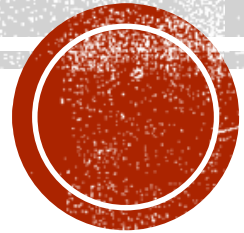
DEVOPS INTRO



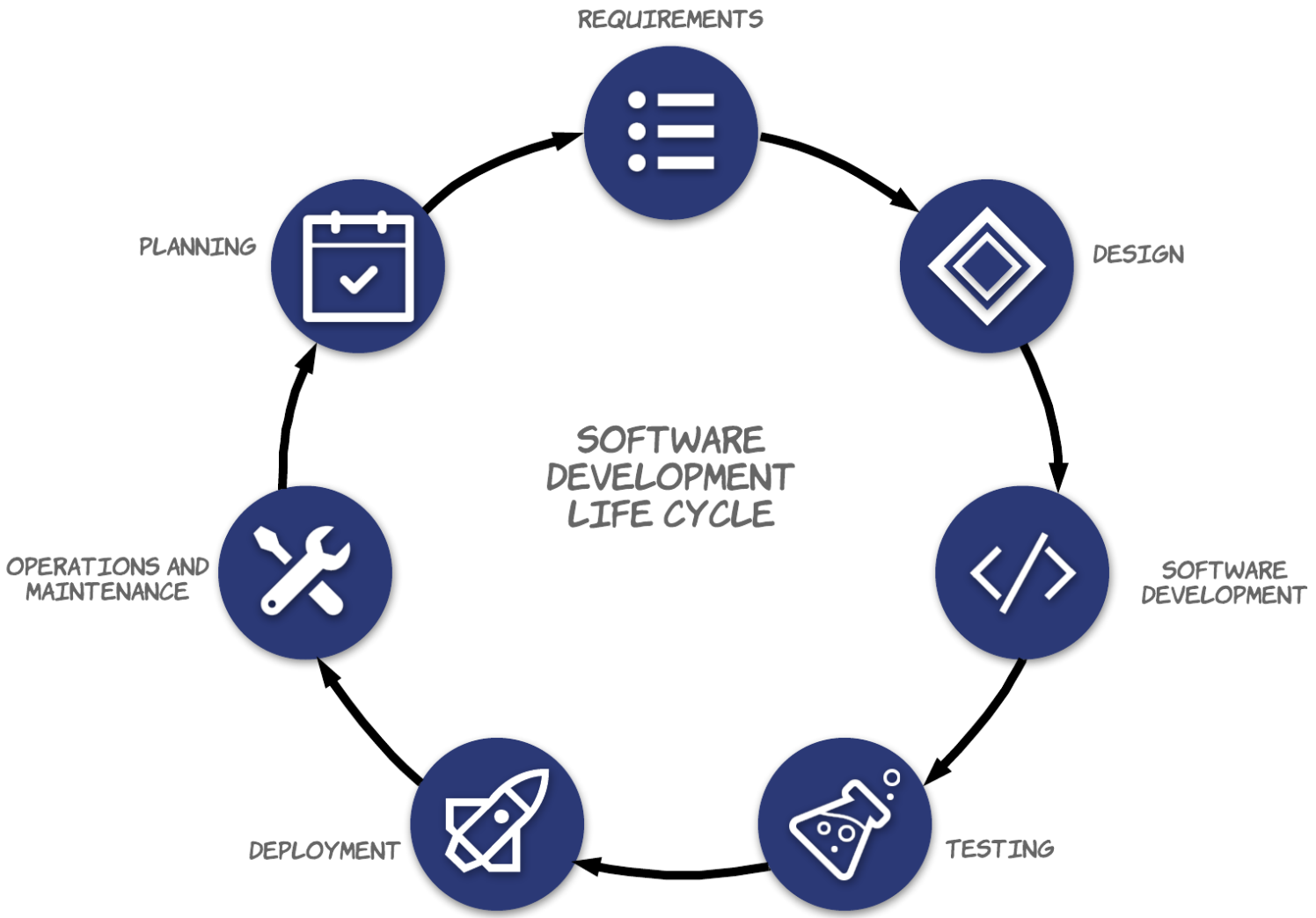
Cloud Technologies Inc

AGENDA

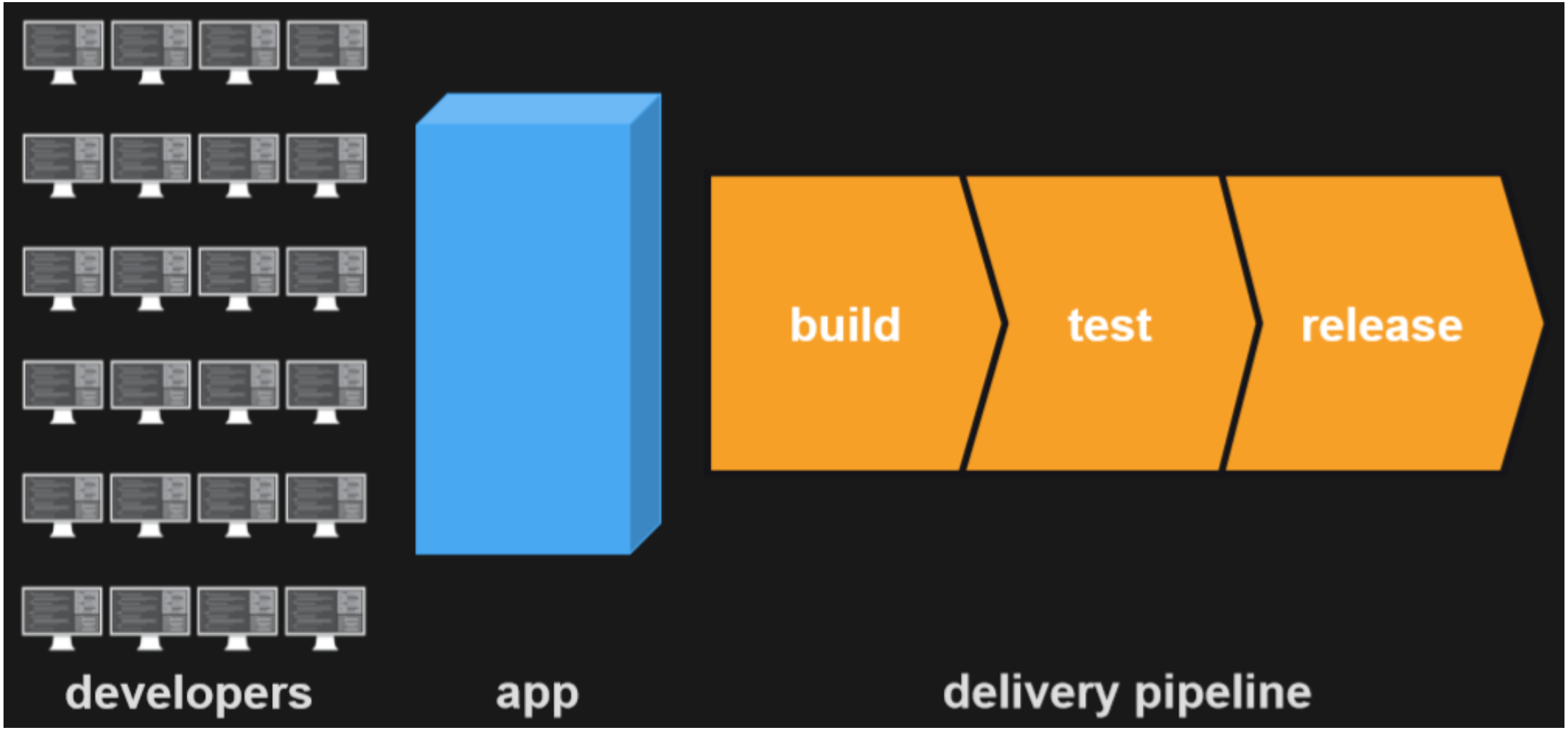
- Software Development Life Cycle / Agile
- What is Devops
- Why Devops?
- Devops vs Release Management
- When to adopt and when not to
- DevOps LifeCycle Phase
- Pipeline Overview

SOFTWARE DEVELOPMENT LIFE CYCLE / AGILE

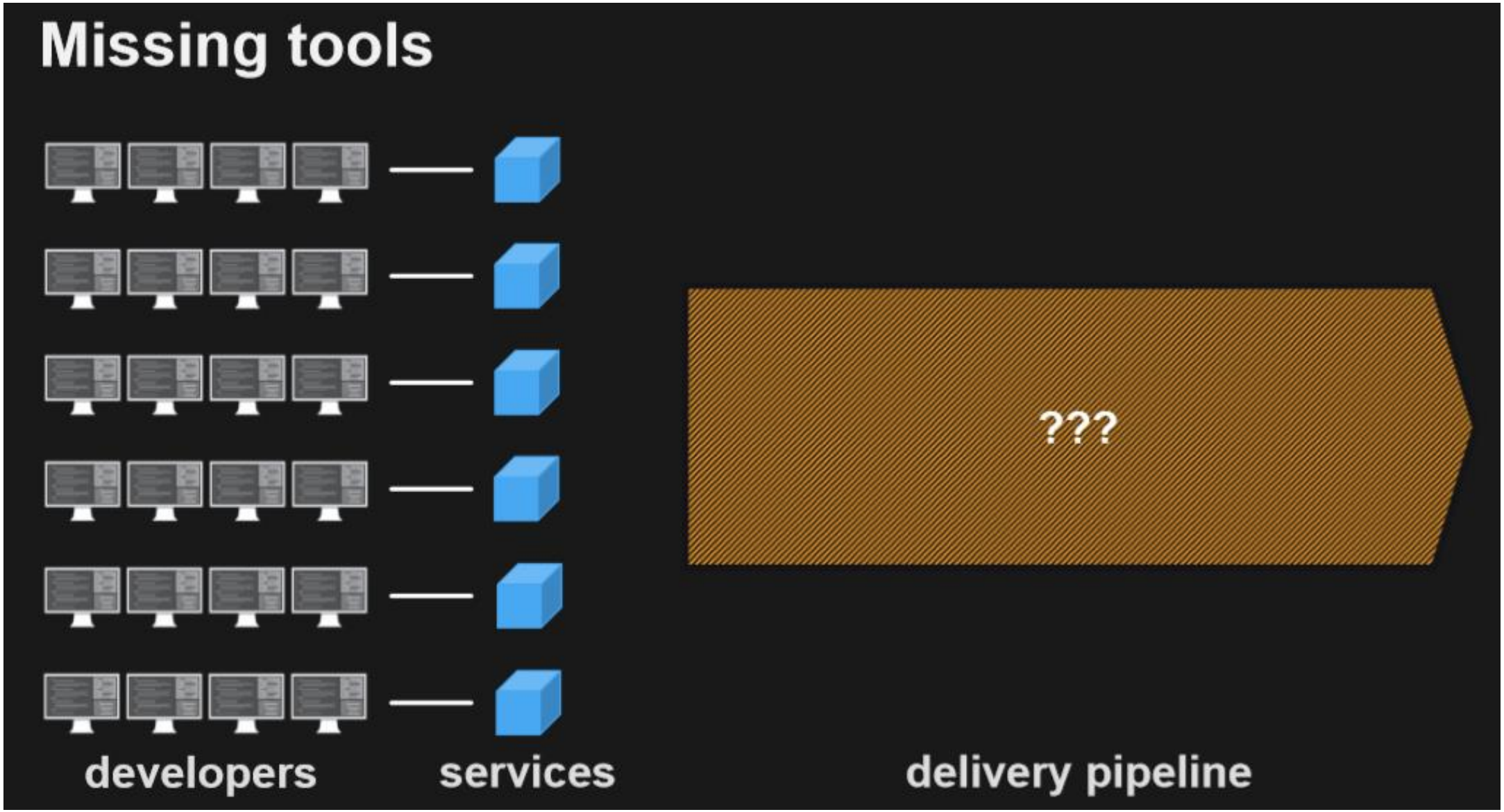
SOFTWARE DEVELOPMENT LIFE CYCLE



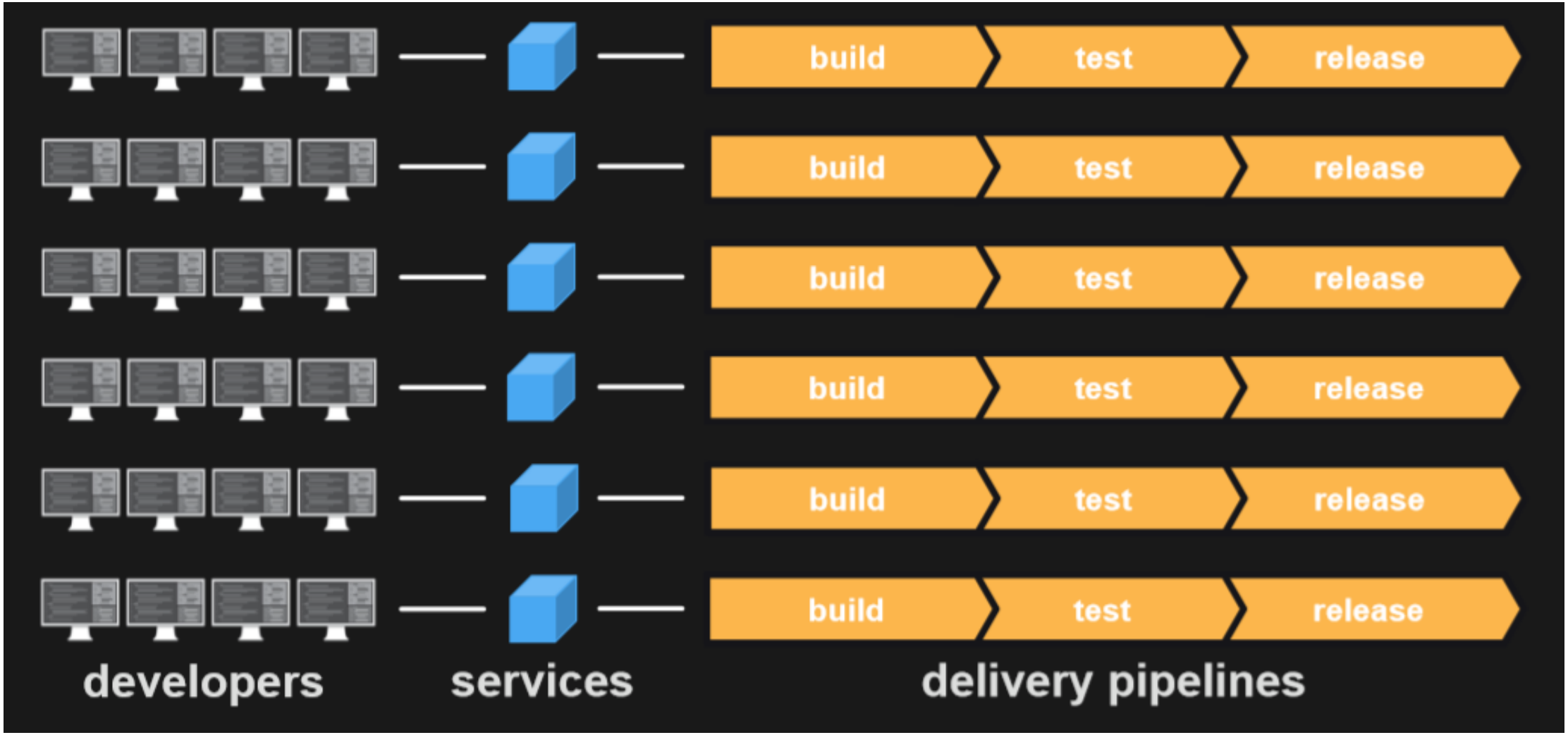
MONOLITH DEVELOPMENT LIFECYCLE



MONOLITH DEVELOPMENT LIFECYCLE



MICROSERVICE DEVELOPMENT LIFECYCLE



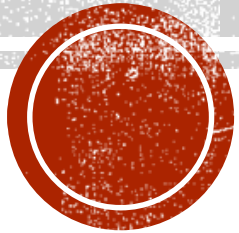
AGILE SOFTWARE DEVELOPMENT

- **Agile software development** is a conceptual framework for software engineering that promotes development iterations throughout the life-cycle of the project.

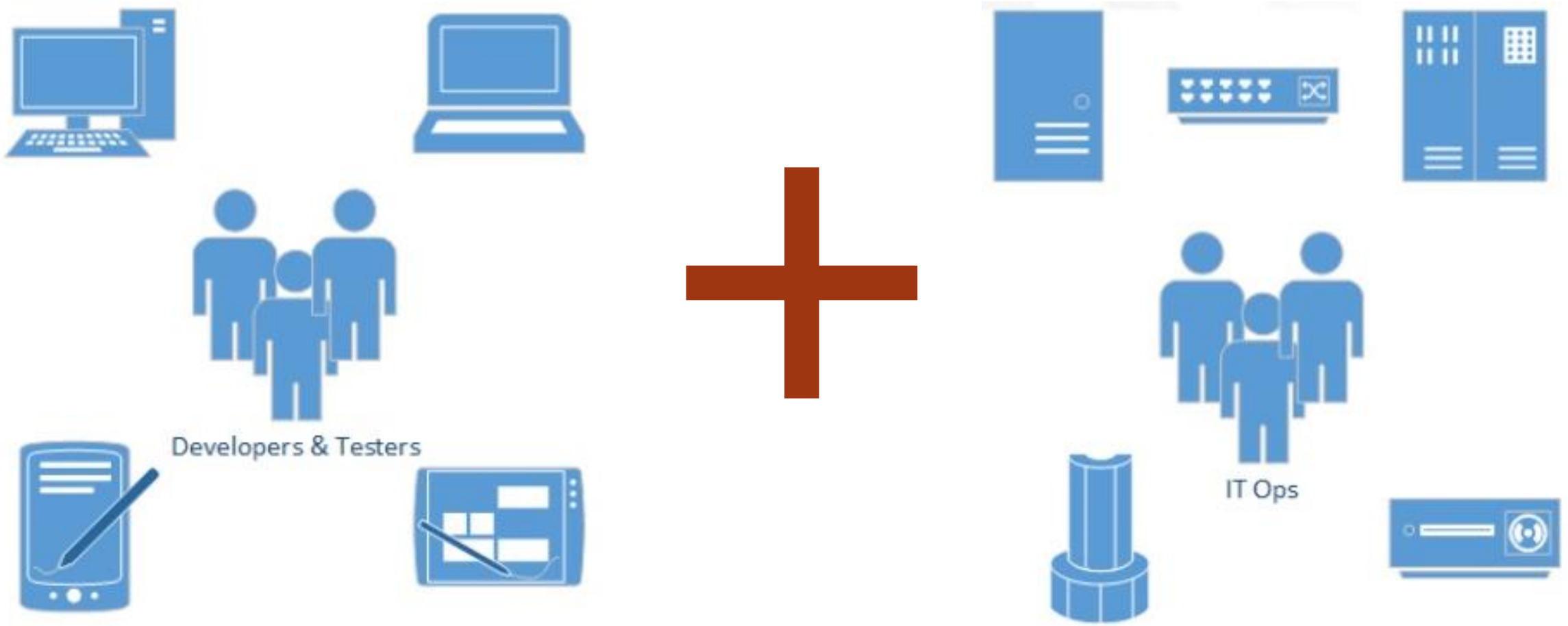
- Software developed during one unit of time is referred to as an iteration, which may last from one to four weeks.

- Agile methods also emphasize working software as the primary measure of progress

WHAT IS DEVOPS?



WHAT IS DEVOPS?
-- EFFICIENCIES THAT SPEED UP LIFECYCLE



WHAT IS DEVOPS?

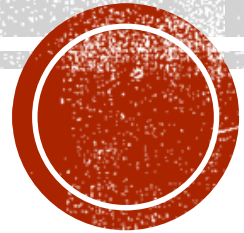
- Devops is a software development method that stresses
  - Communication,
  - Collaboration
  - Integration

  Between Software Developers and Information technology professionals thereby

  Enable rapid evolution of products or services

  Reduce risk, improve quality and reduce costs

WHY DEVOPS?



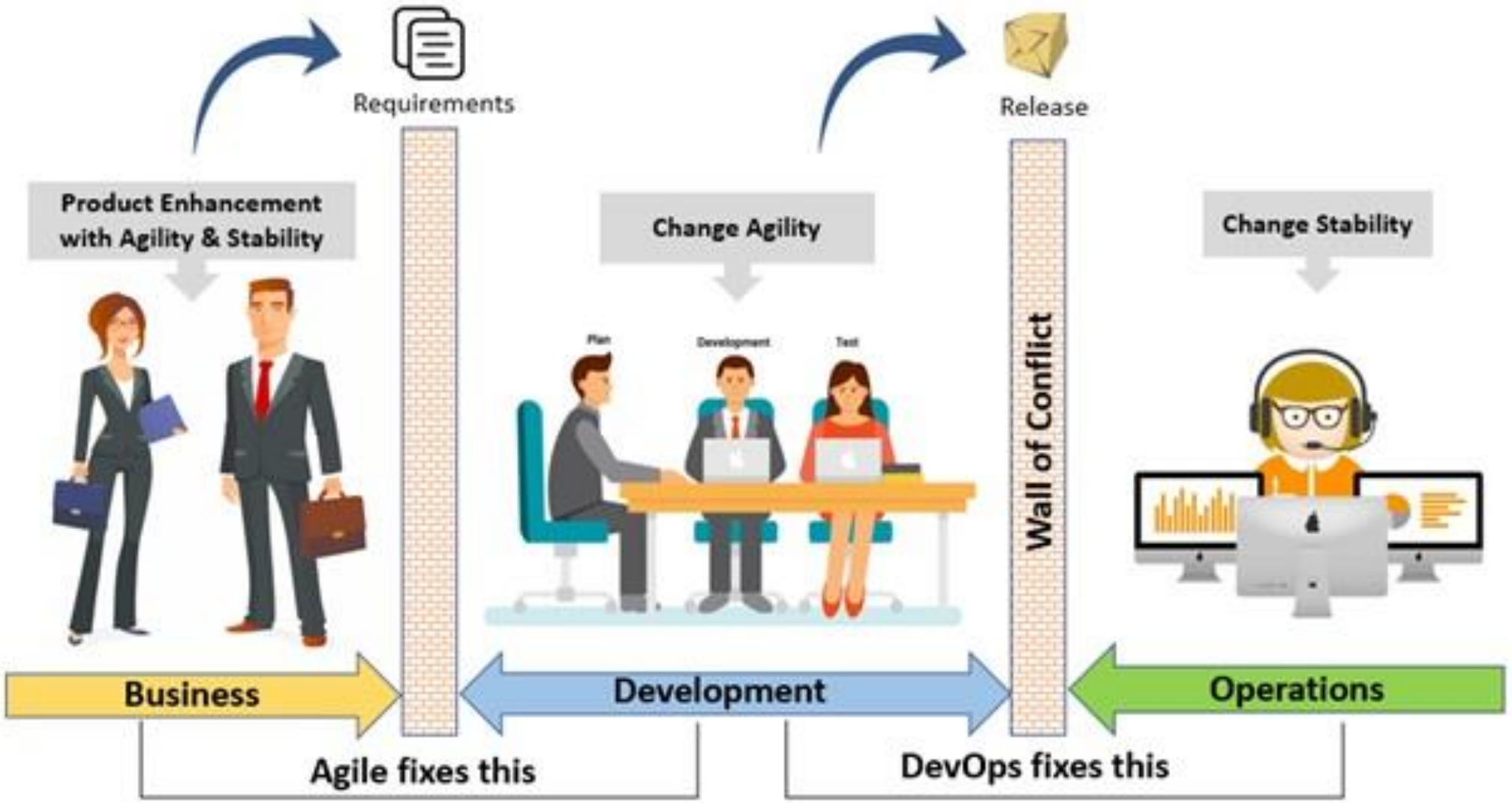
WHY GAPS?

- Developer View
  - Delivers features after testing in development systems
  - Dev system may not be same as production system
  - Developers will have faster turn around time w.r.t features
  - Not much concerned about the infrastructural as well as deployment impact because of the code changes

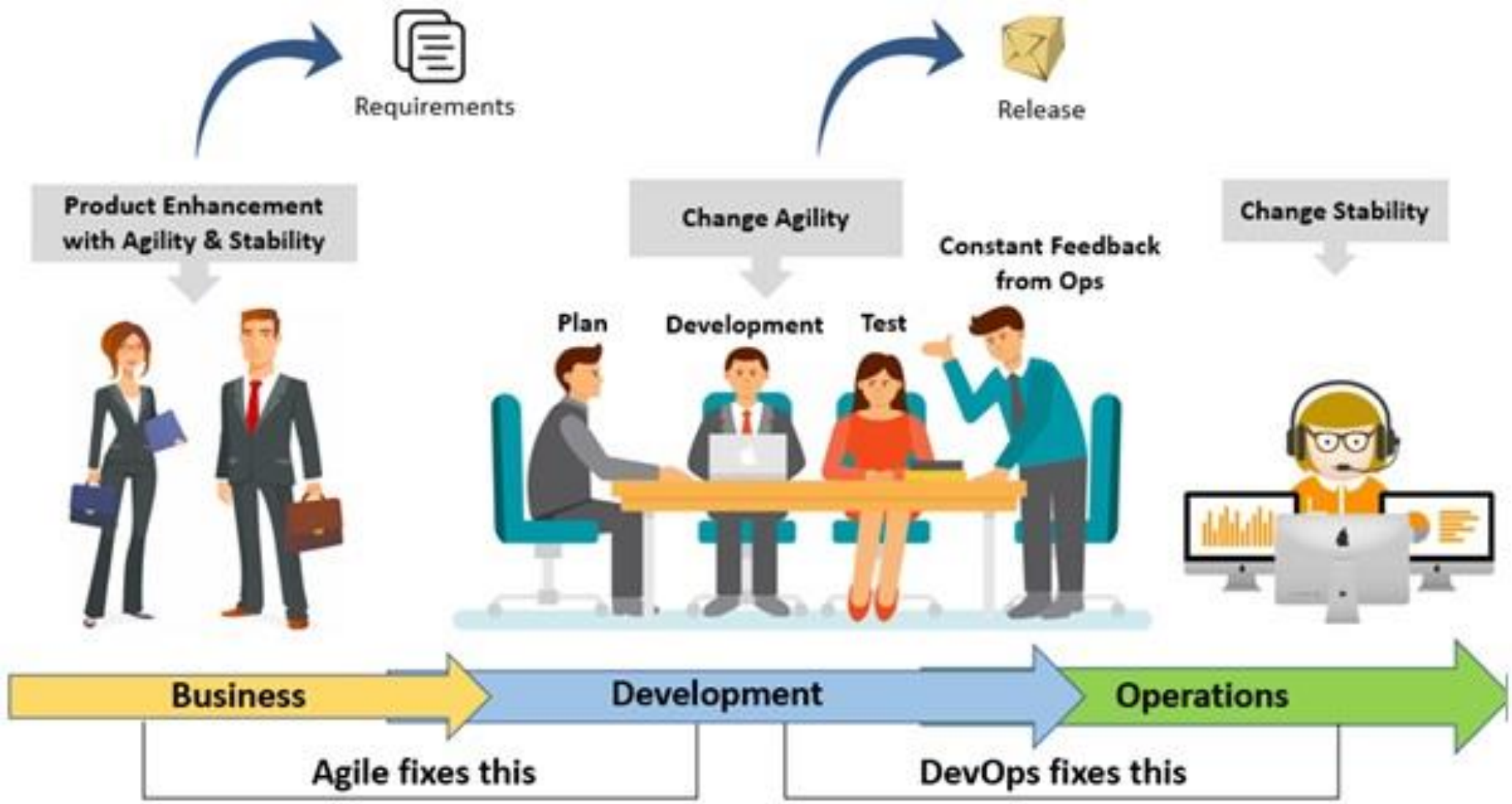
WHY GAPS?

- Ops View
  - Rewarded mainly for uptime
  - Lesser turn around time w.r.t feature deployment and testing due to large number of dev build coming their way
  - Very much concerned about the infrastructural as well as deployment impact because of the code changes

WHY GAPS?


Requirements
Release
Product Enhancement with Agility & Stability
Change Agility
Change Stability
Wall of Conflict
Business
Development
Operations
Agile fixes this
DevOps fixes this

DEV AND OPS


Requirements
Release
Product Enhancement with Agility & Stability
Change Agility
Change Stability
Constant Feedback from Ops
Plan
Development
Test
Business
Development
Operations
Agile fixes this
DevOps fixes this

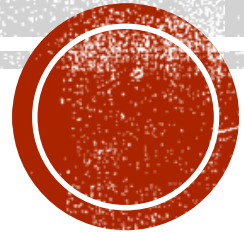
DEV AND OPS

- Developers work with Ops to understand the impact of code changes
- Ops now have more clarity on infra needs
- Closely monitors the DEV – TEST – PROD pipeline for each deployment with immediate feedback

TOP 3 DELIVERY CHALLENGES ADDRESSED

- Release management
  - Better understanding of risks, dependencies, compliance issues
- Release/Deployment Coordination
  - Better tracking of discrete activities, faster escalation of issues, documented process control and granular reporting
- Release/Deployment Automation
  - Usually have existing automation but want to flexibly manage and drive this automation that can be invoked by non-operations resources for non-prod env.

DEVOPS VS RELEASE MANAGEMENT



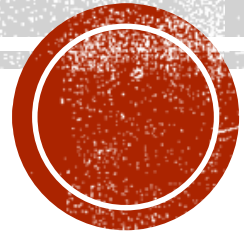
WHEN TO ADOPT AND WHEN NOT TO

WHEN TO ADOPT AND WHEN NOT TO

- When to Adopt:
  - For eCommerce and other web site projects (Amazon, Flickr, etc)
  - Cloud Platform (IaaS and PaaS)

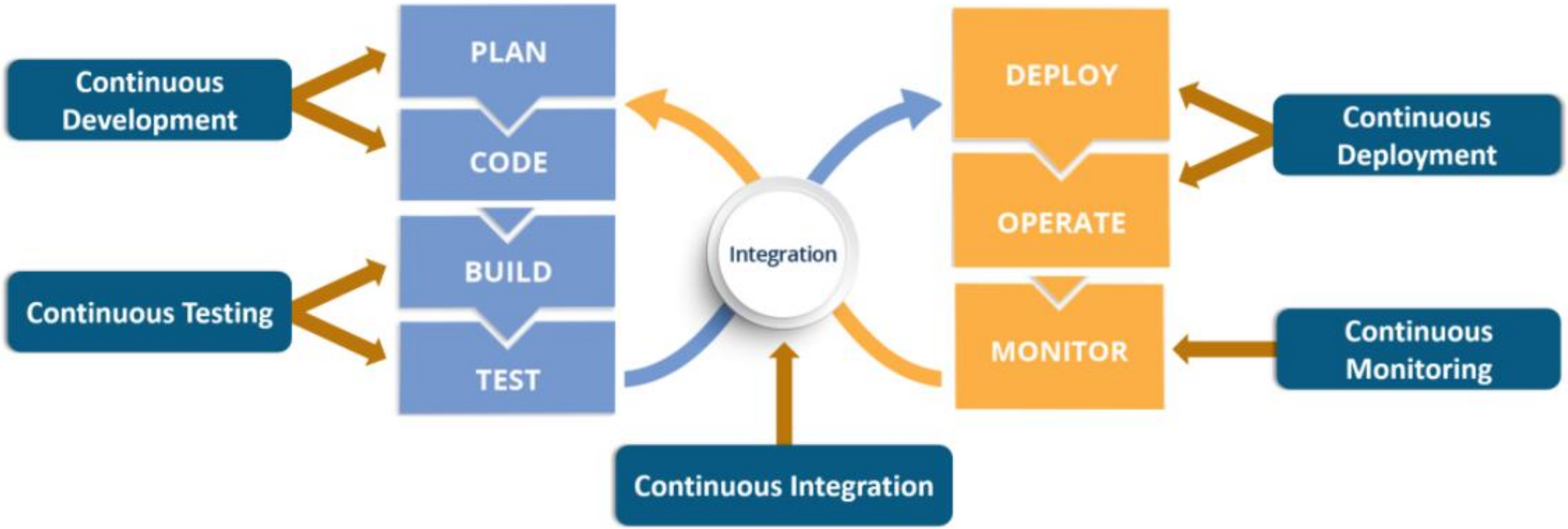
- When NOT to Adopt:
  - Mission critical applications (Banks, Power systems etc)

DEVOPS LIFECYCLE PHASE

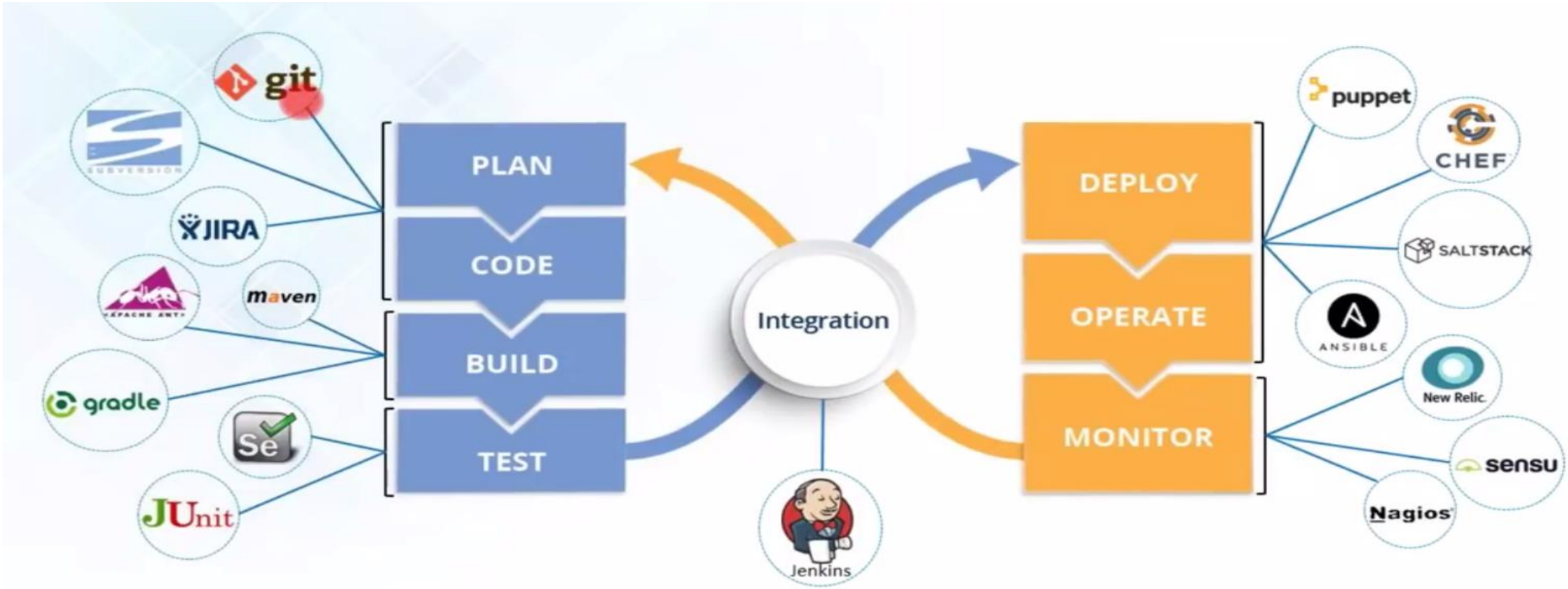


DevOps Life Cycle Phases

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



DevOps LifeCycle Phase



DevOps Tools

CONTINUOUS DEVELOPMENT

- This phase involves ‘planning‘ and ‘coding‘ of the software application’s functionality.
- The vision of the project is decided during the ‘planning’ phase.
- Code can be done in Any Language but maintain by Version Control System.

Why Code Versioning is Imp

- Versions are maintained to hold a single source of Application.
- Using Centralized single source code, Operations can access the same code what they plan to release.
- Easy to Rollout the faulty snippet of code or complete release.

CONTINUOUS TESTING

- This phase involves testing the code's of the software application's functionality.
- Testing could be done in Any Tool, Junit, Selenium…

CONTINUOUS INTEGRATION

- This is the most important phase of the DEVOPS lifecycle.
- All the Devops tools are very well integrated here.
- Scheduling of all the JOBS are done.
- Single point for the Developer and Operations teams to check on the error's and collaborate.

CONTINUOUS DEPLOYMENT

- Configuration Management
- Containerization

Configuration Management

- Establish & Maintain Application's functional requirements and Performance.
- Releasing deployments to Servers.
- Scheduling Updates on all Servers.
- Maintain Configuration Consistency on All Servers.

Containerization

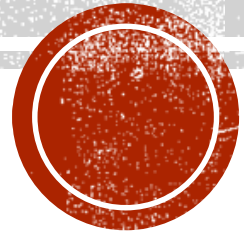
- Containerization tool is a set of Tools which will maintain the consistency across the environments.
- Docker is the First Containerization Tool.
- Scheduling Updates on all Servers.
- Maintain Configuration Consistency on All Servers.

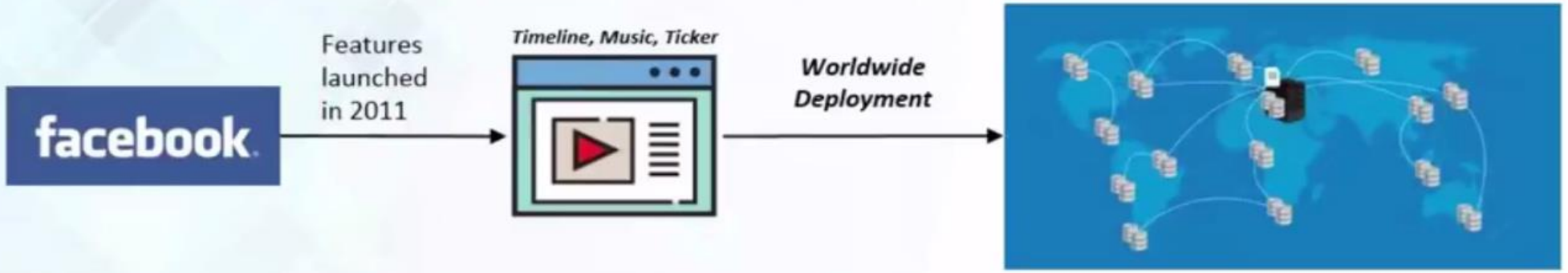
CONTINUOUS MONITORING

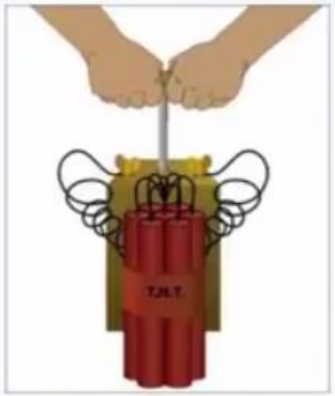
Continuous Monitoring is monitor the System Performance.

Monitor the Product overall Performance.

DEVOPS USECASE







Challenges they faced that day

- Features released to 500 million users → Heavy Website traffic → Server Meltdown
- Mixed responses from users which lead to no conclusion

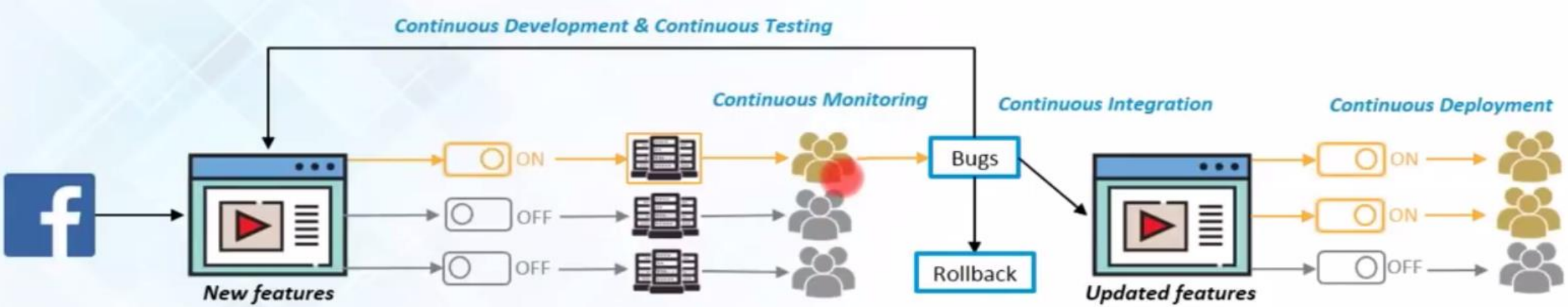
Use Case 2011 : FaceBook

Face Book Comes Up with Dark Launching

According to Dark Launching Technique:

- The new features are first deployed on a smaller & specific user base.
- They are continuously monitored and the feedbacks are continuously developed and tested.
- Once the features are stable, they are deployed on other user bases in multiple releases.

The Dark Launching Techinique



To Implement Dark launching, the below activities are fundamental as they lie at the heart of the DevOps lifecycle:

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

The Dark Launching Techinique

PIPELINE OVERVIEW