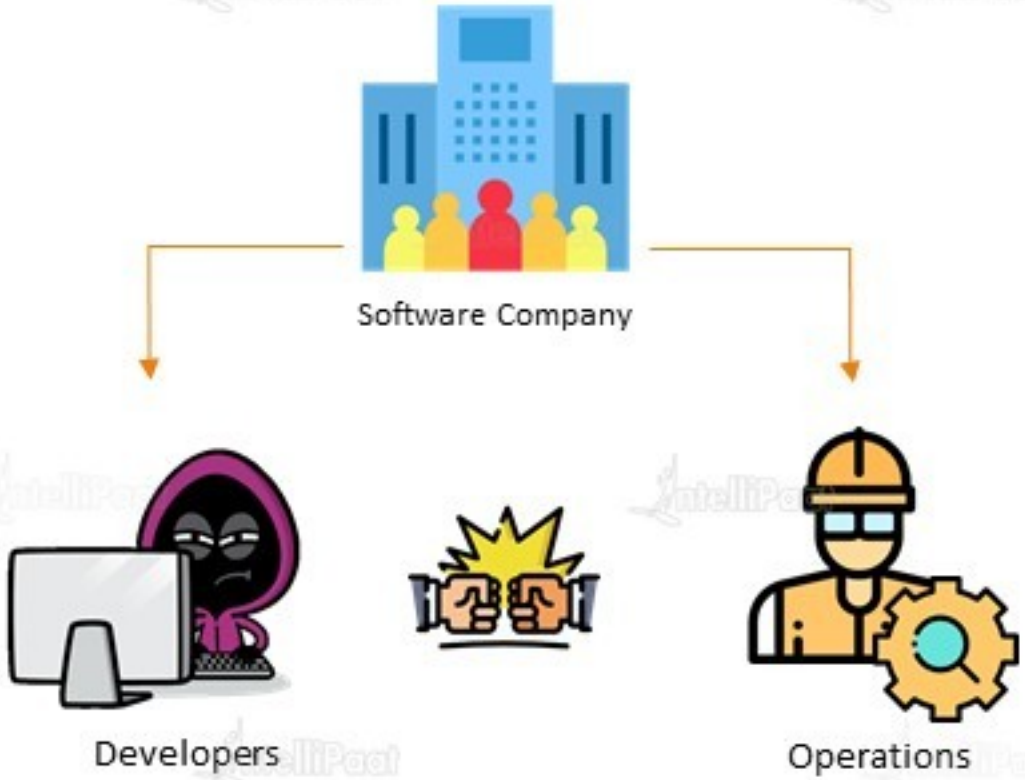
Why DevOps?

Why DevOps?



Although, the software quality was improved. We still had a lack of efficiency among the development team. A typical software development team consists of Developers and Operations employees. Let us understand their job roles

Why DevOps?

A developer's job is to develop applications and pass his code to the operations team

Developer

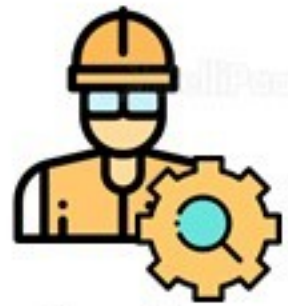
The operations team job is to test the code, and provide feedback to developers in case of bugs. If all goes well, the operations team uploads the code to the build servers

Operations

Why DevOps?

Developer

The developer used to run the code on his system, and then forward it to operations team.



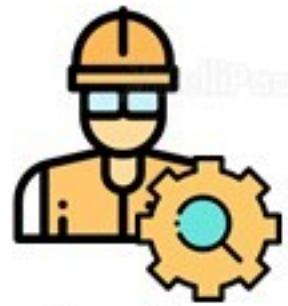
Operations

The operations when tried to run the code on their system, it did not run!

Why DevOps?

Developer

But, the code runs fine on the developer's system and hence he says "It is not my fault!"



Operations

The operations then marked this code as faulty, and used to forward this feedback to the developer

Why DevOps?

Developer

Operations

This led to a lot of back and forth between the developer and the operations team, hence impacted efficiency.

Why DevOps?

Developer

Operations

This problem was solved using Devops!

Traditional IT vs DevOps

Traditional IT	Devops
Less Productive	More Productive
Skill Centric Team	Team is divided into specialized silos
More Time invested in planning	Smaller and Frequent releases lead to easy scheduling and less time in planning
Difficult to achieve target or goal	Frequent releases, with continuous feedback makes achieving targets easy

What is Devops?

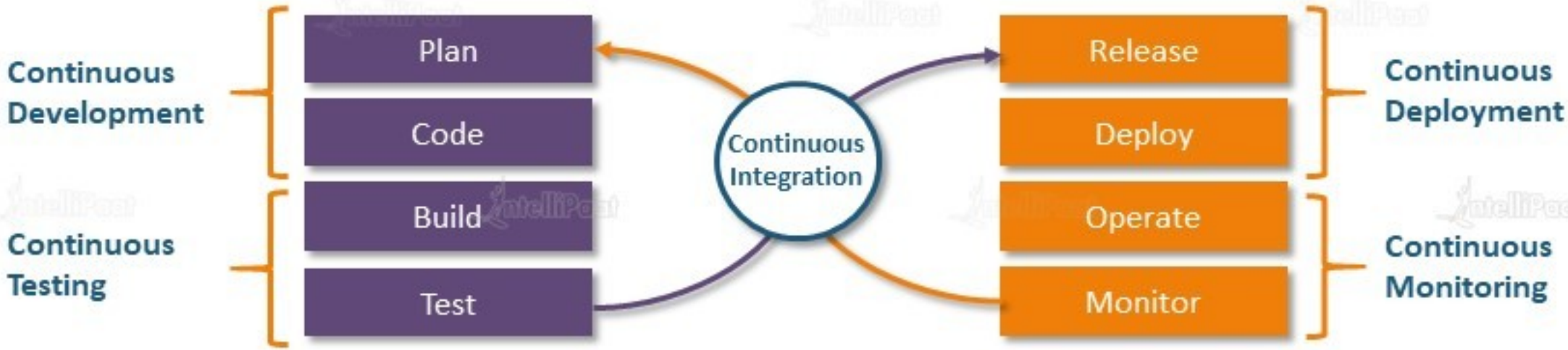
What is DevOps?

Devops is a software development methodology which improves the collaboration between developers and operations team using various automation tools. These automation tools are implemented using various stages which are a part of the Devops Lifecycle

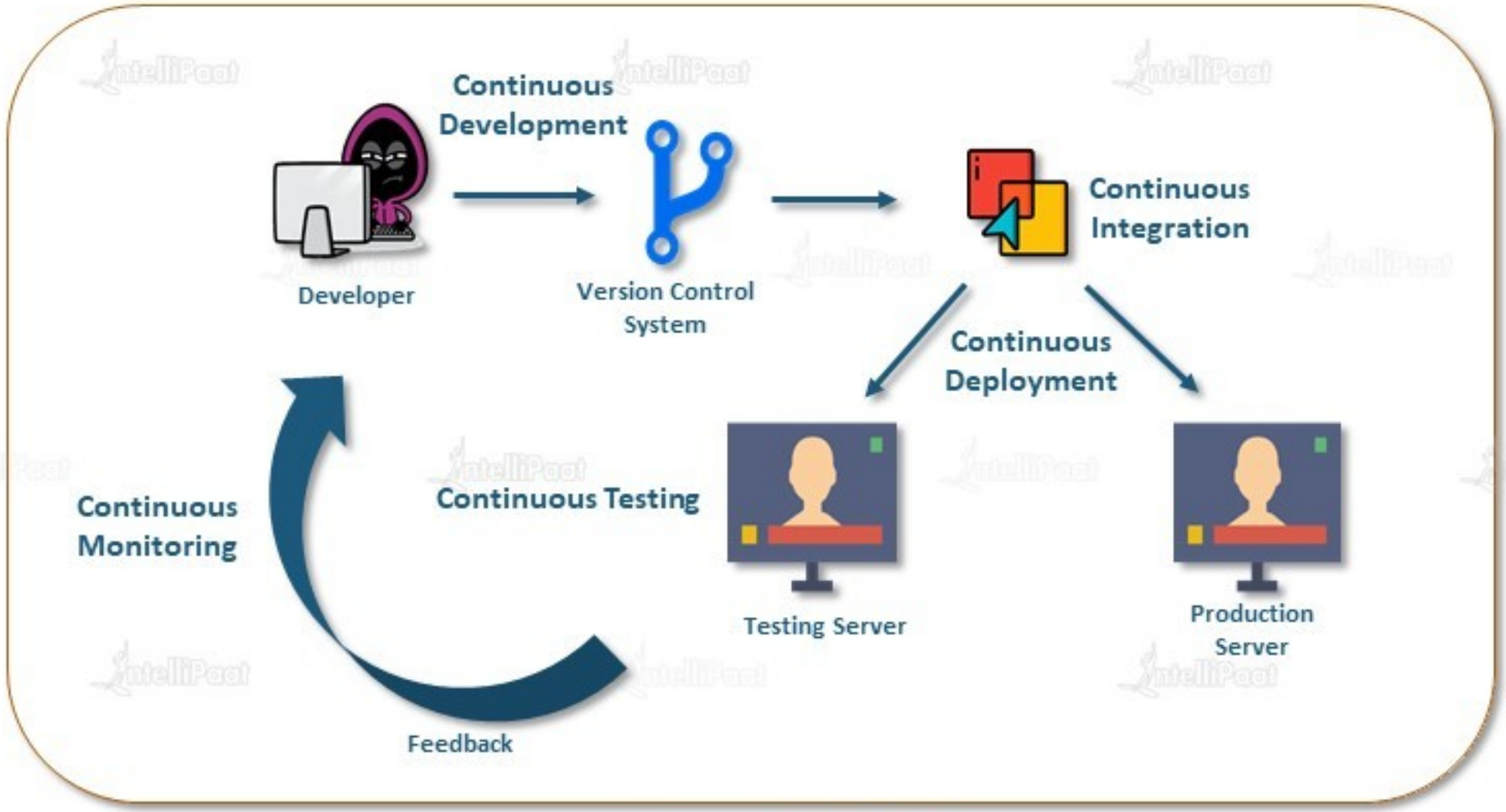
DevOps Lifecycle

How DevOps Works?

The Devops Lifecycle divides the SDLC lifecycle into the following stages:



How DevOps Works?



Automated CI/CD Pipeline

How DevOps Works?

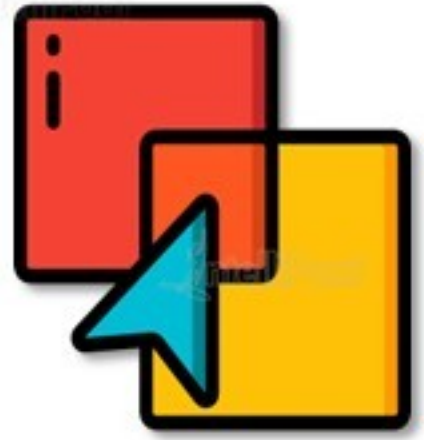
- **Continuous Development**
- Continuous Integration
- Continuous Deployment
- Continuous Testing
- Continuous Monitoring

This stage involves committing code to version control tools such as **Git** or **SVN** for maintaining the different versions of the code, and tools like **Ant**, **Maven**, **Gradle** for building/ packaging the code into an executable file that can be forwarded to the QAs for testing.

How DevOps Works?

- Continuous Development
- Continuous Integration
- Continuous Deployment
- Continuous Testing
- Continuous Monitoring

The stage is a critical point in the whole Devops Lifecycle. It deals with integrating the different stages of the devops lifecycle, and is therefore the key in automating the whole Devops Process



How DevOps Works?

- Continuous Development
- Continuous Integration
- Continuous Deployment
- Continuous Testing
- Continuous Monitoring

In this stage the code is built, the environment or the application is containerized and is pushed on to the desired server. The key processes in this stage are Configuration Management, Virtualization and Containerization

How DevOps Works?

- Continuous Development
- Continuous Integration
- Continuous Deployment
- **Continuous Testing**
- Continuous Monitoring

The stage deals with automated testing of the application pushed by the developer. If there is an error, the message is sent back to the integration tool, this tool in turn notifies the developer of the error. If the test was a success, the message is sent to Integration tool which pushes the build on the production server

How DevOps Works?

- Continuous Development
- Continuous Integration
- Continuous Deployment
- Continuous Testing
- Continuous Monitoring

The stage continuously monitors the deployed application for bugs or crashes. It can also be setup to collect user feedback. The collected data is then sent to the developers to improve the application