

# Topics

---

Following are the topics covered in this module:

- Introduction to DevOps
- Benefits of working in DevOps environment
- DevOps Lifecycle
- DevOps Stages
- DevOps Delivery Pipeline

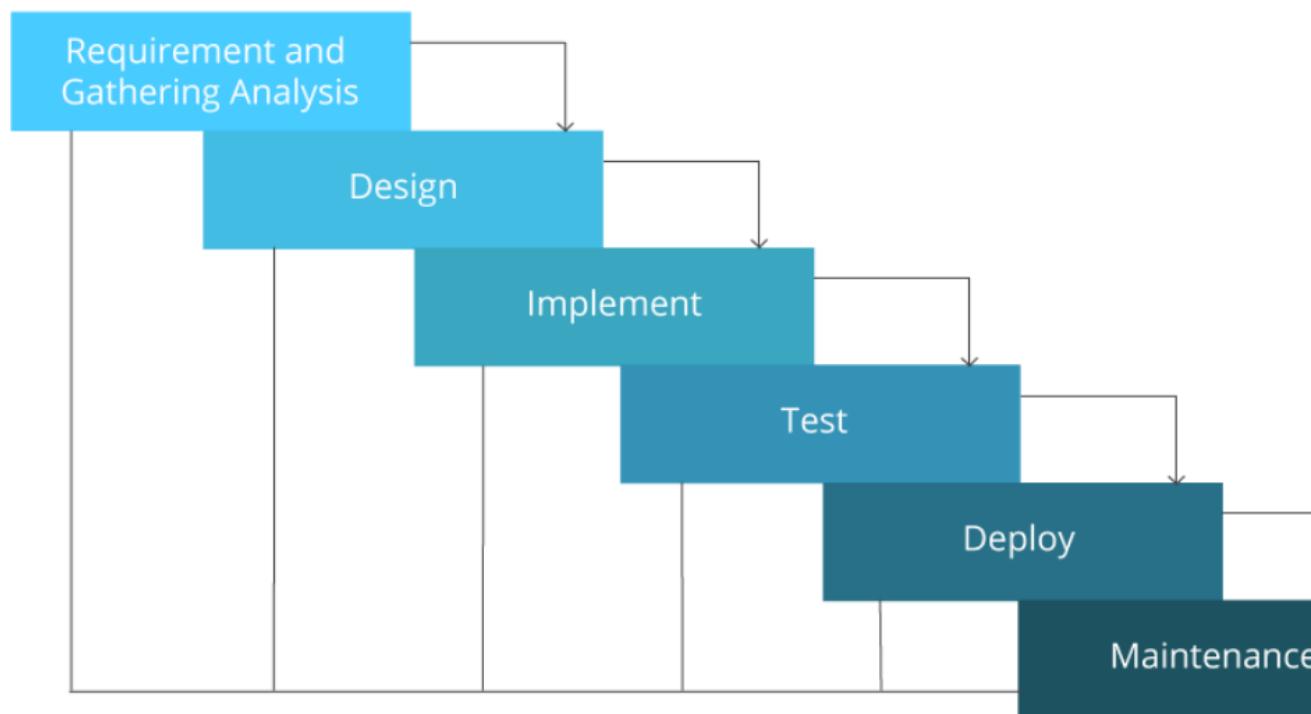
# Objectives

---

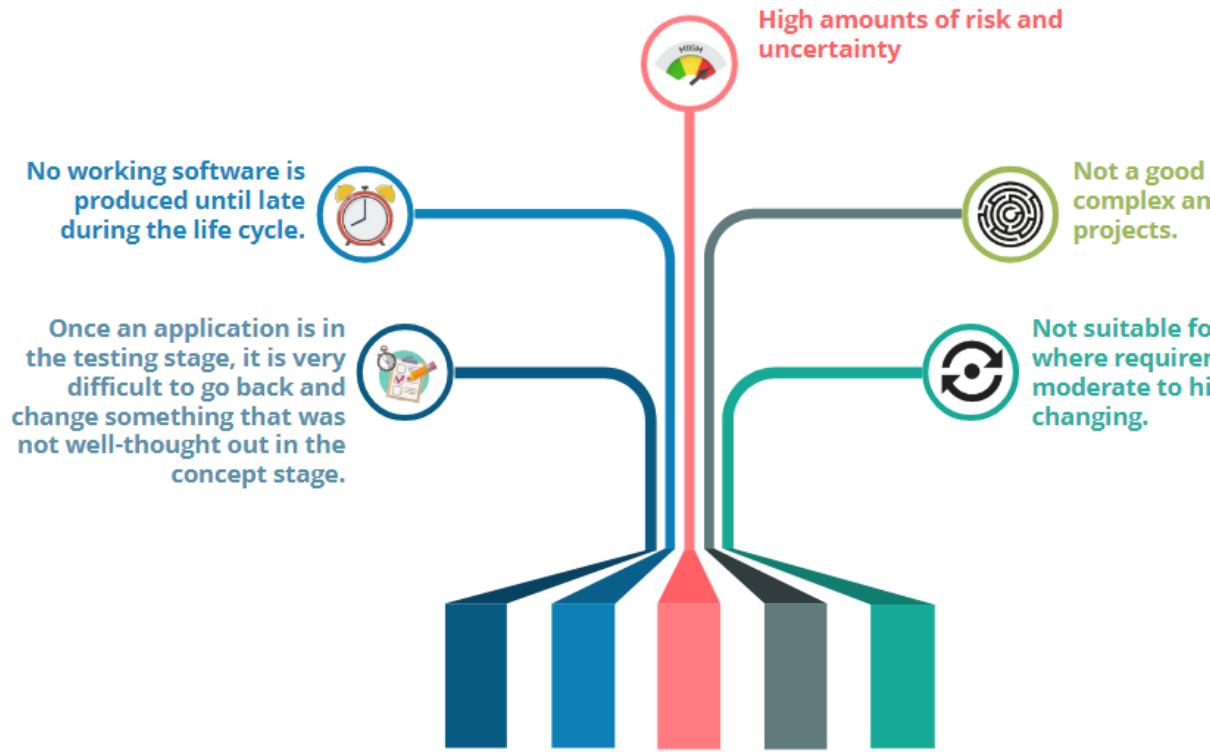
After completing this module, you should be able to:

- Understand the benefits of DevOps over other software development paradigms
- Gain insights into the DevOps environment
- Get an overview of different DevOps Tools
- Get a picture of the working of DevOps Delivery Pipeline

# Traditional Waterfall Model



# Limitations of Waterfall Model



# What is Agile Methodology?

In the Agile Methodology each project is broken up into several 'Iterations'

All Iterations should be of the same time duration (between 2 to 8 weeks)



# Waterfall vs Agile

Waterfall

Analyze

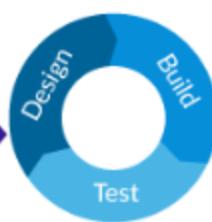
Plan

Design

Bui

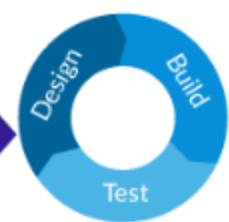
Agile

Analyze Plan



Deploy

Analyze Plan



SDLC - Software Development Lifecycle!!!

Requirement Analysis

Design

Development/Coding

Testing

**Implementation / Deployment**

**Monitoring!**

**Waterfall :::: -- it is Linear in fashion!**

**Requirement Analysis**

**Design**

**Development/Coding**

**java/python**

**Testing**

**Implementation / Deployment**

**Monitoring!**

**To Create a software application/Project!**

**Agile Methodologies!!!**

**Requirement Analysis**

**Design**

**Development/Coding**

**java/python**

**Testing**

**Implementation / Deployment**

**Monitoring!**

**To Create a software application/Project! -- having various modules / iteration.**

**Iteration - 1**

**Module / Function**

**Requirement Analysis**

**Design**

**Development/Coding**

**java/python**

**Testing**

## Implementation / Deployment

## Monitoring!

## Iteration - 2

## Requirement Analysis

## Design

## Development/Coding      java/python

## Testing

## Implementation / Deployment

## Monitoring!

## Iteration - 3

## Requirement Analysis

## Design

## Development/Coding      java/python

## Testing

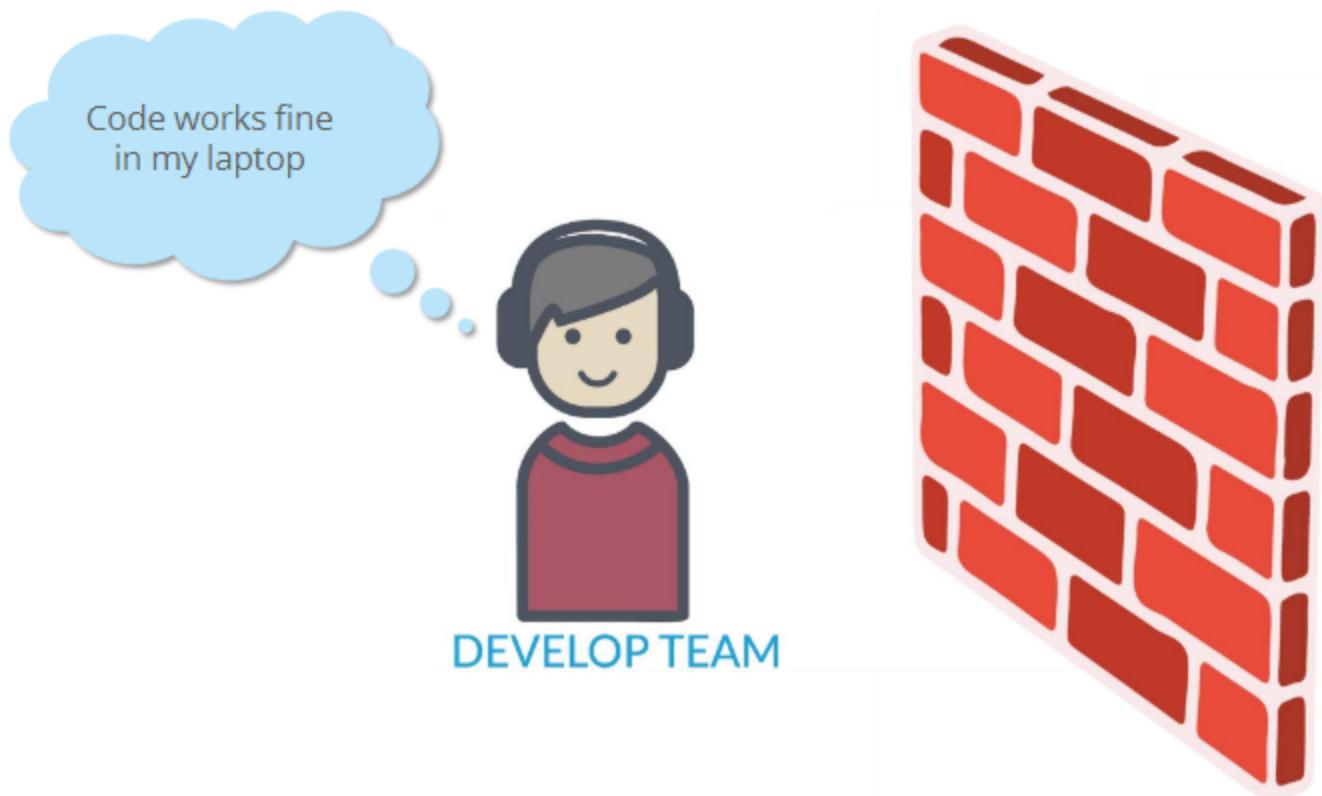
## Implementation / Deployment

## Monitoring!

Finally, all these iterations are integrated and delivered as a software application.

# Limitations of Agile

---

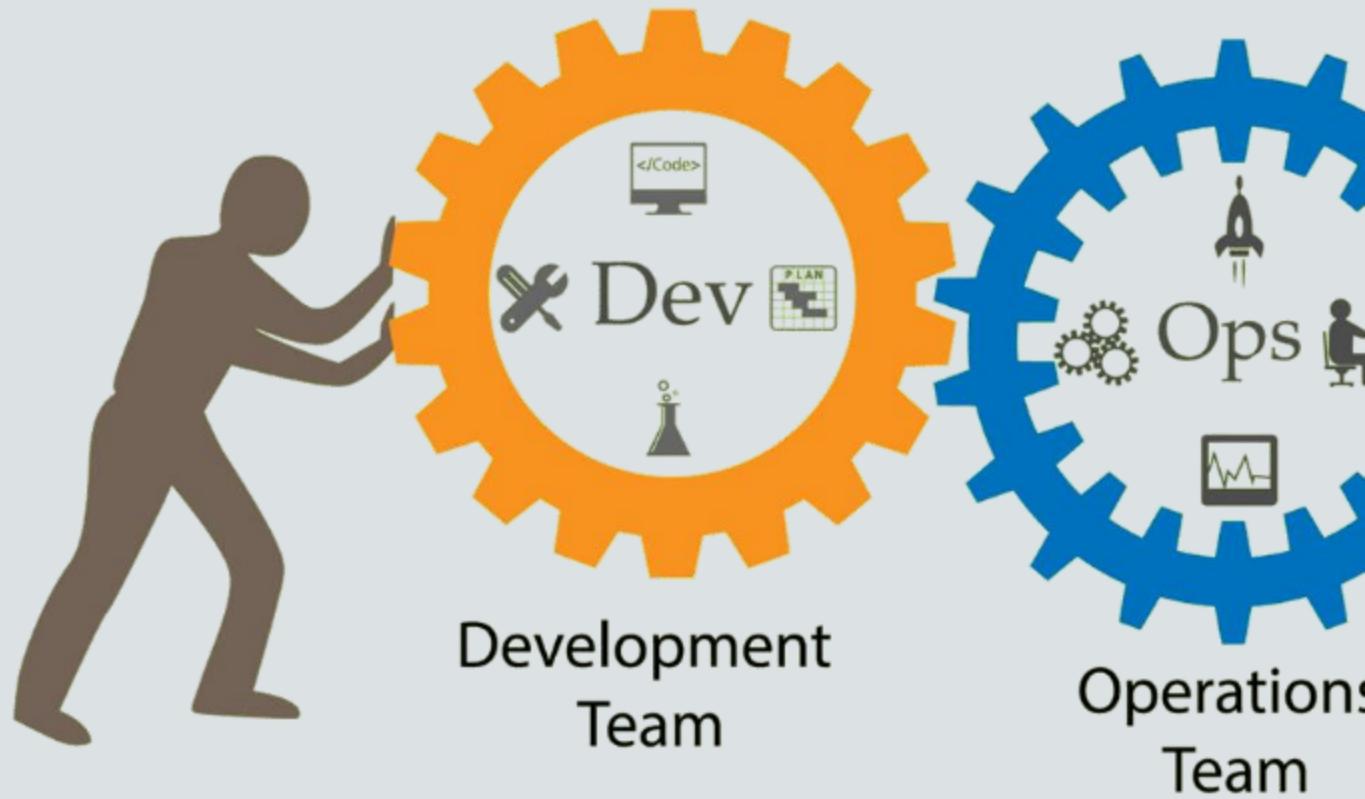


DEVELOP TEAM

Wants Change

WHY DevOps ?

# Solution is DevO



# Development Without DevOps



Release and Deploy  
Mismatch



Unpredictable Issues



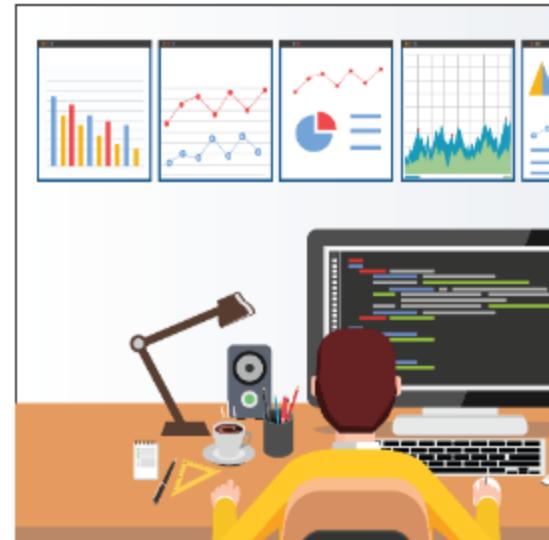
Blame Game

# Development With DevOps Culture

---



Streamlined Deliveries



Continuous Monitoring and

---

# What needs to be done?

## Operations



**Systems should be up and running most of the time**

**Effective monitoring and feedbacks system should be established**

# The Panacea - Devops

## Dev Challenges



Waiting time for code deployment

- Continuous testing



Pressure of work on old, pending and new code

- Since the developer

# The Panacea - Devops

## Ops Challenges



Difficult to maintain uptime of the production environment

Containerization  
environments  
great reliability



Tools to automate infrastructure management are not effective

Configuration  
configuration  
proactively



No. of servers to be monitored increases

Continuous



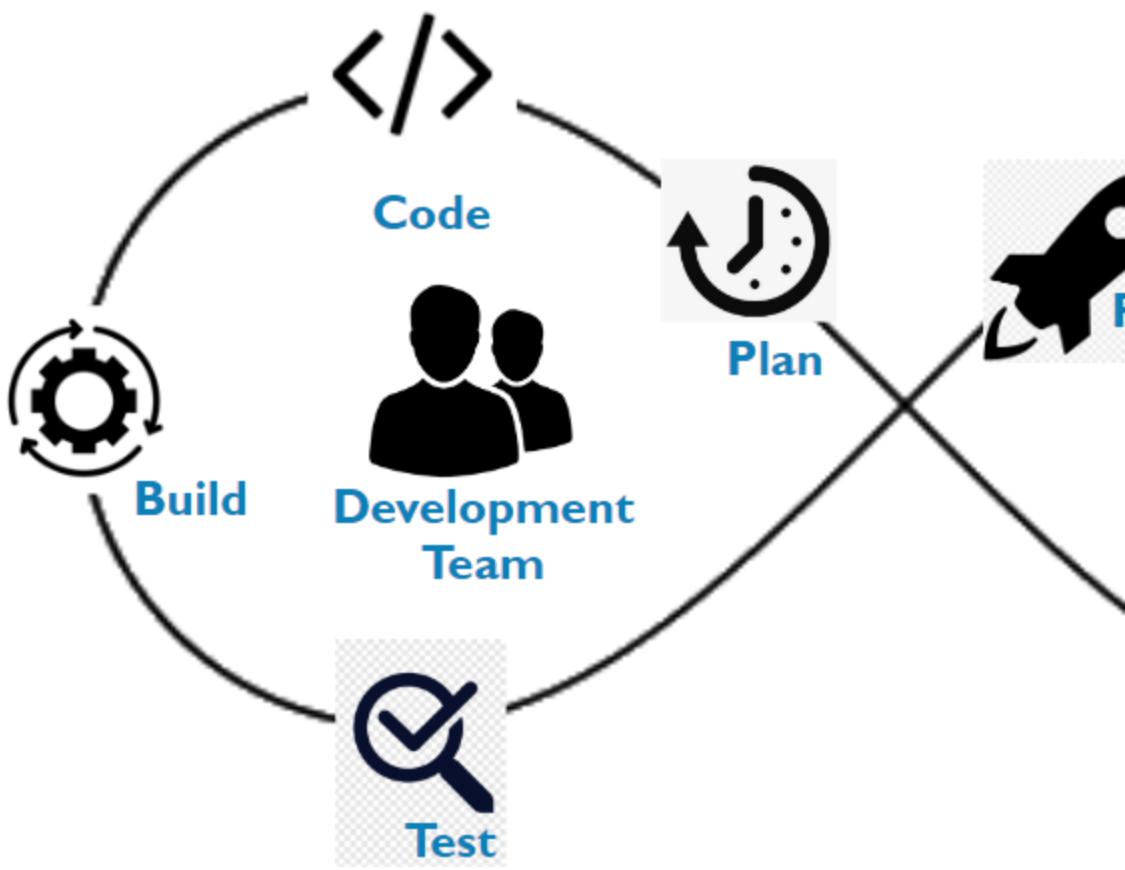
Difficult to diagnose and provide feedback on the product

Effective monitoring  
Nagios  
Thus, effective

# What is DevOps?

DevOps is a Software Development Strategy!

# What Exactly Is DevOps In Real Life



# DevOps: What DevOps Is NOT!!

- 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 about just writing scripts or implementing to

# What Does DevOps Do?

- Integrates developers and operations teams
- Improves collaboration and productivity by:
  - ✓ Automating infrastructure
  - ✓ Automating workflows
  - ✓ Continuously measuring application performance

CODE

Continuous Development !  
Team.

====> related to AppIn. DEV

Continuous Integration !

Continuous Testing !

Continuous Delivery !

Continuous Deployment !

for code development, there are some time frame! --> 2 wks. 1

Integration !!!

Test 2 wks

Software Application/Product!

- Monolith Application Architecture!

All the process are deployment as one Application!!!

Process / tasks / Functions / Modules

- Microservices based Architecture!

Create a service

1 process === 1 service

E-commerce :

www.amazon.com

List of Process / tasks / Functions / Modules :::

User Sign-Up  
the coding is completed(Testing)! -- Developer-1 -- once

User Sign-In

-- Developer-2 --

Search for products

Add the product to cart

Buy the product

fill the form

Make payment

Place the Order

Track the status

Application Owner / Architect

Difference between:

Continuous Integration !

Continuous Delivery !

Continuous Deployment !

Strategy -- Process -- --- Always subjected to Continuous Improvement!!!

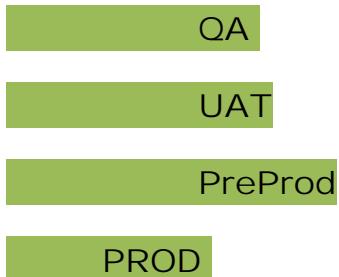
Development Workflow :::

Code the changes / Module / Service  
Java/Python/NodeJS/Angular/.Net (Programming Languages)        ==>

Build -- Compile and Create Artifacts(\*.war/\*.jar/\*.exe)!!!

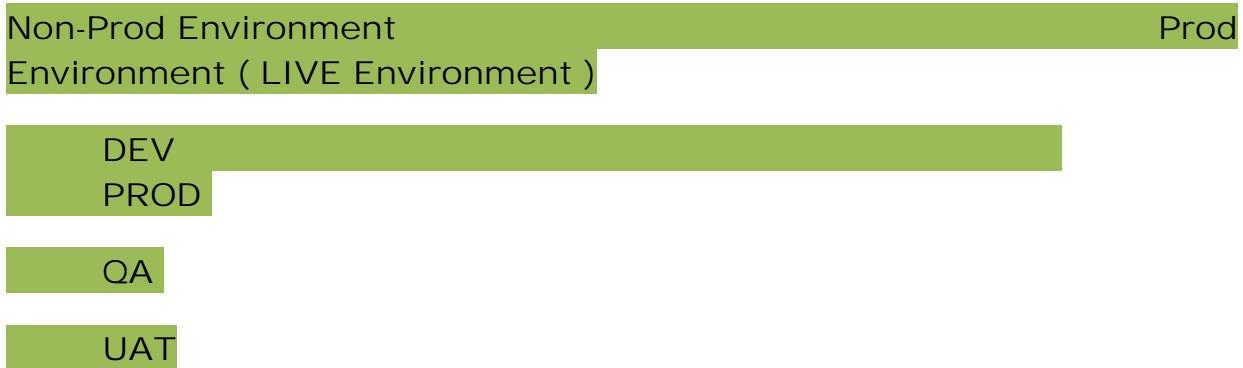
Perform Unit Test

Testing Teams



Continuous Integration :::

Code/Build/Create Artifacts and Integrate for further testing



The Production Release can be done by :

Continuous Delivery : Before the code is promoted to PROD envi. we need Manual Approval from various Teams.

Continuous Deployment : The Code can be automatically promoted to PROD Environment without any Manual Intervention.

As a DevOps Engg. is to automate this end-to-end CI/Cd/CD

Code Build, Create Artifacts, Integrate for further testing(QA/UAT) and PROD Deployment

Banking Appln.

Facebook

Netflix

amazon.com

google

People

Process & Strategies

Tools

Business :::

DevOps Team :::

Identify the various Teams involved in the end-to-end SDLC Process

Infra

Dev

Test

Release Management

Monitoring Team

Prod Support Team

Security Team

Vendors

Creating CI/CD Pipeline

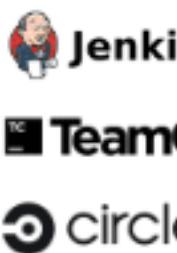
Detailed DevOps assessment

Waterfall --> Agile --> DevOps --> DevSecOps --> SRE -->  
GitOps --> AIOps --> FinOps

DevOps Tools :::

Below the graphic, you'll find helpful descriptions of each of the

## Scripts



## Deploy Tools

## Virtualization & Containerization



Infra-structure Team -- Provision the servers and config the servers - Terraform/Ansible/Puppet/Chef/CF/ARM

Dev (Version control system) -- VS Code / GIT / GITHub

Test -- Selenium/Junit

Release Management -- Ansible/Docker/Kubernetes/

Monitoring Team -- Prometheus/Grafana/Nagios/

Prod Support Team -- AppDynamics  
Security Team -- Static/Dynamic Code Analyzer - Sonarqube

Build Orchestrator or CI /CD Tools:  
Jenkins  
Bamboo  
Gitlab-ci  
Azure Pipeline

## Developer!!!

samplepgm.java index.jsp

Version Controlled!

It should be easily tracked!

Visual Studio Code (IDE) --> to create the source.

whenever we save the program -- it shd be assigned with a version.

index.jsp v1.0

Tested -- successful

index.jsp v1.1

index.jsp v1.2

index.jsp v1.3

index.jsp v1.4

index.jsp v1.5

Tested -- Not working as

expected

GIT --> Distributed VC System

## scripts :::

shell

groovy

yaml

# python scripts

DevOps Engg. write Automation Scripts.

GIT ---> Distributed VCS ===> Local repo & Remote Repo

## Git is a Local Repository

## Remote GIT Repo / Servers

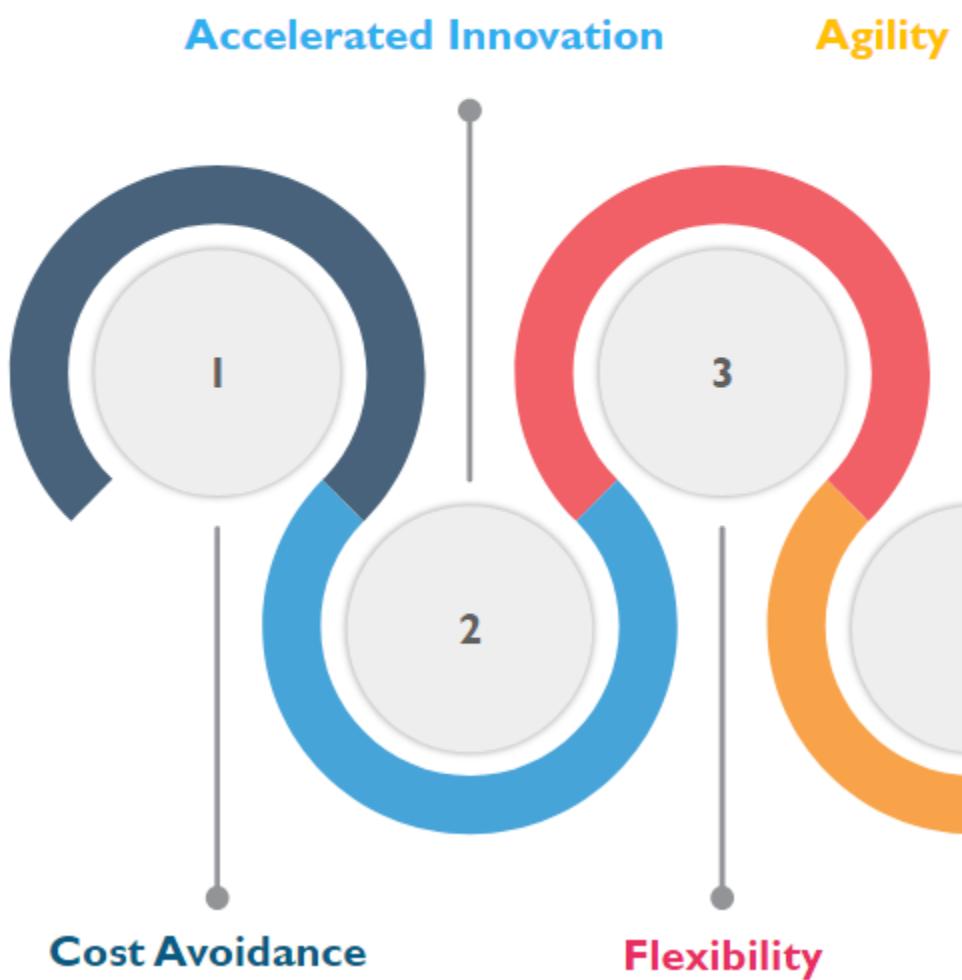
# GitHub is a Remote Git Repository

# Azure Repo

## Bitbucket

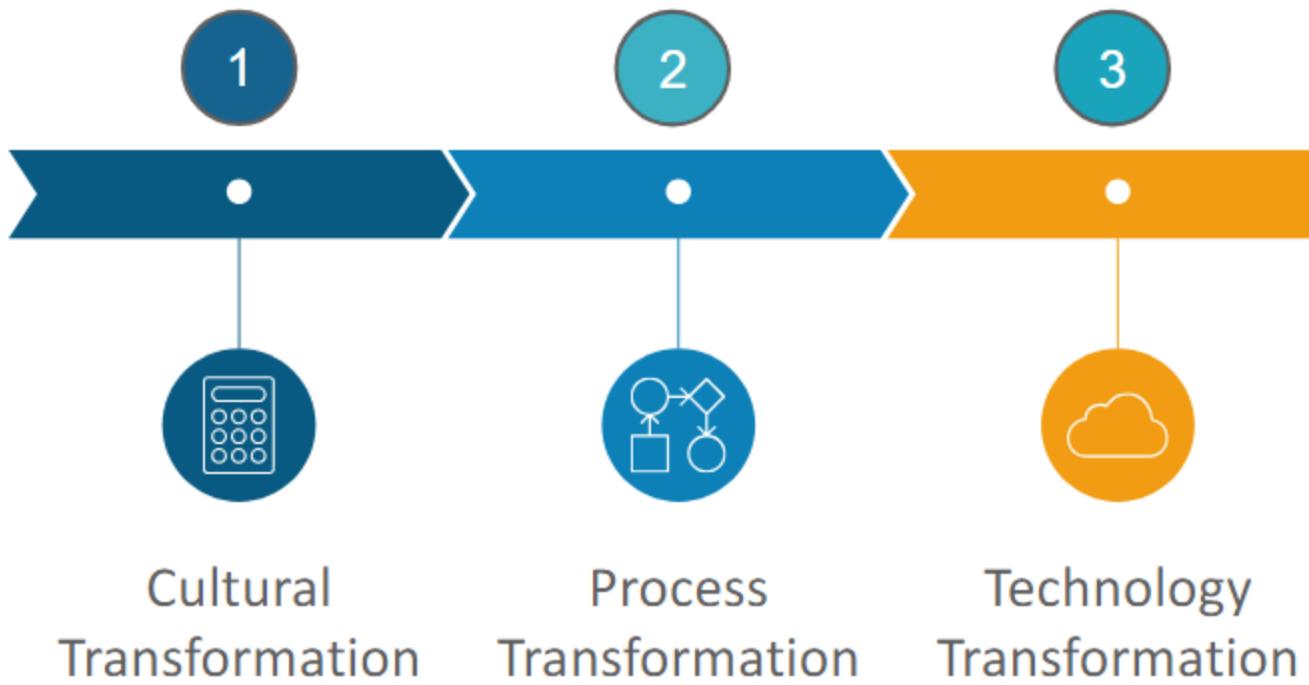
# Gitlab

# Benefits of DevOps



# DevOps Adoption

---



# Skills Of A DevOps Engineer

| Skills                   | Details  |
|--------------------------|--|
| <b>Tools</b>             | <ul style="list-style-type: none"><li>• Version Control - GIT</li><li>• Continuous Integration - Jenkins</li><li>• Virtualization/ Containerization – Docker</li><li>• Configuration Management – Puppet</li><li>• Monitoring – Nagios</li></ul> |
| <b>Networking Skills</b> | <ul style="list-style-type: none"><li>• General networking skills – Establishing connections between hosts, containers/Port Forwarding/ Container Networking</li></ul>   |
| <b>Other Skills</b>      | <ul style="list-style-type: none"><li>• People Skills</li><li>• Process Skill</li><li>• Customer Skill and Empathy</li><li>• Cloud Awareness</li></ul>   |

# DevOps Lifecycle: Plan

“First stage of DevOps cycle, where you **Plan**, **Track**, **Visualize** and **Summarize** your Project before working/starting it.”

Planning Too



# DevOps Lifecycle: Code

---

“Second stage of DevOps cycle, where the developers write their code”



# DevOps Lifecycle: Test

---

“Process of executing automated tests as part of the software delivery pipeline in order to obtain feedback on the business risks associated with a software release as rapidly as possible”



# DevOps Lifecycle: Release

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



# DevOps Lifecycle: Deploy

“Manage and maintain development and deployment of software systems and servers in any computational environment”



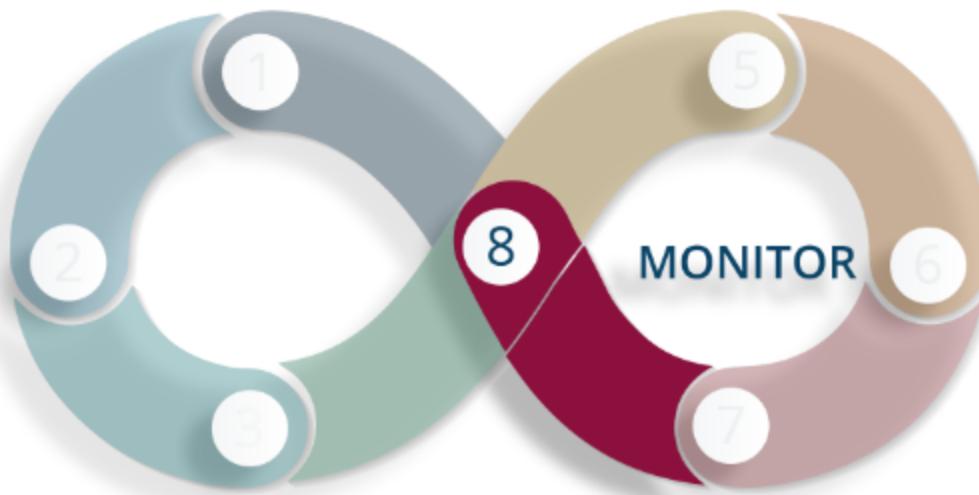
# DeployDevOps Lifecycle: Operate

“This phase is to keep the system upgraded with the latest update”



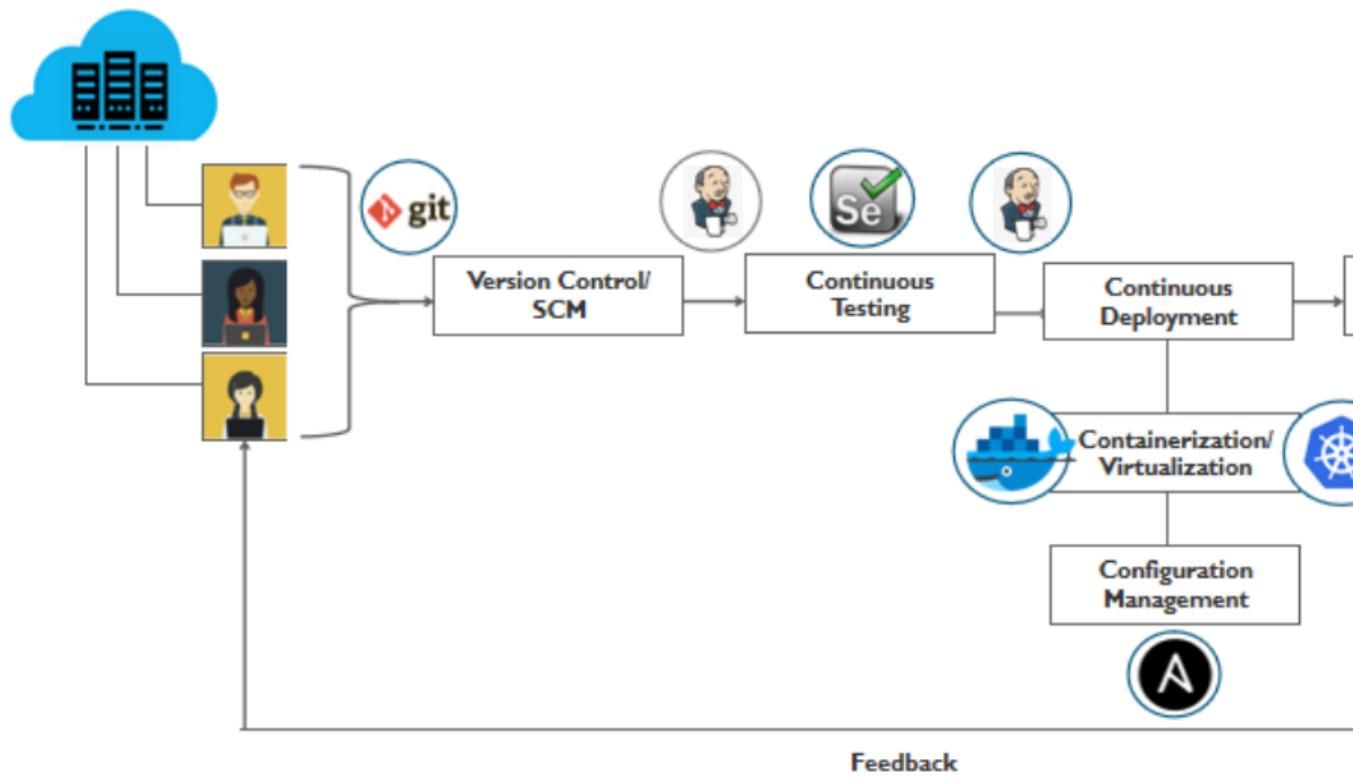
# DevOps Lifecycle: Monitor

“It ensures that the application is performing as desired and the environment is stable. It quickly determines when a service is unavailable and understand the underlying causes”



# DevOps Stages

---



# What Is Continuous Integration?

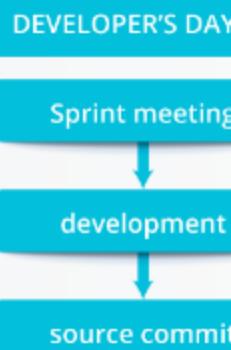
---

A development practice that requires developers to integrate code into a shared repository several times a day. By integrating regularly, you can detect errors quickly, and locate them more easily.

# What Is Continuous Testing?

Process of executing automated tests as part of the software delivery pipeline to obtain immediate feedback

Con



# What Is Continuous Deployment

---

**Continuous Deployment** is a **DevOps** practice where the code changes are automatically built, tested, and prepared for a release to production



# What Is Containerization?

---

**Containerization** is a lightweight alternative to full machine virtualization which involves encapsulating an application in a container with its working environment.



# What Is Configuration Management?

---

Process of standardizing resource configurations and enforcing the state across IT infrastructure in an automated yet agile manner.

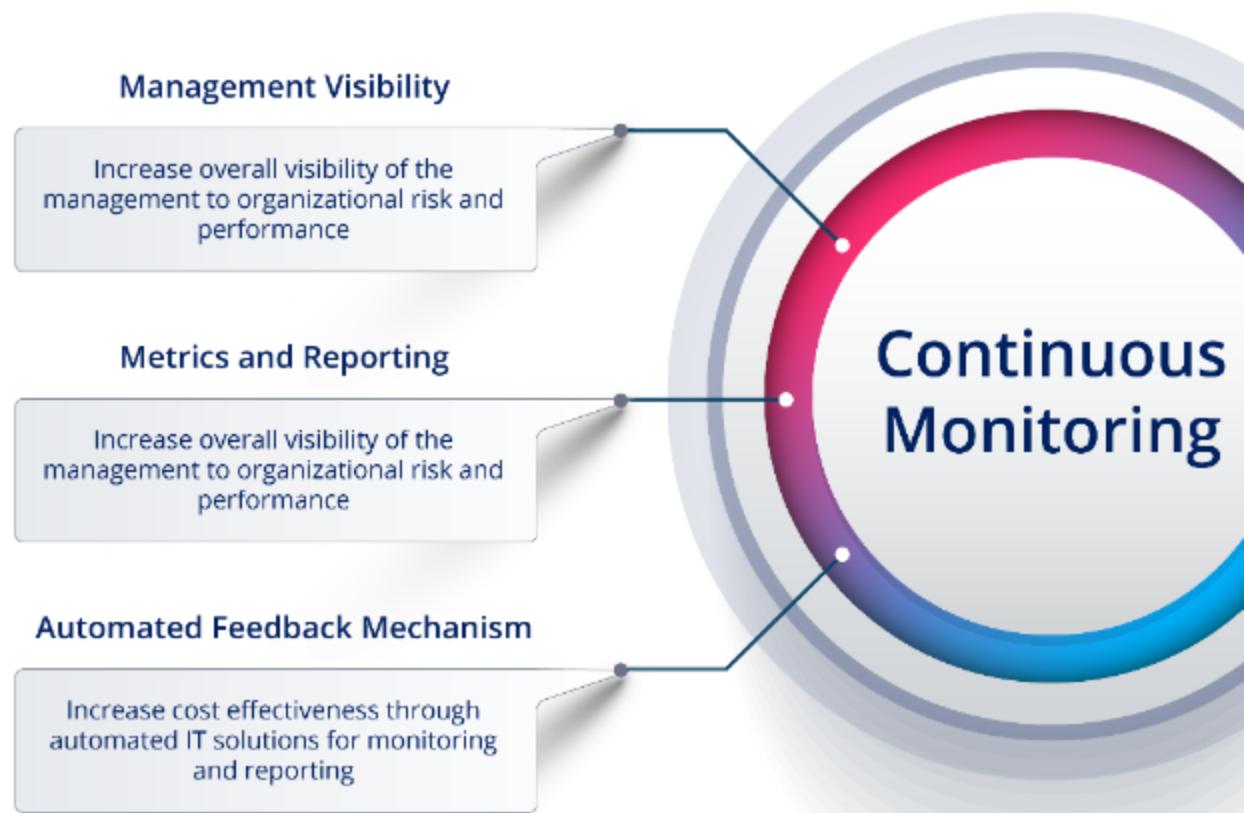
# Configuration Management Using Ansible

- Ansible is an open source IT Configuration Management, Deployment & Orchestration tool
- It aims to provide large productivity gains to a wide variety of automation challenges



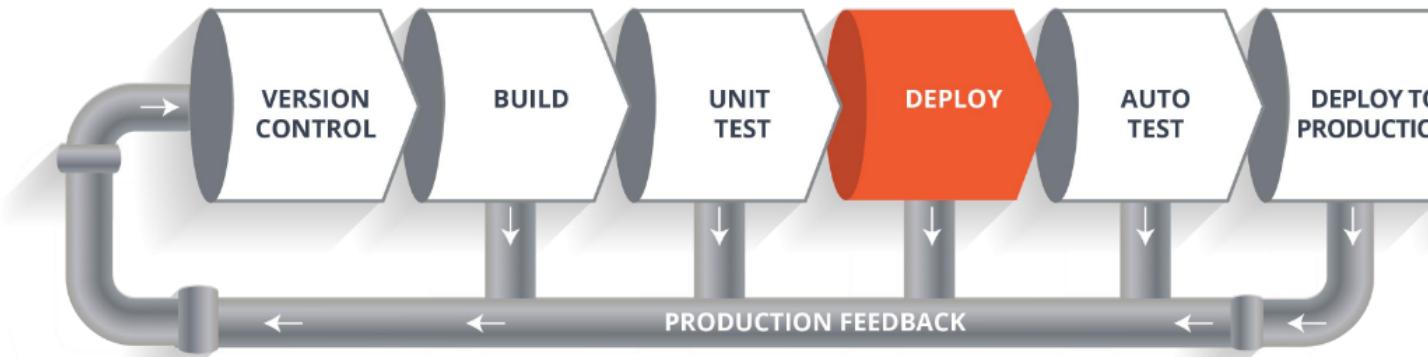
A N S

# What Is Continuous Monitoring?



# DevOps Delivery Pipeline

---



k

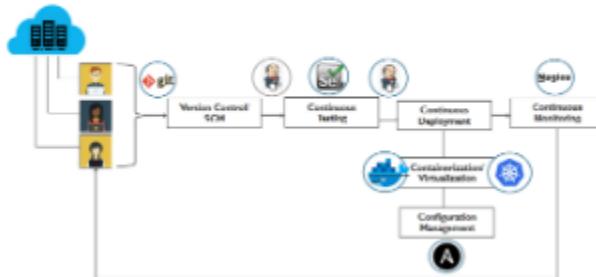
# Summary

## What Is DevOps?

DevOps is a practice that allows a single team to manage the entire application development life cycle, that is, development, testing, deployment, and monitoring.



## DevOps Stages



## Limitations of Waterfall Model



## What Is Configuration Management?

Process of standardizing resource configurations and enforcing their state across IT infrastructure in an automated yet agile manner.

