
DevOps
Introduction

OUR AGENDA

1. WHAT

1. What is DevOps ?
2. What is SDLC?
3. What is Agile?
4. Waterfall vs Agile vs DevOps
5. Evolution of DevOps

2.WHY

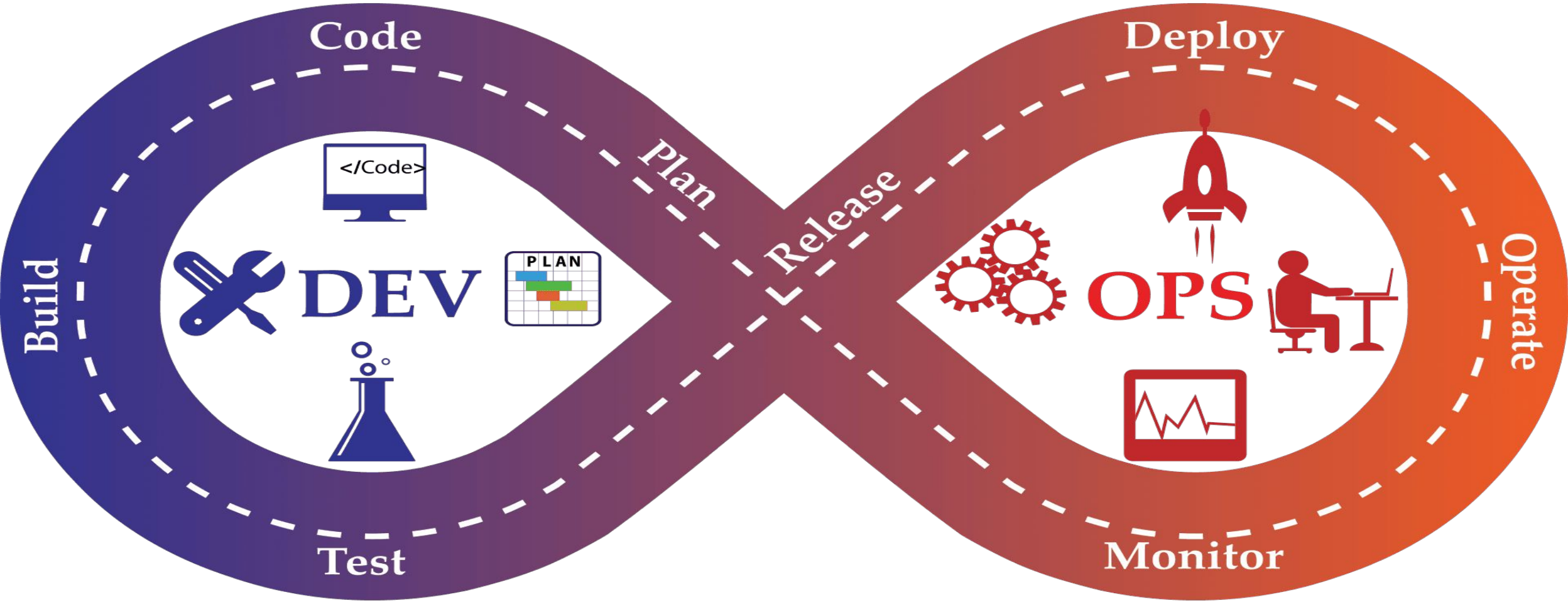
Why we need to apply Devops to projects

3.WHO

Who can learn DevOps

4.Benefits

5.Functionalities

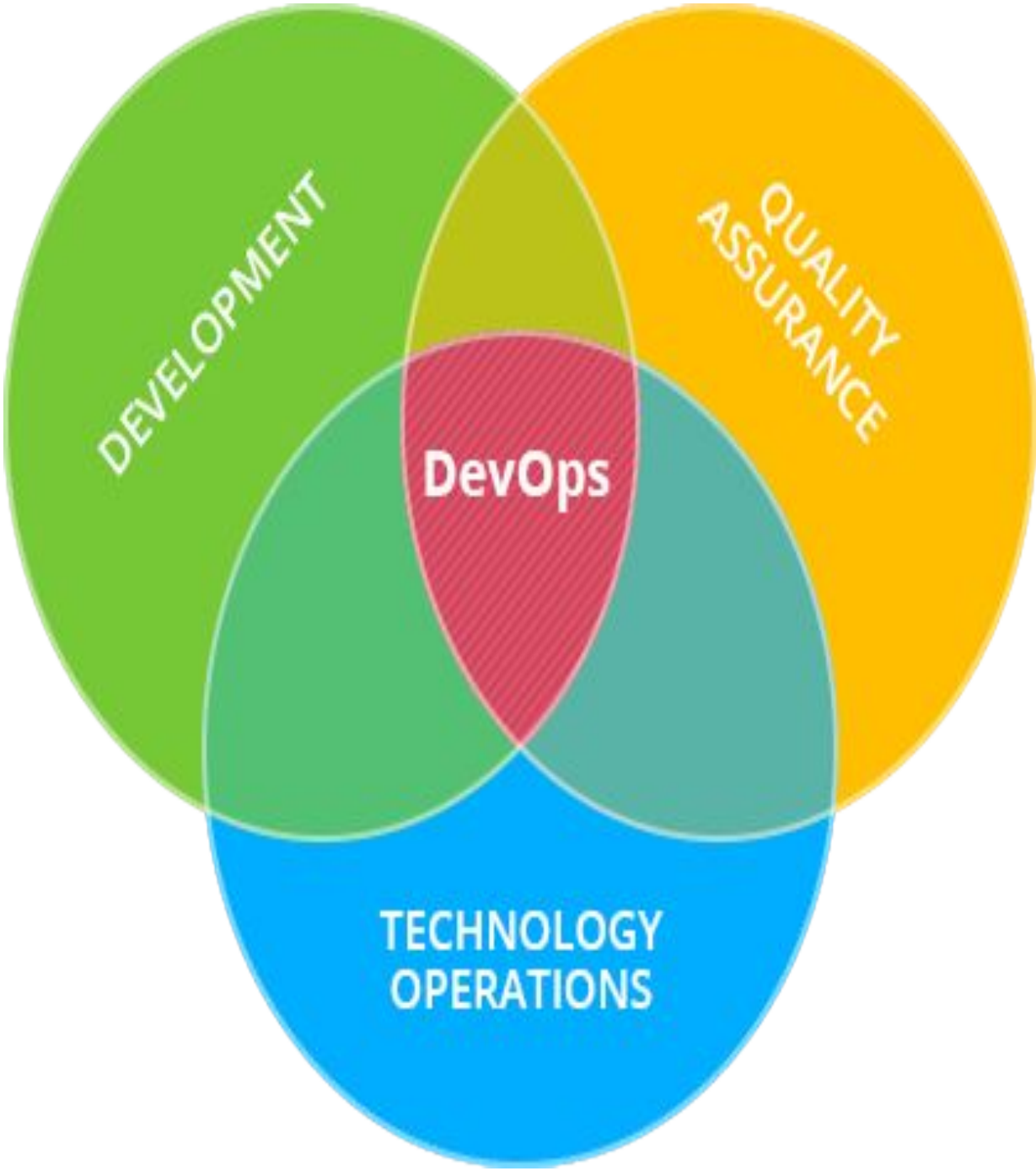

Code
Plan
Build
Test
Release
Deploy
Operate
Monitor
</Code>
DEV
PLAN
OPS

1.1 What is DevOps?

DevOps is the combination of cultural philosophies, practices, and tools that increases an organization’s ability to deliver applications and services at high velocity (Source: AMAZON)

DevOps (a clipped compound of "development" and "operations") is a software development methodology that combines software development (Dev) with information technology operations (Ops). (Source: Wiki)

DevOps is a set of practices that automates the processes between software development and IT teams, in order that they can build, test, and release software faster and more reliably. The concept of DevOps is founded on building a culture of collaboration between teams that historically functioned in relative siloes. ource: A(Stlassian)


DEVELOPMENT
QUALITY ASSURANCE
DevOps
TECHNOLOGY OPERATIONS

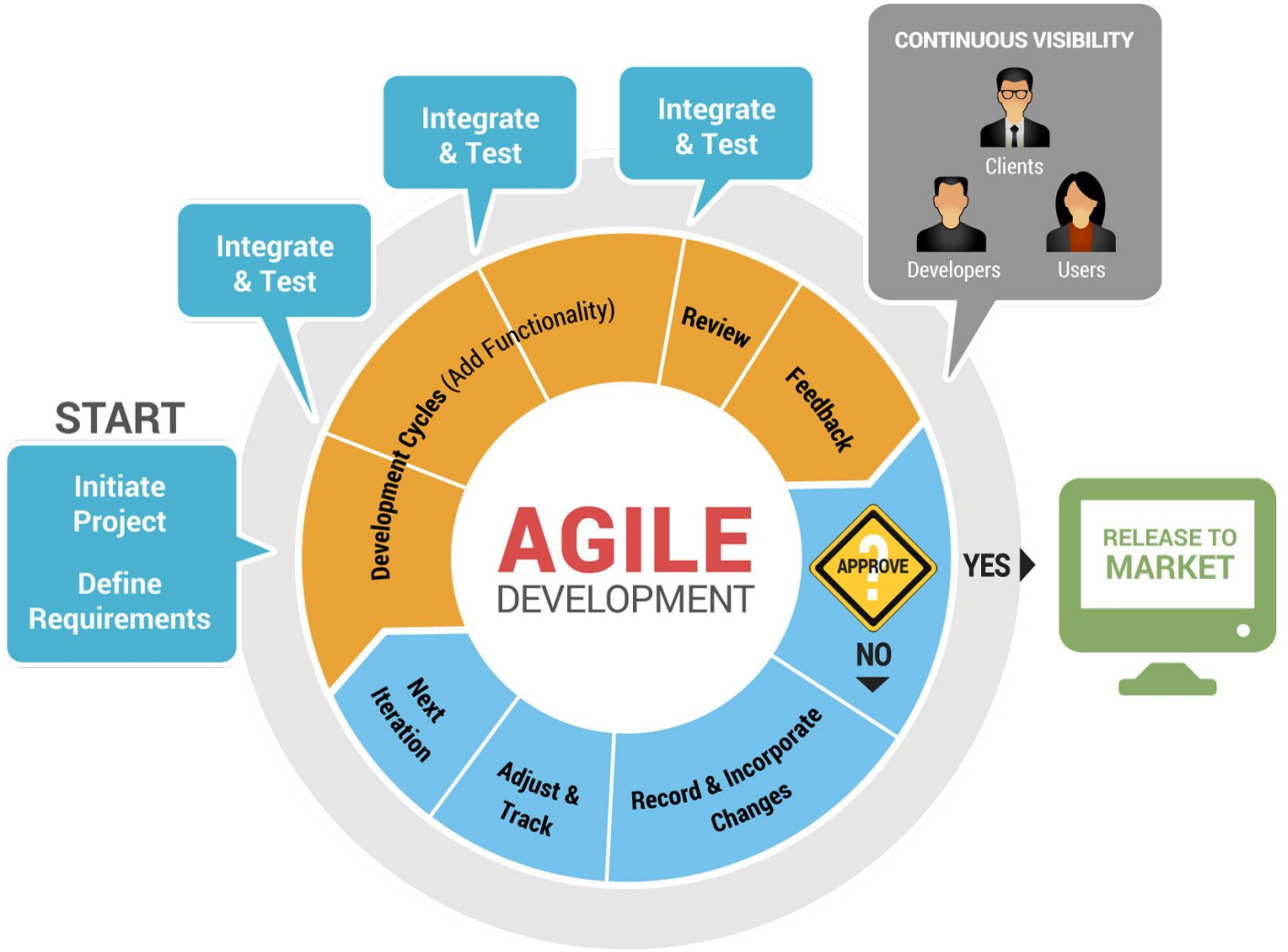

Communication
Collaboration
Integration
Automation
DEV
OPS
Agile Dev + IT Ops = DevOps!

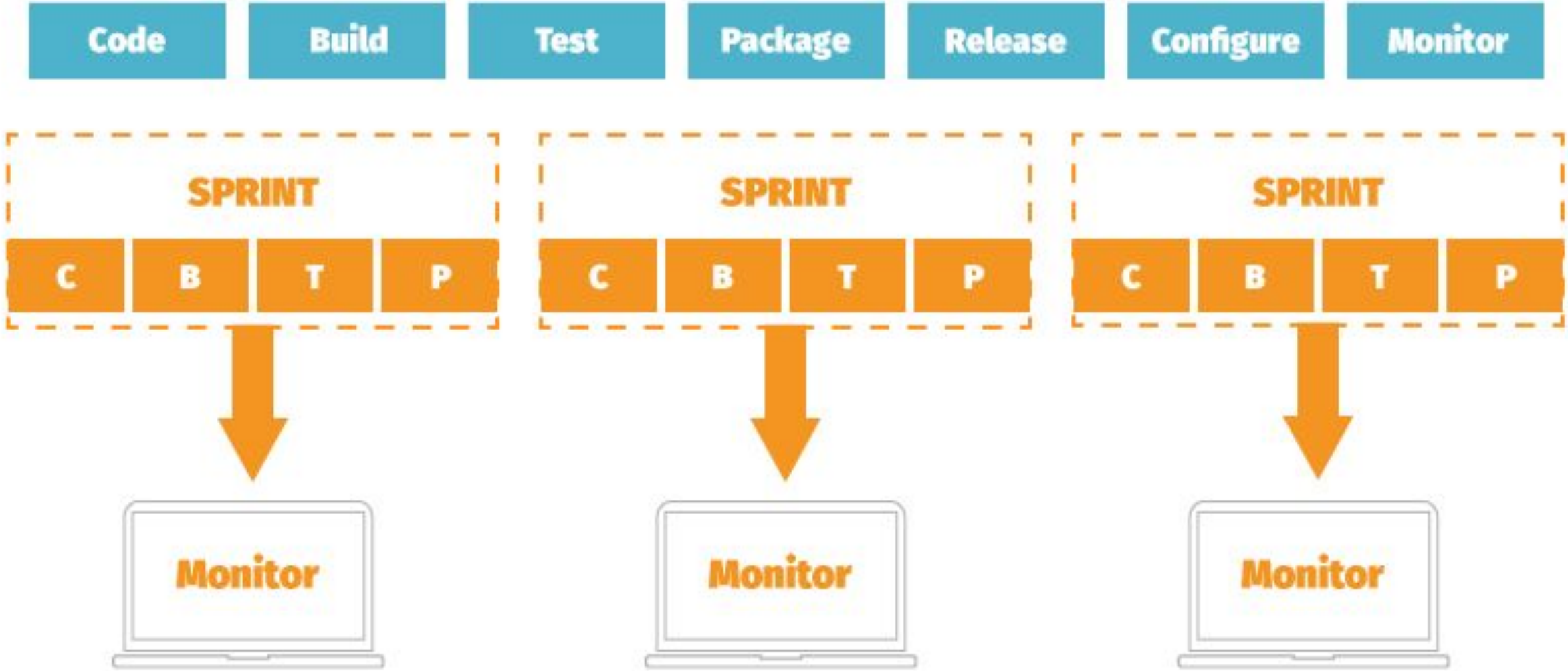
1.2 What is SDLC?

SDLC is a process followed for a software project, within a software organization. It consists of a detailed plan describing how to develop, maintain, replace and alter or enhance specific software. The life cycle defines a methodology for improving the quality of software and the overall development process.

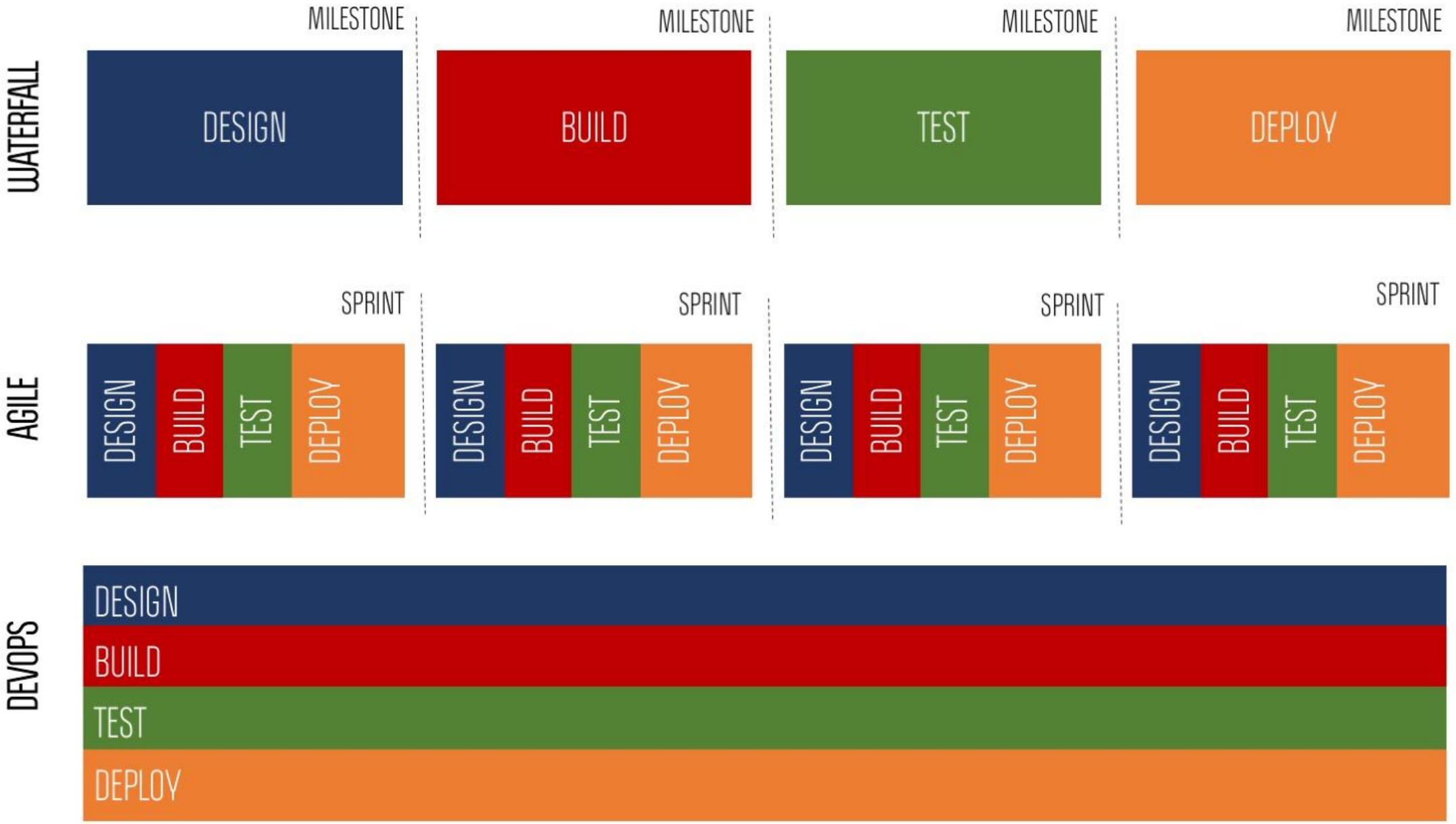
1.3 What is Agile ?

Agile software development is an approach to software development under which requirements and solutions evolve through the collaborative effort of self-organizing and cross-functional teams and their customer/end user.

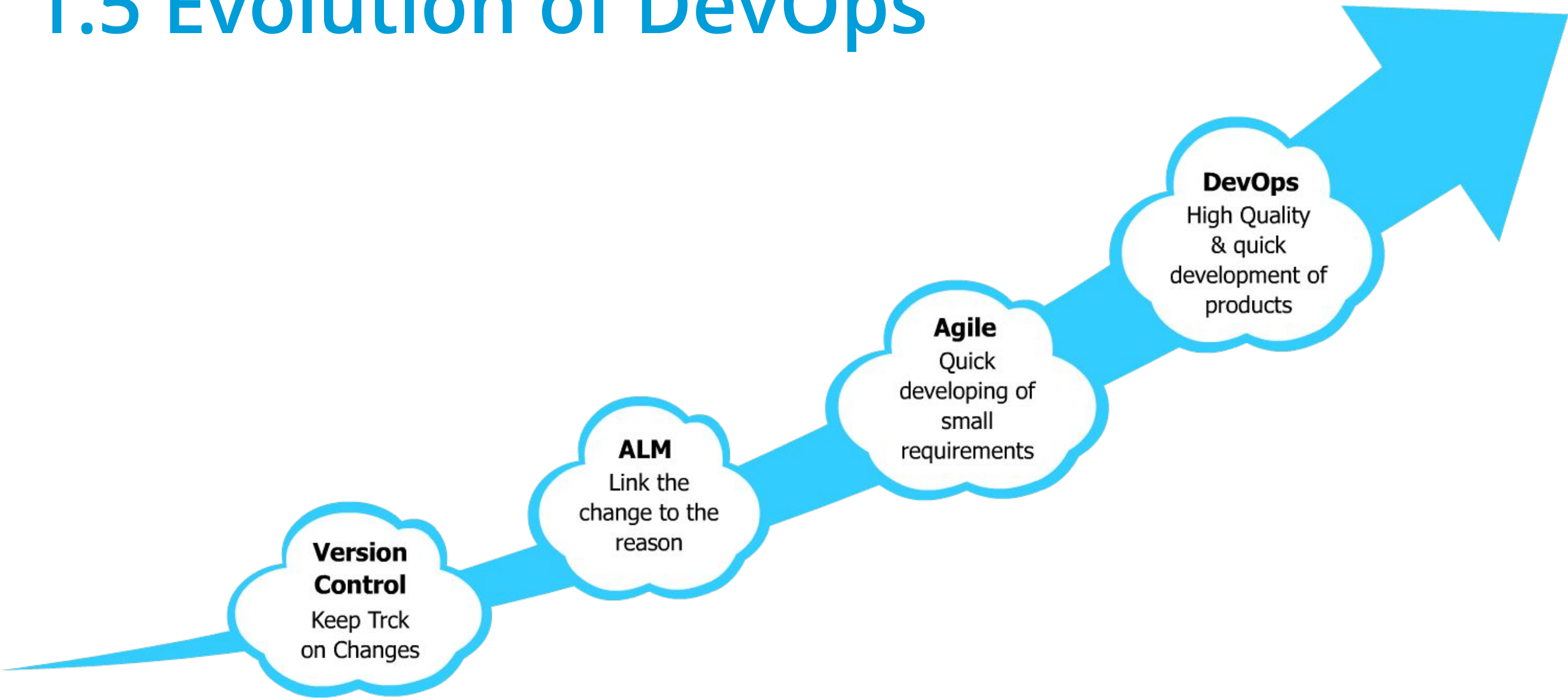



Code
Build
Test
Package
Release
Configure
Monitor
SPRINT
C
B
T
P
Monitor
SPRINT
C
B
T
P
Monitor
SPRINT
C
B
T
P
Monitor

Waterfall vs Agile Vs DevOps



1.5 Evolution of DevOps



2. Why DevOps?

1) Shorter Development Cycles, Faster Innovation

a) With a combined development and operations team, applications are ready for use much more quickly. This is important, since companies succeed based on their ability to innovate faster than their competitors do. In fact, Kevin Murphy from Red Hat estimates that shorter development cycles translate to bringing an application to market 60 percent faster than with traditional approaches.

2) Reduced Deployment Failures, Rollbacks, and Time to Recover

a) Time to recover is an important issue, because some failure has to be expected. But recovery is much faster when the development and operations teams have been working together, exchanging ideas and accounting for both teams' challenges during development.

3) Improved Communication and Collaboration

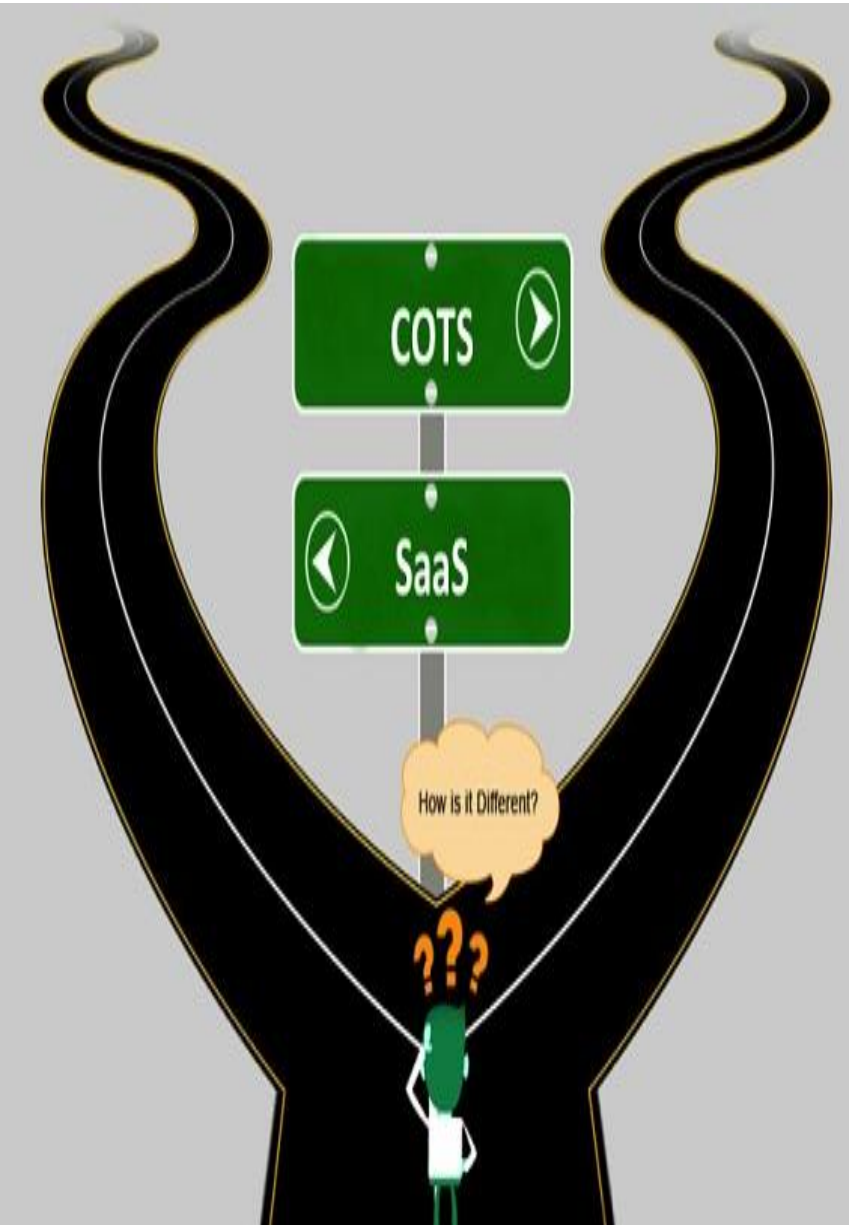
a) DevOps improves the software development culture. Combined teams are happier and more productive. The culture becomes focused on performance rather than individual goals. When the teams trust each other, they can experiment and innovate more effectively. The teams can focus on getting the product to market or into production, and their KPIs should be structured accordingly.

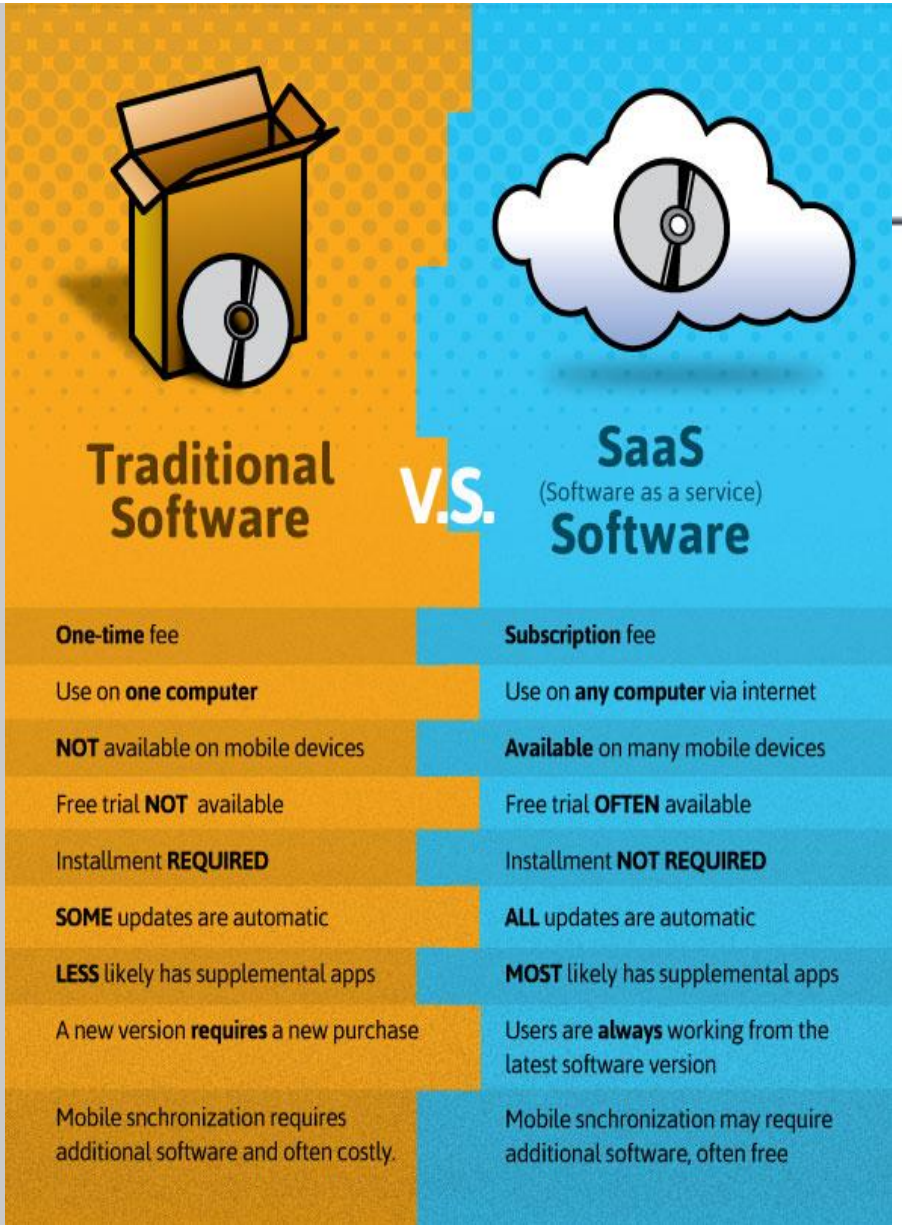
4) Increased Efficiencies

a) Scalable infrastructures, such as cloud-based platforms, increase the access the team has to hardware resources. As a result, testing and deployment operations speed up.
b) Build acceleration tools can be used to compile code more quickly.
c) Parallel workflows can be embedded into the continuous delivery chain to avoid delays; one team waits for another to complete its work.
d) Using one environment avoids the useless task of transferring data between environments. This means you don't have to use one environment for development, a different environment for testing, and a third for deployment.

5) Reduced Costs and IT Headcount

a) All of the DevOps benefits translate to reduced overall costs and IT headcount requirements. According to Kevin Murphy from Red Hat, DevOps development teams require 35 percent less IT staff and 30 percent lower IT costs.





Some (Extreme) Examples

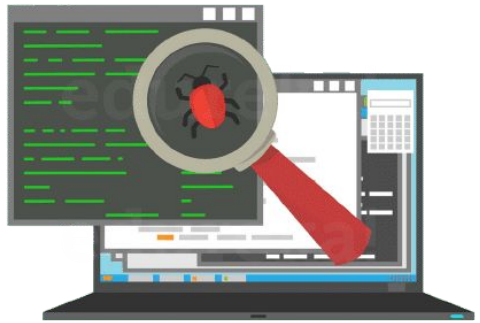
flickr — 10+ deployments per day

Etsy — 25+ deployments per day

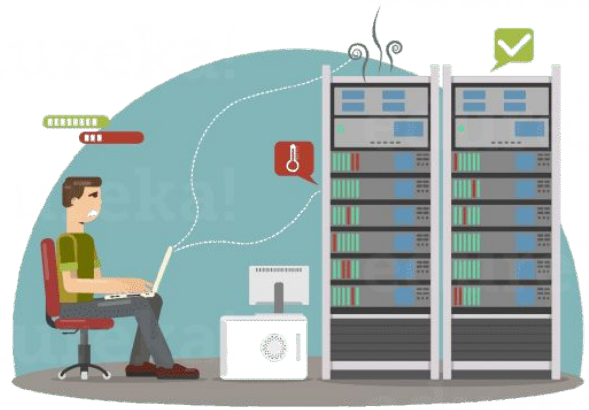
amazon.com — Deploy every 11.6 seconds

3. Who can learn DevOps

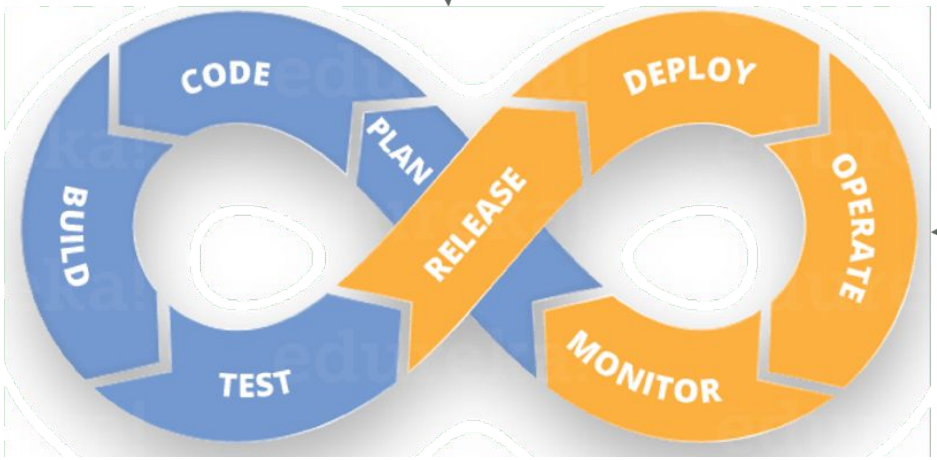
Developer



Tester



Admin



4.Benefits

Container devops benefits

Agility	Consistency	Utility
• Containers are for applications	• Application packaging with dependencies	• Containers are fast to start than VMs
• Accelerate application development and deployment	• Increased consistency between environments due to Images	• Use containers over for short-living VMs (Integration Testing)
• Deploy anywhere	• Leverage images by vibrant community	• Support microservice architectures
• Build, ship, run	• Better utilize resources with increased density	

5.DevOps Functionalities

- Source Code Management
- CI/CD
  - Build Tools
  - Code Coverage
  - Repo Management
- Configuration Management
- Container Management

Functionality	Tools	Aws Services	Azure Services
SCM(V.C)	GIT,Mercurial,Bitbucket	Code Commit	Azure Repo
CI/CD	Jenkins,GITLAB	Code Pipelines	Azure Pipelines
Build	Maven, Gradle	Code Build	Azure Artifacts
Repo Management	Nexus OSS, JFrog	S3 Buckets	Blob Storage
Configuration Management	Puppet, Ansible, Chef	CloudFormation	
Containers	Docker	ECS	ACS