



Key Takeaways

Introduction to DevOps

Different IT Roles in Software Development & Delivery Process

Software Development



- software **programmed by developers**
- in different programming languages, like Java, Python, JavaScript
- new functionality & bugfixes

Software Testing



- **test** new features & old functionality
- done by developers & dedicated testers
- manual & automated testing

Release of Software

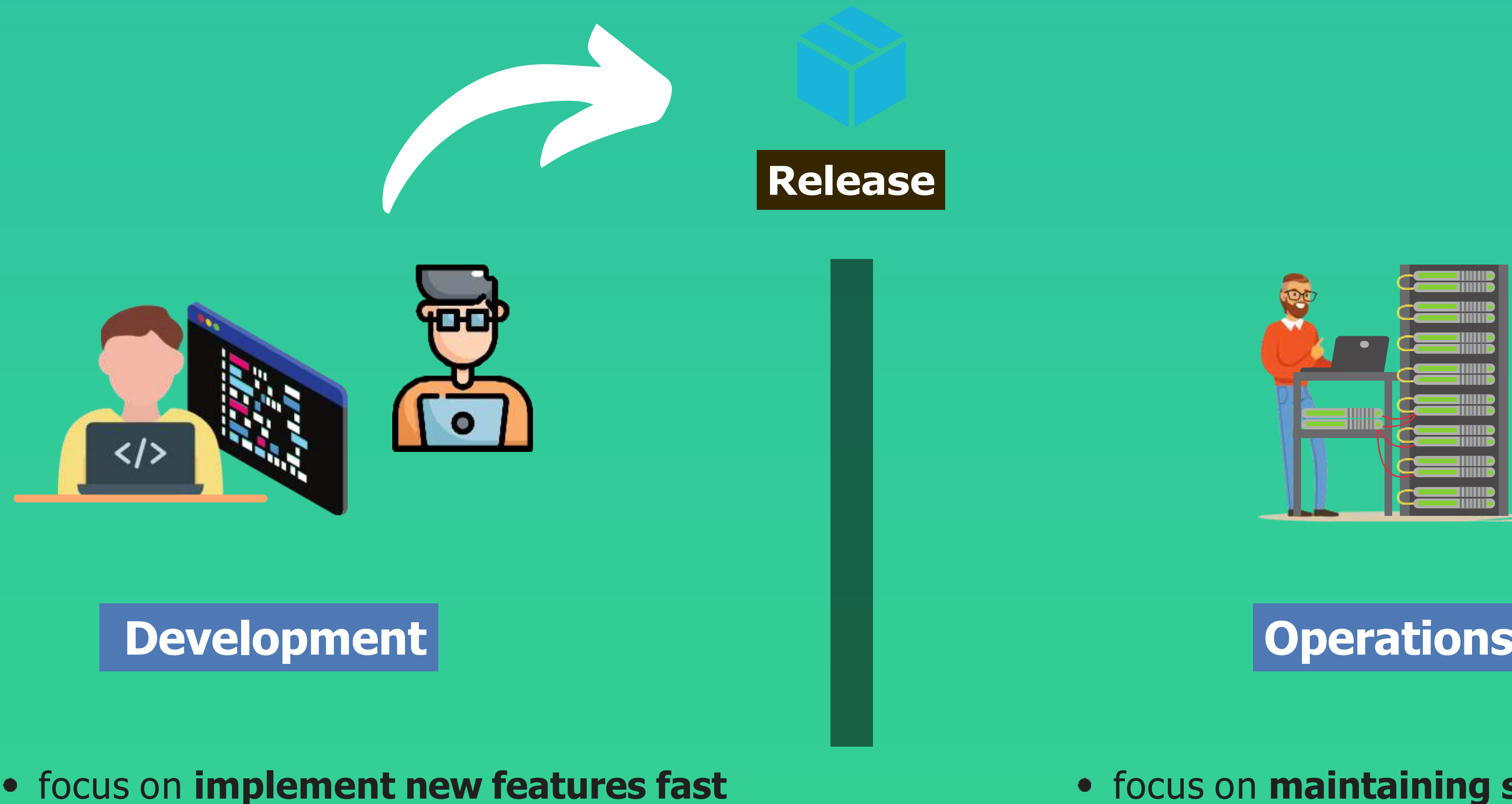


Operations



- **build** application
- **deploy** on servers
- **upgrade** existing software
- run software in production
- done by operations team

Traditional: Development vs Operations - 1



Traditional: Development vs Operations - 2



**Different
Responsibilities**

**Different
Technical Knowledge**

**Different
Toolsets**

Development

- programming languages
- test frameworks
- databases
- version control

Operations

- OS, mostly Linux
- command-line
- scripting
- monitoring tools

Traditional: Development vs Operations - 3



Miscommunication and
Lack of Collaboration



- Deployment requires configurations & environment needs to be prepared, so communication is important
- In reality **silos between those 2 departments**



Conflict of Interest



- **DEV Focus** on releasing new features fast
- **OPS Focus** on maintaining systems' stability



Manual
Work & Checklists



- **Manual checks:** does new change affect systems' stability or security
- **Manual deployment**
- **Manual configurations** of deployment environment



Slows down the release process!!



Solution: DevOps Culture - 1

Simplest Definition:

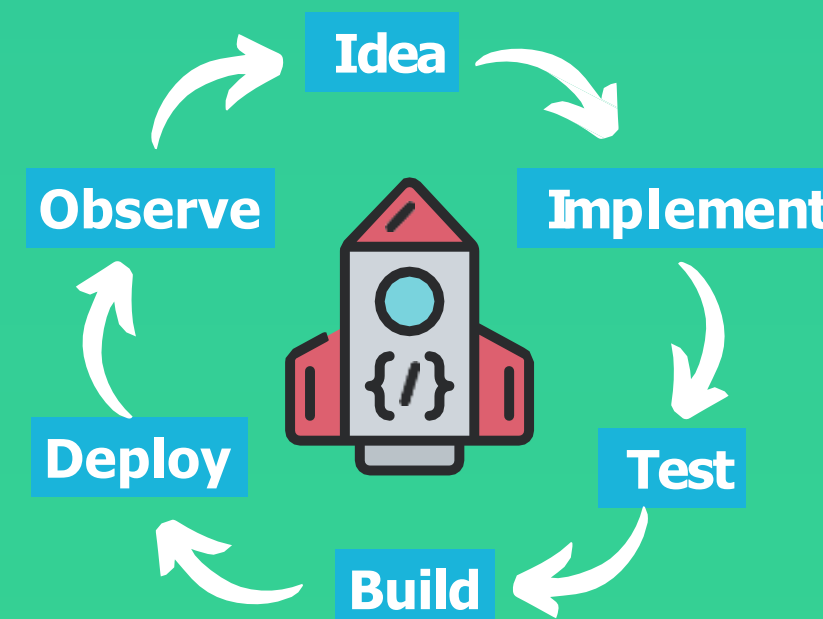
Intersection of Development & Operations

- DevOps was just a way of working between **DEV**'s and **OP**'s
- Common language to communicate

Better Definition:

DevOps is about making the process of continuous delivery fast and with minimal errors

- DevOps tries to **remove all these roadblocks** and things that slow down the release process
- Instead of manual inefficient processes helps create fully **streamlined processes for release cycles**





Solution: DevOps Culture - 2



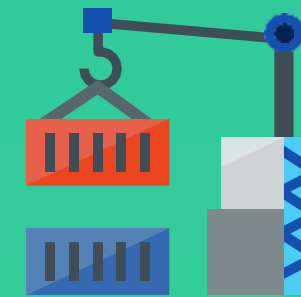
- Different companies implemented DevOps in different ways
- But, gradually it got a more concrete form with common patterns



**Source Code
Management**



**Continuous
Integration/Deployment**



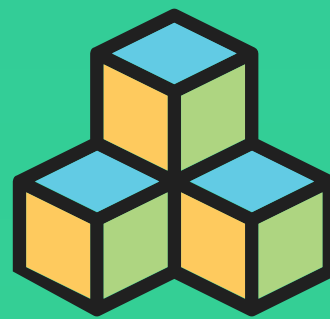
**Container
Orchestration**



**Continuous
Monitoring**



**Planning &
Collaboration**



**Package
Management**



**Treat Infrastructure
as Code**



Cloud



DevOps Engineer

DevOps as a separate Role



- DevOps evolved into an **actual role**: "DevOps Engineer"
- Where either Developers, Operations or a separate role is responsible for DevOps



Developers
doing DevOps



Doing only
DevOps



Operations
doing DevOps

- **DevOps tools**: Set of technologies used to implement the DevOps Principles became DevOps technologies
- DevOps Engineer responsible for creating a streamlined fully automated release process



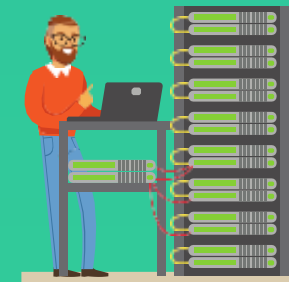
DevOps Tasks and Responsibilities - 1



Need **some know-how from DEV and OPS team**



Development



Operations

DEV OPS

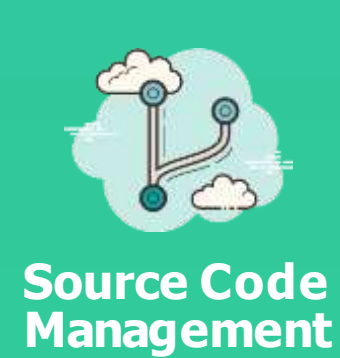


Additional DevOps specific skills and know-how



DevOps Tasks and Responsibilities - 2

- DevOps tools that are known today:
- Replacing human tasks with automation tools



Source Code Management

Git



Package Management

Docker



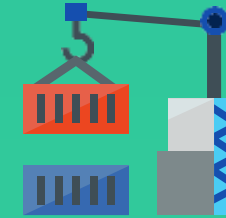
Infrastructure as Code

Terraform,
Ansible



Continuous Integration/Delivery

Jenkins,
Gitlab



Container Orchestration

Kubernetes



Cloud

AWS, Azure,
Google Cloud



Continuous Monitoring

Prometheus

- At the core of DevOps: **CI/CD Pipeline for an automated release process**





High Level Overview & Big Picture of DevOps

Throughout Bootcamp: Zoom in into each part and understand it in detail and be able to implement the whole DevOps process