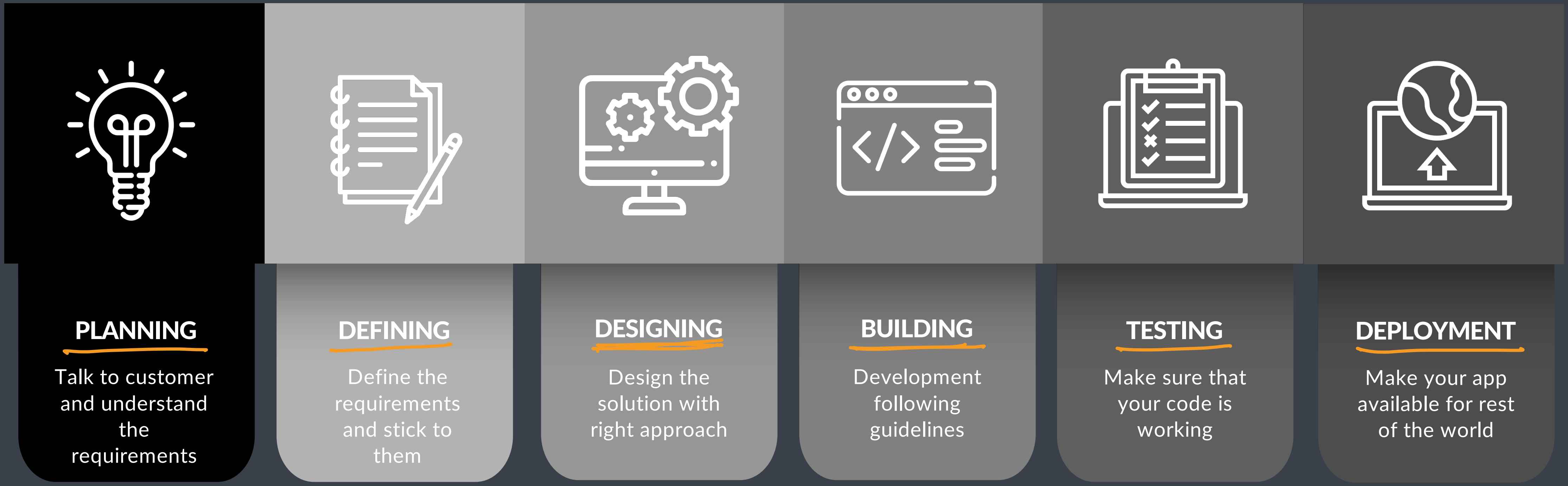
DevOps

Software Development Lifecycle



PLANNING

Talk to customer and understand the requirements

DEFINING

Define the requirements and stick to them

SRS

DESIGNING

Design the solution with right approach

DDS

BUILDING

Development following guidelines

program
↓
package

TESTING

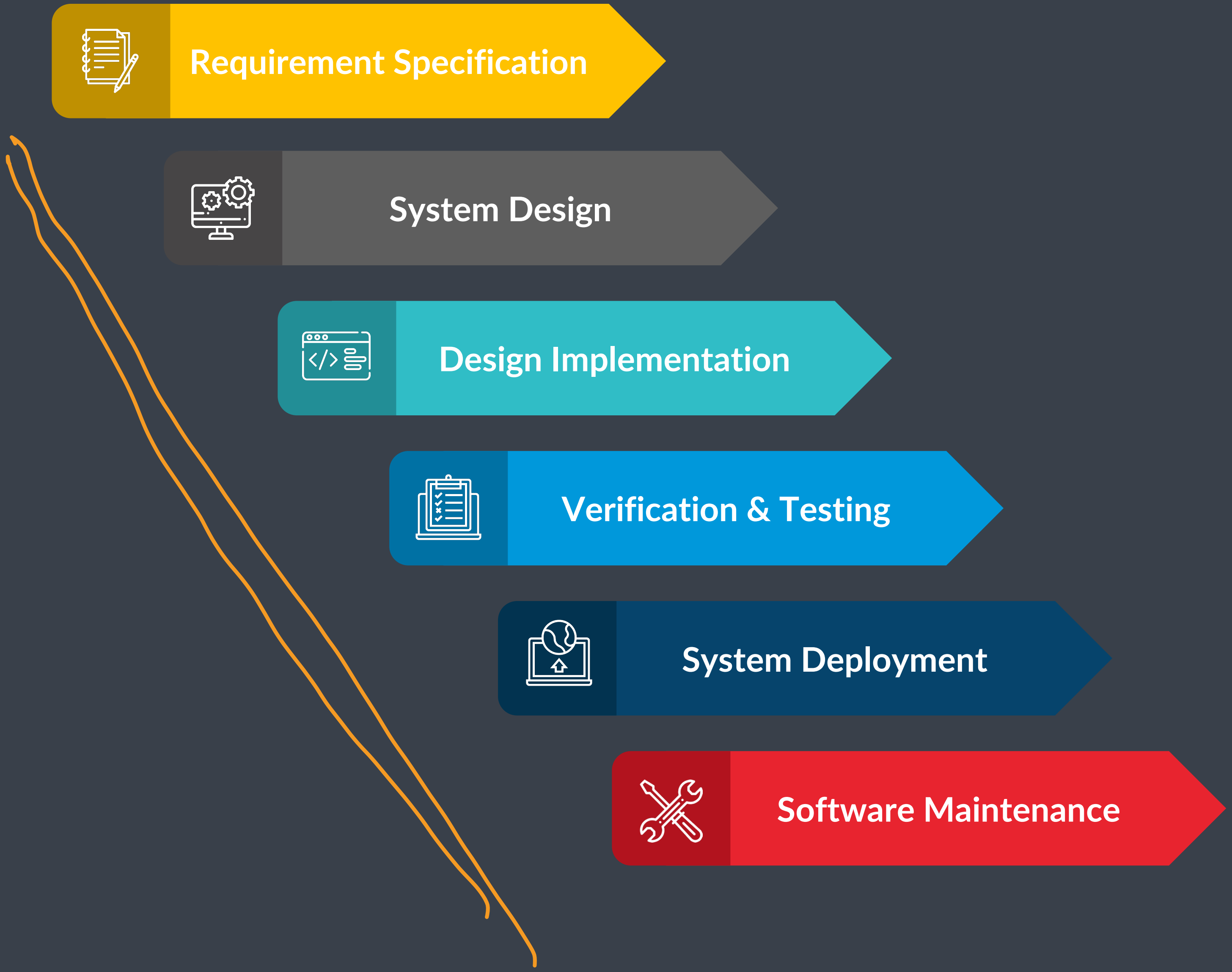
Make sure that your code is working

DEPLOYMENT

Make your app available for rest of the world

↓

Waterfall Model



Requirement Specification

System Design

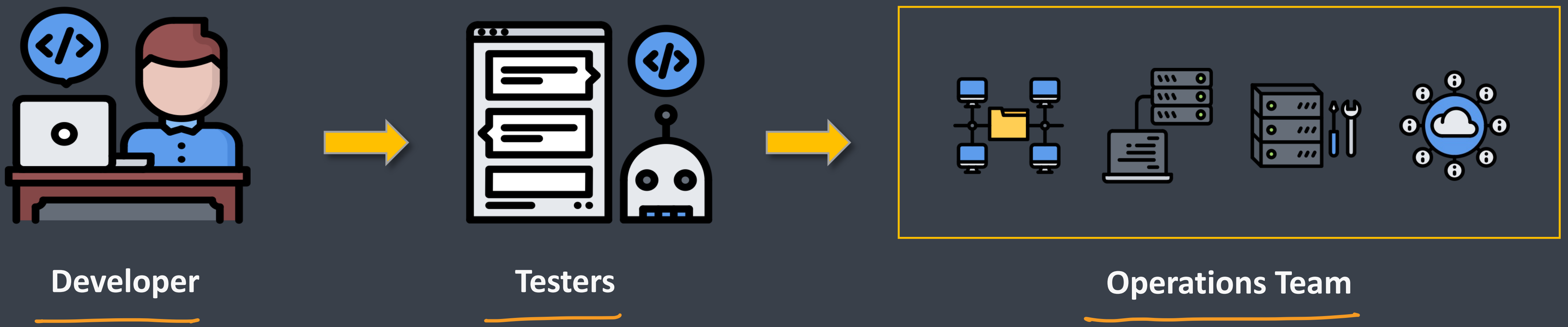
Design Implementation

Verification & Testing

System Deployment

Software Maintenance

Entities involved



Developer

Testers

Operations Team

Responsibilities

Dev Team

Developers and Testers

- Developers
  - Develop the application
  - Package the application
  - Fix the bugs
  - Maintain the application
- Testers
  - Thoroughly test the application manually or using test automation
  - Report the bugs to the developer

package => deployable package

program

resources

dependencies

doc

Ops

Operations Team

- Make all the necessary resources ready
- Deploy the application
- Maintain multiple environments
- Continuously monitor the application
- Manage the resources

Challenges

Developers and Testers

- The process is slow
- The pressure to work on the newer features and fix the older code
- Not flexible

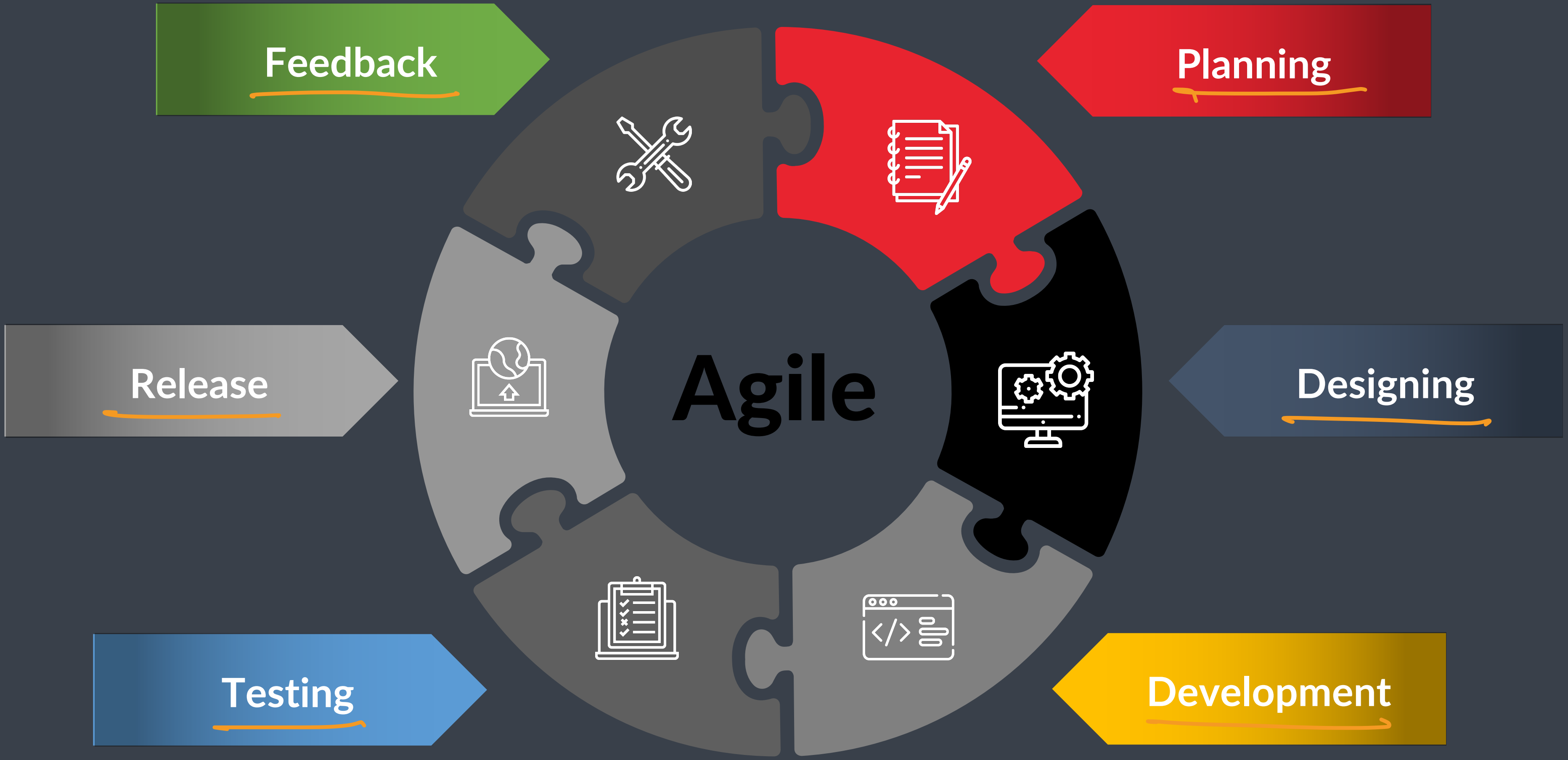
Operations Team

- Uptime
- Configure the huge infrastructure
- Diagnose and fix the issue

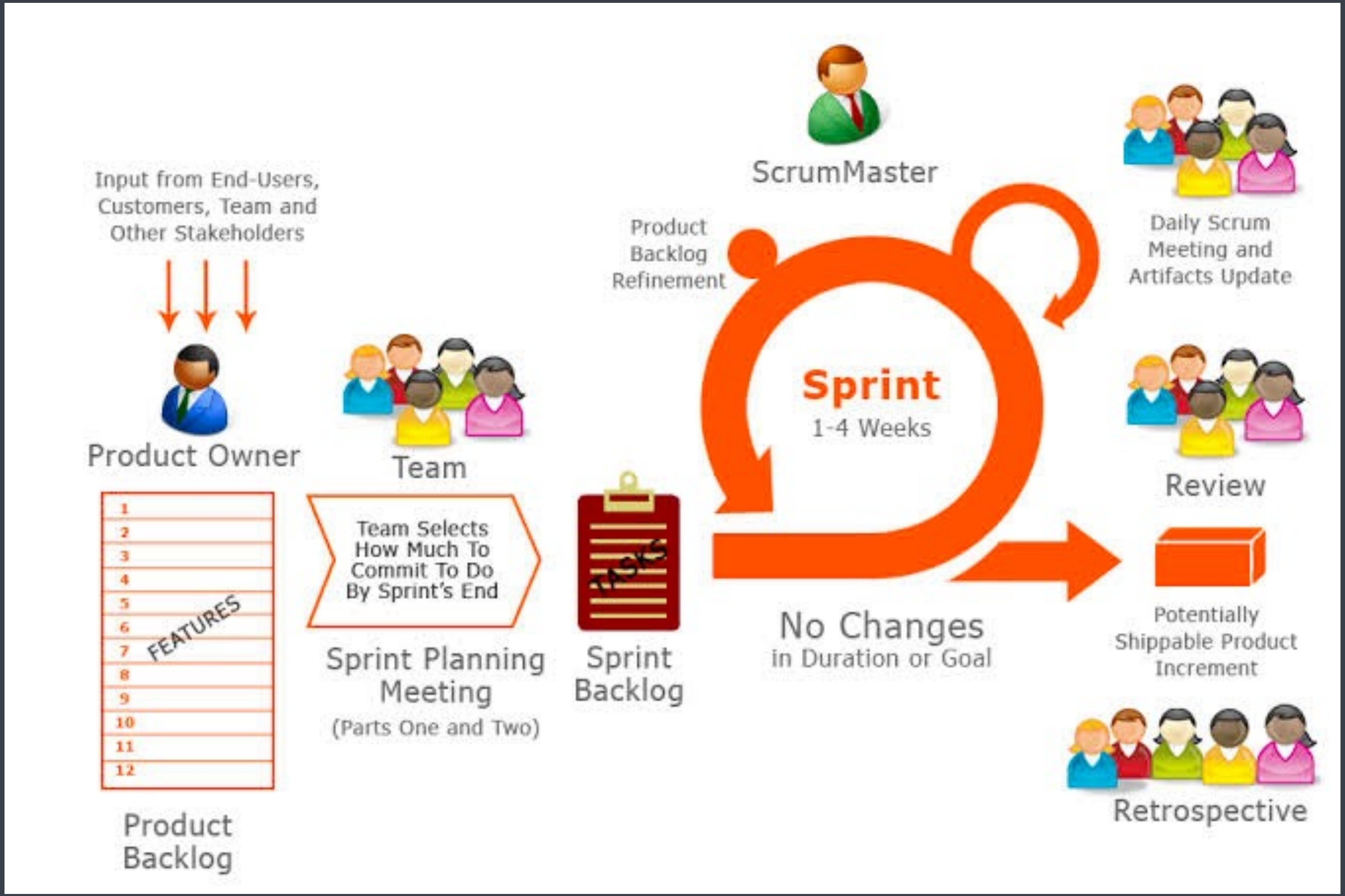
- storage
- servers
  - physical
  - virtual
  - containers
- network
- permission

Agile Development

sprint



Scrum Process


Input from End-Users, Customers, Team and Other Stakeholders
Product Owner
1
2
3
4
5
6
7
8
9
10
11
12
FEATURES
Product Backlog
Team
Team Selects How Much To Commit To Do By Sprint's End
Sprint Planning Meeting
(Parts One and Two)
TASKS
Sprint Backlog
ScrumMaster
Product Backlog Refinement
Sprint
1-4 Weeks
No Changes
in Duration or Goal
Daily Scrum Meeting and Artifacts Update
Review
Potentially Shippable Product Increment
Retrospective

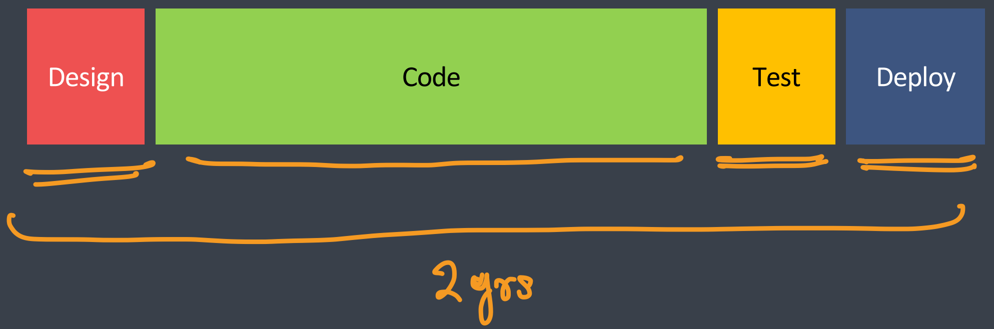
Waterfall Vs Agile

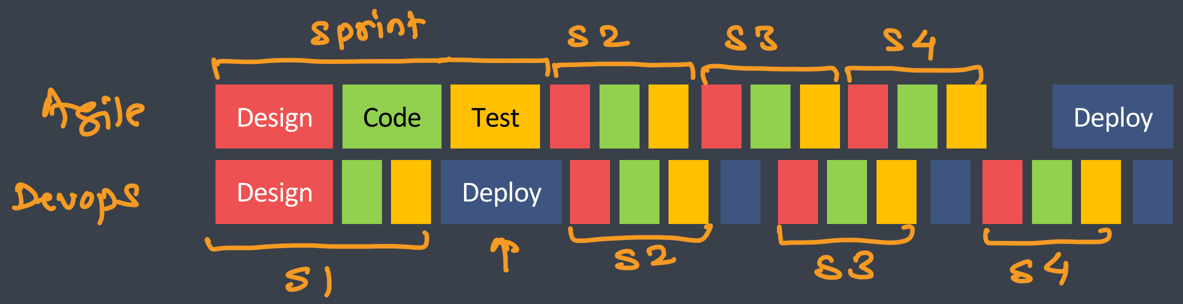
The Waterfall Process

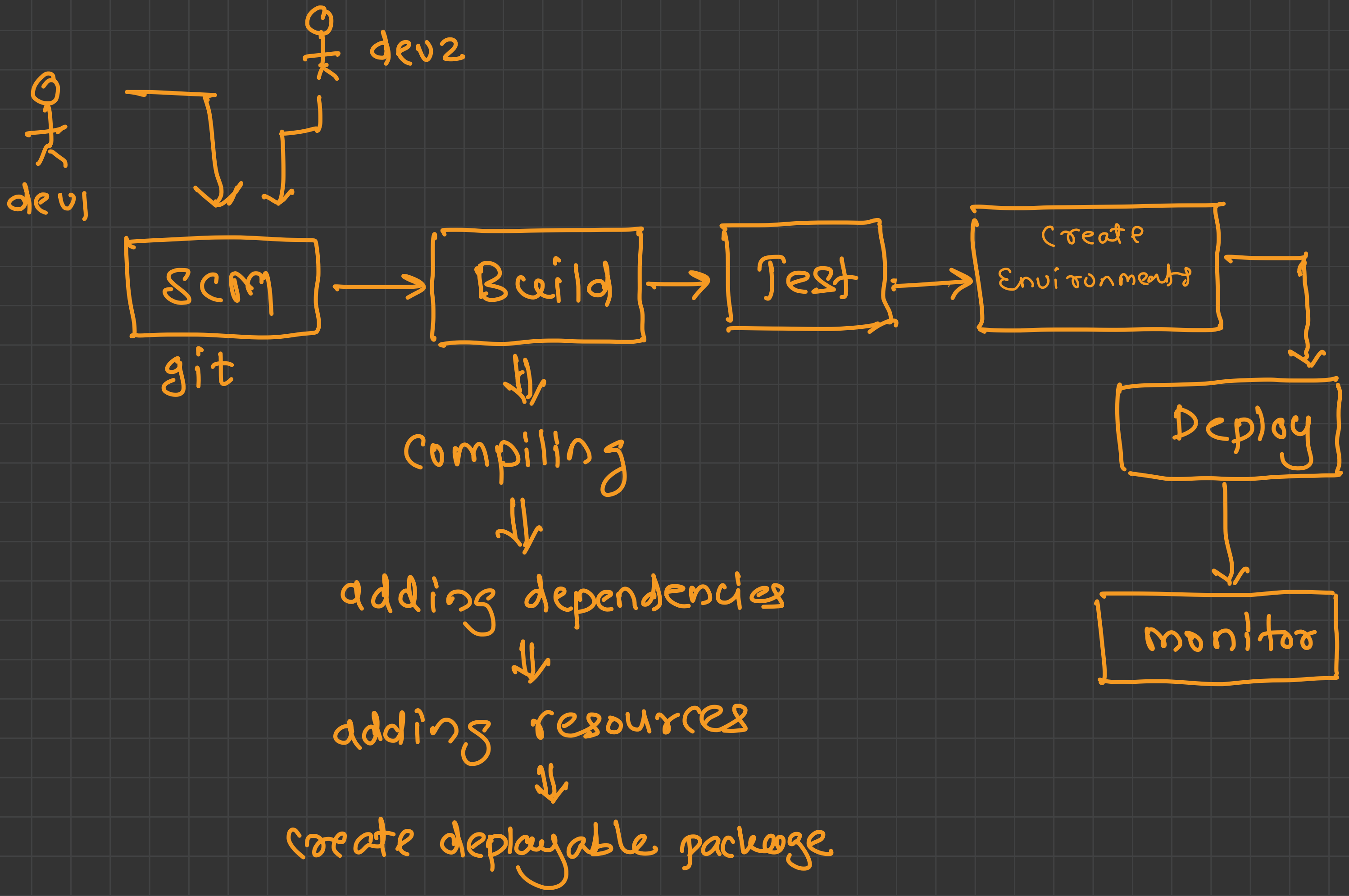
The Agile Process

This project has got so big.
I am not sure I will be able to deliver it!

It is so much better delivering
this project in bite-sized sections






dev2
dev1
SCM
git
Build
Test
Create
Environments
Deploy
monitor
Compiling
adding dependencies
adding resources
create deployable package

Problems

- Managing and tracking changes in the code is difficult
- Incremental builds are difficult to manage, test and deploy
- Manual testing and deployment of various components/modules takes a lot of time
- Ensuring consistency, adaptability and scalability across environments is very difficult task
- Environment dependencies makes the project behave differently in different environments

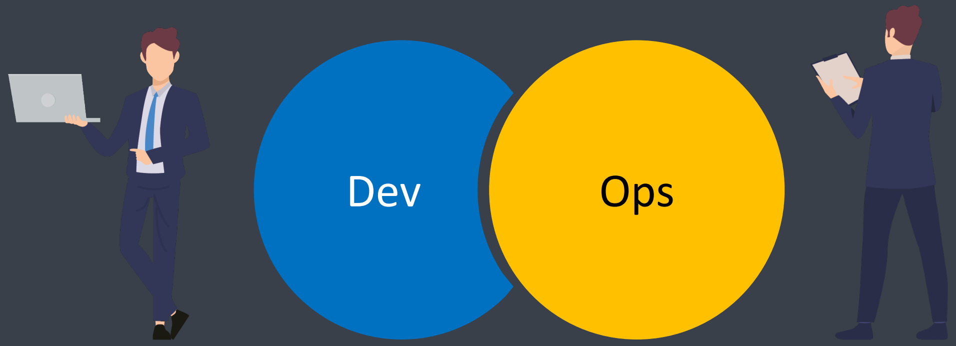
Solutions to the problem

- Managing and tracking changes in the code is difficult: **SCM tools** → versions
- Incremental builds are difficult to manage, test and deploy: **Jenkins** → CI/CD pipeline
- Manual testing and deployment of various components/modules takes a lot of time: **Selenium** : test automation
- Ensuring consistency, adaptability and scalability across environments is very difficult task: **Puppet** : configuration tools
- Environment dependencies makes the project behave differently in different environments: **Docker** : containerization tools

What is DevOps ?

Dev Team

- DevOps is a combination of two words development and operations
- Promotes collaboration between Development and Operations Team to deploy code to production faster in an automated & repeatable way
- DevOps helps to increases an organization's speed to deliver applications and services
- It allows organizations to serve their customers better and compete more strongly in the market
- Can be defined as an alignment of development and IT operations with better communication and collaboration
- DevOps is not a goal but a never-ending process of continuous improvement
- It integrates Development and Operations teams
- It improves collaboration and productivity by
  - Automating infrastructure
  - Automating workflow
  - Continuously measuring application performance



Why DevOps is Needed?

- Before DevOps, the development and operation team worked in complete isolation
- Testing and Deployment were isolated activities done after design-build. Hence they consumed more time than actual build cycles.
- Without using DevOps, team members are spending a large amount of their time in testing, deploying, and designing instead of building the project.
- Manual code deployment leads to human errors in production
- Coding & operation teams have their separate timelines and are not in synch causing further delays

Common misunderstanding

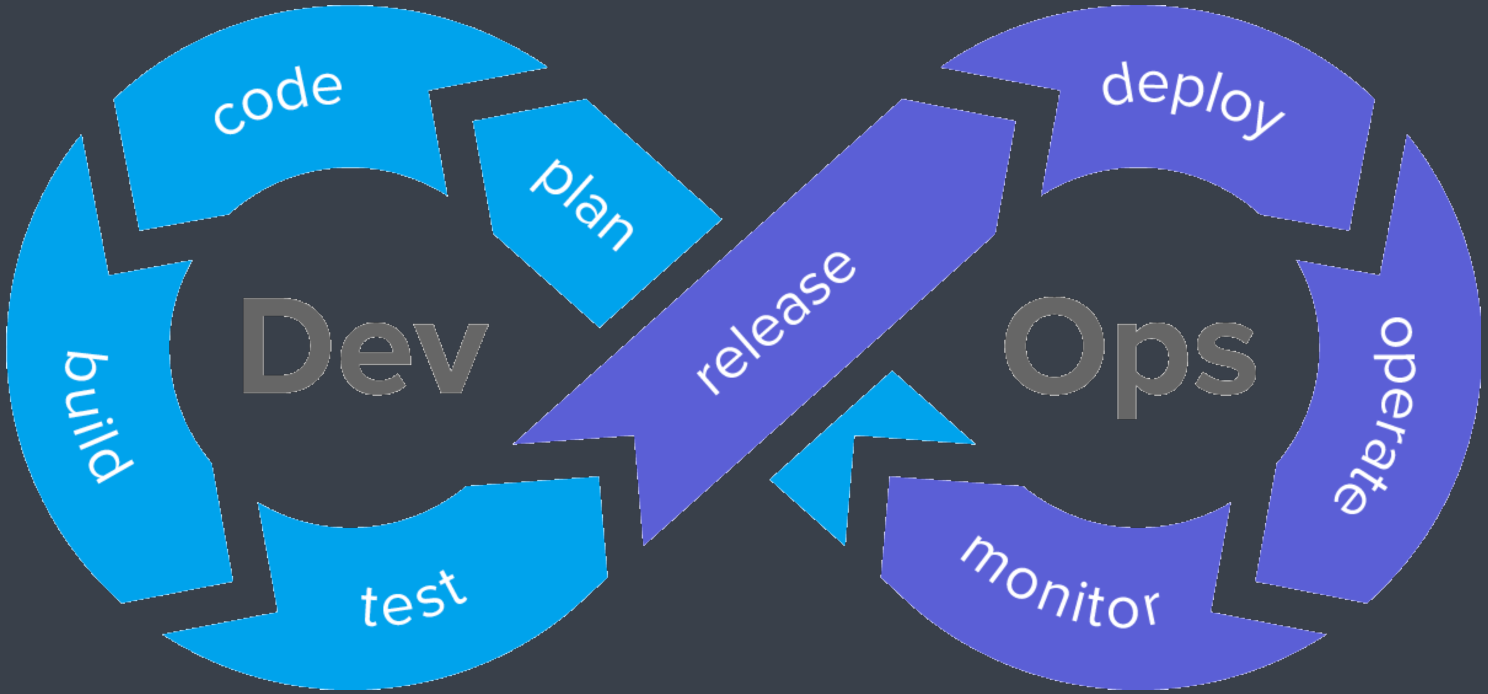
- DevOps is not a role, person or organization
- DevOps is not a separate team
- DevOps is not a product or a tool
- DevOps is not just writing scripts or implementing tools

Reasons to use DevOps

- **Predictability**
  - DevOps offers significantly lower failure rate of new releases
- **Reproducibility**
  - Version everything so that earlier version can be restored anytime
- **Maintainability**
  - Effortless process of recovery in the event of a new release crashing or disabling the current system
- **Time to market**
  - DevOps reduces the time to market up to 50% through streamlined software delivery
  - This is particularly the case for digital and mobile applications
- **Greater Quality**
  - DevOps helps the team to provide improved quality of application development as it incorporates infrastructure issues
- **Reduced Risk**
  - DevOps incorporates security aspects in the software delivery lifecycle. It helps in reduction of defects across the lifecycle
- **Resiliency**
  - The Operational state of the software system is more stable, secure, and changes are auditable

DevOps Lifecycle

continuous → never ending



DevOps Lifecycle - Plan

- First stage of DevOps lifecycle where you plan, track, visualize and summarize your project before you start working on it

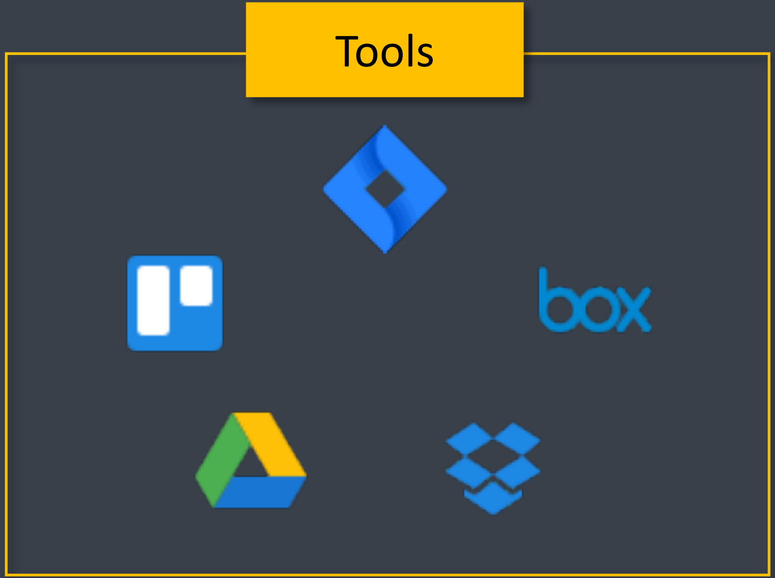
1] text files

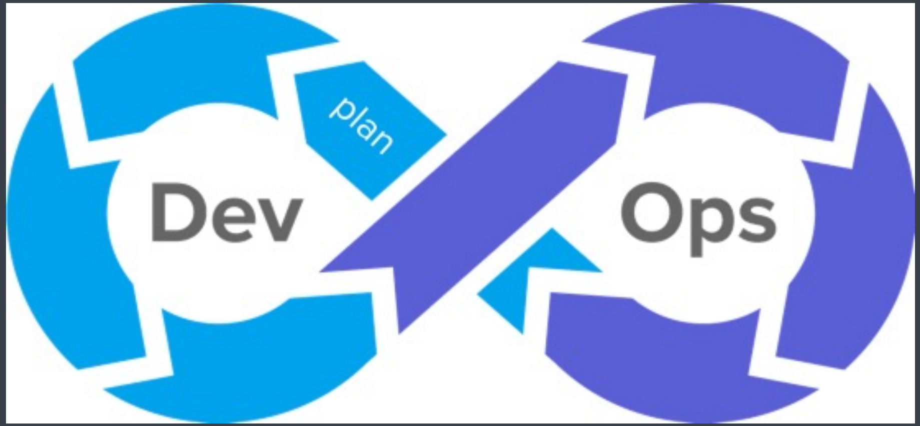
2] Drives → google drive, One drive, box,

3] Excel

4] Jira ←

5] Trello ←





DevOps Lifecycle - Code

- Second stage where developer writes the code using favorite programming language

1] SCM : git, SVN, CVS, bazaar

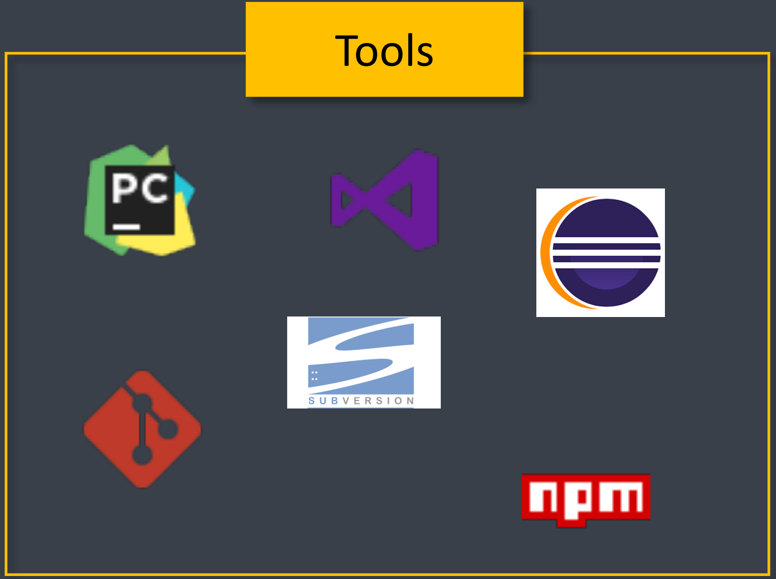
2] IDE : VS, pycharm, webstorm, CLion, Eclipse, netbeans, IDea

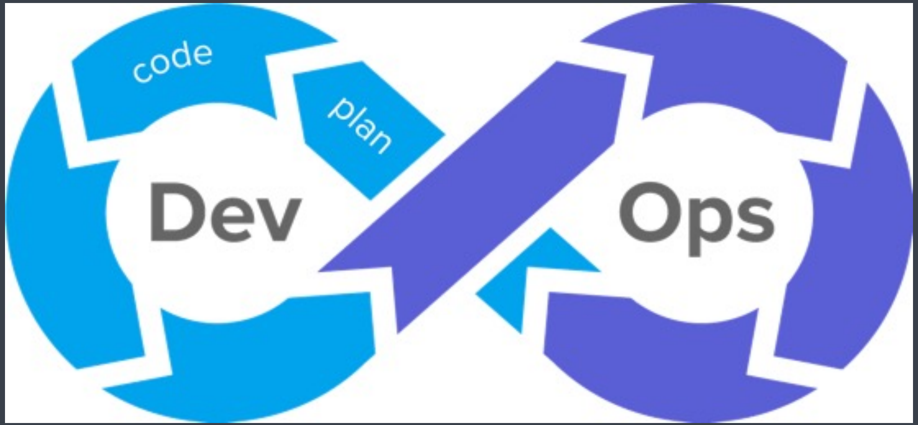
3] Editor: vim, VS code, geditor

4] language: C, C++, C#, Java, pearl, php, golang, kotlin, JavaScript, TS, ruby, python, swift,

5] SDK : JDK, .Net, Android, iOS

6] Stacks : MERN, MEAN, LAMP, WISA, MAMP, WAMP, XAMP





DevOps Lifecycle -Build

- Integrating the required libraries
- Compiling the source code
- Create deployable packages

compilation + dependencies + resources ⇒ package

dependencies → libraries, frameworks

Android ⇒ .apk

ios ⇒ .ipa

web ⇒ .bundle
.webpack

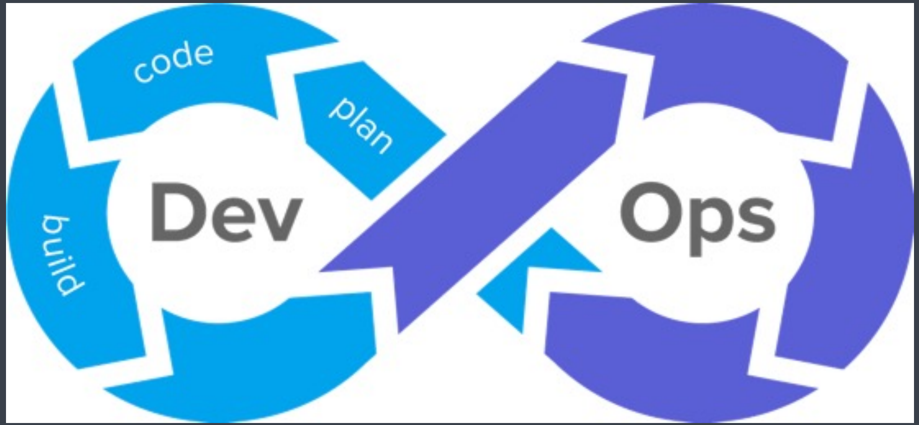
java ⇒ .jar, .war

tools → ant (deprecated)

tools → maven

tools → gradle ***

Tools
maven



DevOps Lifecycle - Test

- Process of executing automated tests
- The goal here is to get the feedback about the changes as quickly as possible

① unit testing : Junit, Nunit, Jest / Jasmine

② UI testing : selinium , cypress

③ system testing : Jest (e2e testing)

④ load testing : win loader, JMeteter

⑤ stress testing : stress tester

Tools

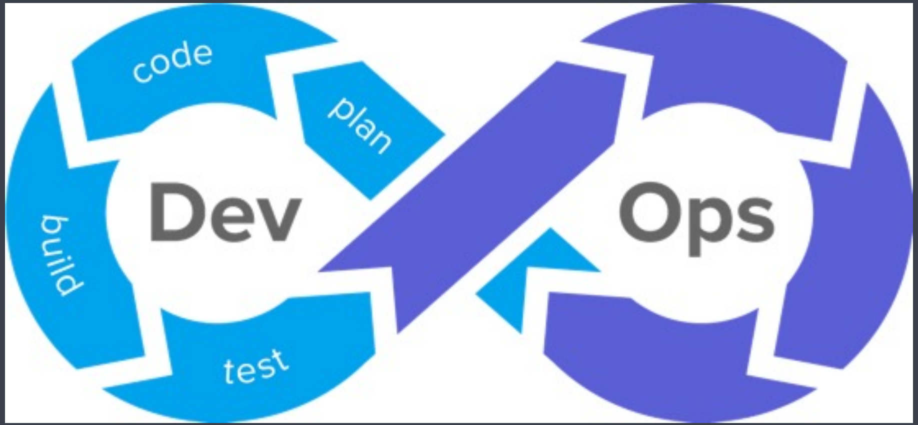
JUnit

nunit

Se

APACHE JMeter™

TestNG



DevOps Lifecycle - Release (automate)

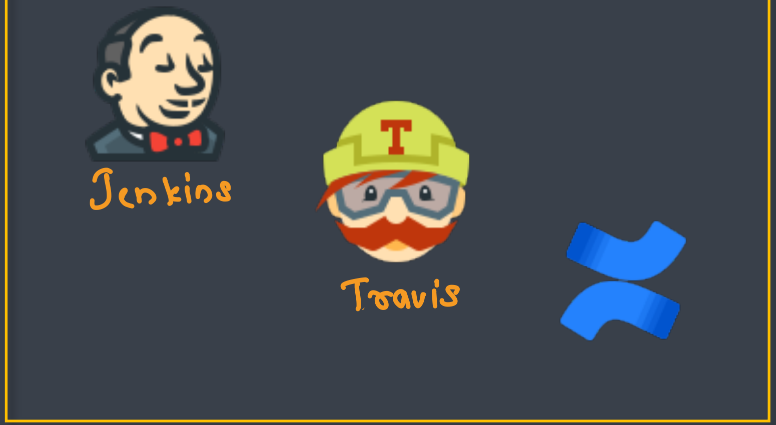
- This phase helps to integrate code into a shared repository using which you can detect and locate errors quickly and easily

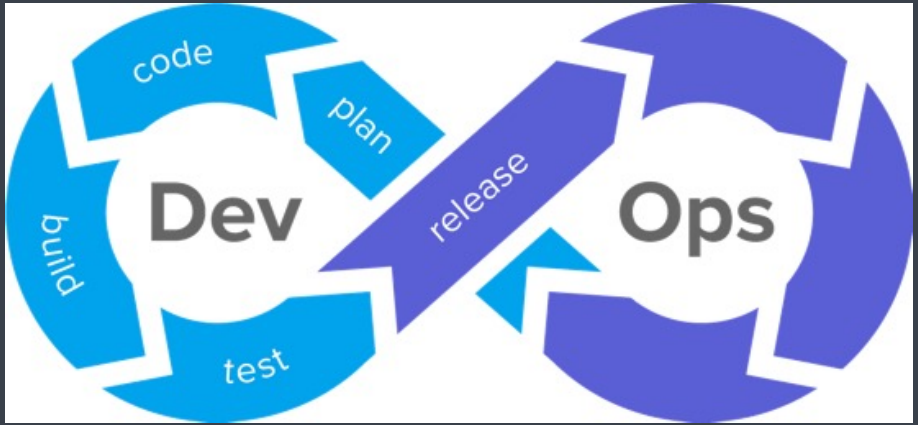
CI/CD tools : Jenkins, Bamboo

CI tools : TravisCI

CD tools : ArgoCD

Tools





DevOps Lifecycle - Deploy

- Manage and maintain development and deployment of software systems and server in any computational environment

1] traditional deployment = physical machines (deprecated)

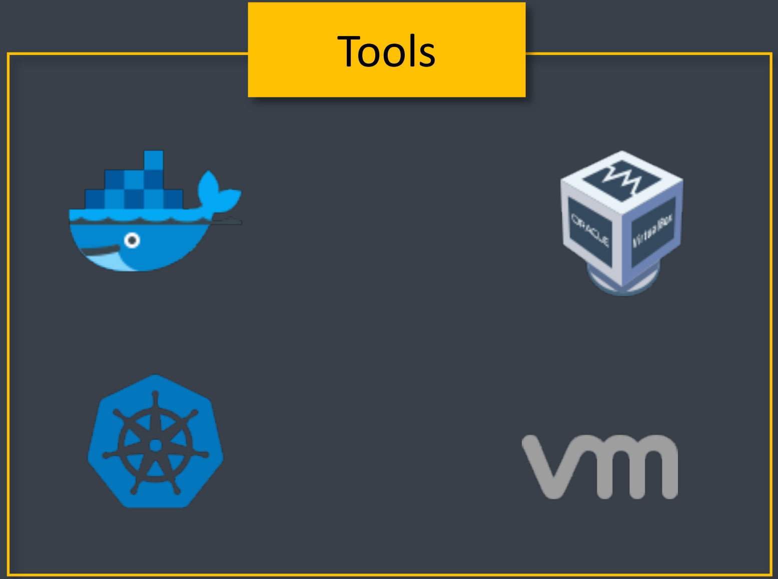
2] virtualized deployment = virtual machines ⇒ on-prem ⇒ vmware, virtual box, parallels

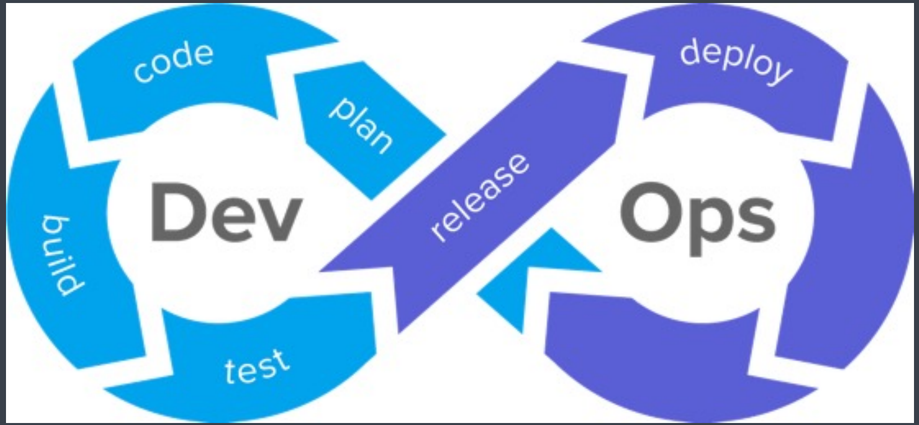
cloud ⇒ AWS, Azure, GCP, Digital Ocean

3] containerized deployment = containers

→ tools : docker, crio, rkt, podman

→ orchestration tools : kubernetes, docker swarm, marathon, mesos





DevOps Lifecycle - Operate

- This stage where the updated system gets operated

configuration tools : Ansible, puppet, chef,
vagrant, terraform

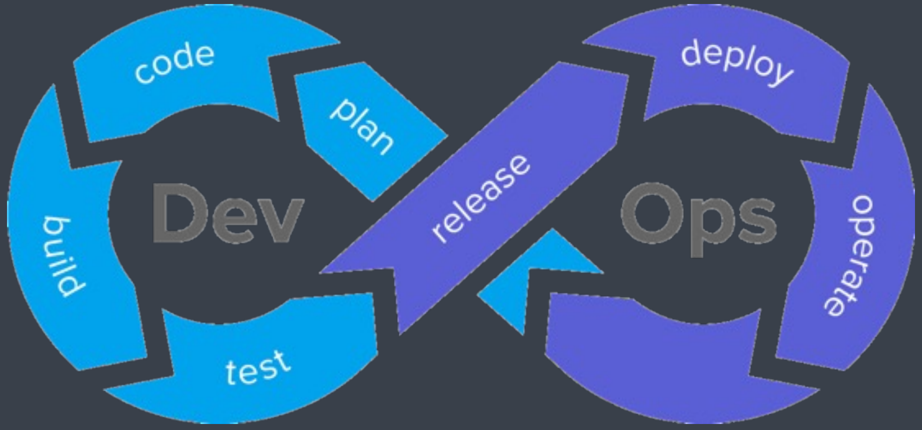
① virtual machines

② networking

③ resource configuration

④ environment configuration

Tools



DevOps Lifecycle - Monitor

- It ensures that the application is performing as expected and the environment is stable
- It quickly determines when a service is unavailable and understand the underlying causes

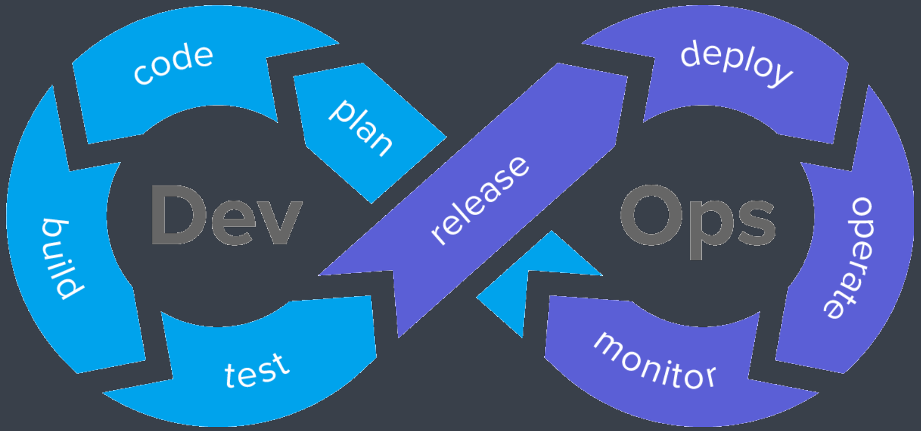
tools : Nagios , data Dog , splunk , new relic

Tools

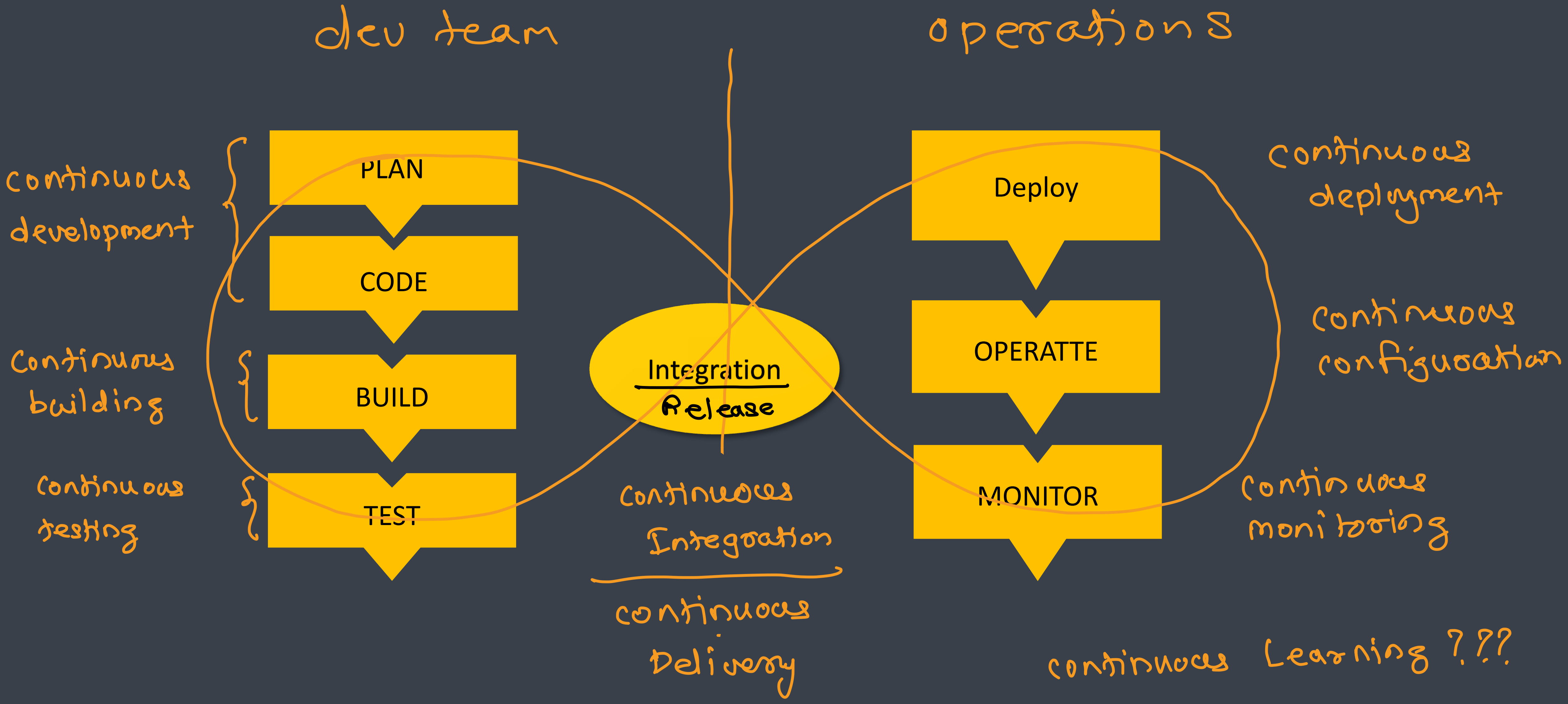
splunk>

DATADOG

Nagios®



DevOps Terminologies



dev team

operations

continuous development

PLAN

CODE

continuous building

BUILD

continuous testing

TEST

Integration

Release

continuous Integration

continuous Delivery

Deploy

OPERATTE

MONITOR

continuous deployment

continuous configuration

continuous monitoring

continuous Learning ???

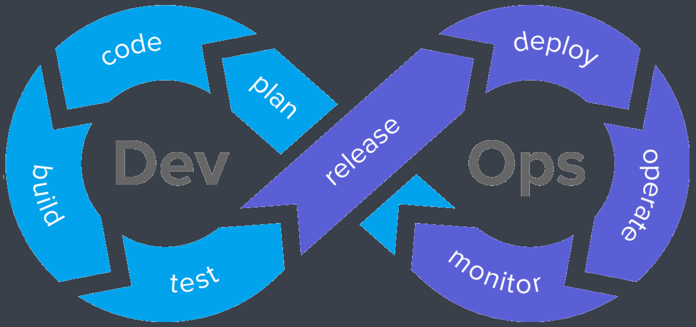
Responsibilities of DevOps Engineer

Be an excellent sysadmin

Deploy Virtualization

Hands-on experience in network and storage

Introduction to coding ×



Soft skills

Automation tools

Software Testing knowledge

IT security

Skills of a DevOps Engineer

Skills	Description
Tools	• Version Control – Git/SVN • Continuous Integration – Jenkins • Virtualization / Containerization – Docker/Kubernetes • Configuration Management – Puppet/Chef/Ansible • Monitoring – Nagios/Splunk
Network Skills	• General Networking Skills • Maintaining connections/Port Forwarding
Other Skills	• Cloud: AWS/Azure/GCP • Soft Skills • People management skill