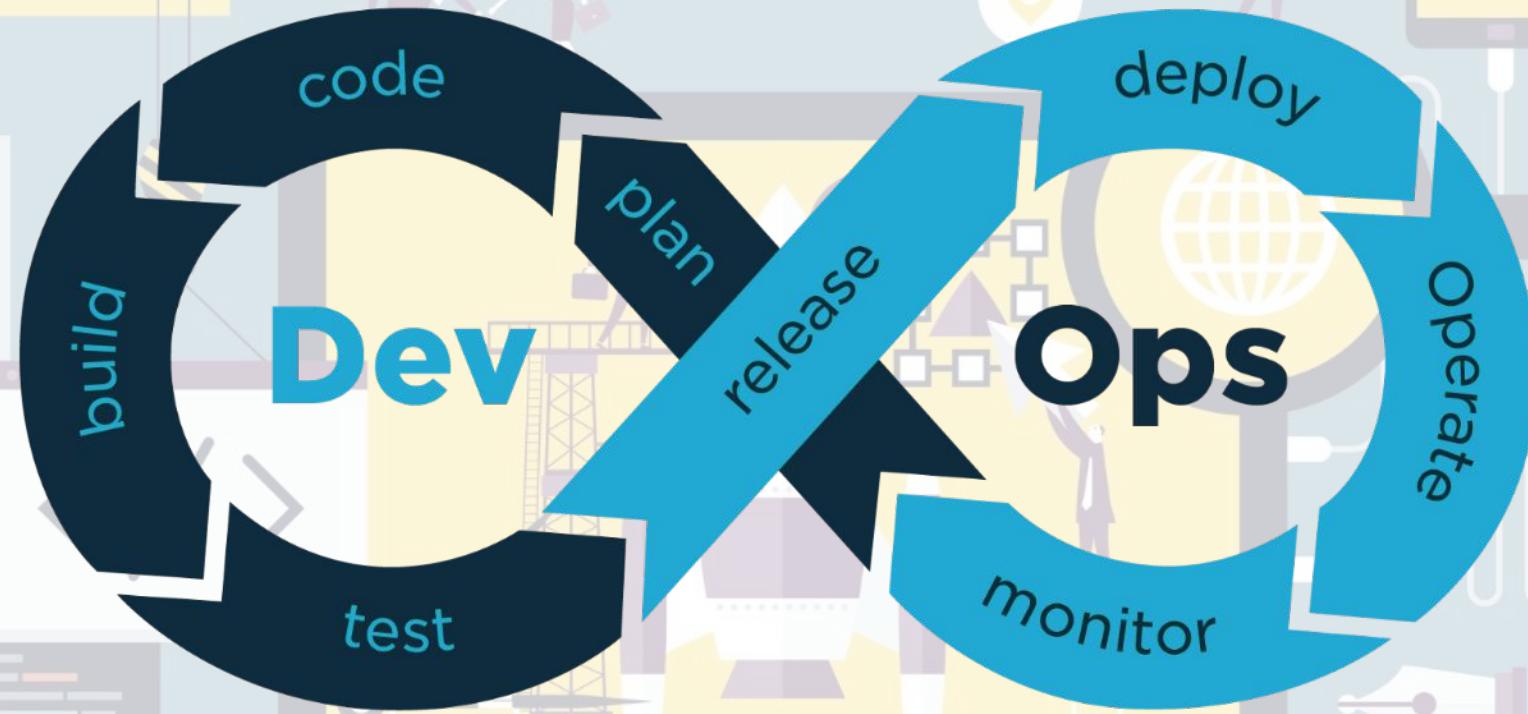
✓ Integration



Wall of Confusion

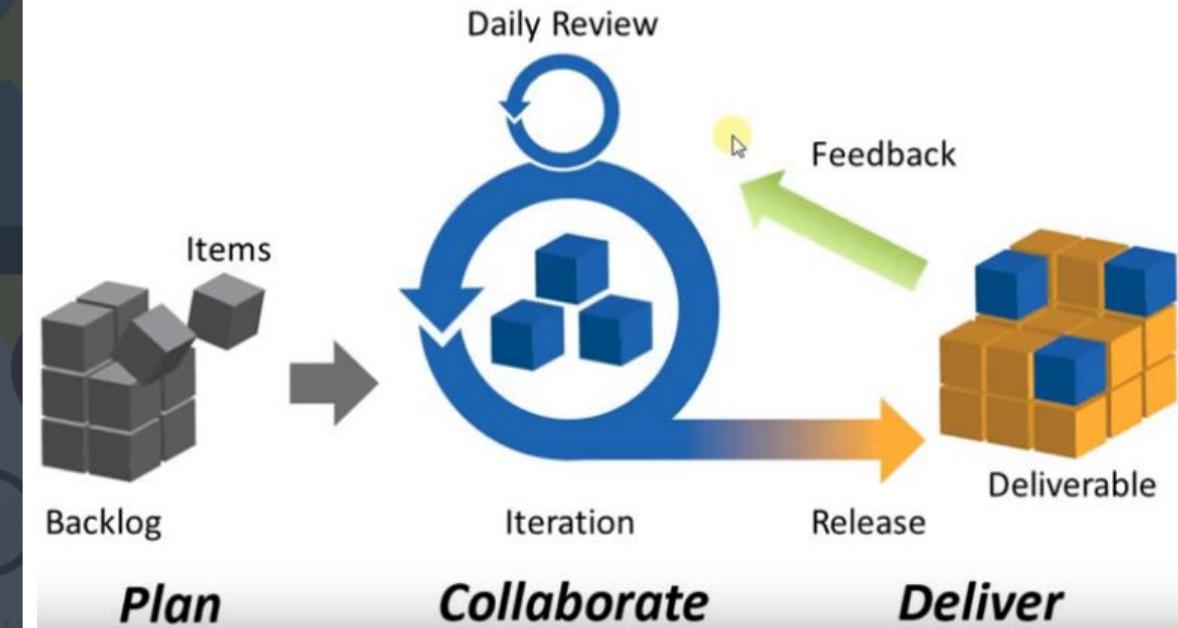


Process



- Continuous iteration of development and testing in the SDLC process.
- Emphasizes on iterative, incremental development.
- Agile breaks the product into smaller pieces and integrates them for final testing.

AGILE



AGILE VS DEVOPS

Parameter

Agile

DevOps

Feedback

Feedback is given by the customer.

Feedback comes from the internal team.

Automation

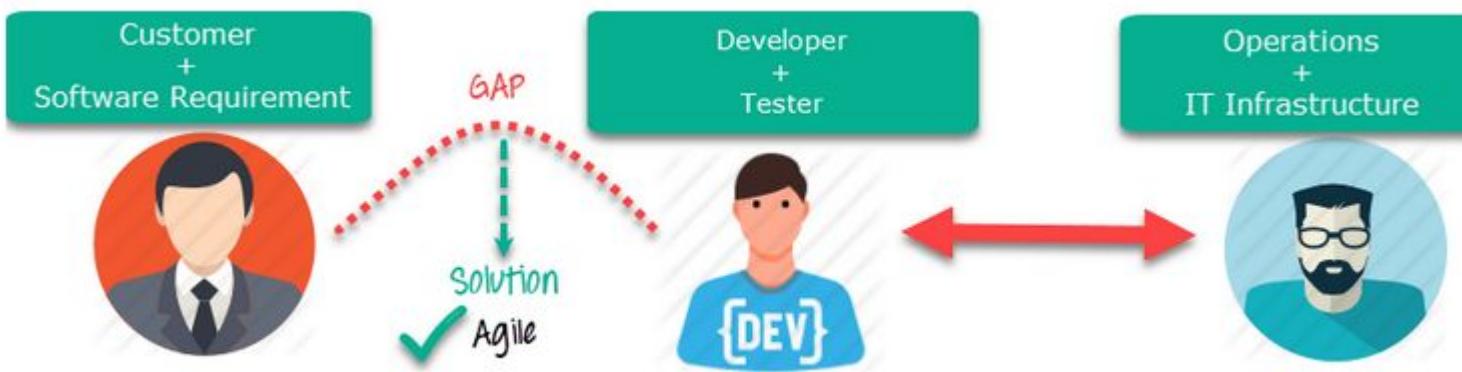
Agile doesn't emphasize on automation. Though it helps.

Automation is the primary goal of DevOps. It works on the principle to maximize efficiency when deploying software.

Importance

Developing software is inherent to Agile.

Developing, testing and implementation all are equally important.





Spotify®

NETFLIX

amazon

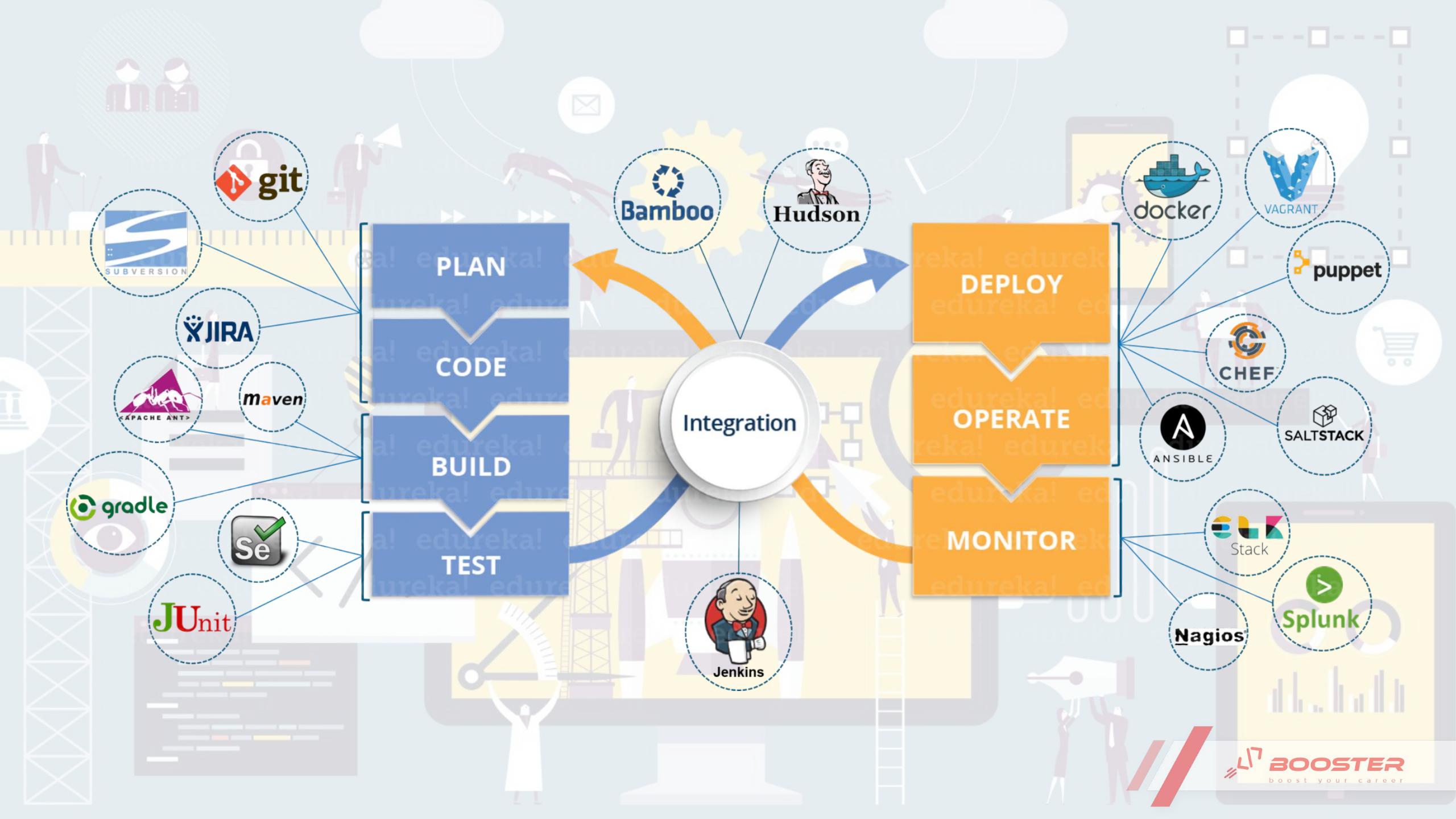
intel®

A®
Adobe

YAHOO!

P PayPal

COMPANIES

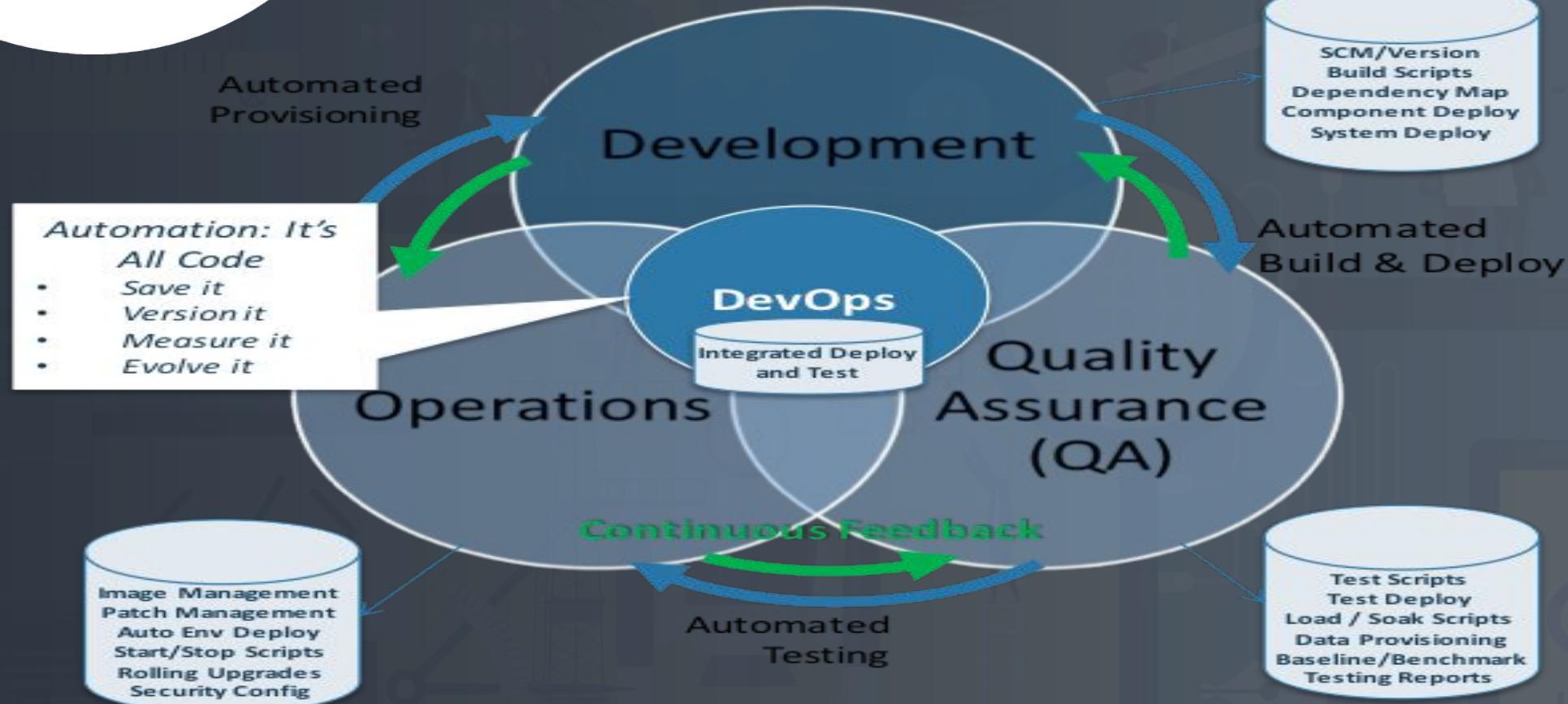


- ❖ A squad contains multiple teams such as: Development, operations, testing, DevOps and so on.
- ❖ All the squad are responsible for achieving one objective.
- ❖ All engineers work across the entire application lifecycle to automate processes, that historically have been manual and slow.

**DevOps
Team**



DevOps Engineer



Principles of DevOps!

Customer-Centric Action

To be able to meet these customers' requirements.

1



Create with the End in Mind

It's all about knowing your destination from the beginning.

2



End-To-End Responsibility

In a DevOps environment, all team members are responsible for everything from start to end, which greatly enhances the level of responsibility felt and the quality of the products engineered.

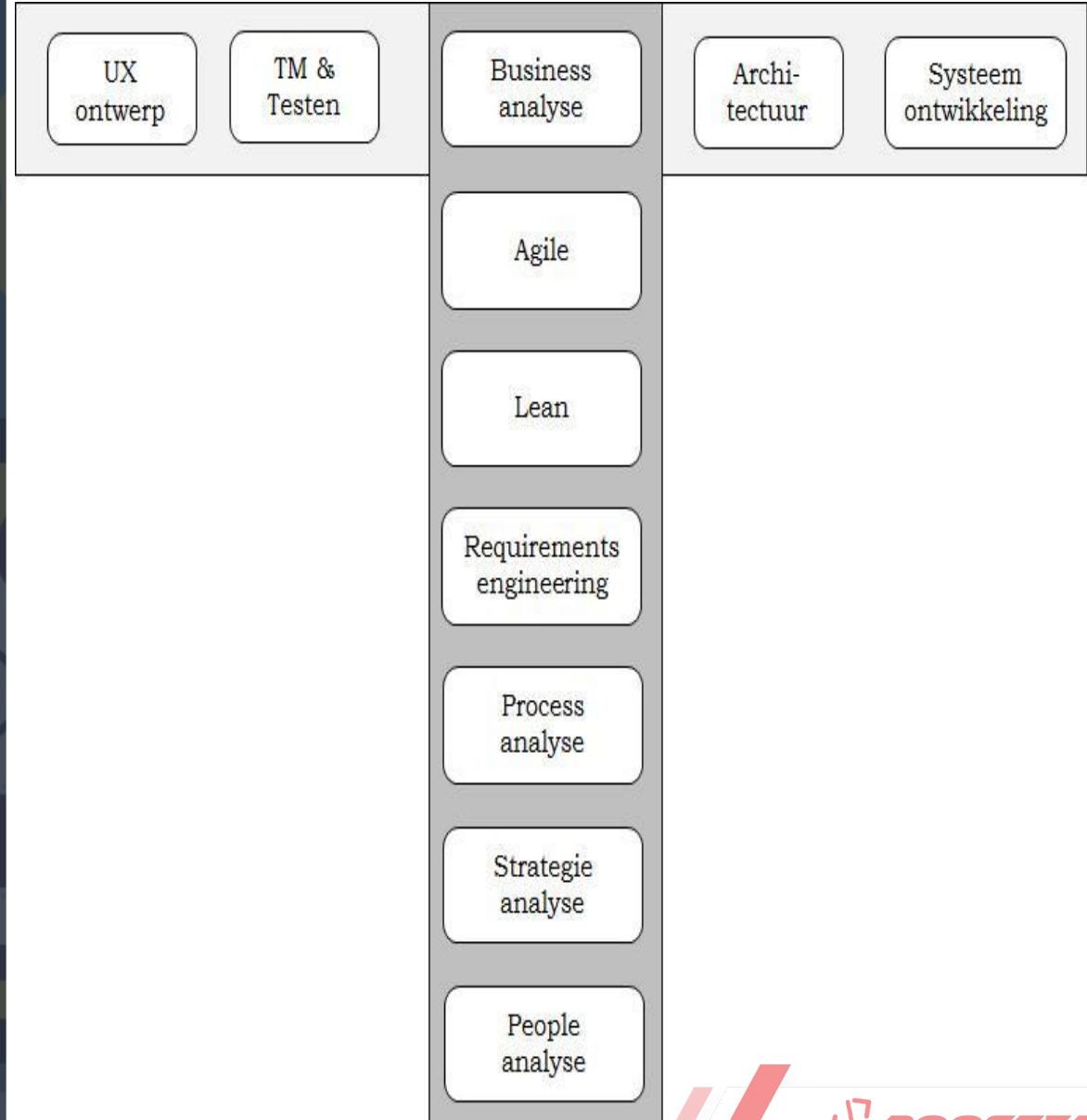
3



Cross-Functional Autonomous Teams

That requires a balanced set of skills unlike old-school IT specialists who are only knowledgeable or skilled in for example testing, requirements analysis or coding.

4



Continuous Improvement

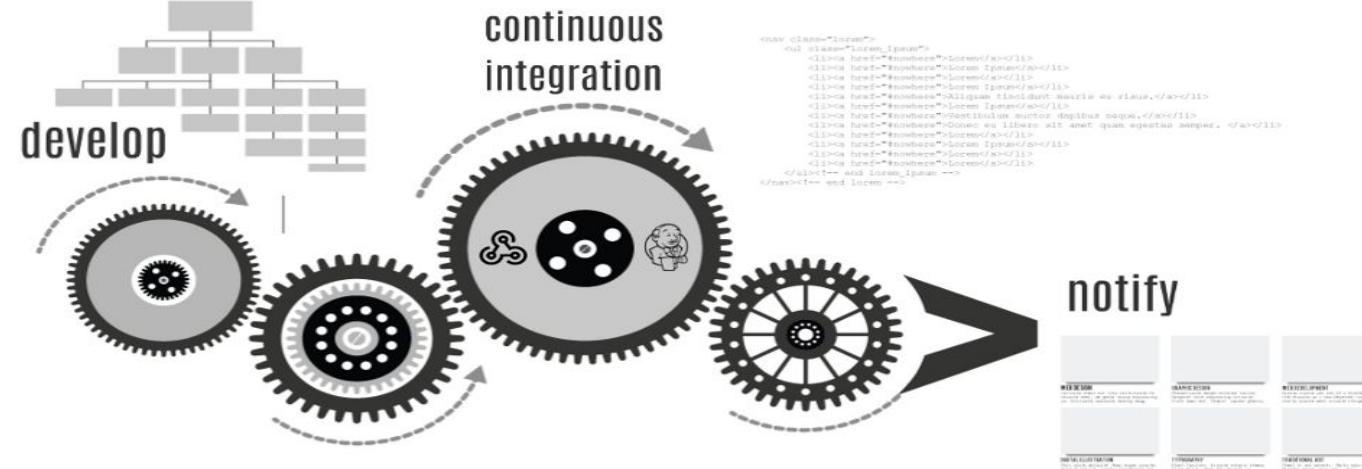
In a DevOps culture, a strong focus is put on continuous improvement to minimize waste, optimize for speed, costs, and ease of delivery, and to continuously improve the products/services offered.

A good rule to live by in that respect is "if it hurts, do it more often".

5



Automate Everything You Can



6



DevOps cycle comprises of 5 different phases

1

Continuous Development

2

Continuous Integration

3

Continuous Testing

4

Continuous Deployment

5

Operation & Monitoring



1

Continuous Development

PLAN & CODING

Requirement and analysis

- Observation
- Focusing group
- Survey
- Questionnaire
- Interviews

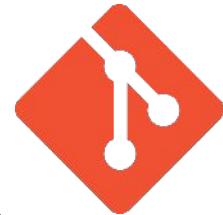
Design

PLAN

CODING

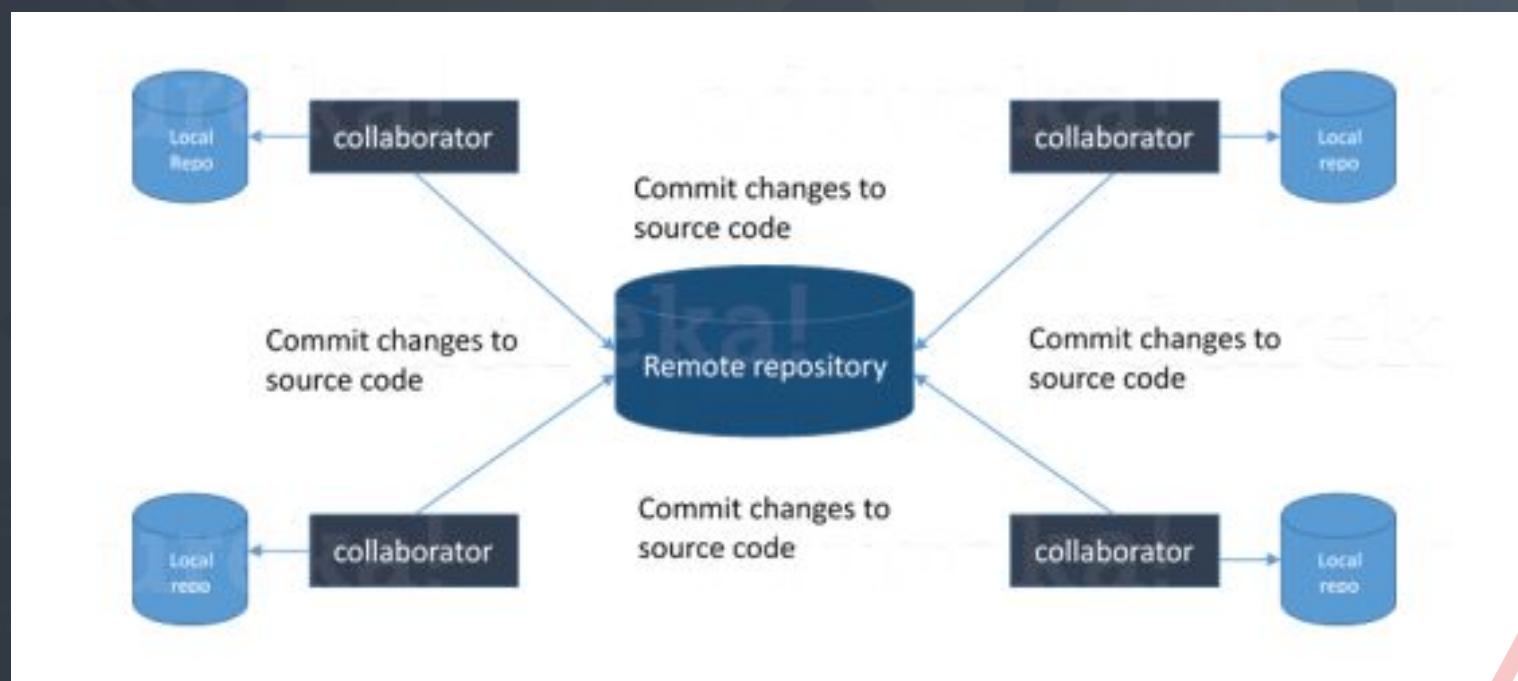
As a developer we have two pillars





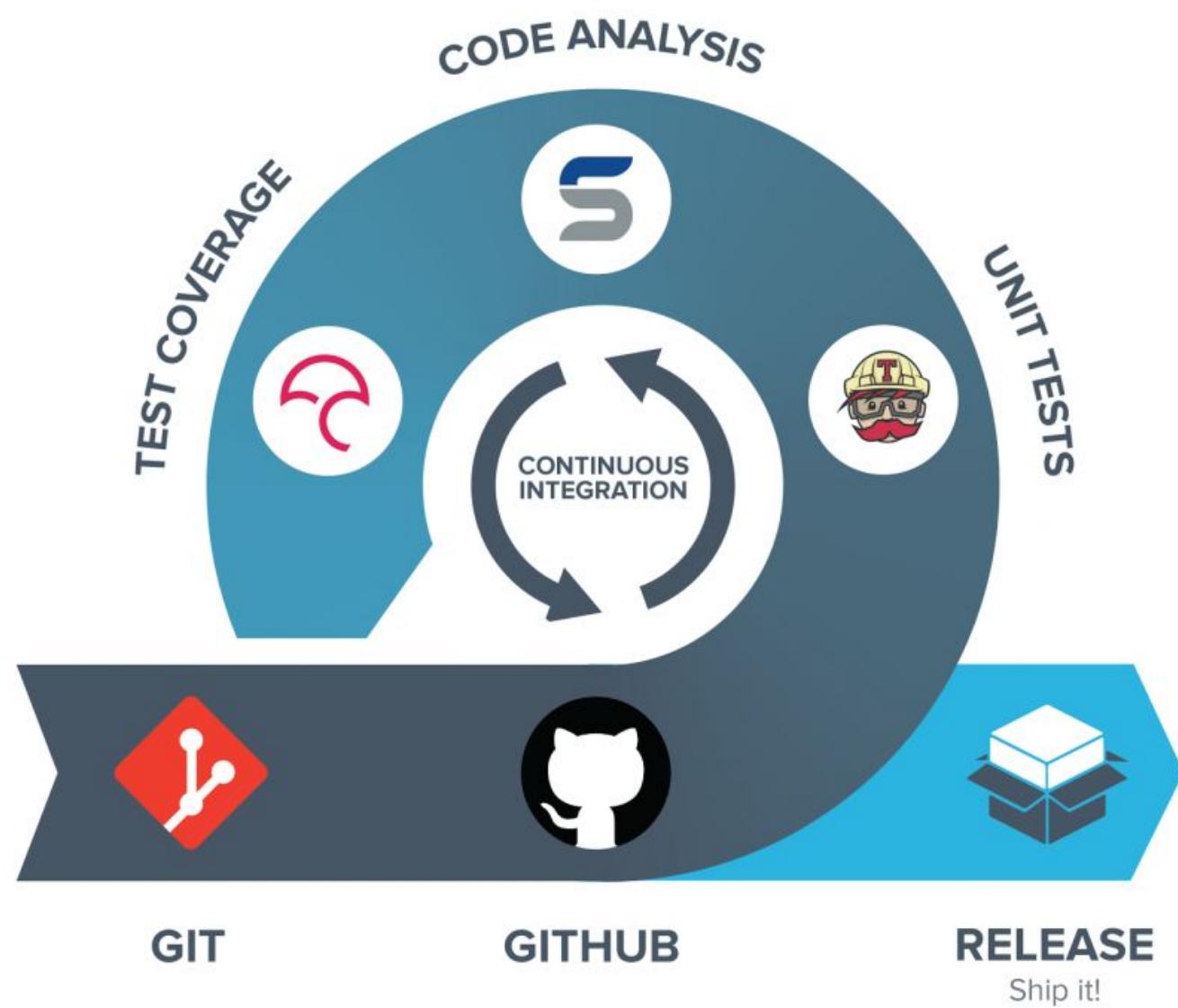
git

Version Control System



2

Continuous Integration



PROCESS

- The team commit changes
- Every commit is then built
- Unit testing
- Merge
- Integration test to detect any problems earlier
- Deploy

BUILD TOOLS

CI SERVERS

Tasks	maven			gradle			ivy		
	Jenkins/ Hudson	Bamboo	Team City	Jenkins/ Hudson	Bamboo	Team City	Jenkins/ Hudson	Bamboo	Team City
Run build scripts	yes	yes	yes	yes	yes	yes	yes	yes	yes
multimodule project detection/incremental builds	yes	yes	yes	yes	yes	yes	yes	no	no
automatic JUnit test results publishing	yes	yes	yes	no*	no*	yes	no*	no*	no*
automatic archiving/publishing of artifacts	yes	yes	yes	no**	yes	yes	no**	no**	no**
detection of new builds of dependencies	yes	yes	yes	no	no	yes	yes	no	no
Enablement	built-in	plugin	built-in	plugin	plugin	built-in	Ant : built-in Ivy : plugin	plugin	built-in

*CI needs to be told where junit report files are located

**CI needs to be told where are the newly built artifacts

TOOLS

Differences between tools

- Ease of installation process
- Number of plugins available
- The cost of the software itself
- Popularity (community support)

CI tools



8

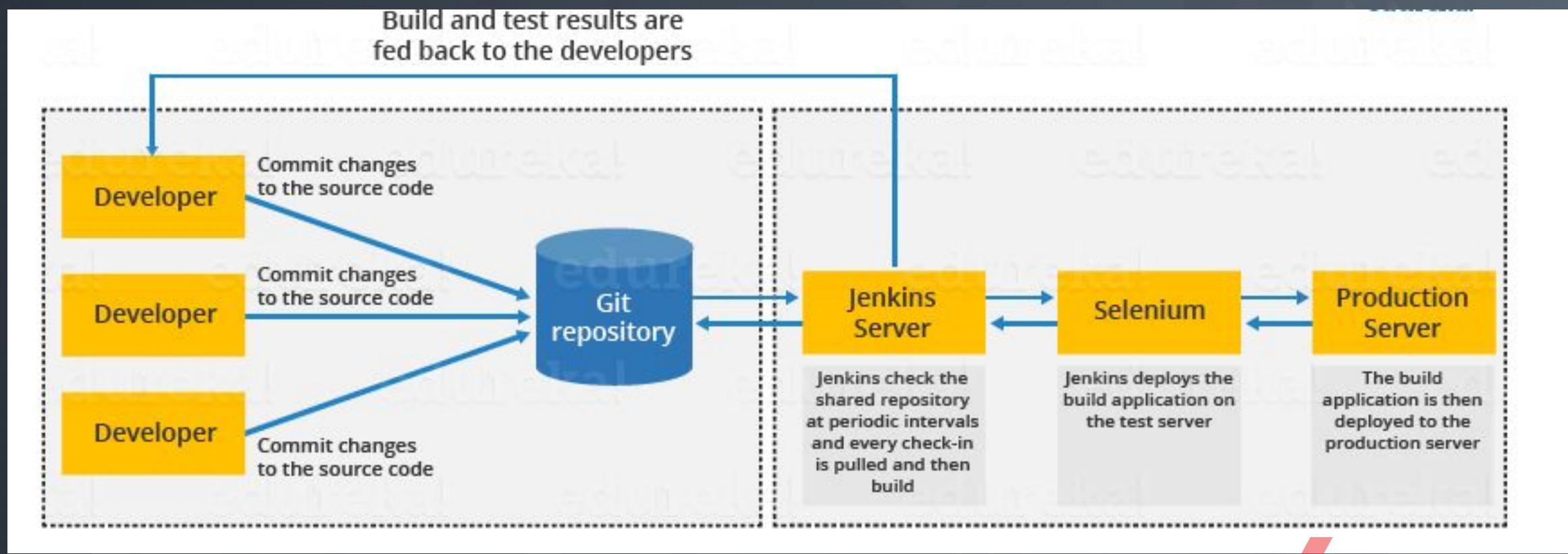
- Open source tool with great community support.
- Easy to install.
- Has 1000+ plugins to ease your work.
- Free of cost.
- Built with Java and hence, it is portable to all the major platforms.

Jenkins



Jenkins

CI BY Jenkins



B&A Jenkins

Before Jenkins	After Jenkins
The entire source code was built and then tested. Locating and fixing bugs in the event of build and test failure was difficult and time consuming, which in turn slows the software delivery process.	Every commit made in the source code is built and tested. So, instead of checking the entire source code developers only need to focus on a particular commit. This leads to frequent new software releases.
Developers have to wait for test results	Developers know the test result of every commit made in the source code on the run.
The whole process is manual	You only need to commit changes to the source code and Jenkins will automate the rest of the process for you.

3

Continuous Testing



**OLD
SCHOOL**

Mostly apply waterfall model.

RESULT..

Conflict between development team and build engineers.



TEST TOOL

- Open source
- Automating testing tool
- Automates web browsers. Mainly Used for function and testing and regression testing
- Supports different PL, OS and different browsers



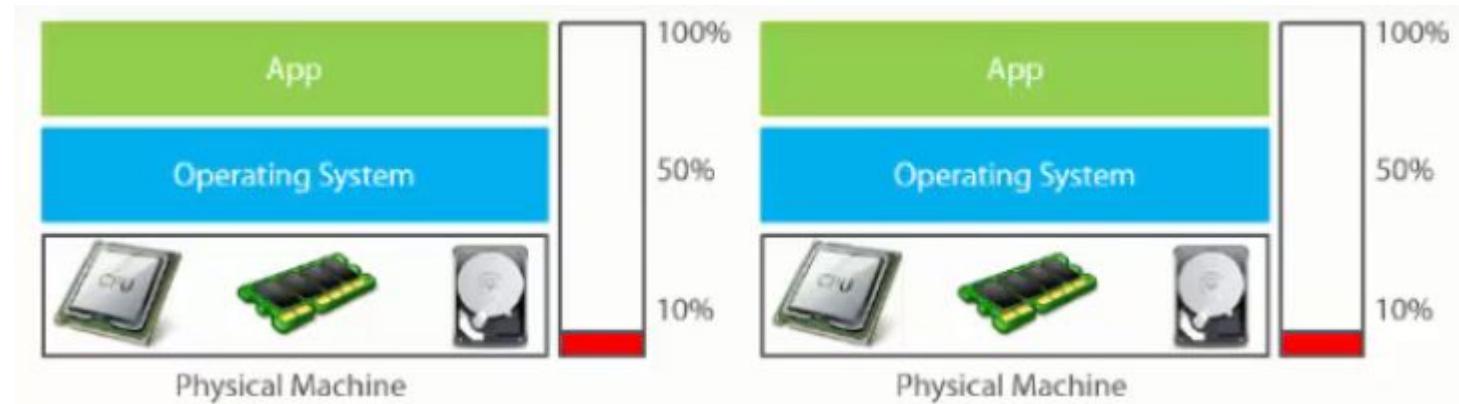
DEPLOYMENT

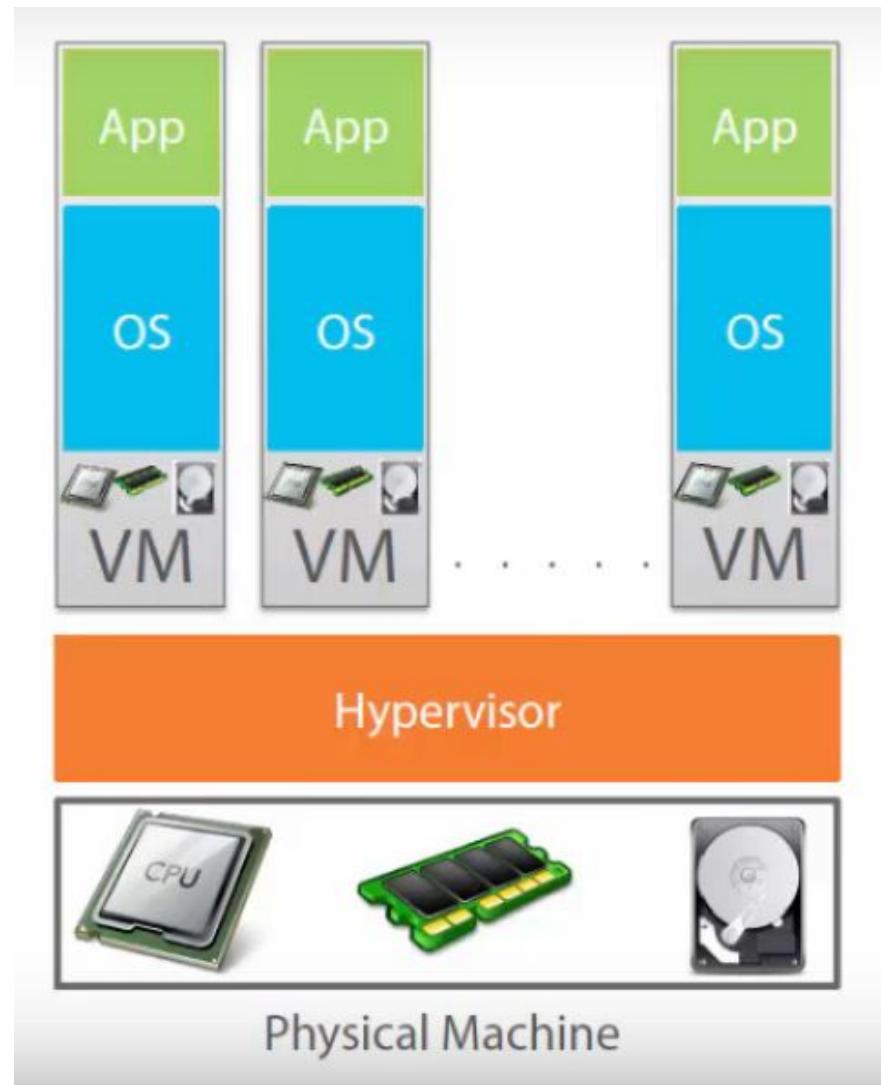


Continuous Deployment



1 APP – 1 SERVER

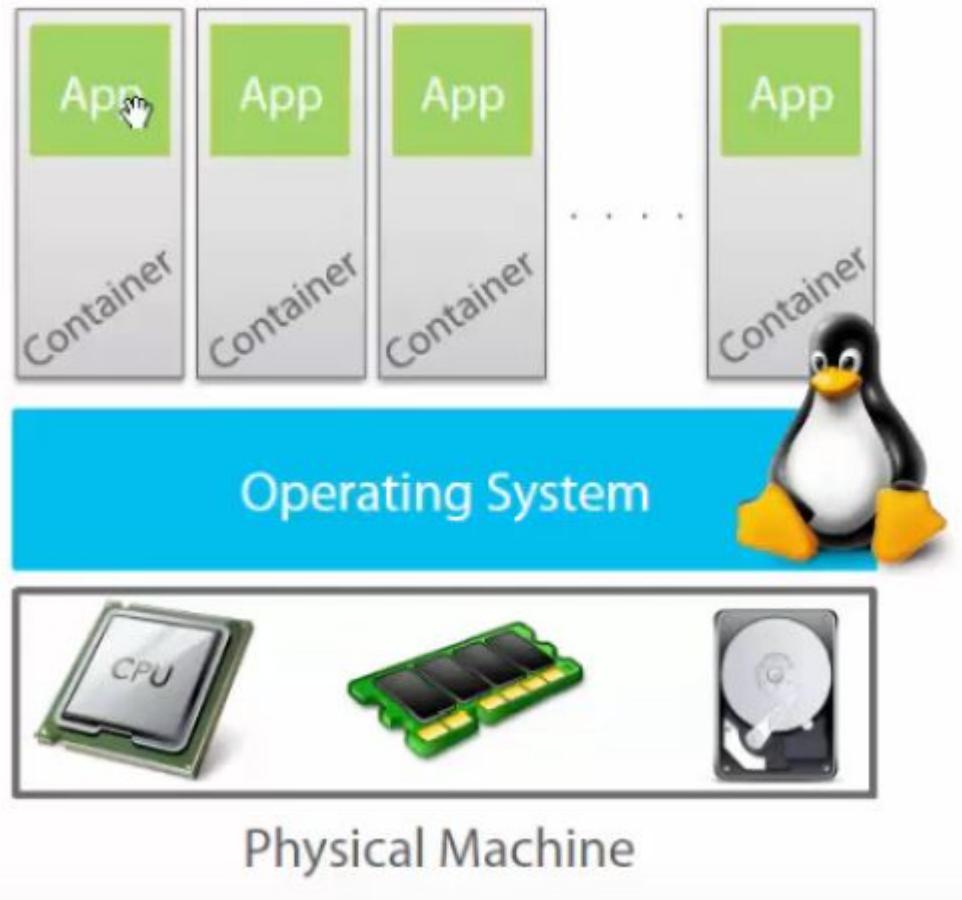




VIRTUAL MACHINE



CONTAINER!



5

Operation & Monitoring

WHY?
WHAT?
TOOLS!

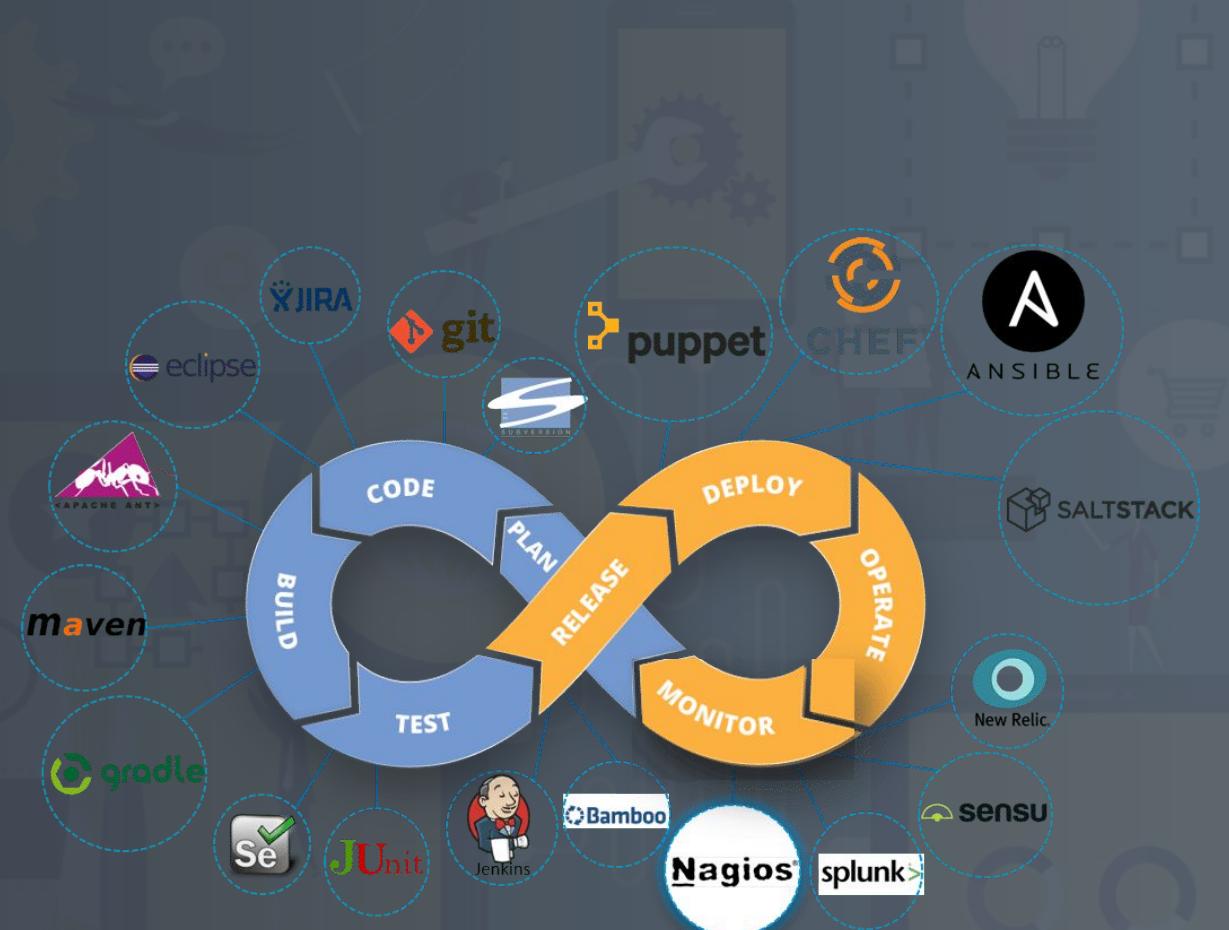


WHY?

- Continuous Monitoring Tools resolve any system errors (low memory, unreachable server etc.) before they have any negative impact on your business productivity.
- Discover problems (Server & Network)
- Keep (Security & Availability) for servers

WHAT?

- Ability of organization to (Discover, Report, Contain) attack in infrastructure.
- Last step?
- New?



Prometheus

An open-source monitoring system with a dimensional data model, flexible query language, efficient time series database and modern alerting approach.



DevOps Objective

Speed up Time To Market

Accelerate Digital transformation by moving fast without breaking things

Enhance software development lifecycle by reducing time and minimize potential human error

continuously develop, test and release high quality software to achieve a true competitive advantage

Demo

time!



Thank you!

Questions?!