

Get the best out of Live Sessions HOW?



Check your Internet Connection

Log in 10 mins before, and check your internet connection to avoid any network issues during the LIVE session.



Speak with the Instructor

By default, you will be on mute to avoid any background noise. However, if required you will be **unmuted by instructor**.



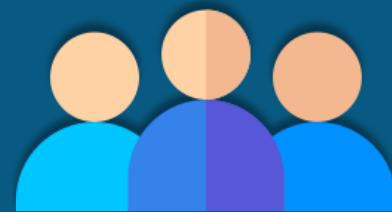
Clear Your Doubts

Feel free to clear your doubts. Use the “**Questions**” tab on your webinar tool to interact with the instructor at any point during the class.



Let us know if you liked our content

Please share feedback after each class. It will help us to enhance your learning experience.



edureka!



DevOps Certification Training

COURSE OUTLINE

MODULE 01

1. Introduction to DevOps

2. Version Control with Git

3. Git and Jenkins

4. Continuous Integration with Jenkins

5. Configuration Management using Ansible

6. Containerization using Docker Part - I

7. Containerization using Docker Part - II

8. Container Orchestration Using
Kubernetes Part-I

9. Container Orchestration Using
Kubernetes Part-II

10. Monitoring Using Prometheus and
Grafana

11. Provisioning Infrastructure using
Terraform Part - I

12. Provisioning Infrastructure using
Terraform Part - II



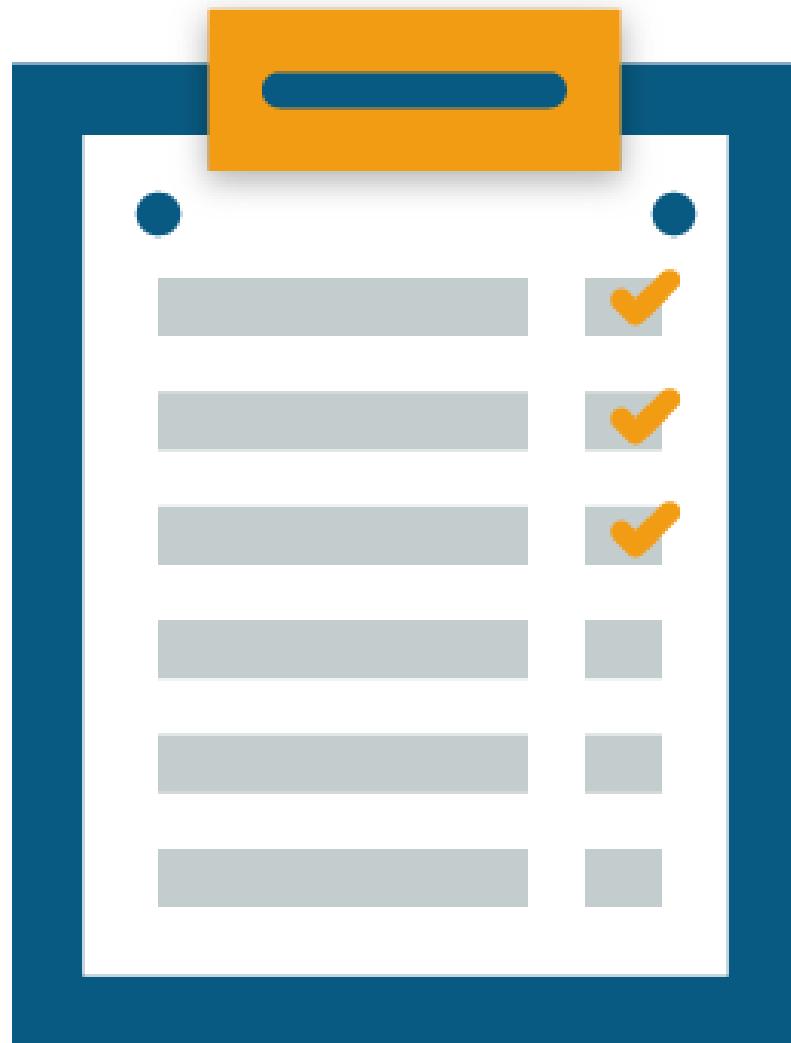
edureka!

Module 1-Introduction to DevOps

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

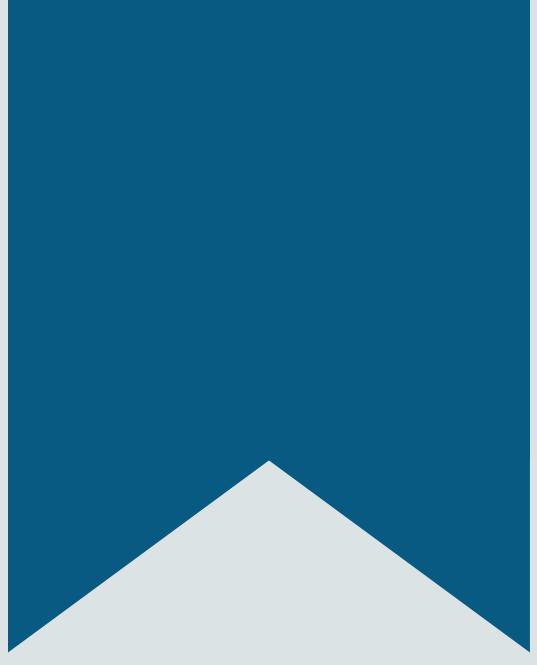


Objectives

After completing this module, you should be able to:

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

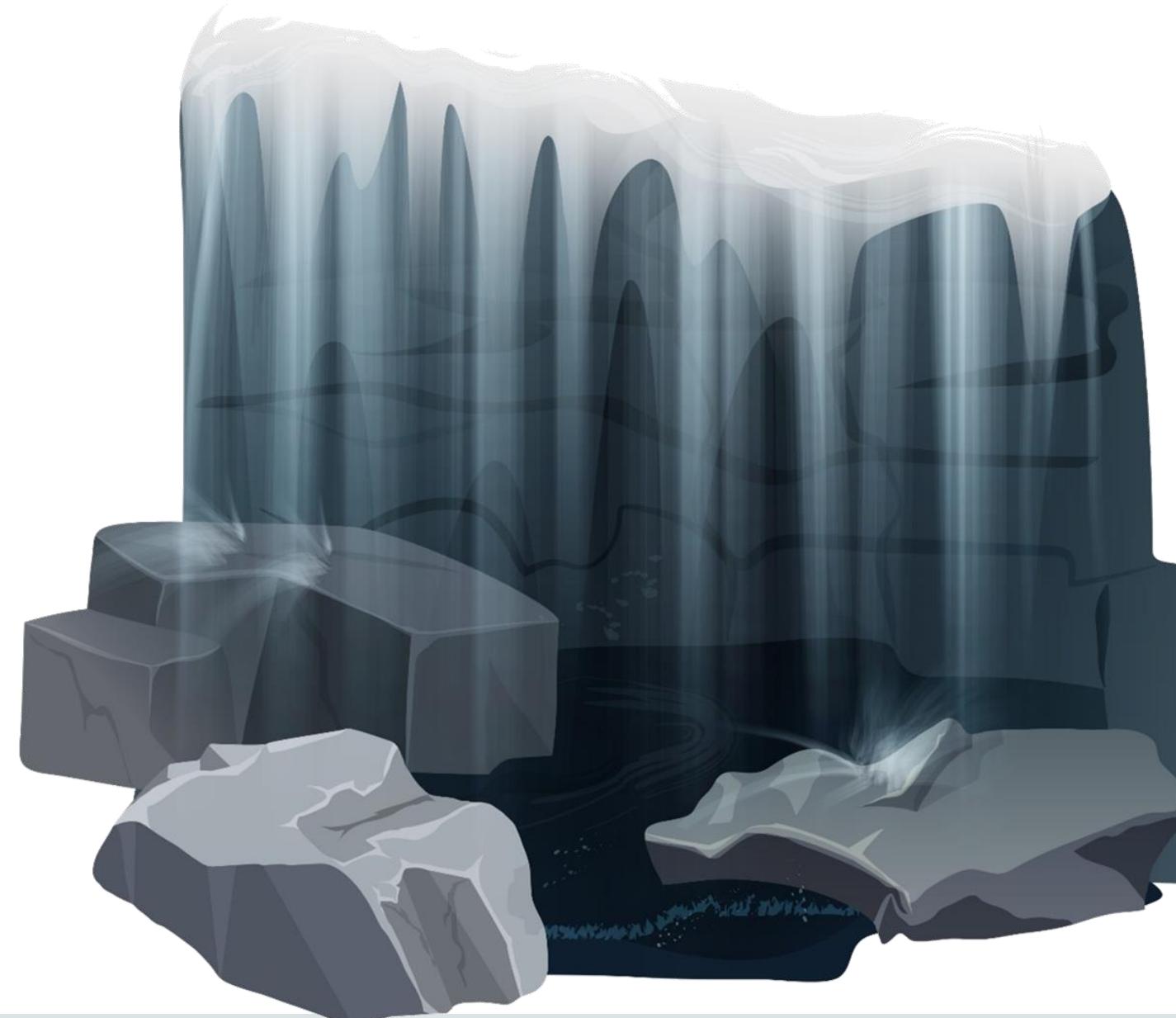




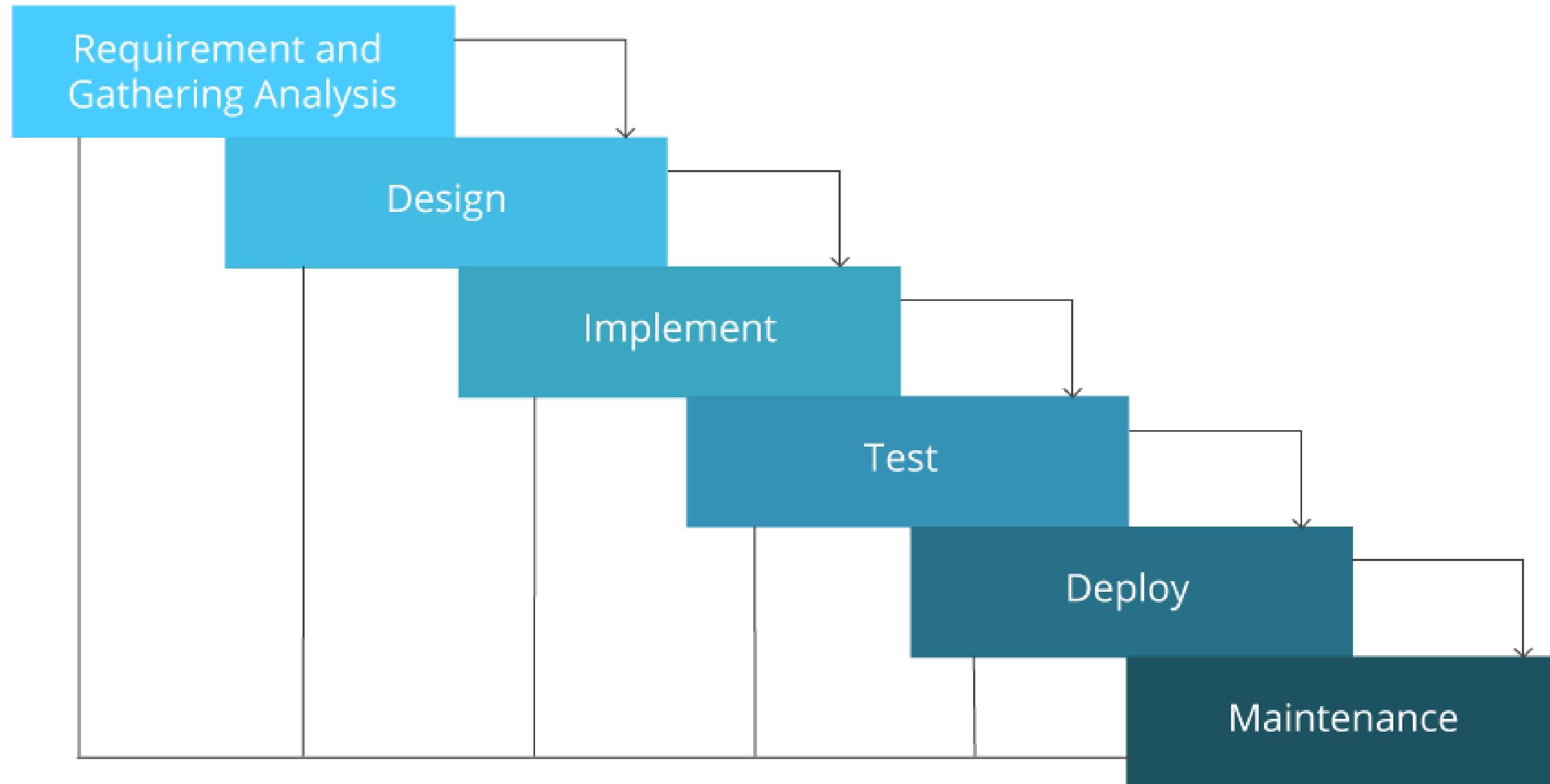
Why DevOps?

Waterfall Model

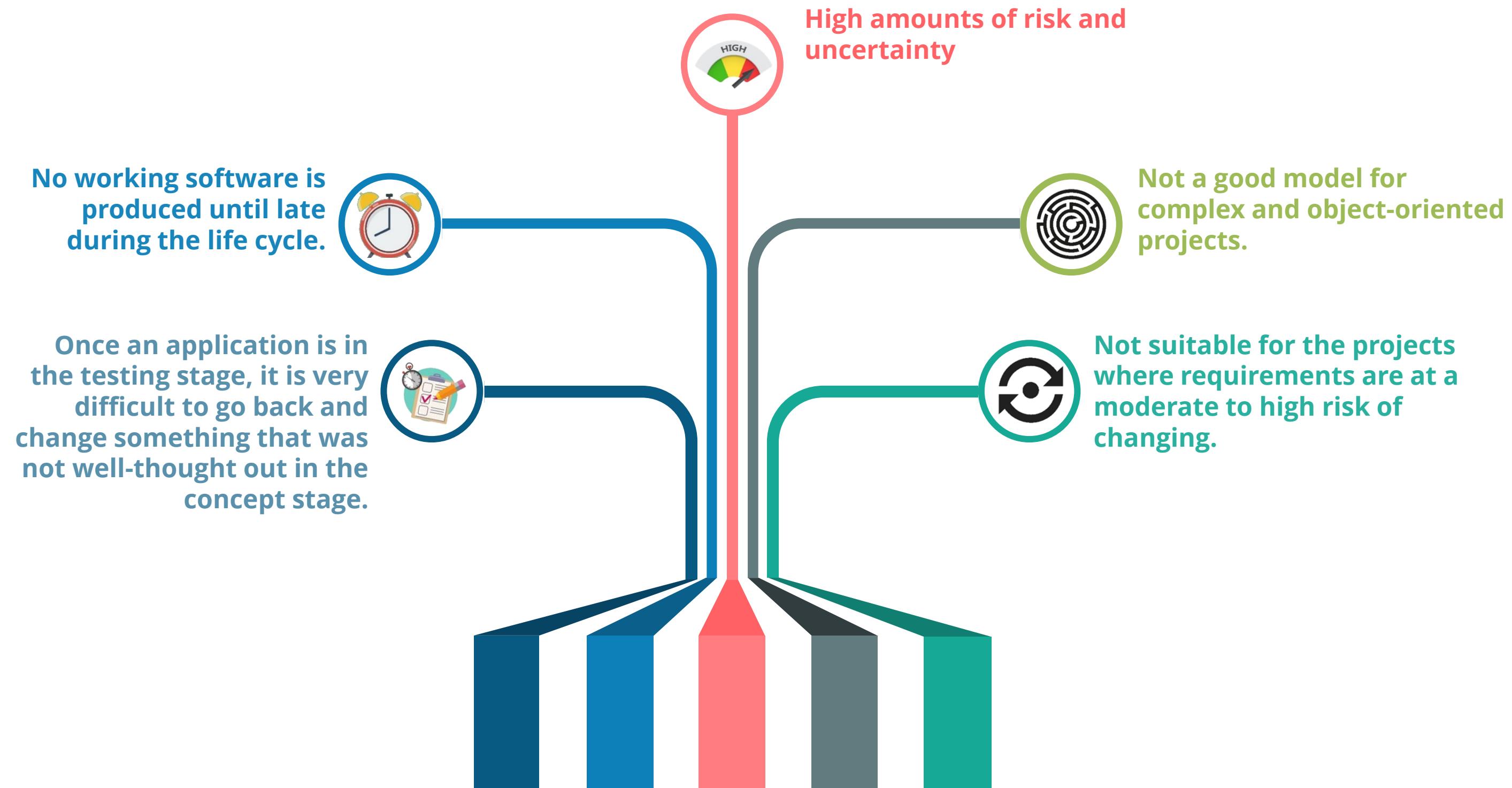
It has distinct goals for each phase of development. Imagine a waterfall on the cliff of a steep mountain. Once the water has flowed over the edge of the cliff, it cannot turn back.



Traditional Waterfall Model



Limitations of Waterfall Model

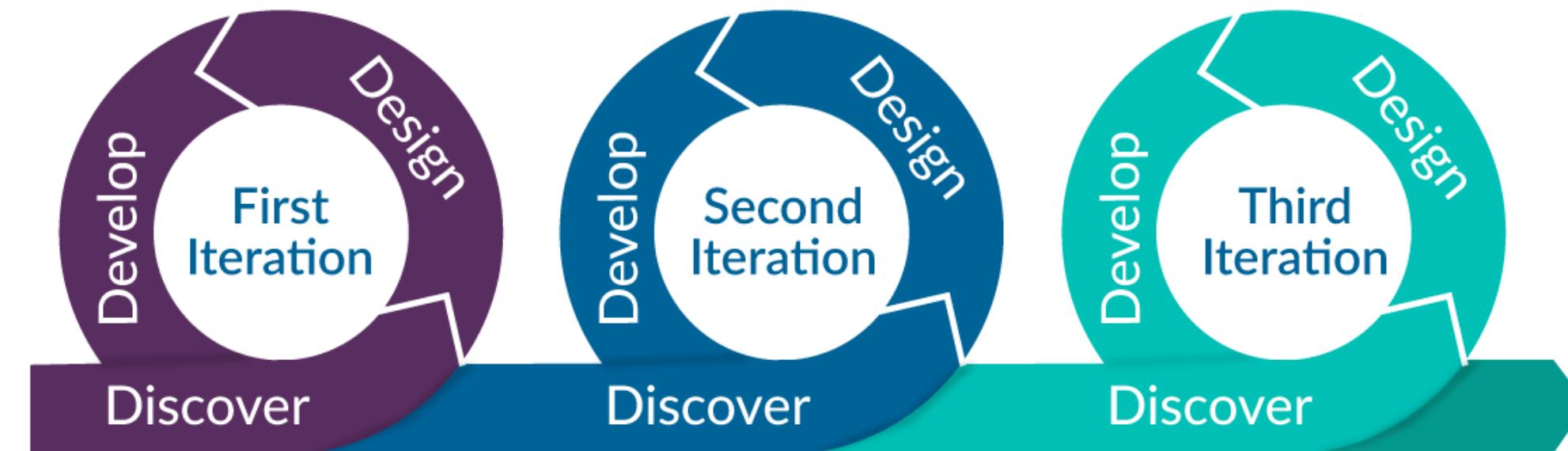


What is Agile Methodology?

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)

At the end of each iteration, a working product should be delivered

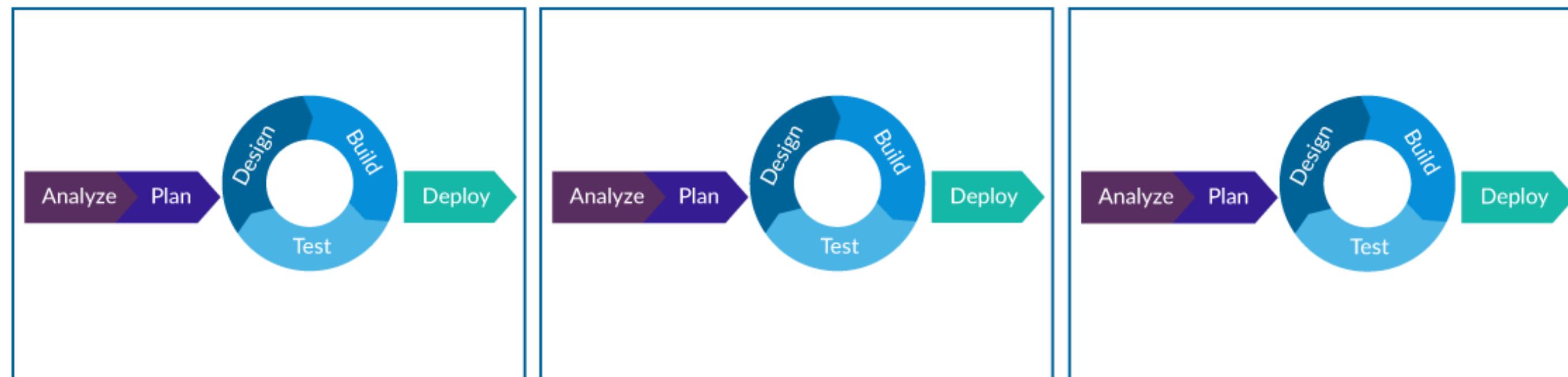


Waterfall vs Agile

Waterfall



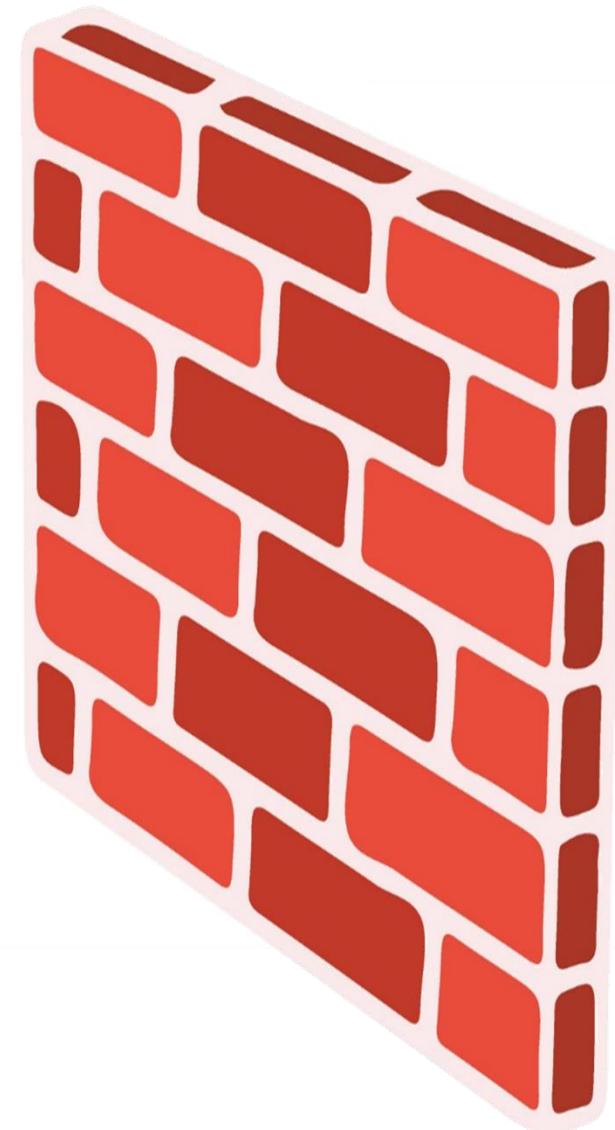
Agile



Limitations of Agile



DEVELOP TEAM



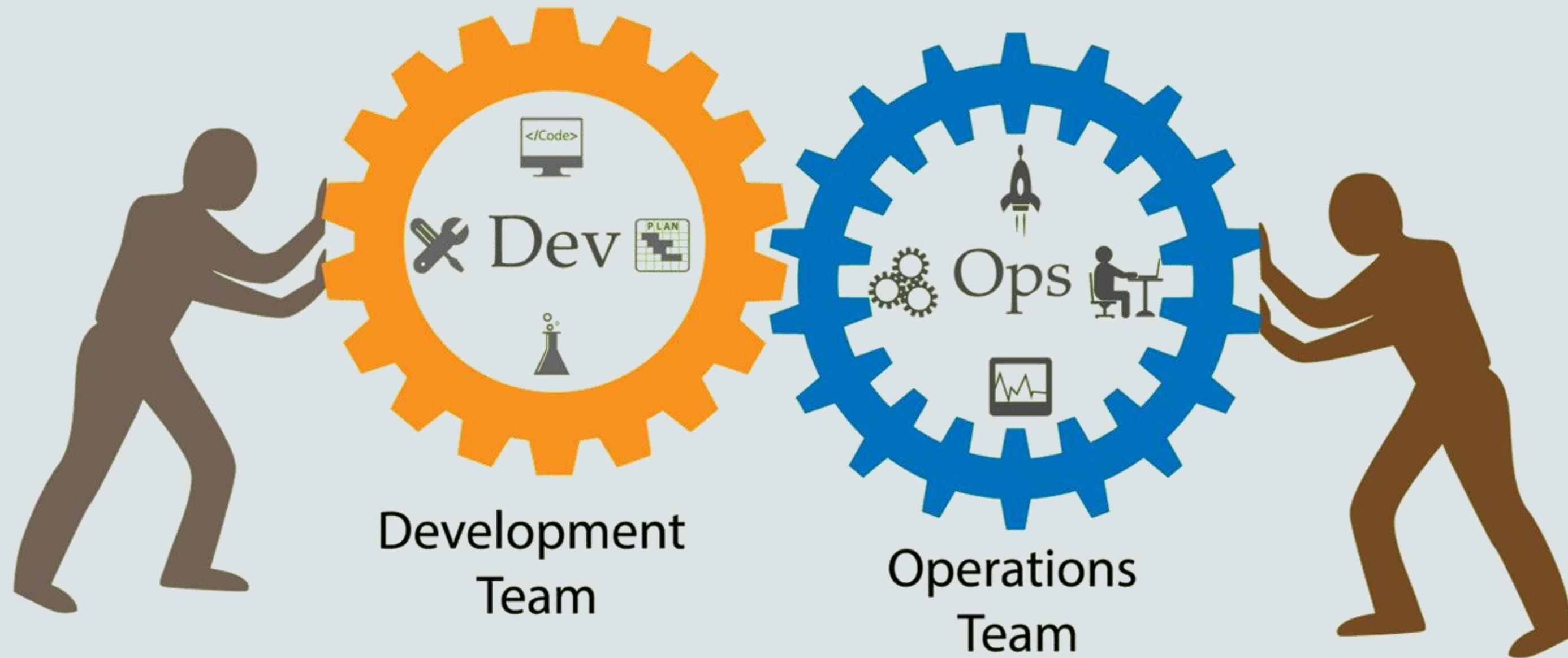
OPERATION TEAM

There is some problem with the code, it does not work in production`

Wants Change

Wants Stability

Solution is DevOps



Development Without DevOps Culture



Release and Deploy
Mismatch



Unpredictable Issues



Blame Game

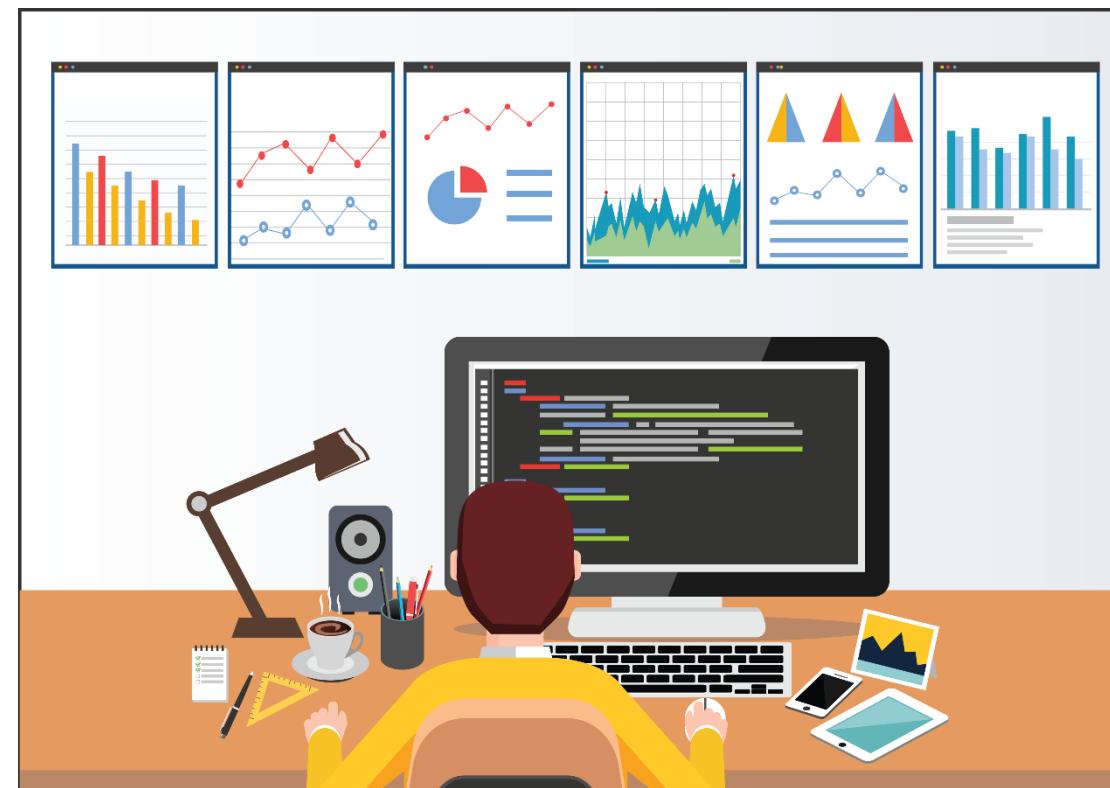


Lack of monitoring and
Feedback

Development With DevOps Culture



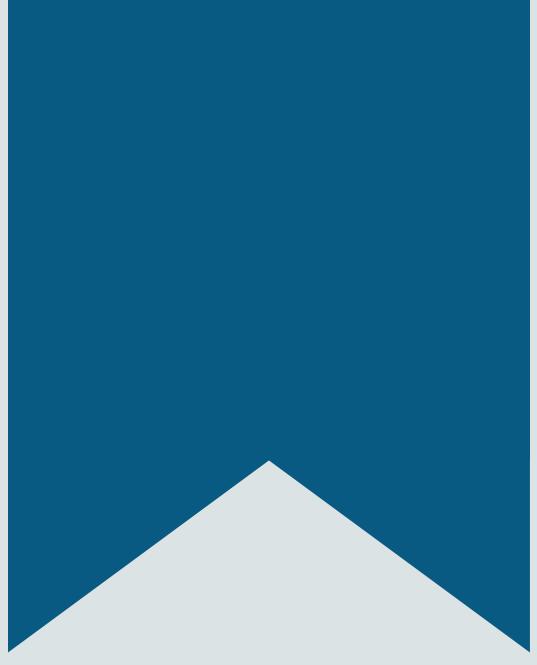
Streamlined Deliveries



Continuous Monitoring and Feedback



Team Work in Collaboration

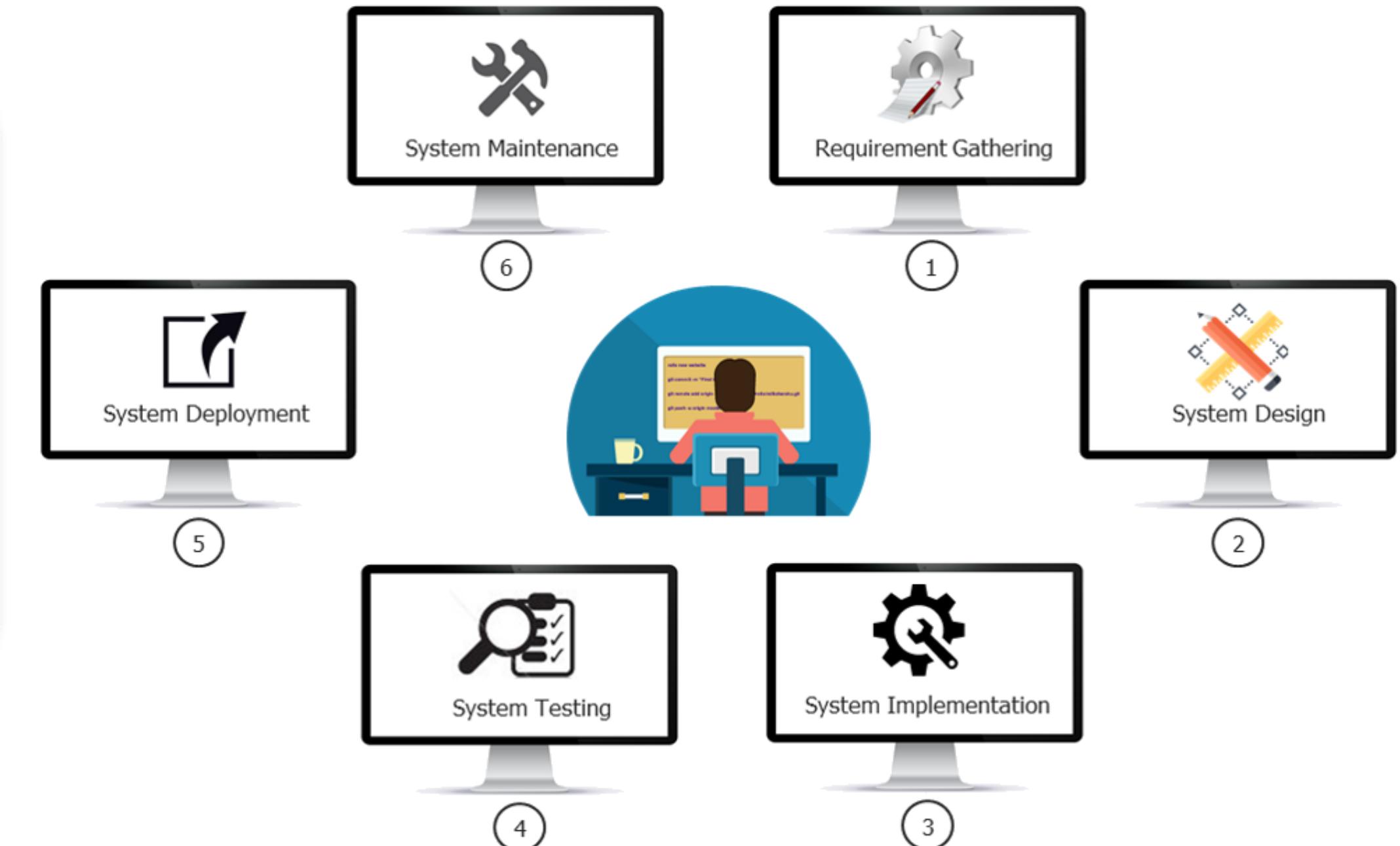


Understanding DevOps: Scenario

Traditional Development Model Challenges

Consider developing software in a traditional way using a [Waterfall model](#)

Waterfall Model Challenge: New phase in the development process begins only if the previous phase is complete



Waterfall Model Challenges

Developers



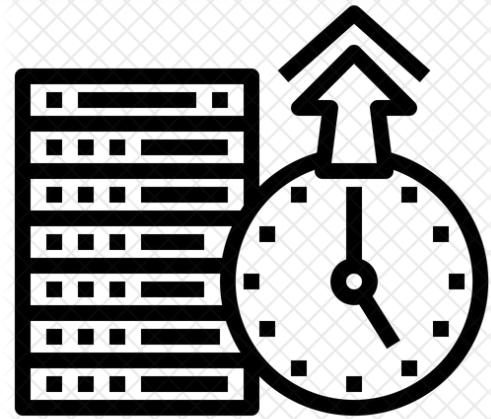
Huge waiting time for code deployment

Pressure of work on old and pending code



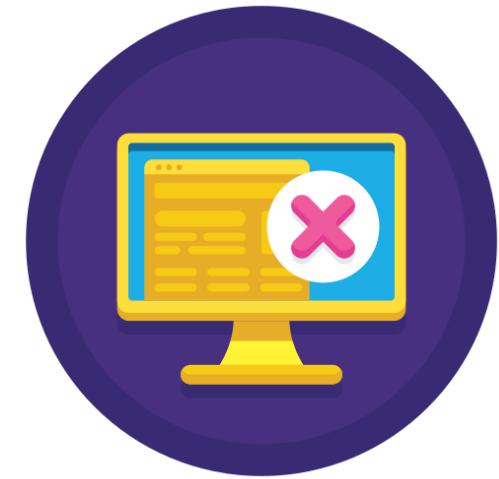
Waterfall Model Challenges

Operations



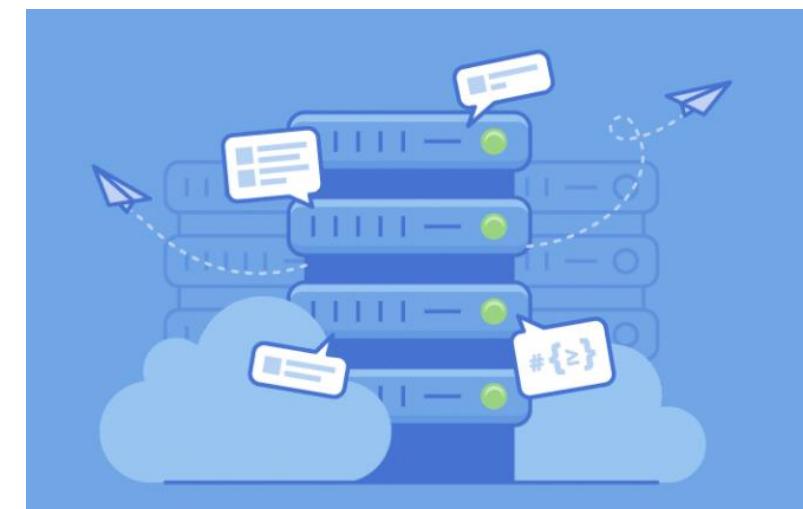
Difficult to maintain uptime
of the production
environment

Tools to automate
infrastructure management are
not effective



Difficult to diagnose
and provide feedback
on the product

No. of servers to be
monitored
increases



What needs to be done?

Developers



Use system with little or no waiting time

Use system with updated code



What needs to be done?

Operations

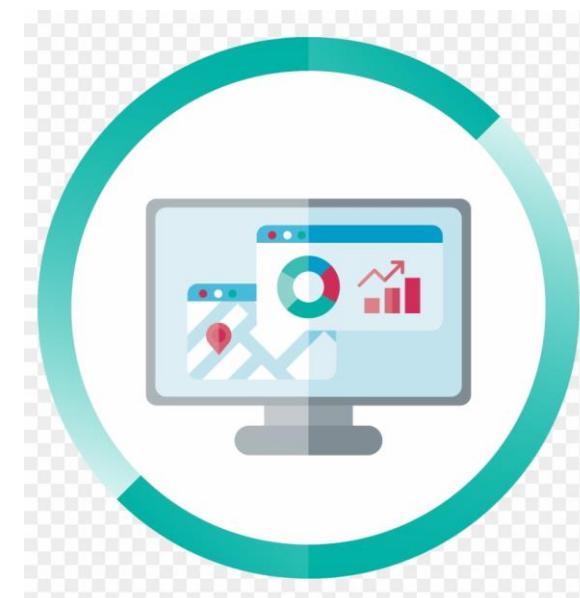


Systems should be up and running most of the time

System required for easy administration



Effective monitoring and feedbacks system should be established



The Panacea - Devops

Dev Challenges



Waiting time for code deployment



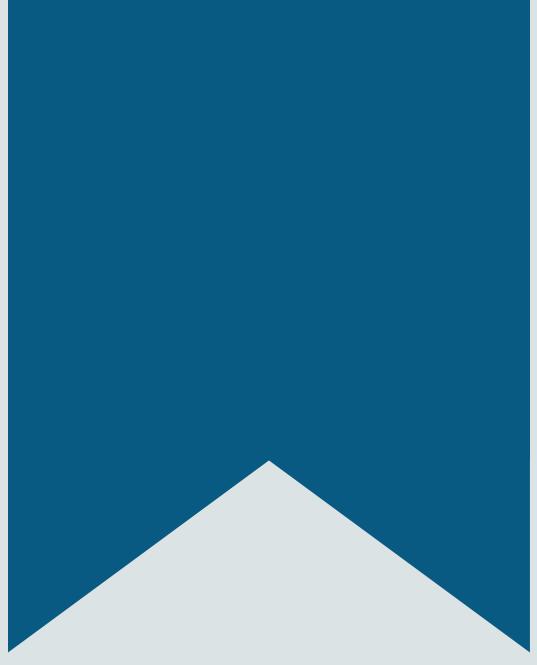
Pressure of work on old, pending and new code

DevOps Solution

- Continuous Integration ensures quick deployment, faster testing and speedy feedback mechanism
- Since there is no waiting time to deploy the code. Hence the developer focuses on building the current code

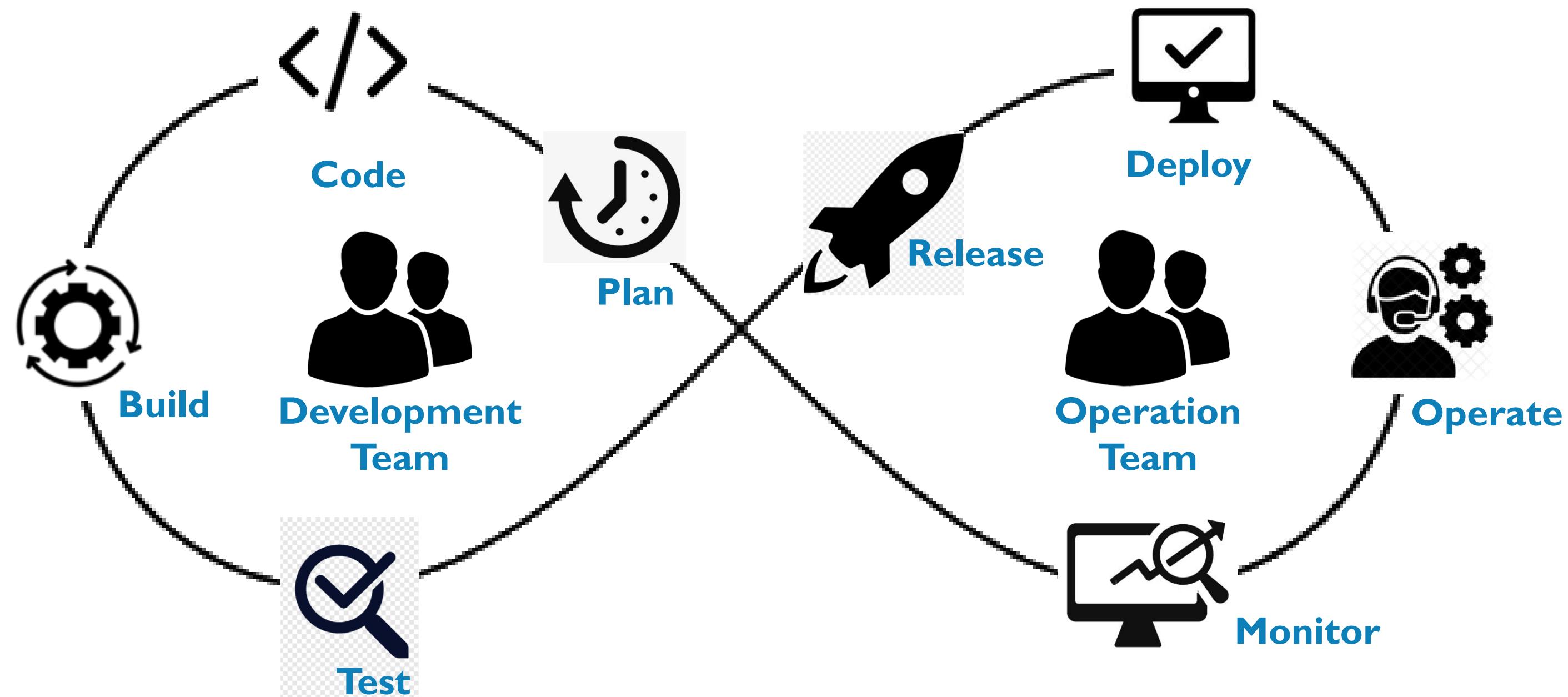
The Panacea - Devops

Ops Challenges	DevOps Solution
 <p>Difficult to maintain uptime of the production environment</p>	<p>Containerization / Virtualization ensures a simulated environment to run the software as containers in turn offering great reliability for service uptime</p>
 <p>Tools to automate infrastructure management are not effective</p>	<p>Configuration Management helps you to organize and execute configuration plans, consistently provision the system, and proactively manage their infrastructure</p>
 <p>No. of servers to be monitored increases</p>	<p>Continuous Monitoring</p>
 <p>Difficult to diagnose and provide feedback on the product</p>	<p>Effective monitoring and feedback system is established through Nagios Thus, effective administration is assured</p>



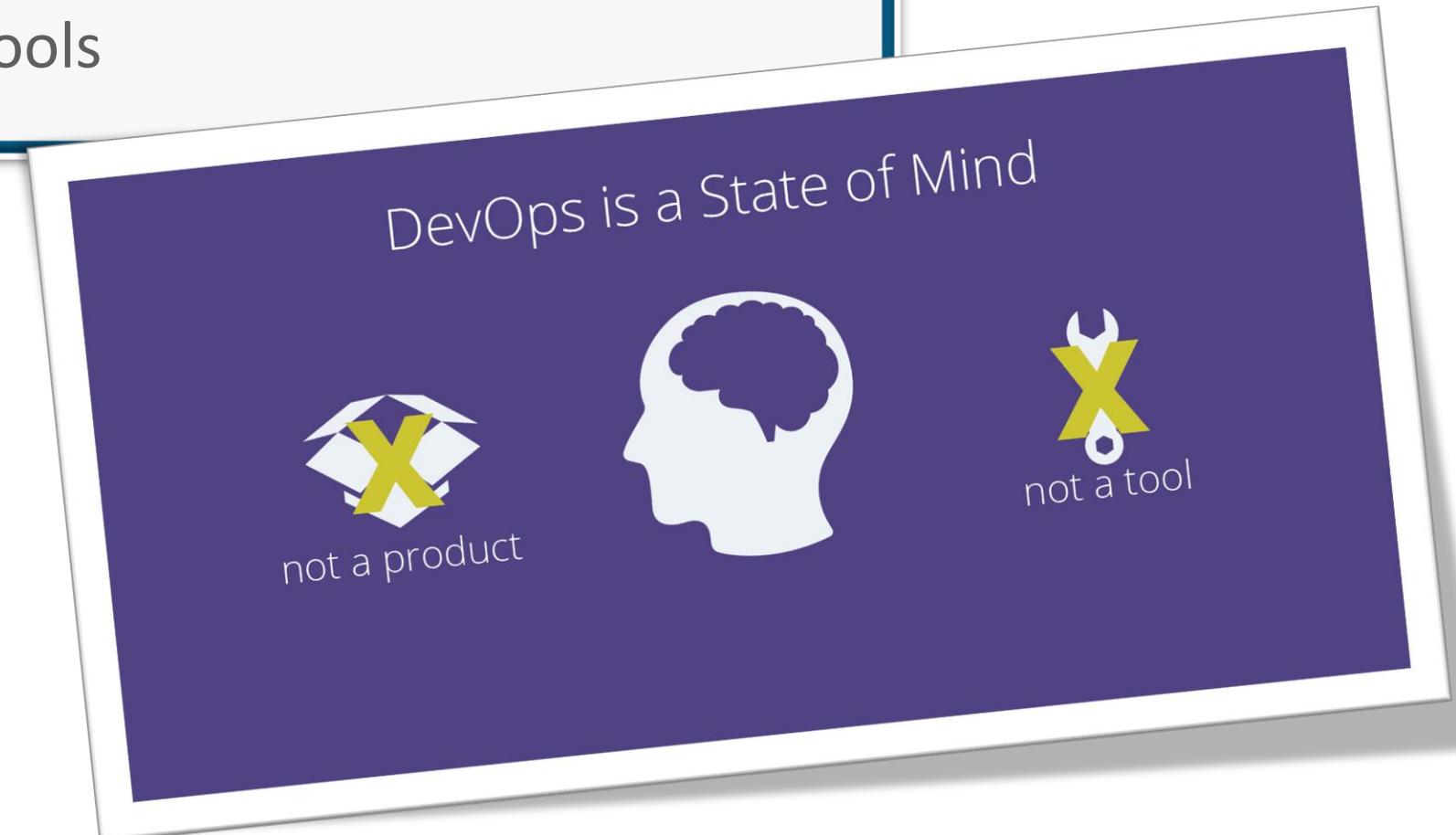
What is DevOps?

What Exactly Is DevOps In Real Life?



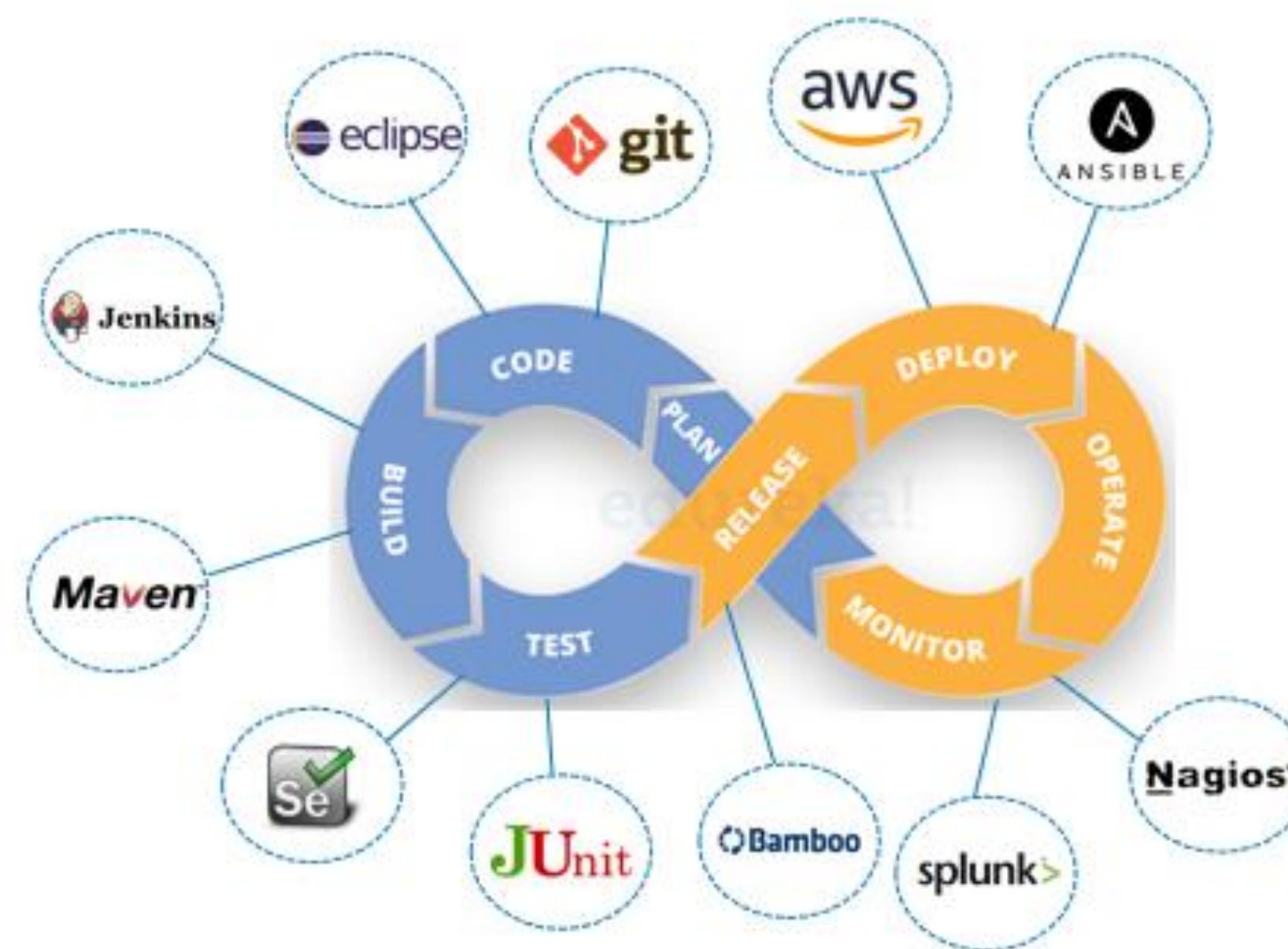
DevOps: What DevOps Is NOT!!

- 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 tools



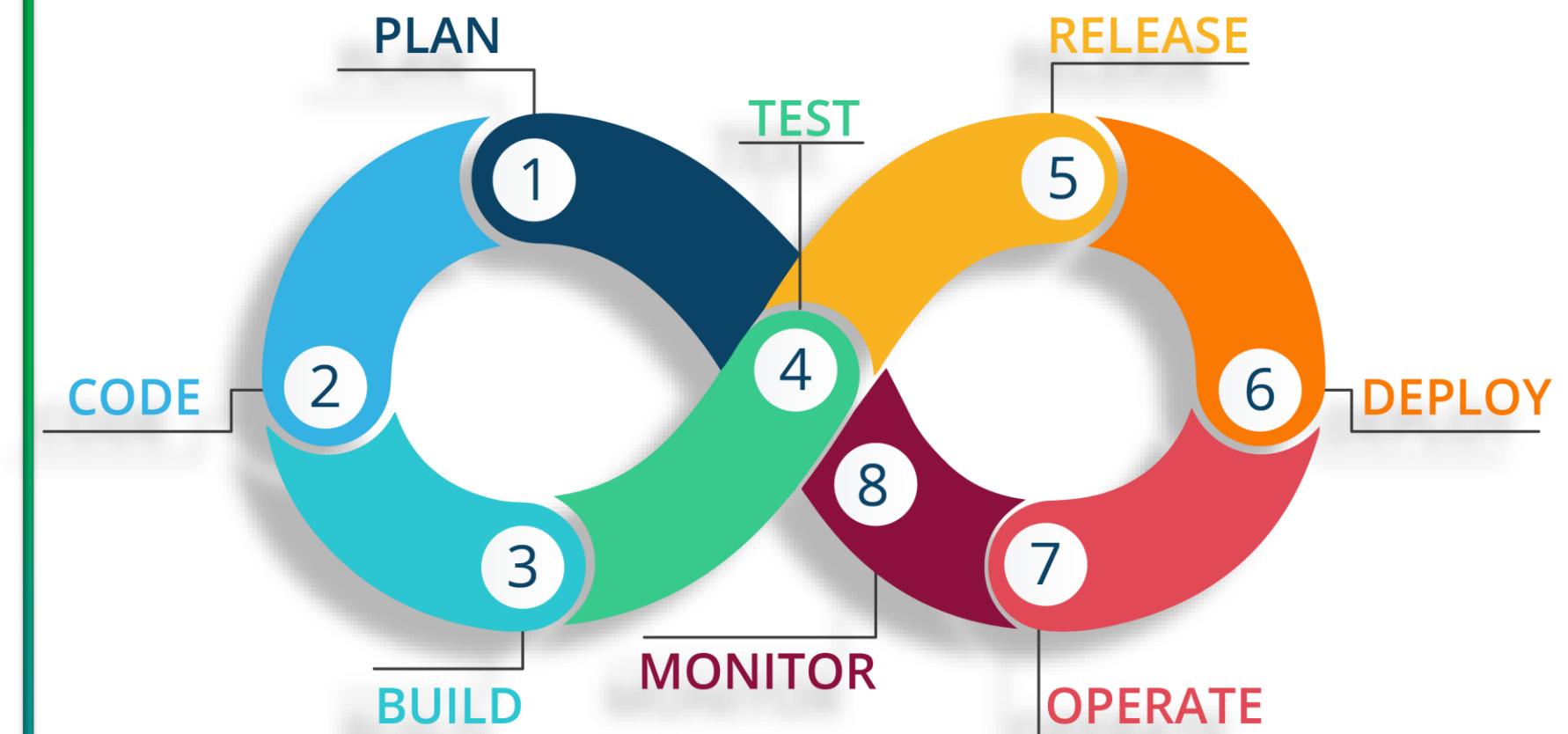
What Is DevOps?

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

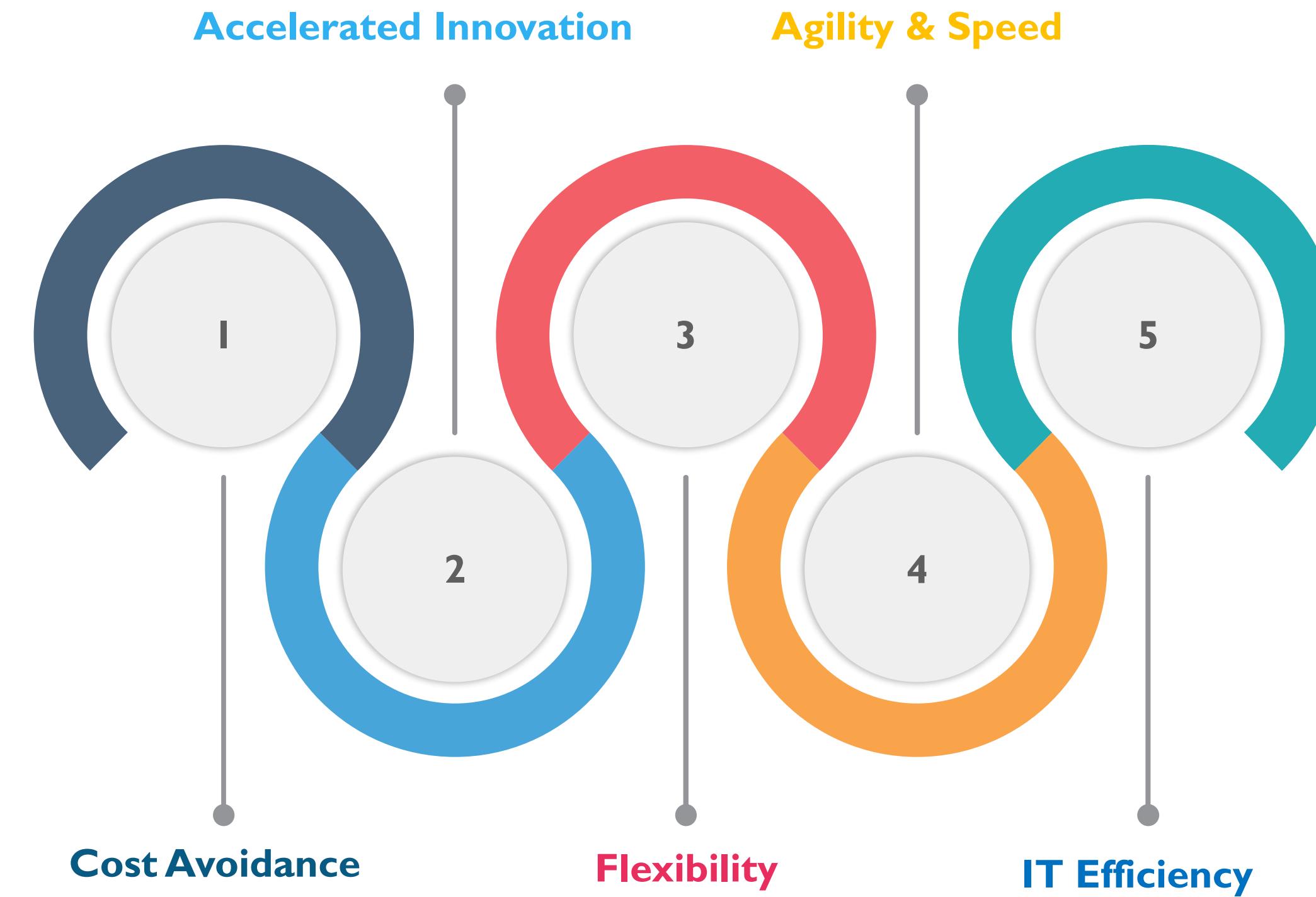


What Does DevOps Do?

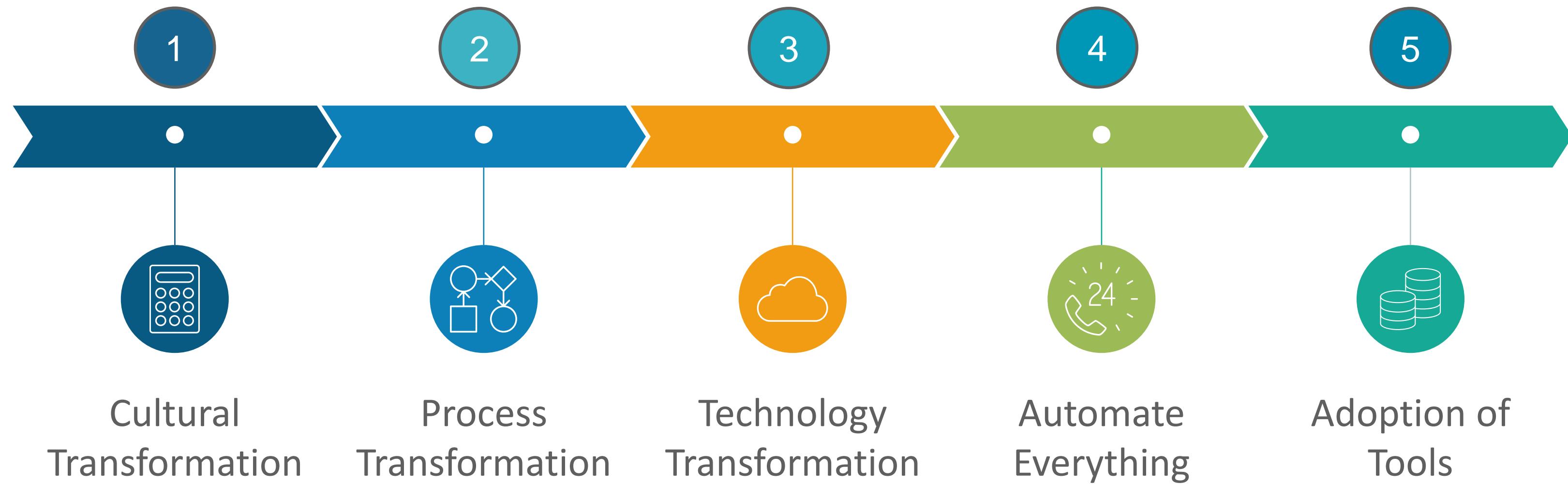
- Integrates developers and operations teams
- Improves collaboration and productivity by:
 - ✓ Automating infrastructure
 - ✓ Automating workflows
 - ✓ Continuously measuring application performance



Benefits of DevOps



DevOps Adoption



Cultural Transformation

- The workplace culture undergoes a major transformation while implementing cultural changes with DevOps.
- It is a long-term process that requires patience and endurance to maintain a positive and transparent atmosphere in the workplace with the changes being implemented.



Process Transformation

Adopting DevOps becomes challenging for companies which follow specific guidelines for software development, as DevOps doesn't have any fixed framework stating procedures that employees can follow to reach their desired goals.



Technology Transformation

- Organisations require updated hardware and software systems to avoid performance issues such as instability, slow load and processing times, and security lapses.
- They should keep up with the latest technology trends on a regular basis, so that new systems can co-exist with the existing systems.



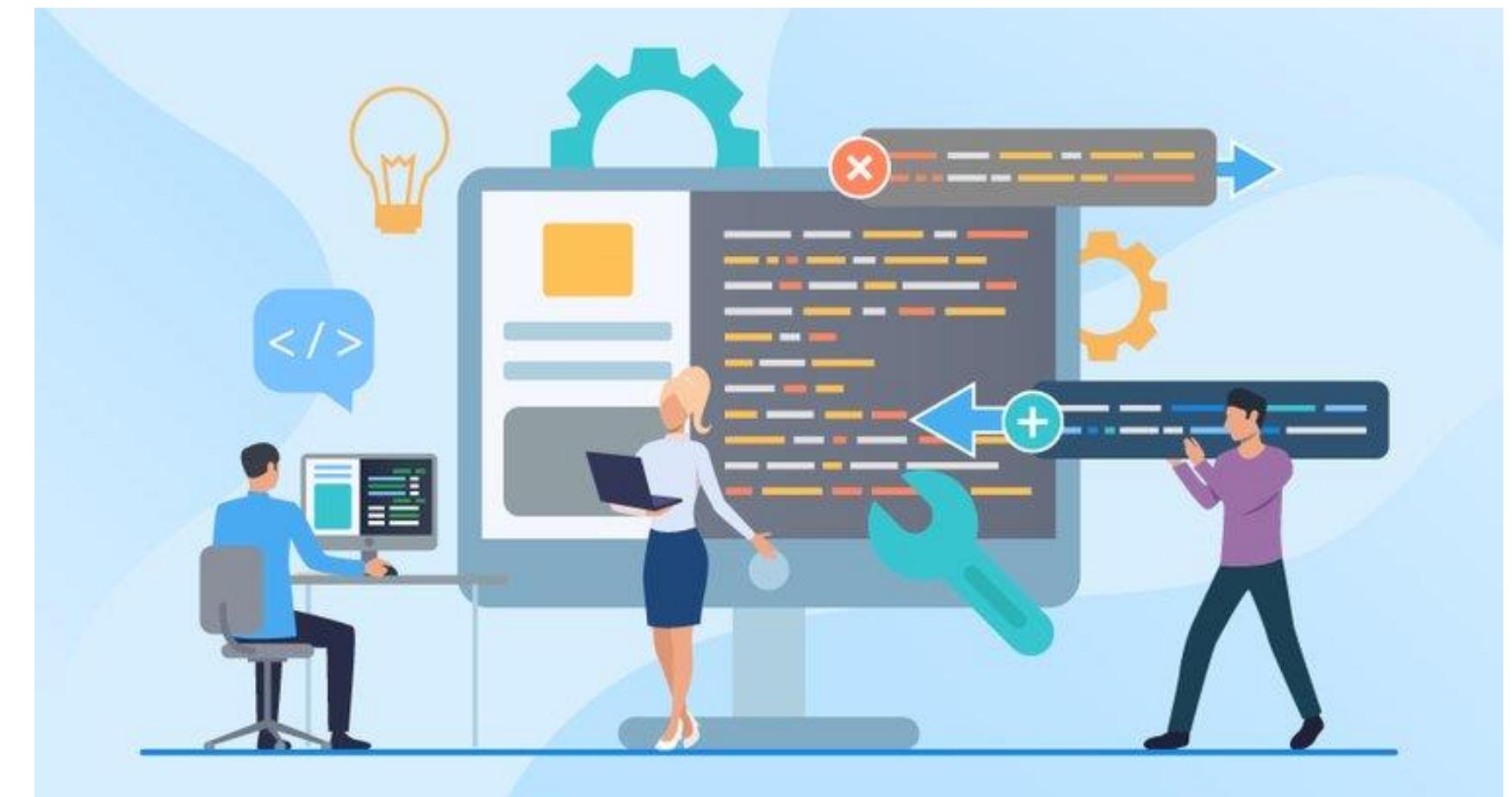
Automate Everything

- Most of the legacy tools and systems used by organizations may not be conducive to automation and collaboration.
- Automation is essential since continuous testing and development are necessary for smooth deployment.



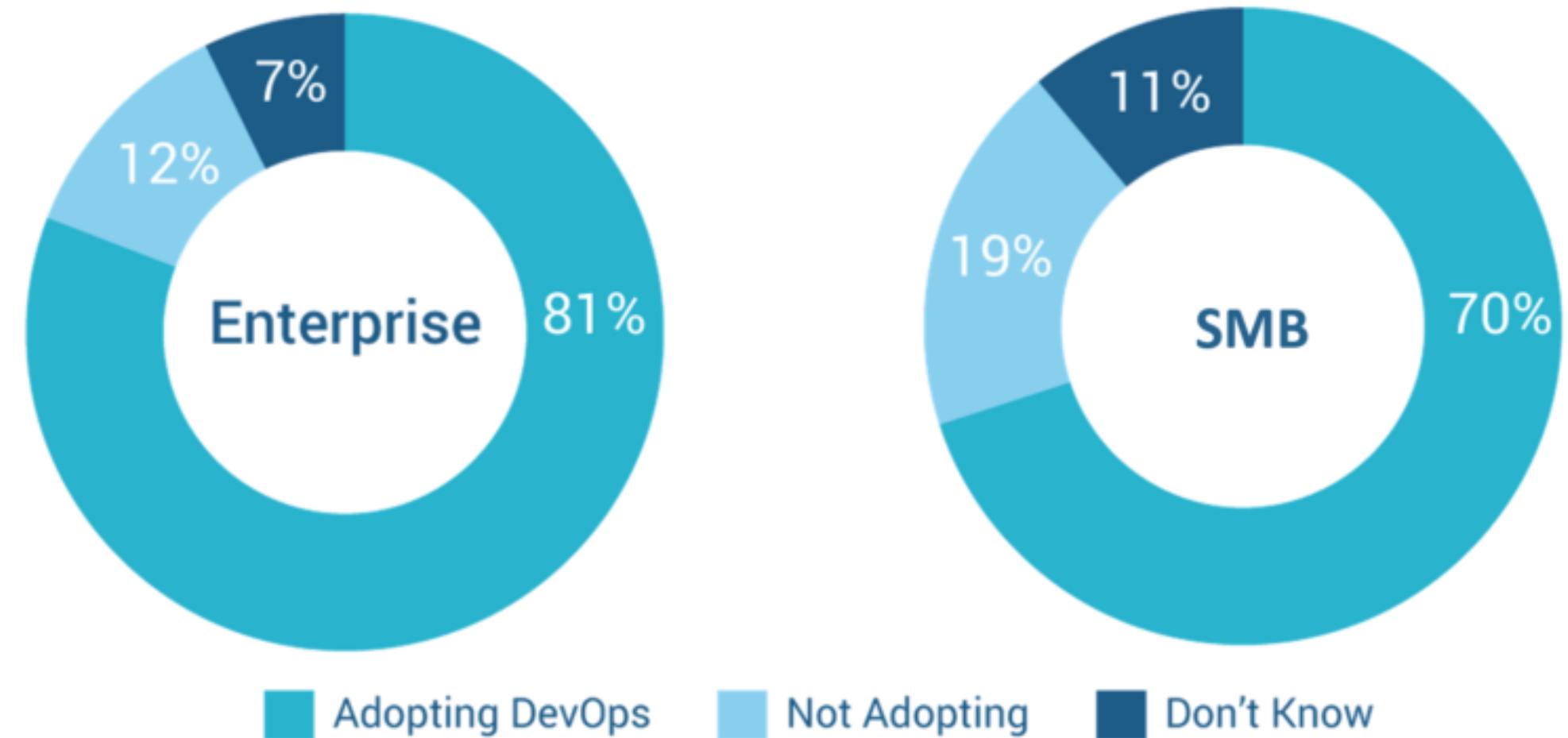
Adoption Of Tools

- The Dev and Ops teams have separate toolsets and metrics.
- It is necessary to integrate all the tools properly to make testing, deployment, and building all work together in a continuous manner.



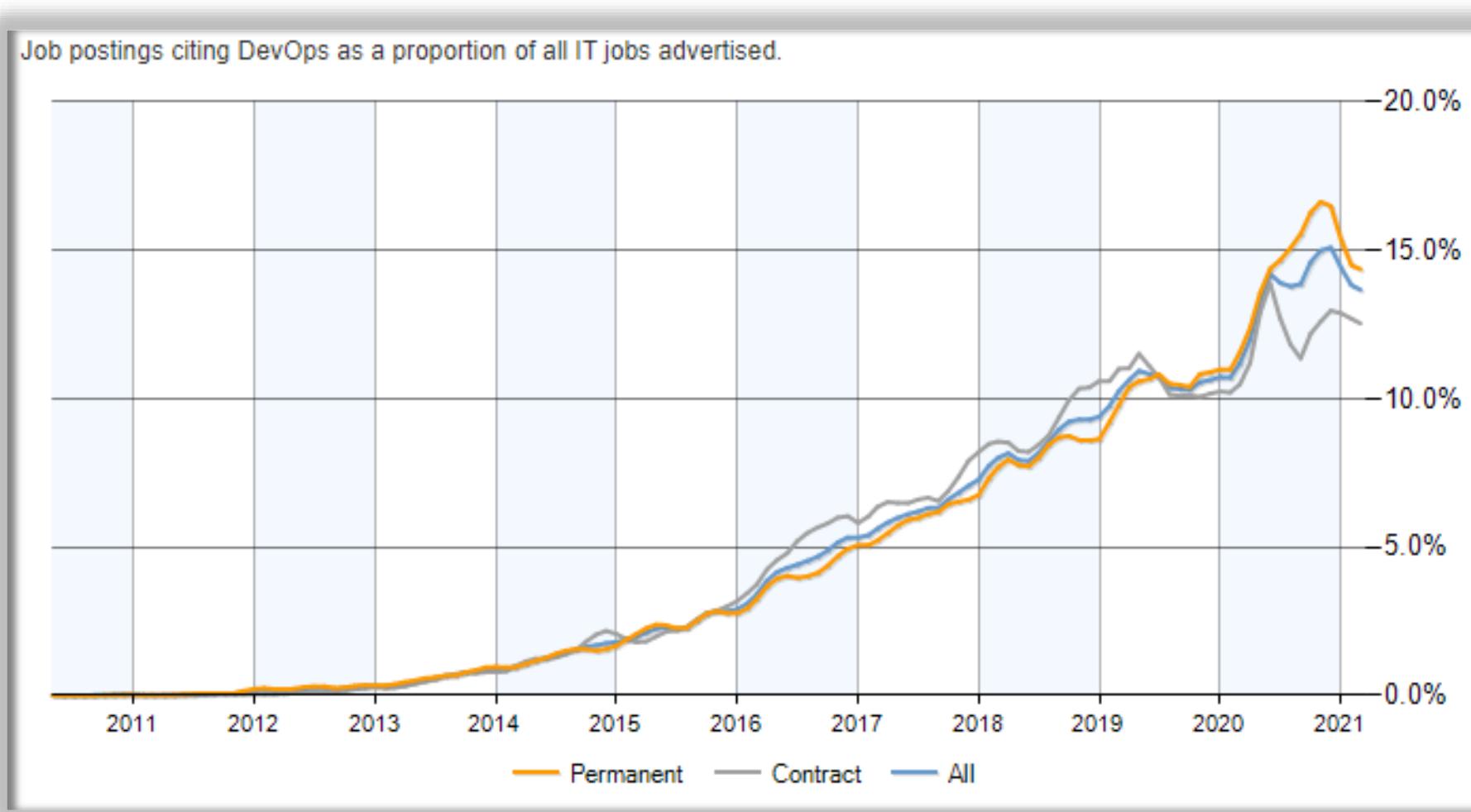
DevOps Current Scenario

“Considering the changing pace of IT landscape,
almost all the companies require fast paced
development environment”



DevOps Current Scenario: Job & Salary Trend

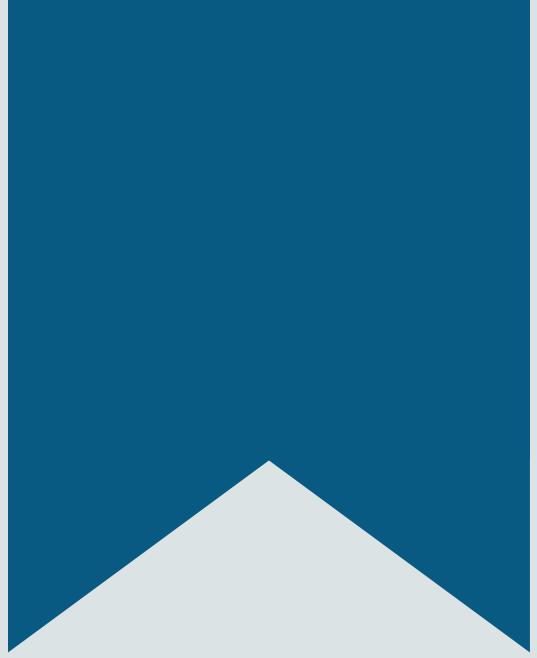
“The demand trend of jobs that feature **DevOps Engineer** in the job title”



According to **PayScale.com**, salaries for professionals working on DevOps related jobs in California more than \$127,000

Skills Of A DevOps Engineer

Skills	Description
Tools	<ul style="list-style-type: none">• Version Control - GIT• Continuous Integration - Jenkins• Virtualization/ Containerization – Docker/Kubernetes• Configuration Management – Puppet/Ansible• Monitoring – Nagios
Networking Skills	<ul style="list-style-type: none">• General networking skills – Establishing connection between the containers/Port Forwarding/ Container Orchestration
Other Skills	<ul style="list-style-type: none">• People Skills• Process Skill• Customer Skill and Empathy• Cloud Awareness

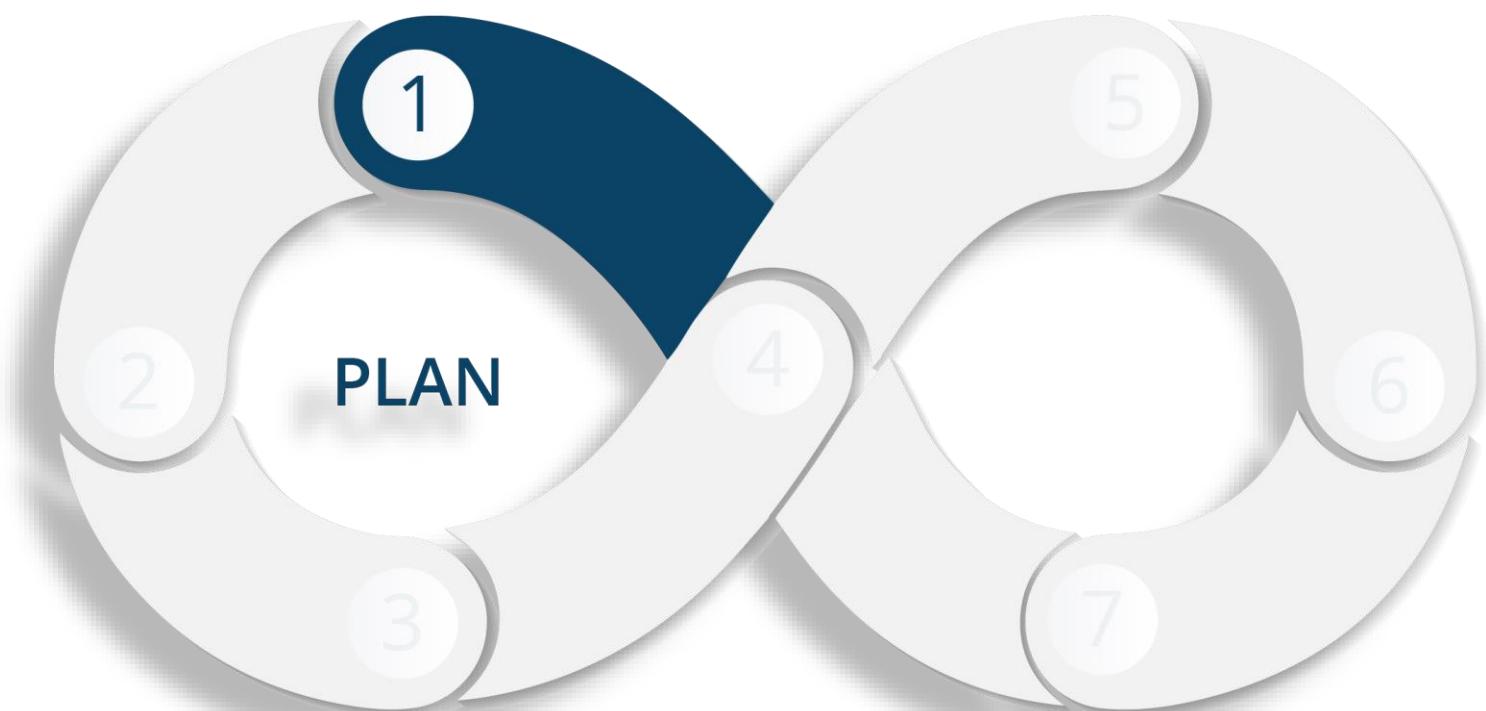


DevOps Lifecycle

DevOps Lifecycle: Plan

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

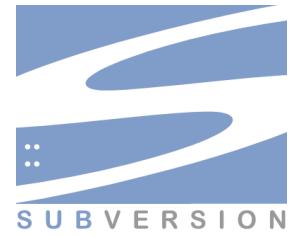
Planning Tools



DevOps Lifecycle: Code

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

Coding Tools



DevOps Lifecycle: Build

“Build is a pre-release version and is identified by a build number, rather than by a release number”



Building Tools



Maven™



Jenkins



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”



Testing Tools



JUnit



DevOps Lifecycle: Release

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

Releasing Tools



Travis CI



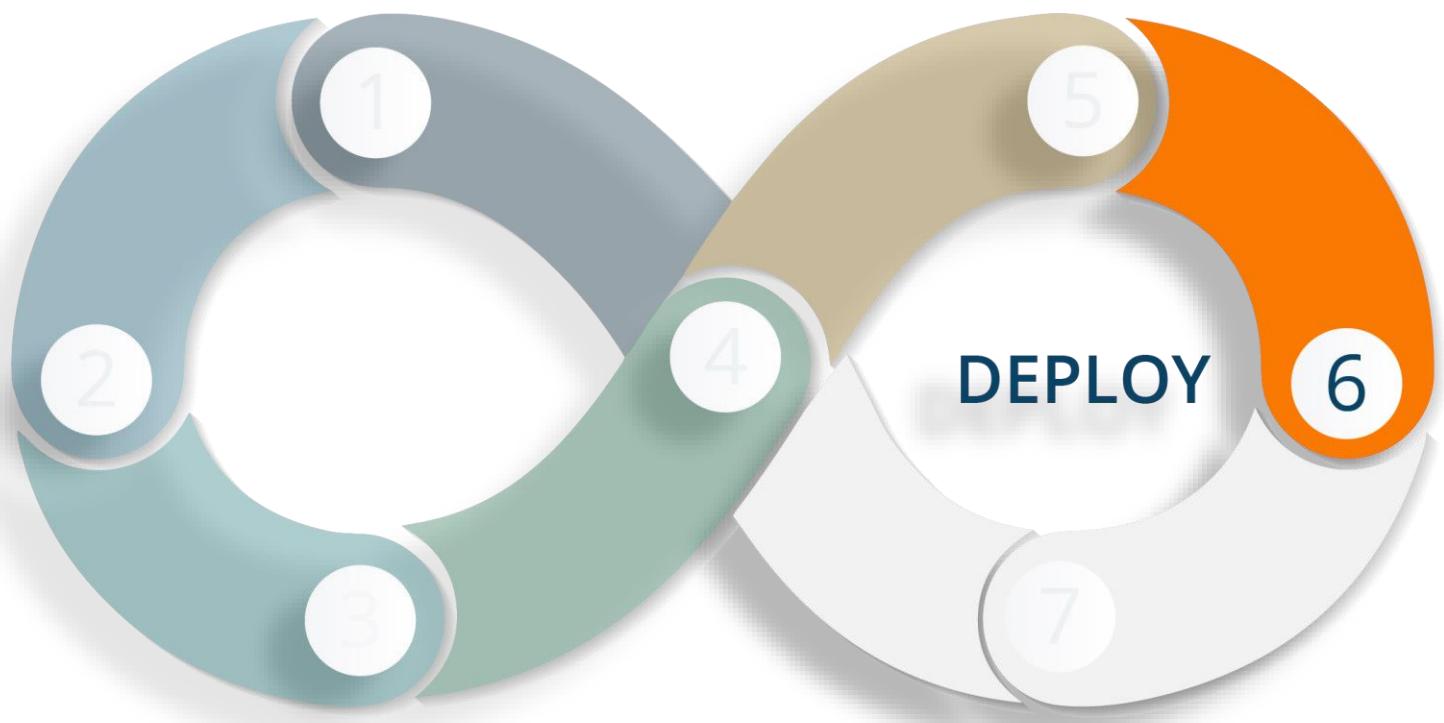
GitLab



DevOps Lifecycle: Deploy

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

Deploying Tools



DeployDevOps Lifecycle: Operate

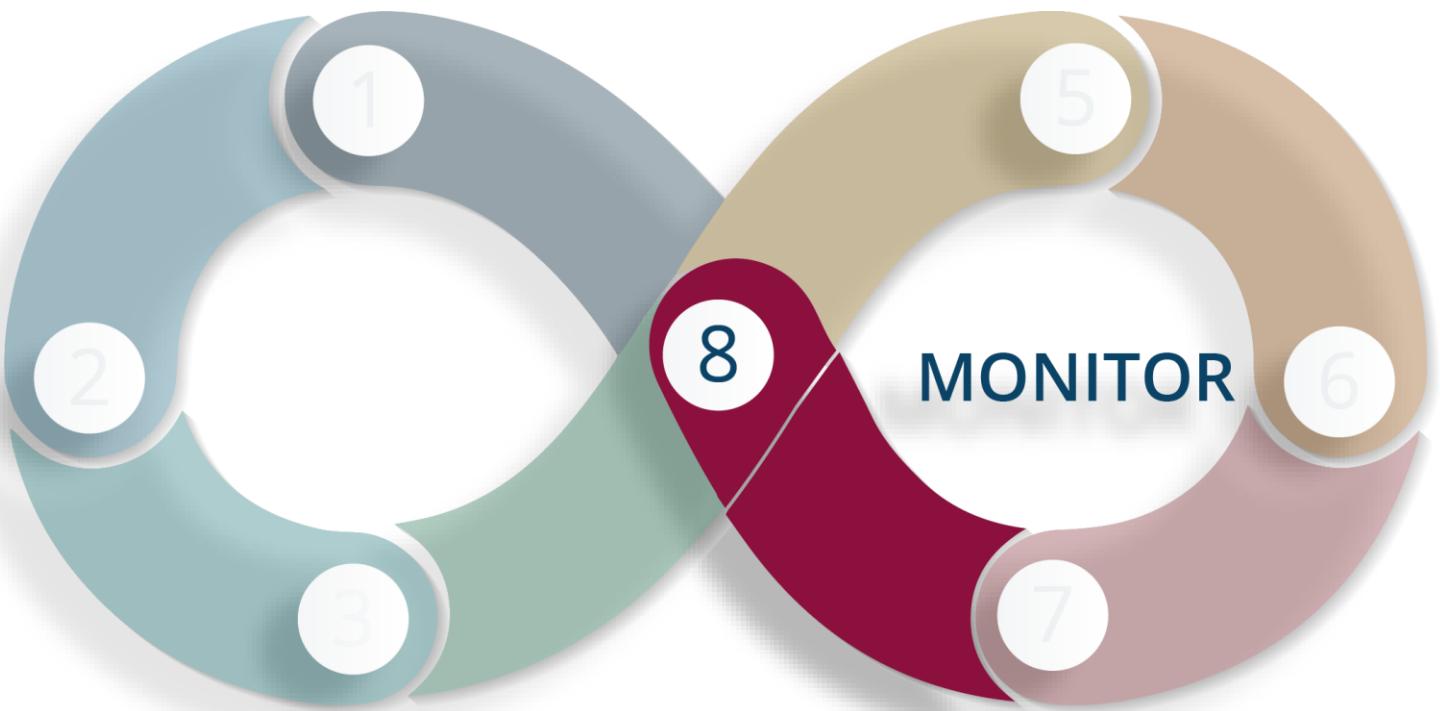
“This phase is to keep the system upgraded with the latest update”

Operating Tools



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”

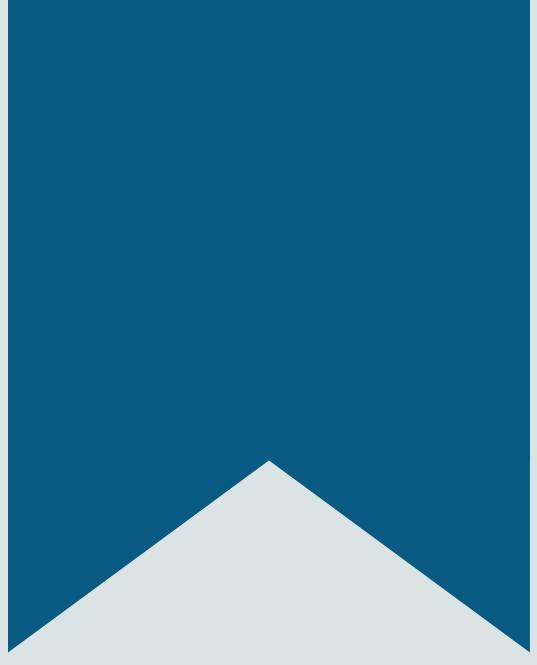


Monitoring Tools

Nagios[®]

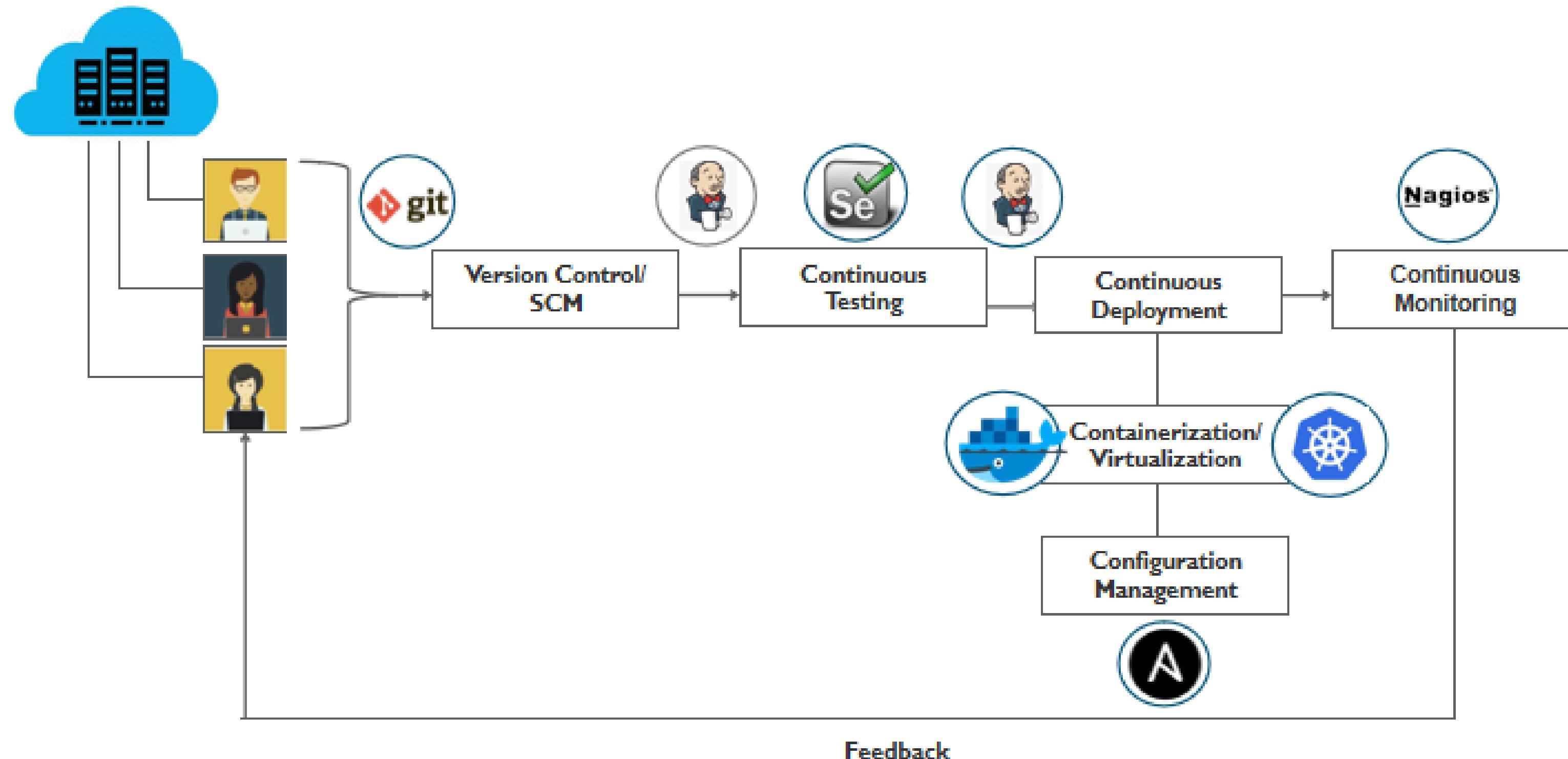


splunk[®]



DevOps Stages

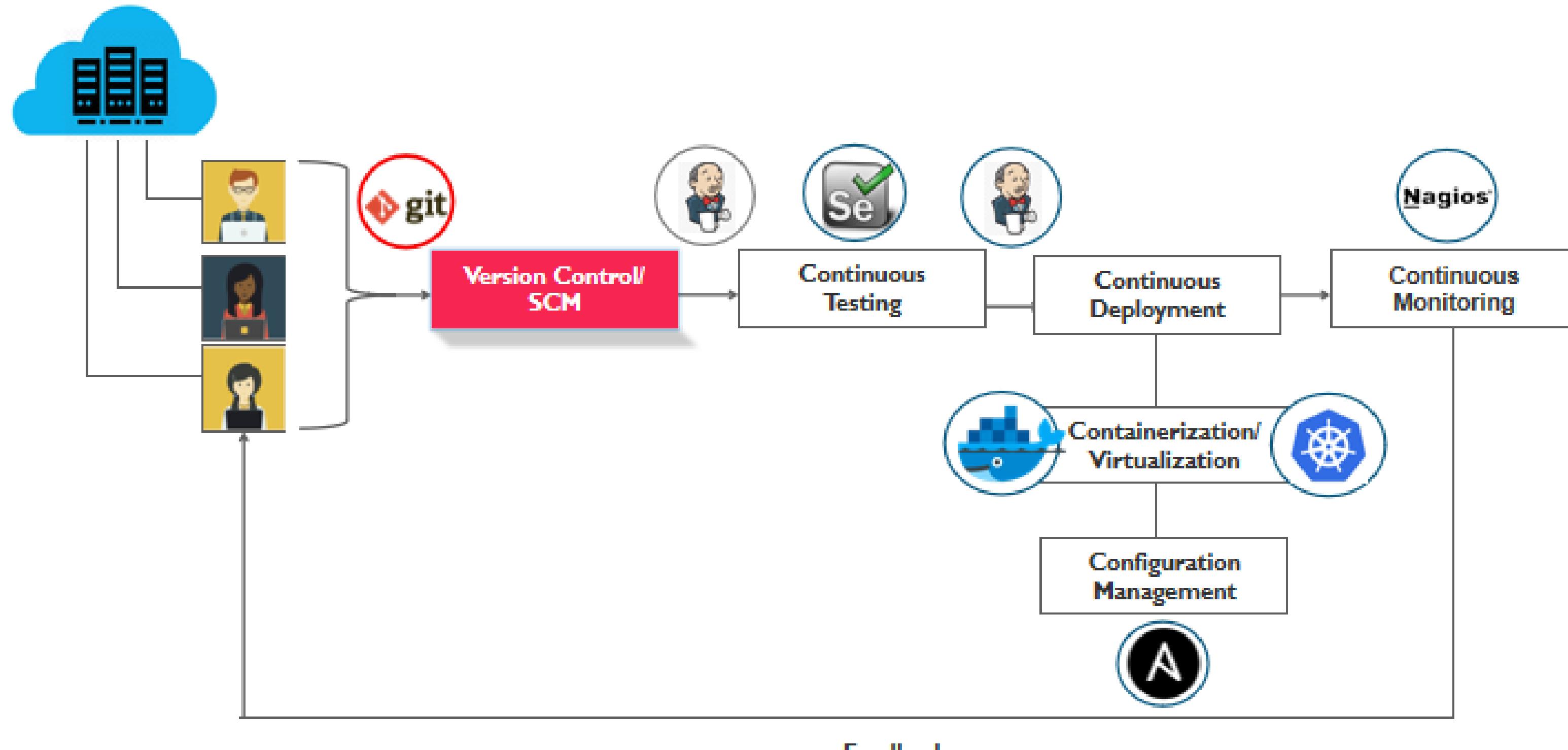
DevOps Stages





Version Control / Source Code Management

Version Control / Source Code Management

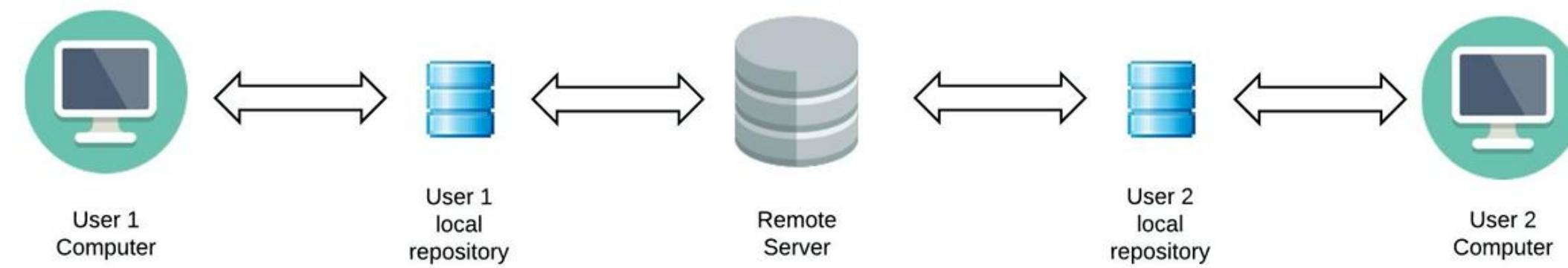


Version Control / Source Code Management

What Is Version Control?

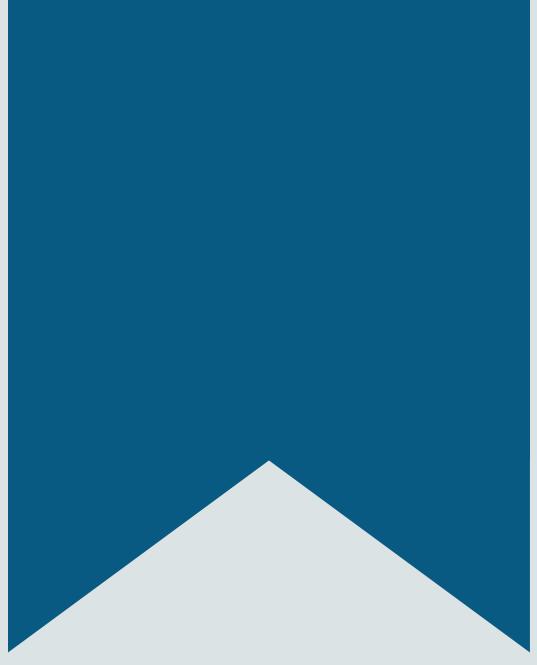
System that records changes to documents, computer programs, large websites and other collection of information over time

Decentralised Version Control



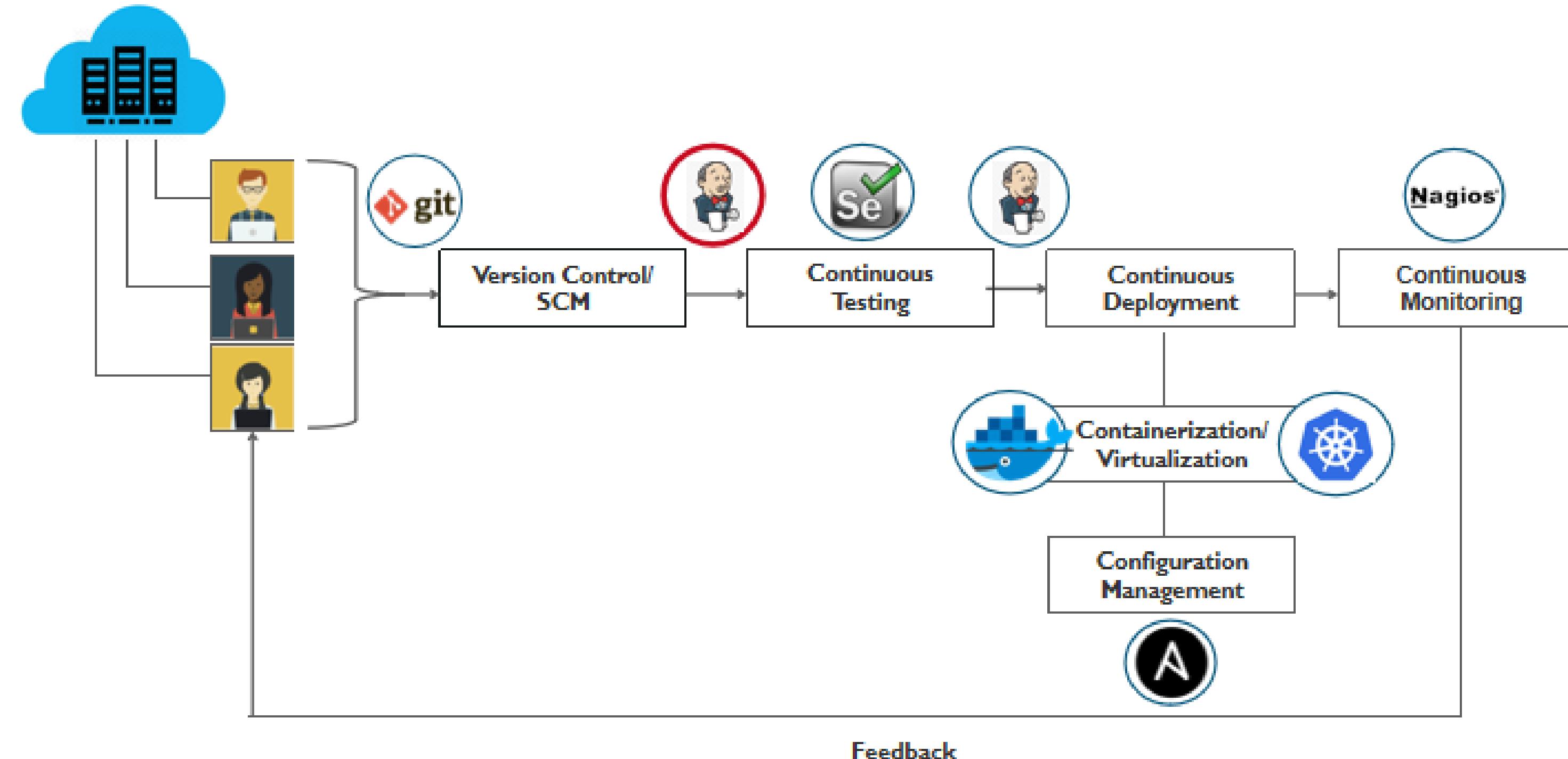
Centralised Version Control





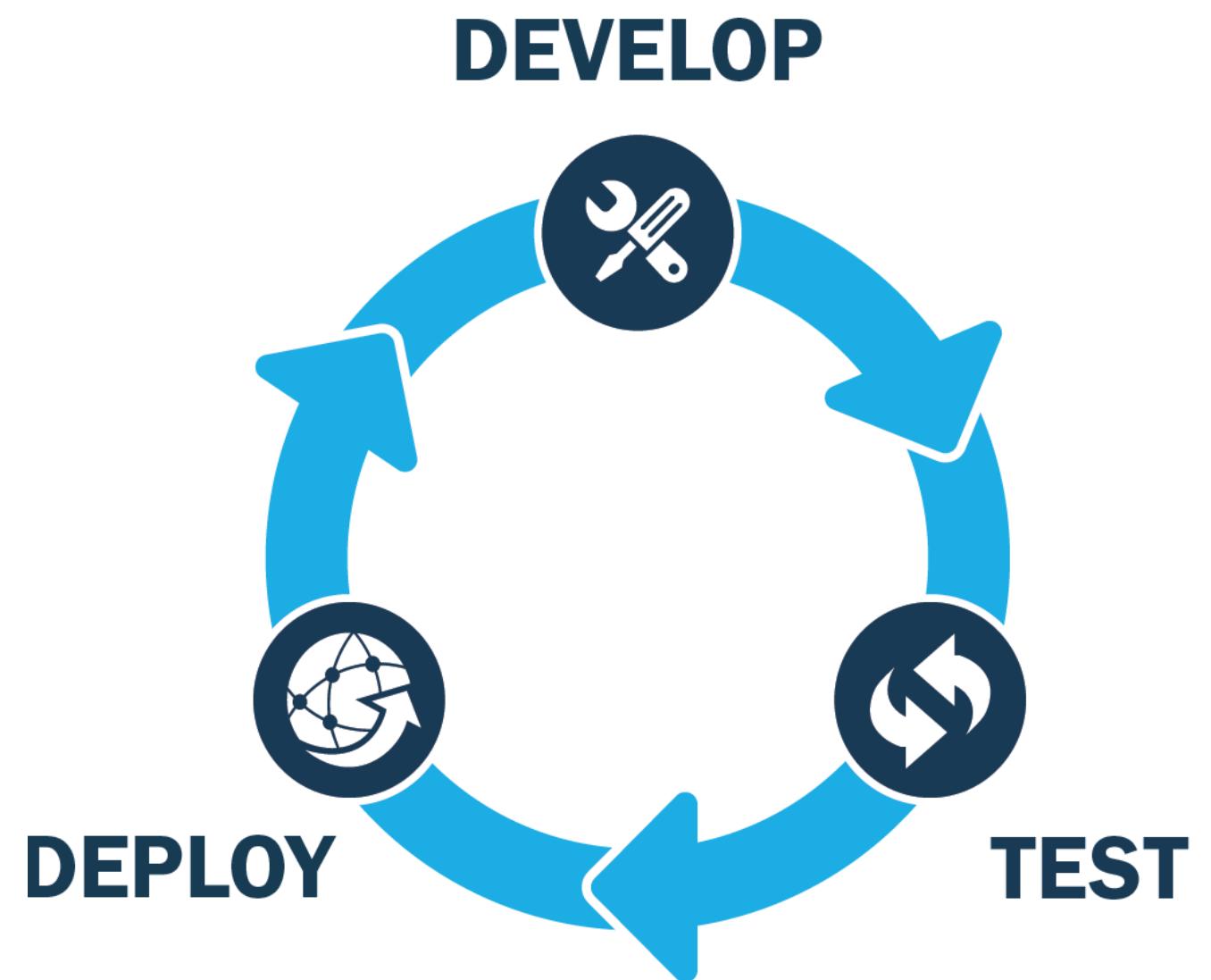
Continuous Integration

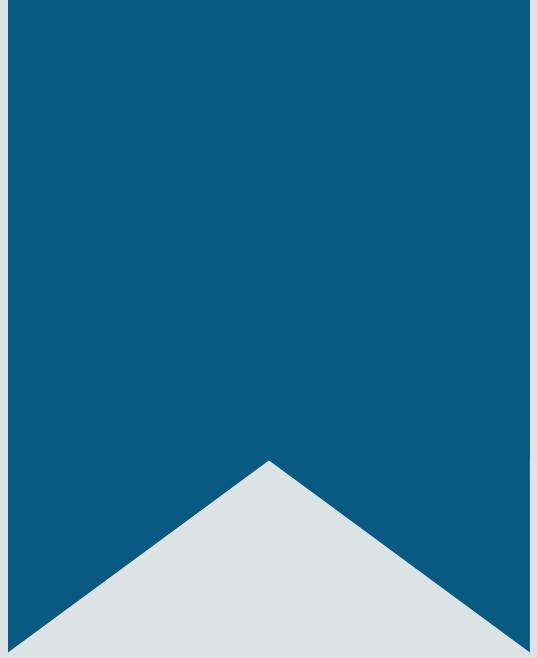
Continuous Integration



What Is Continuous Integration?

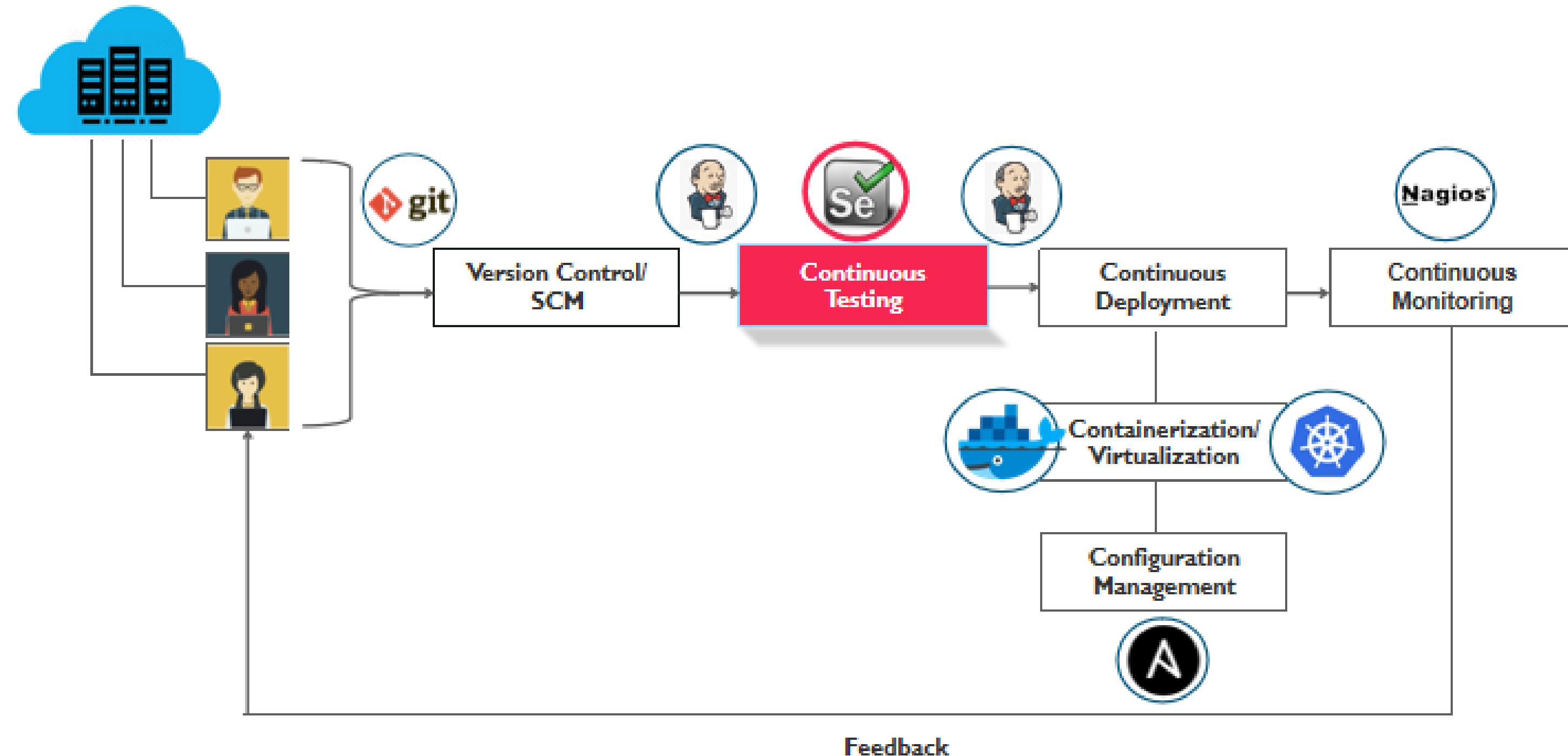
A development practice that requires developers to integrate code into a shared repository several times a day. By integrating regularly, you can detect errors quickly, and locate them more easily.





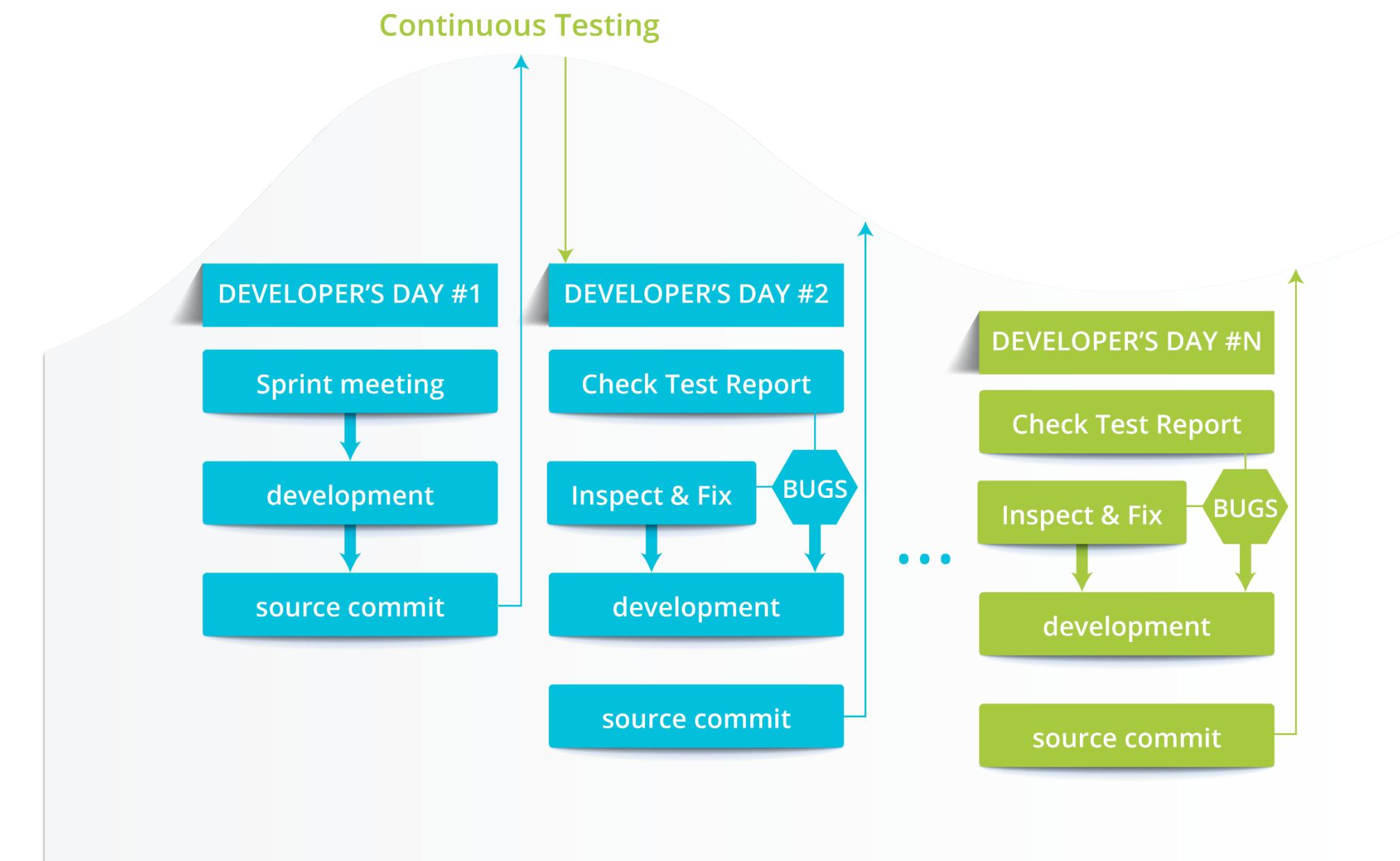
Continuous Testing

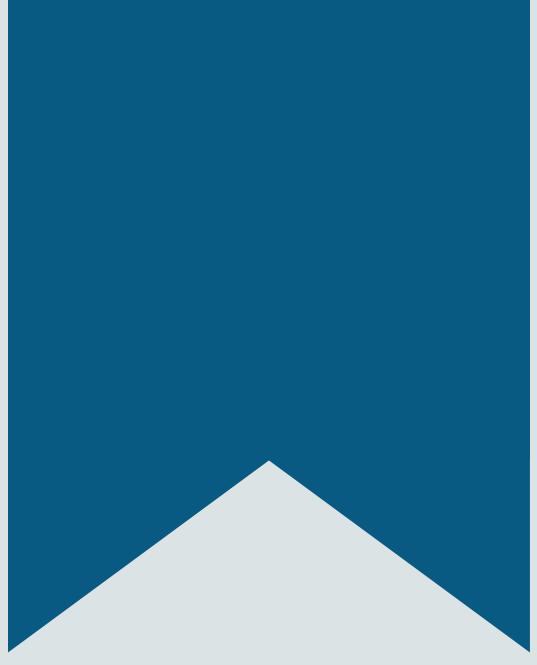
Continuous Testing



What Is Continuous Testing?

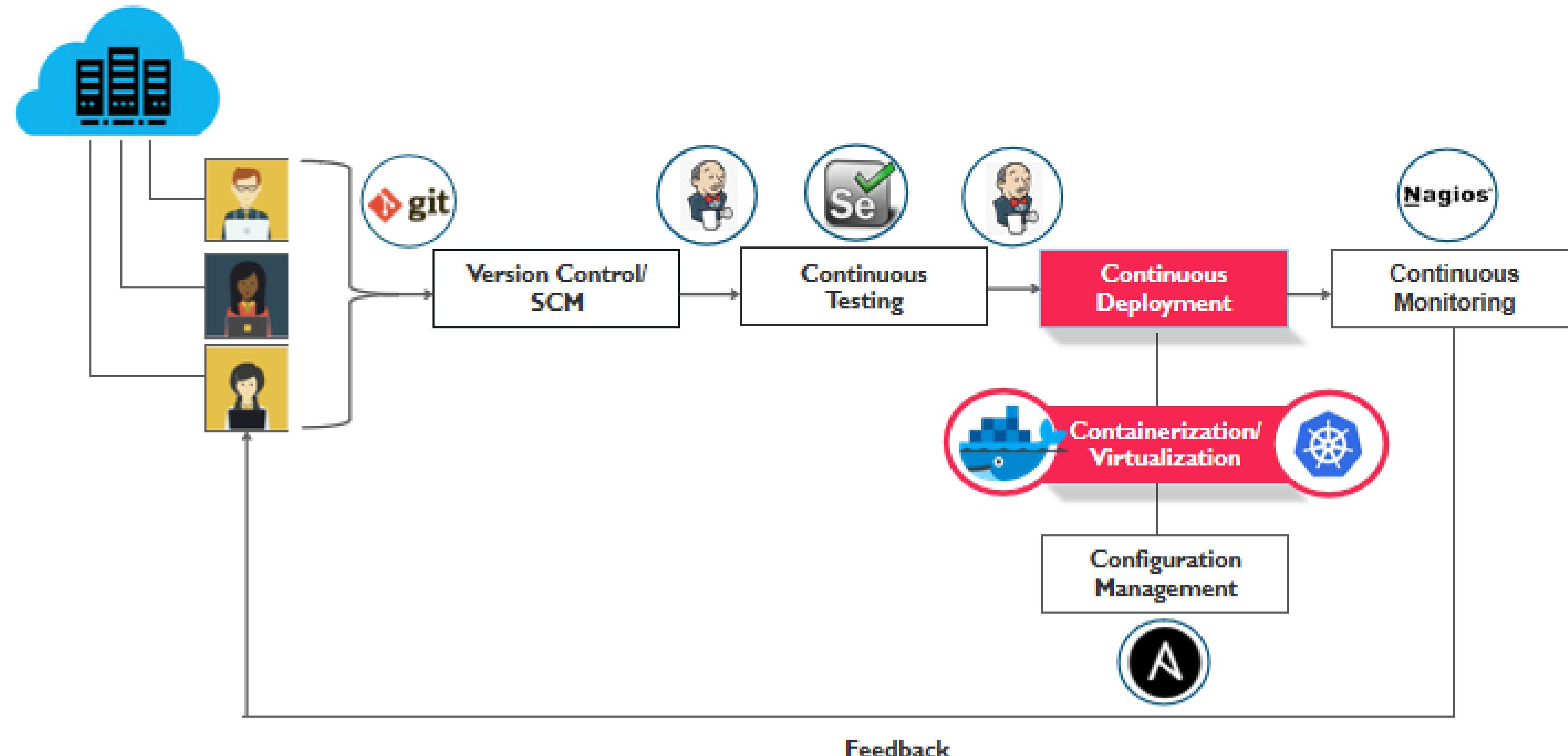
Process of executing automated tests as part of the software delivery pipeline to obtain immediate feedback





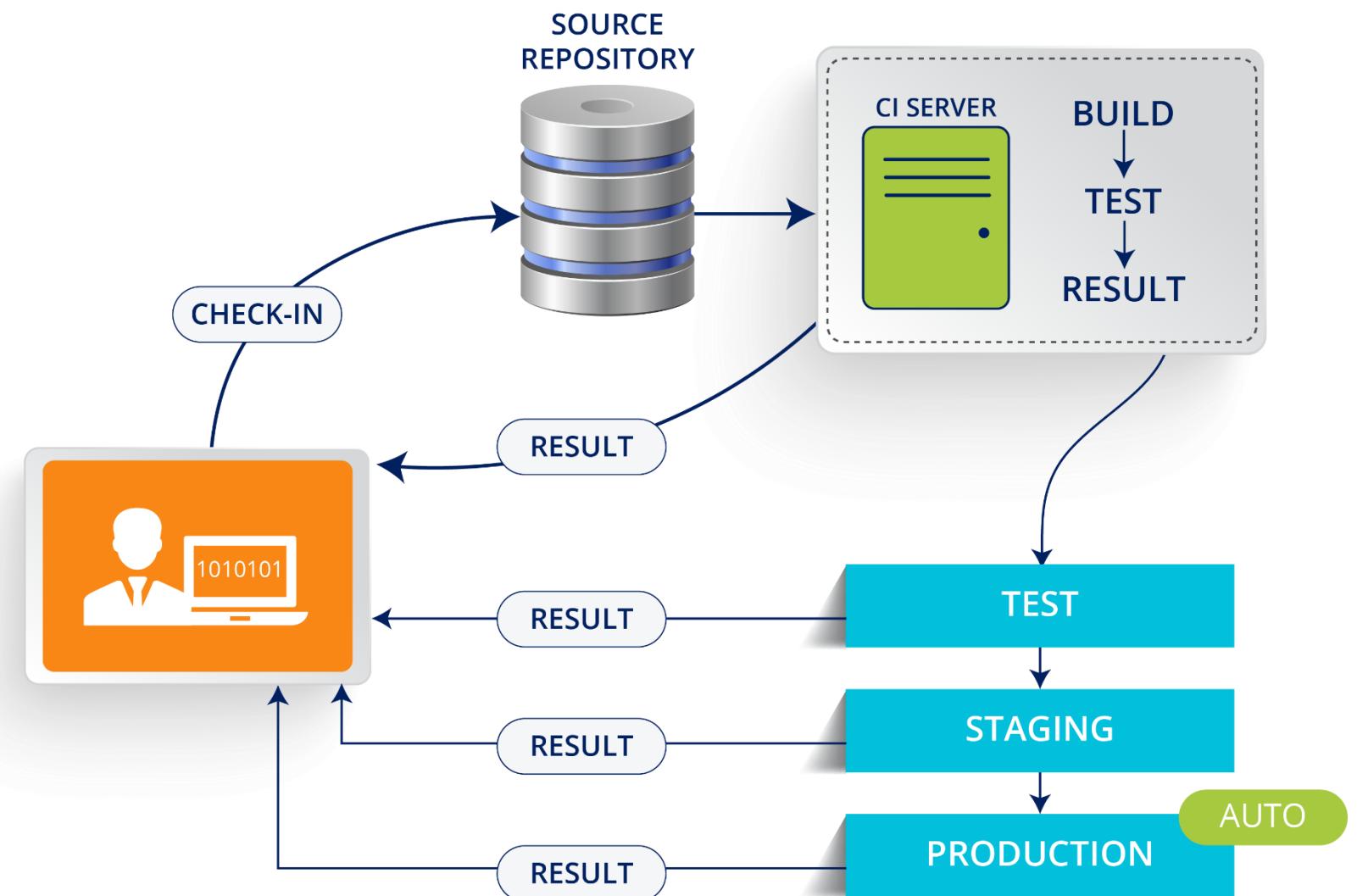
Continuous Deployment

Continuous Deployment



What Is Continuous Deployment?

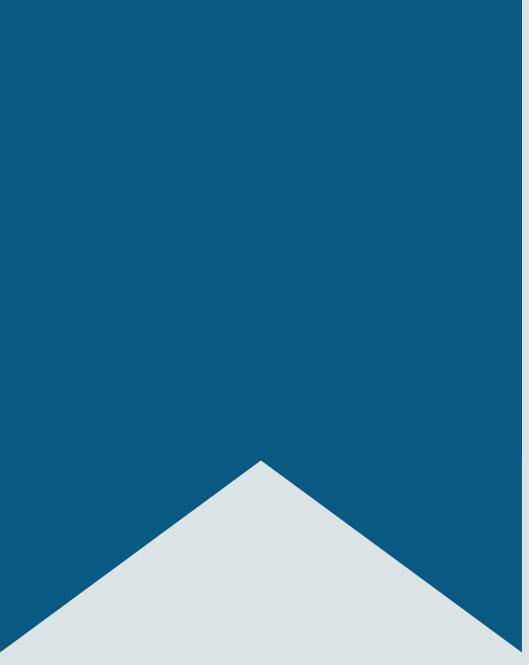
Continuous Deployment is a DevOps practice where the code changes are automatically built, tested, and prepared for a release to production



What Is Containerization?

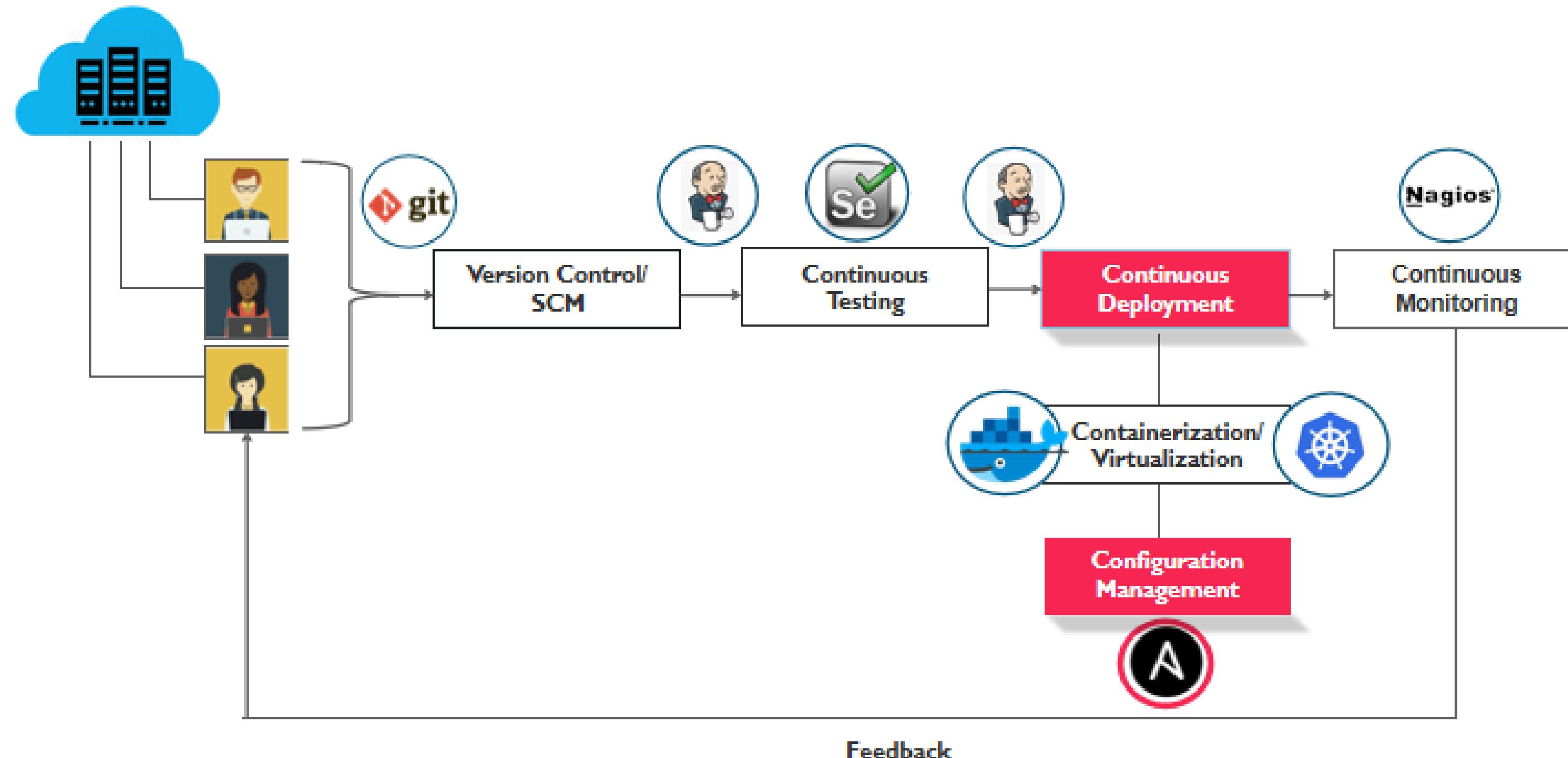
Containerization is a lightweight alternative to full machine virtualization which involves encapsulating an application in a container with its working environment.





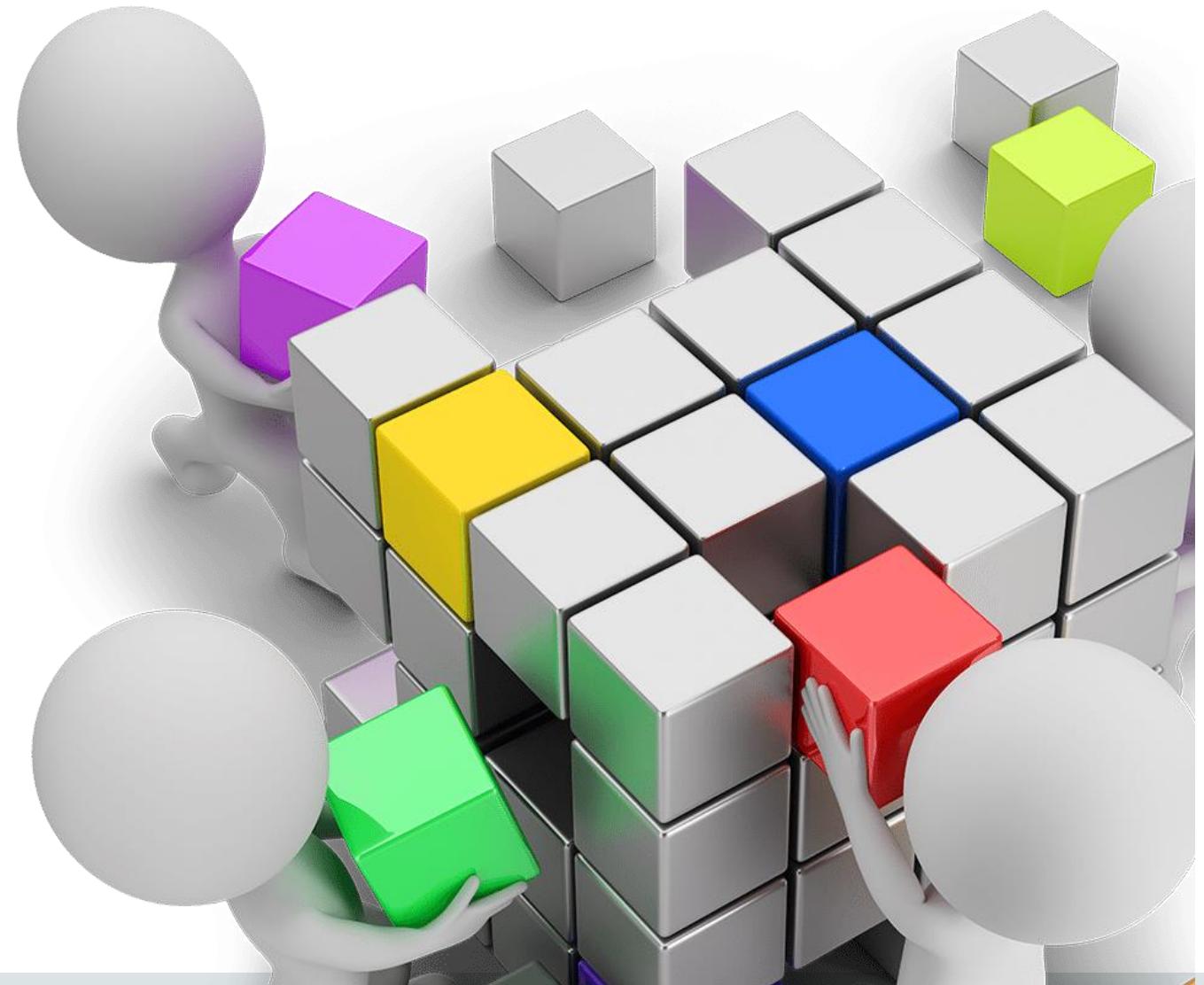
Configuration Management with Puppet & Ansible

Configuration Management



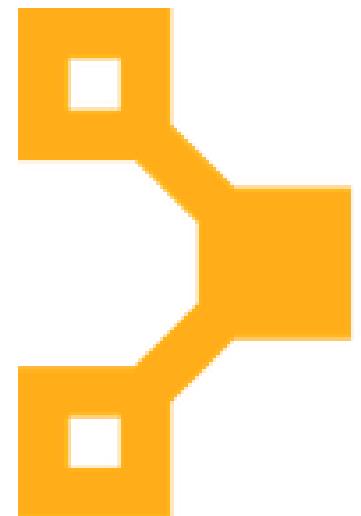
What Is Configuration Management?

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



Configuration Management Using Puppet

Puppet is a configuration management tool used to manage and maintain the development and deployment of software systems and servers in any computational environment



puppet

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

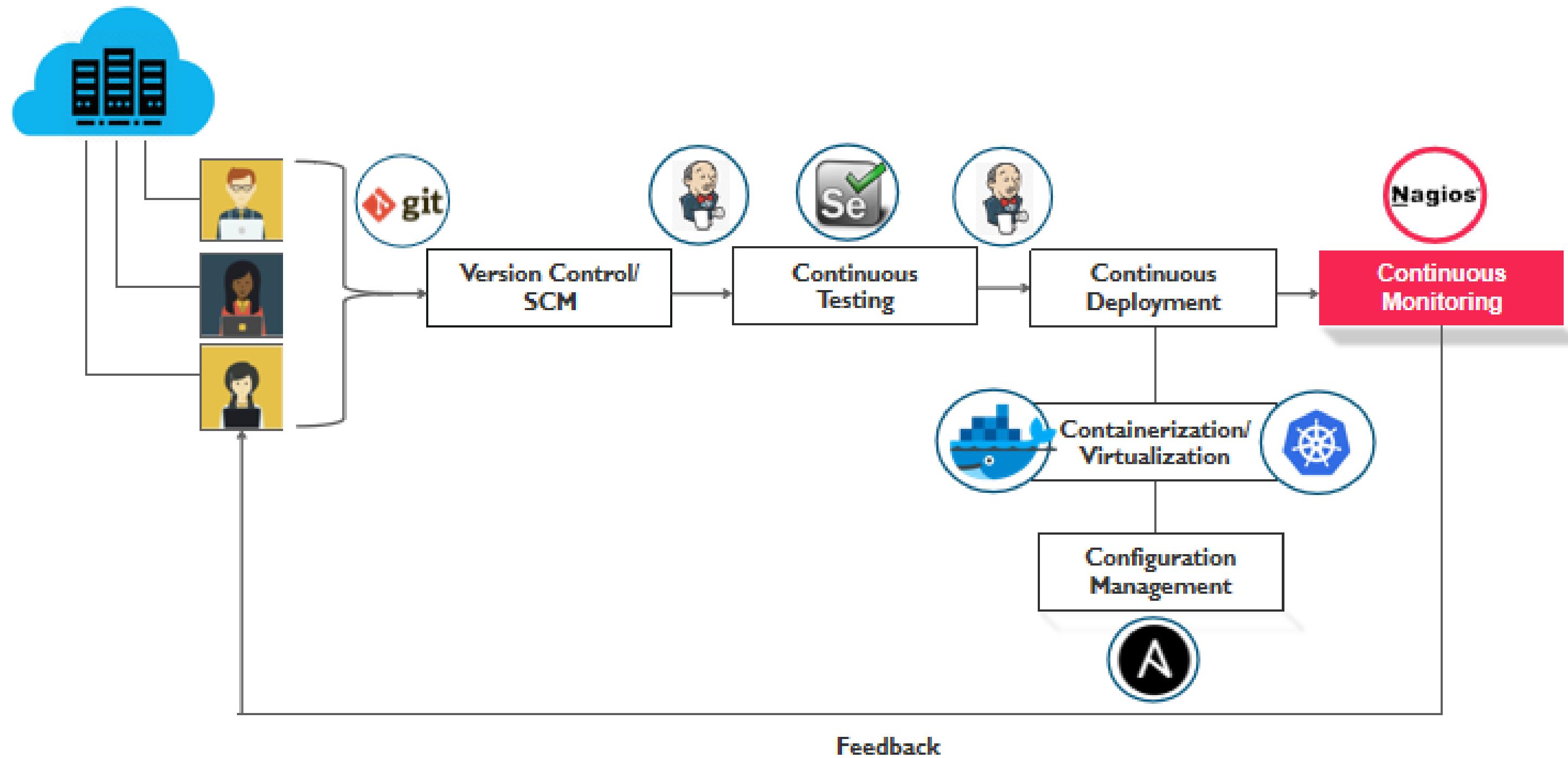


ANSIBLE



Continuous Monitoring

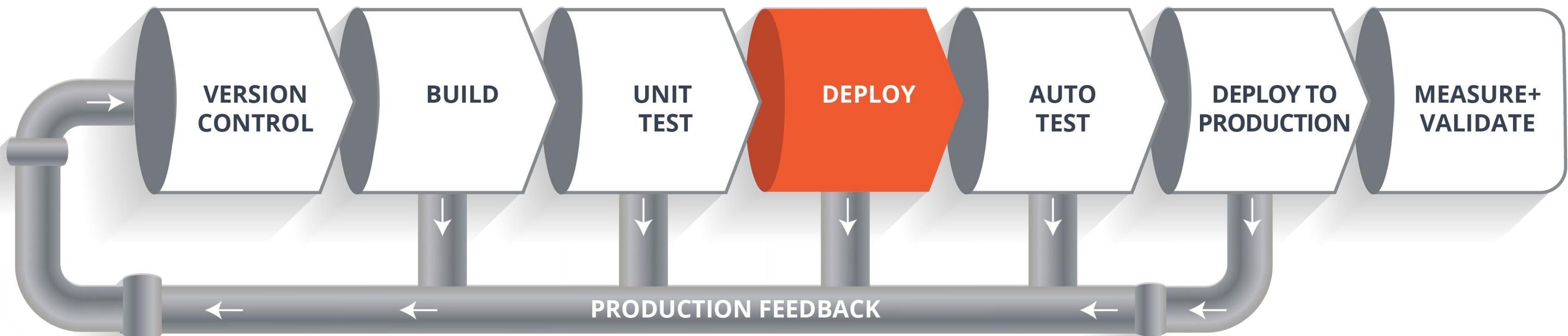
Continuous Monitoring



What Is Continuous Monitoring?



DevOps Delivery Pipeline



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

Copyright © edureka and/or its affiliates. All rights reserved.

Limitations of Waterfall Model

edureka!

Copyright © edureka and/or its affiliates. All rights reserved.

DevOps Lifecycle: Plan

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

Planning Tools

edureka!

Copyright © edureka and/or its affiliates. All rights reserved.

DevOps Stages

Feedback

edureka!

Copyright © edureka and/or its affiliates. All rights reserved.

What Is Configuration Management?

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

edureka!

Copyright © edureka and/or its affiliates. All rights reserved.

DevOps Delivery Pipeline

edureka!

Copyright © edureka and/or its affiliates. All rights reserved.

Questions

FEEDBACK



Survey



Ratings



Ideas



Comments



Suggestions



Likes



Thank You

For more information please visit our website
www.edureka.co