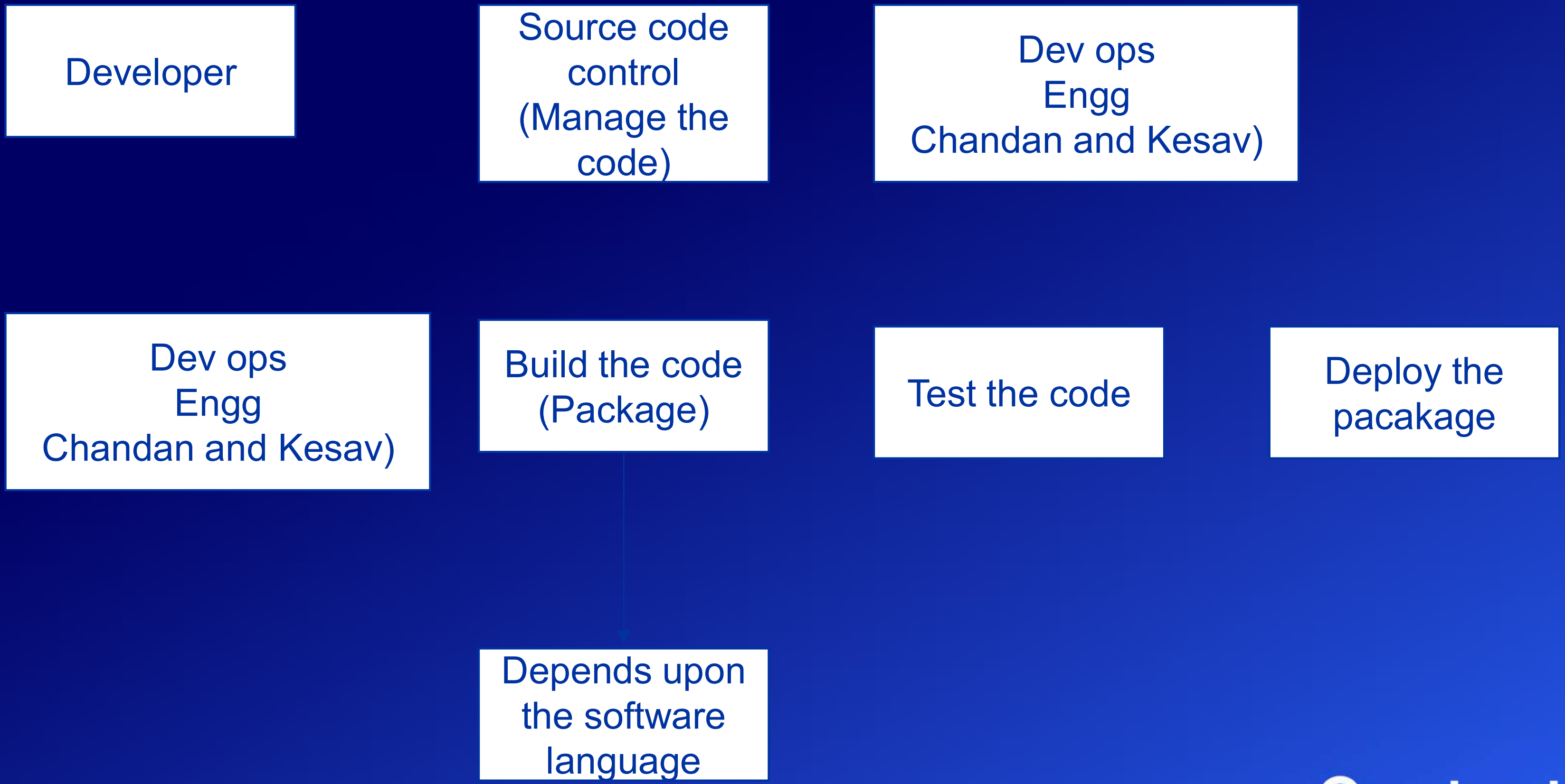
Cognizant Academy

DevOps Session

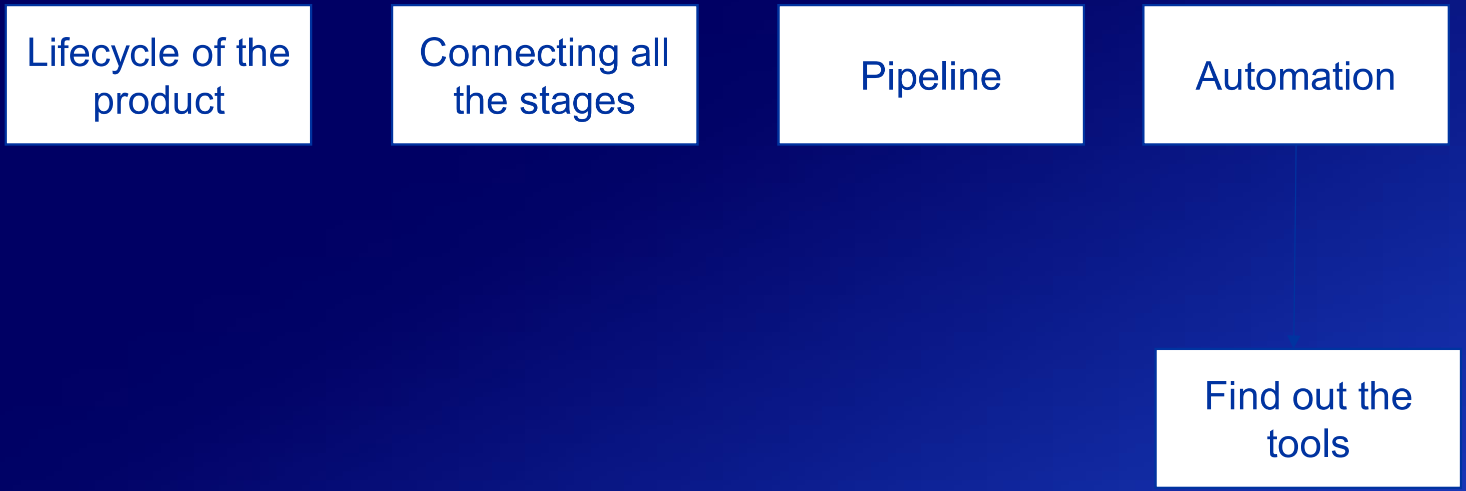
Sep 2024

Software Lifecycle

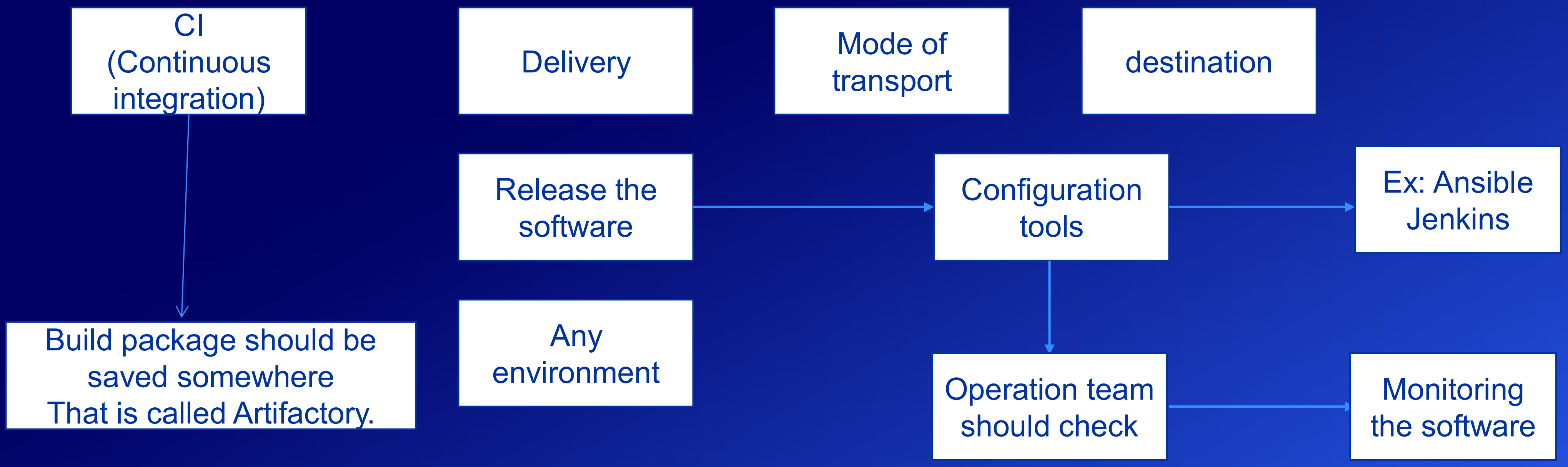

Developer
Source code control (Manage the code)
Dev ops Engg Chandan and Kesav)
Dev ops Engg Chandan and Kesav)
Build the code (Package)
Test the code
Deploy the pacakage
Depends upon the software language

Cognizant Academy

Automation


Lifecycle of the product
Connecting all the stages
Pipeline
Automation
Find out the tools

CICD



Cognizant Academy

Server - Infrastructure

On-Prem

Application servers
Web servers
Database servers

Cloud

Cloud will have all infrastructure
All the services available

Version control

Lost code
Multiple people are working and pushing their code into central code repo
Merge all the required code
Managing versions
Restoring version whenever it is needed

Branching startegy
(Devops create a branches for different project teams)
    Master or main branch – Which are managed by Devops engg

Distributed version control (DVCS)

Version control

- Working directory
  (We have to decide which project directory are we going to work for our project)
- Staging area
- Repository (Git Repo)

- git add – Pushing our changes to staging area
- ex: git add sample.txt

- Initializing the repository
- git init

Cognizant Academy

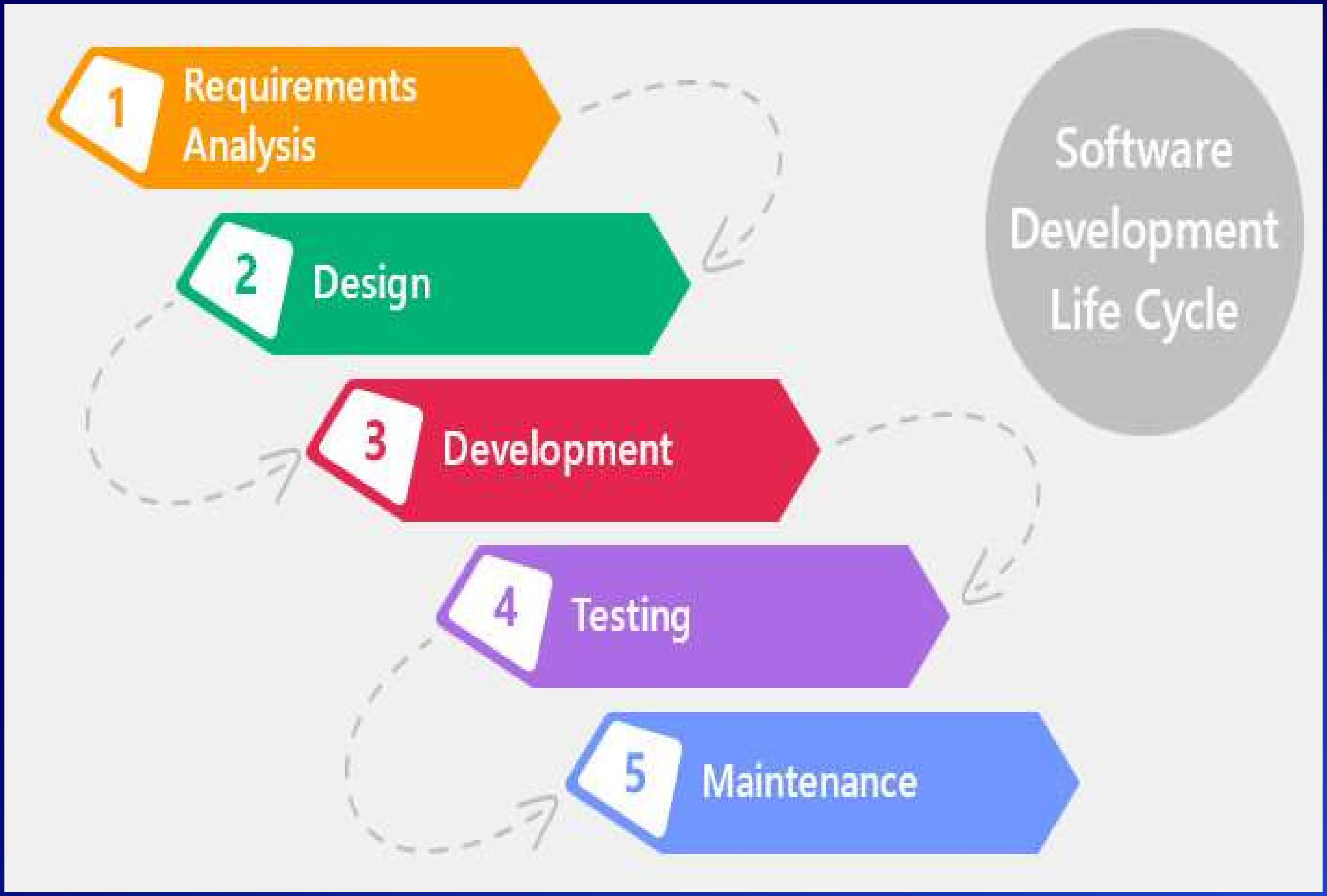
Version control

```
git init
git status
touch <<file_name>> - create a file
git add .
git commit
git status
git log
git show <<commit number>>

stash:
```

Cognizant Academy

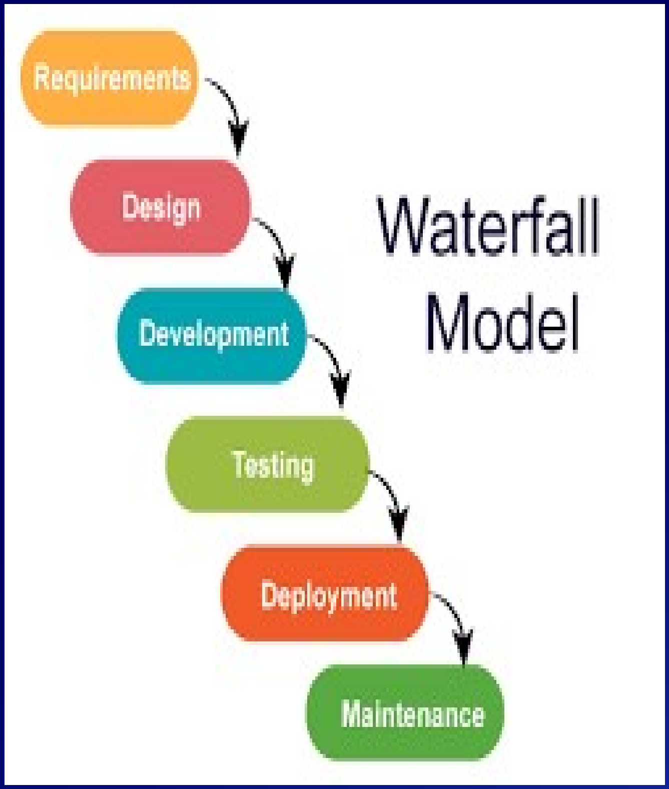
Software Lifecycle

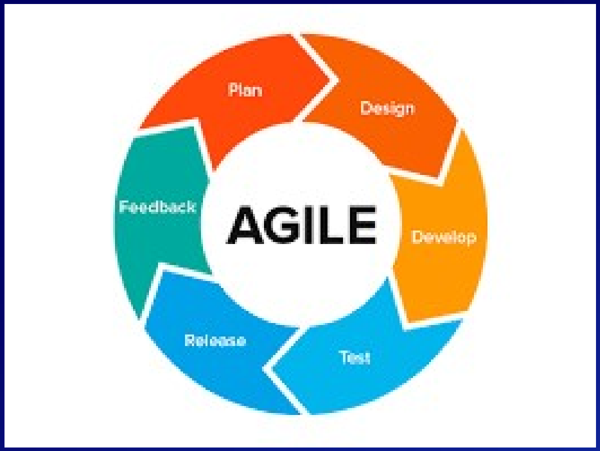


Ex: Prime video
Product Planning
Design
Developer tools (dev team)
Different Environments
Release manager (Discussion)

Cognizant Academy

Waterfall vs Agile

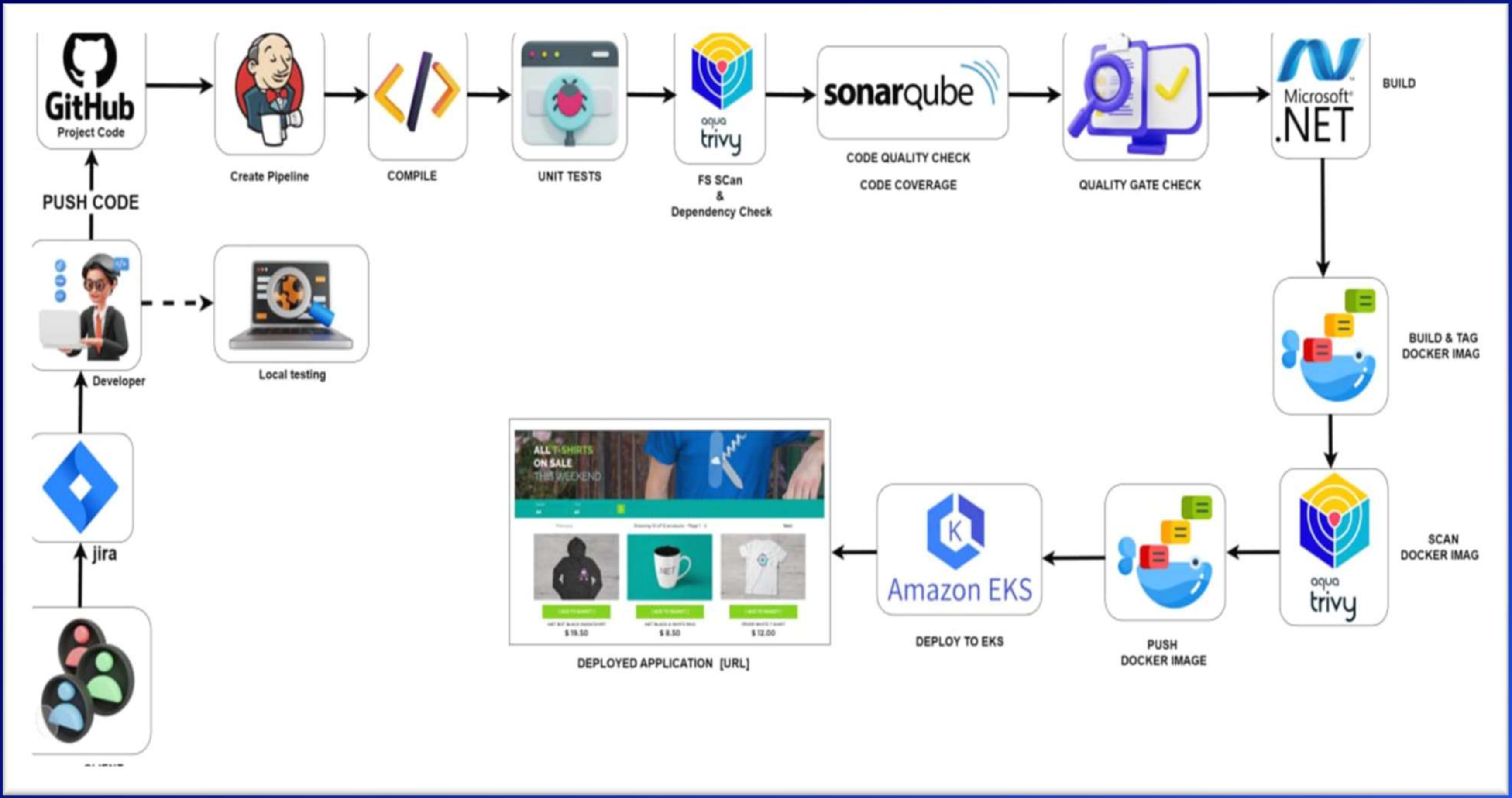




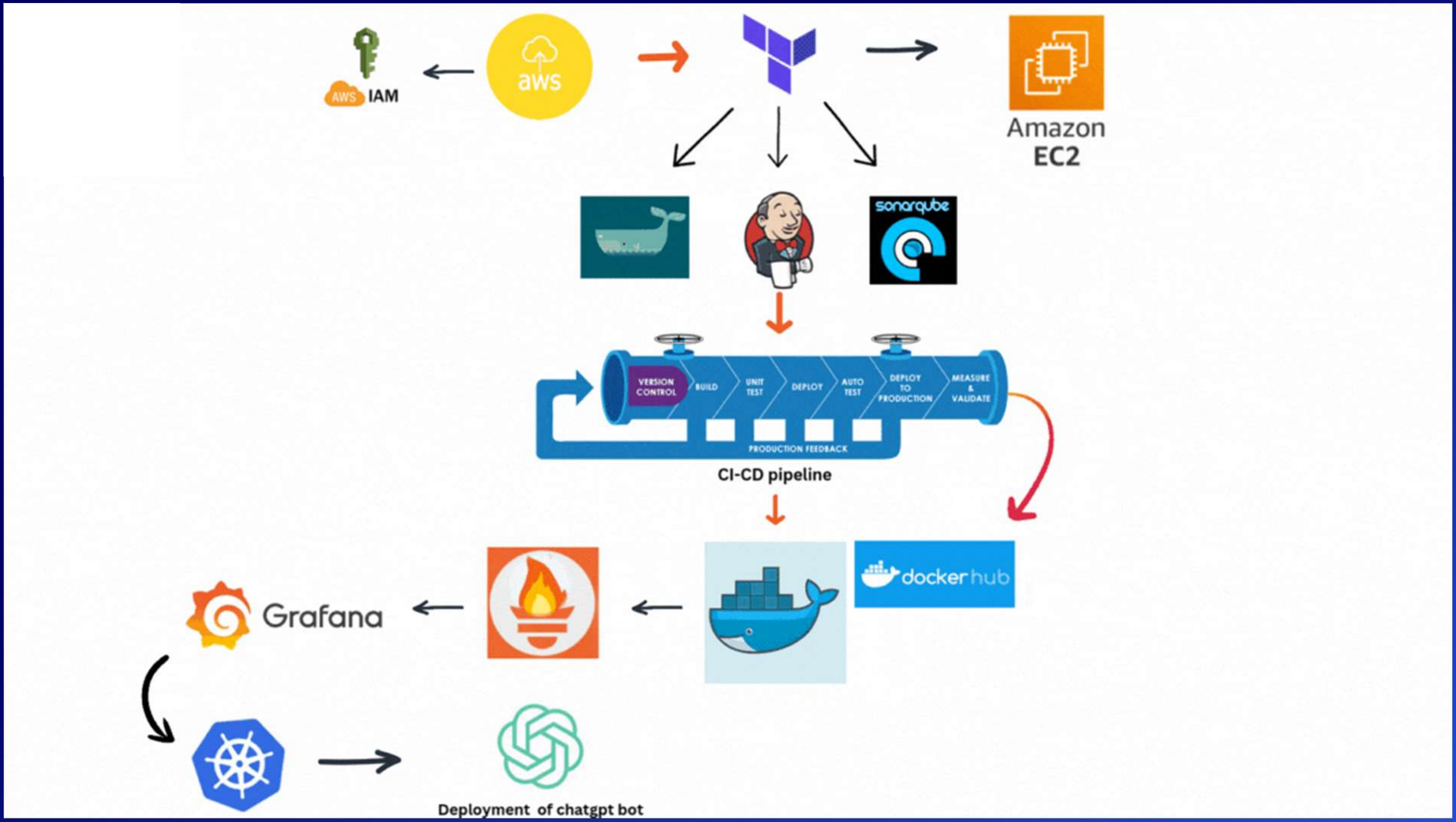
Waterfall
❖ No life cycle concept in waterfall
❖ For small scale projects we implement waterfall method.
Agile
❖ Long term product
❖ Continuous cycles
❖ Customer involvement

Cognizant Academy

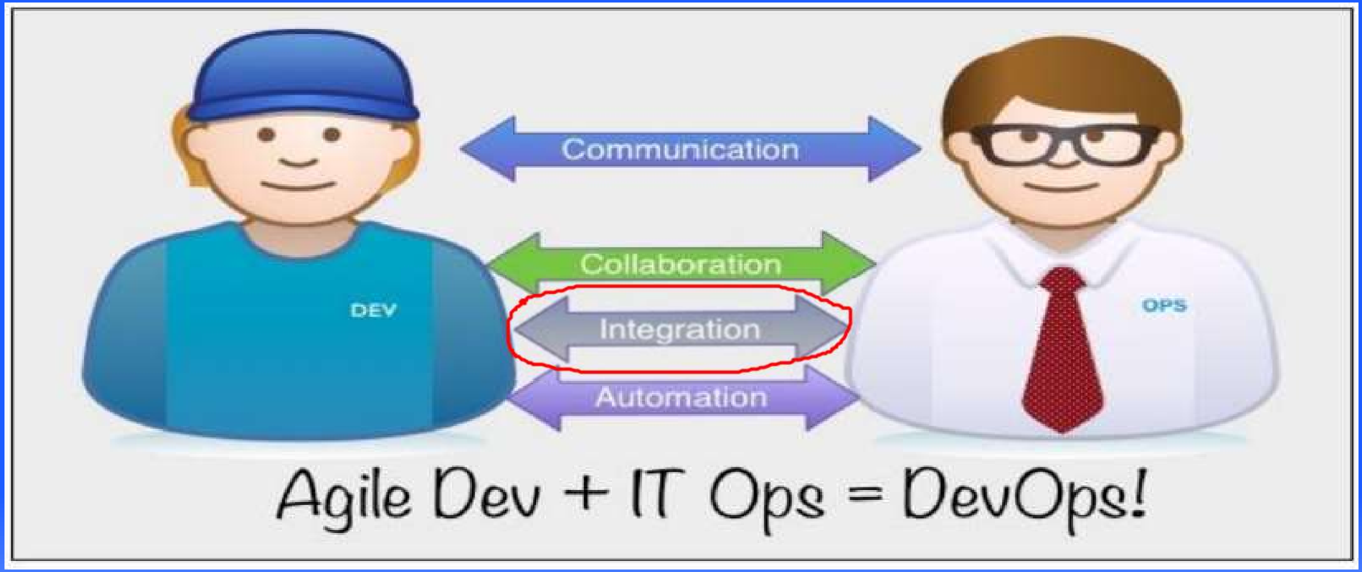
Phases



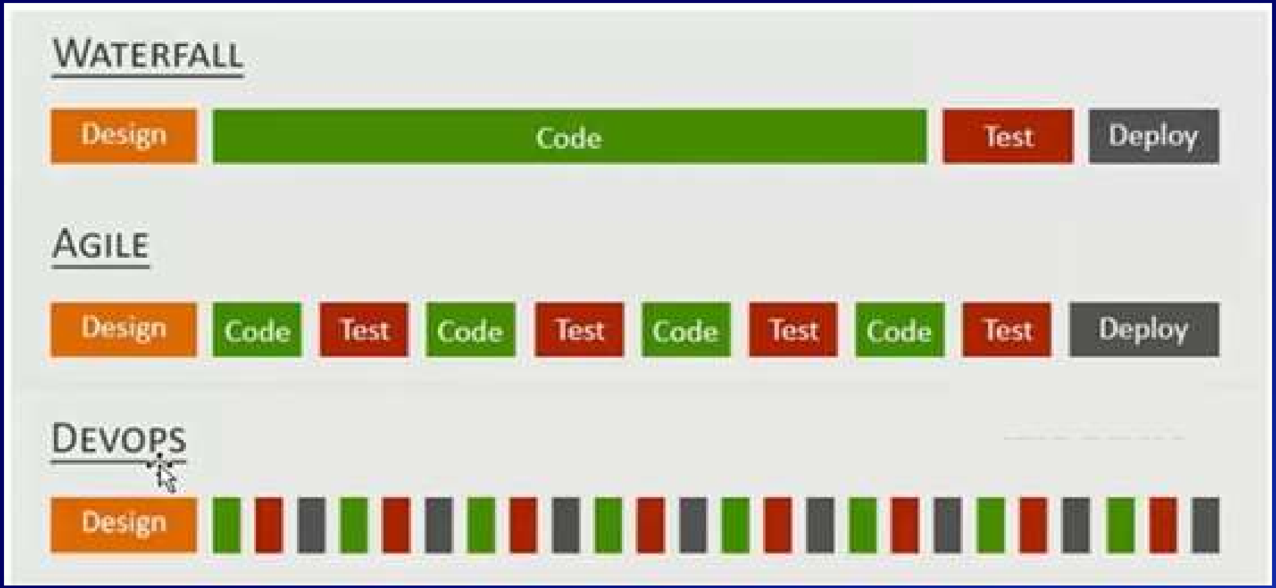
GitHub
Project Code
Create Pipeline
COMPILE
UNIT TESTS
aqua trivy
FS SCan & Dependency Check
sonarqube
CODE QUALITY CHECK
CODE COVERAGE
QUALITY GATE CHECK
Microsoft .NET
BUILD
PUSH CODE
Developer
Local testing
BUILD & TAG DOCKER IMAG
SCAN DOCKER IMAG
aqua trivy
PUSH DOCKER IMAGE
Amazon EKS
DEPLOY TO EKS
ALL T-SHIRTS ON SALE THIS WEEKEND
DEPLOYED APPLICATION [URL]
jira


AWS IAM
aws
Amazon EC2
sonarqube
VERSION CONTROL
BUILD
UNIT TEST
DEPLOY
AUTO TEST
DEPLOY TO PRODUCTION
MEASURE & VALIDATE
PRODUCTION FEEDBACK
CI-CD pipeline
docker hub
Grafana
Deployment of chatgpt bot

Devops Overview



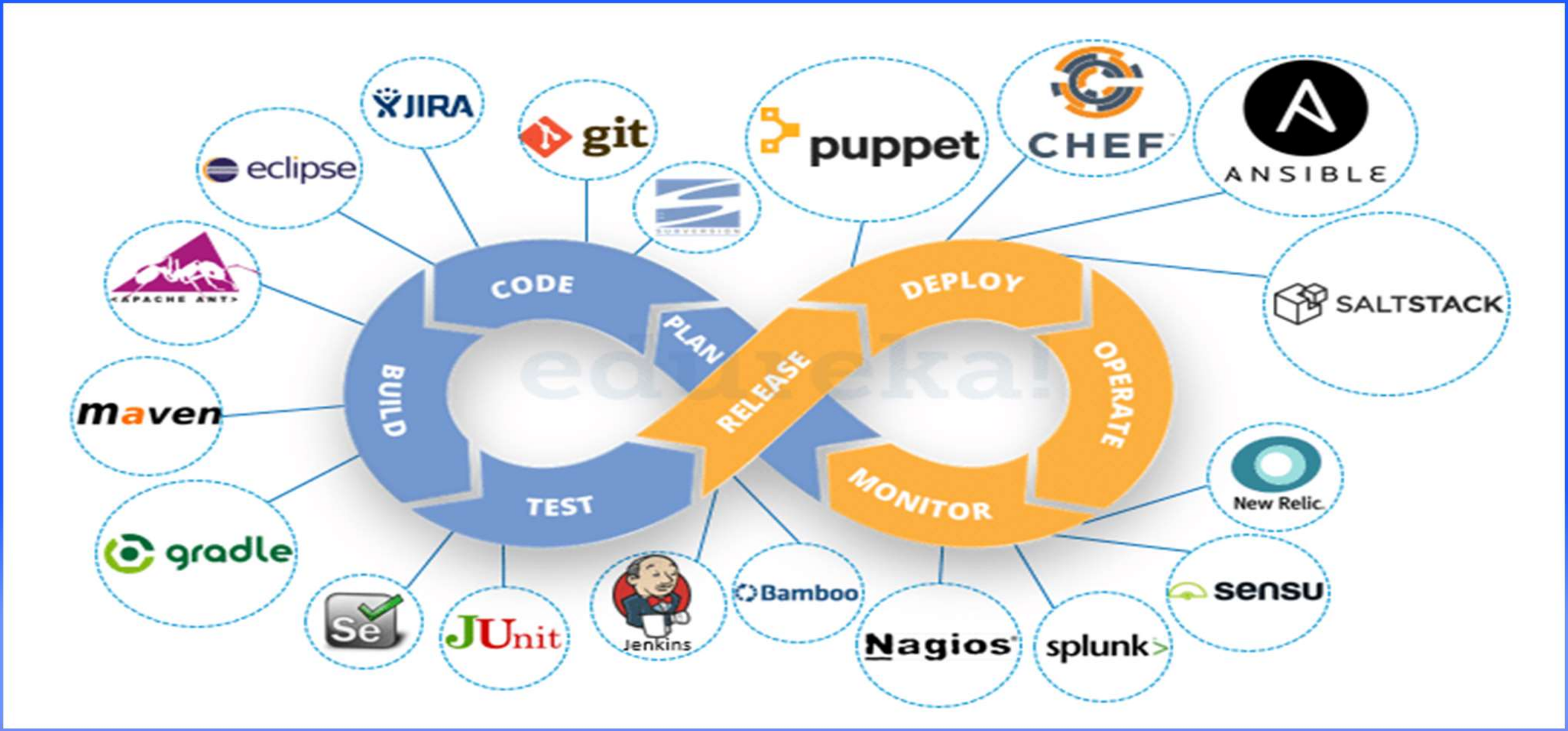
Why devops?



How DevOps differ from SDLC?
❖ Frequent releases ❖ Through Devops we achieve faster releases
❖ Speed-up our development process
Co-ordinate with development team
❖ Automation process
❖ Integrate with release management
❖ Increase quality and efficiency
Tools and technologies

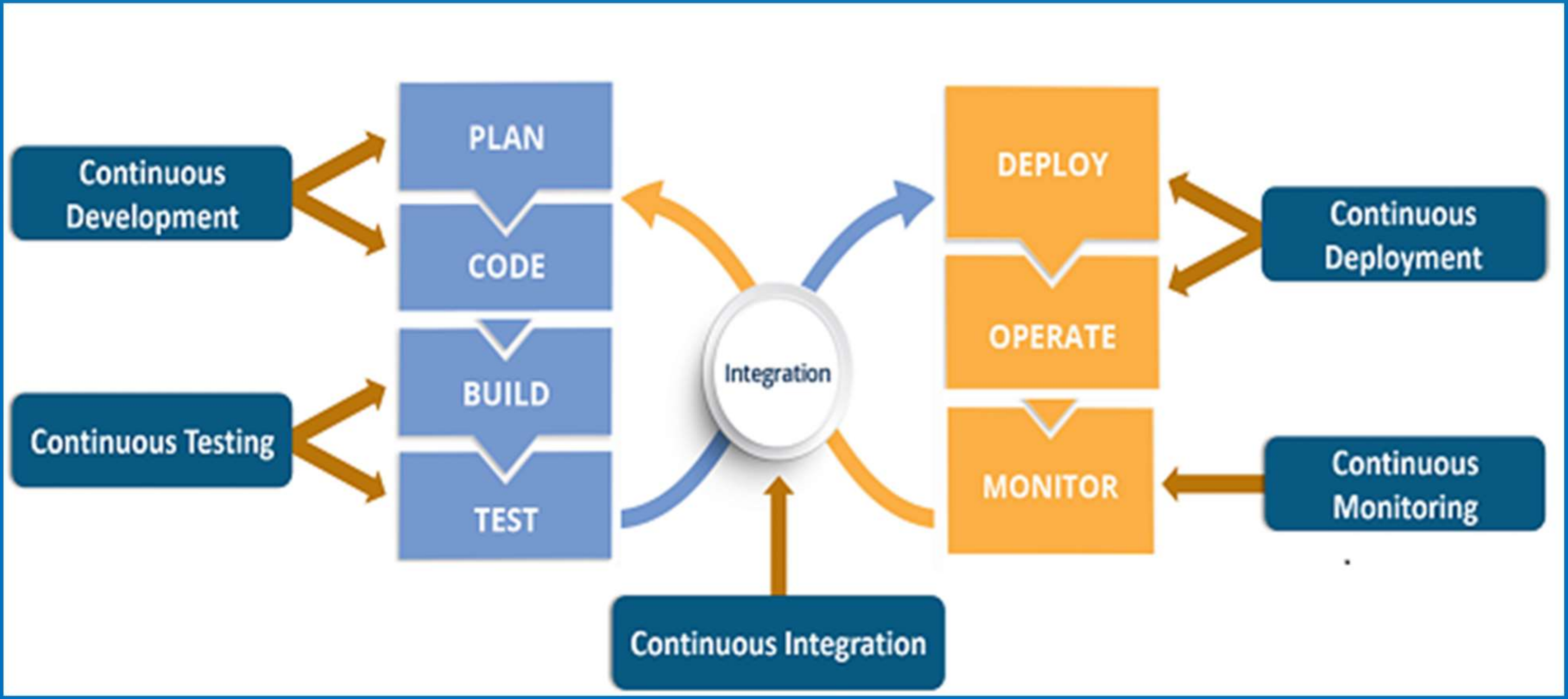
Cognizant Academy

Devops Tools Overview



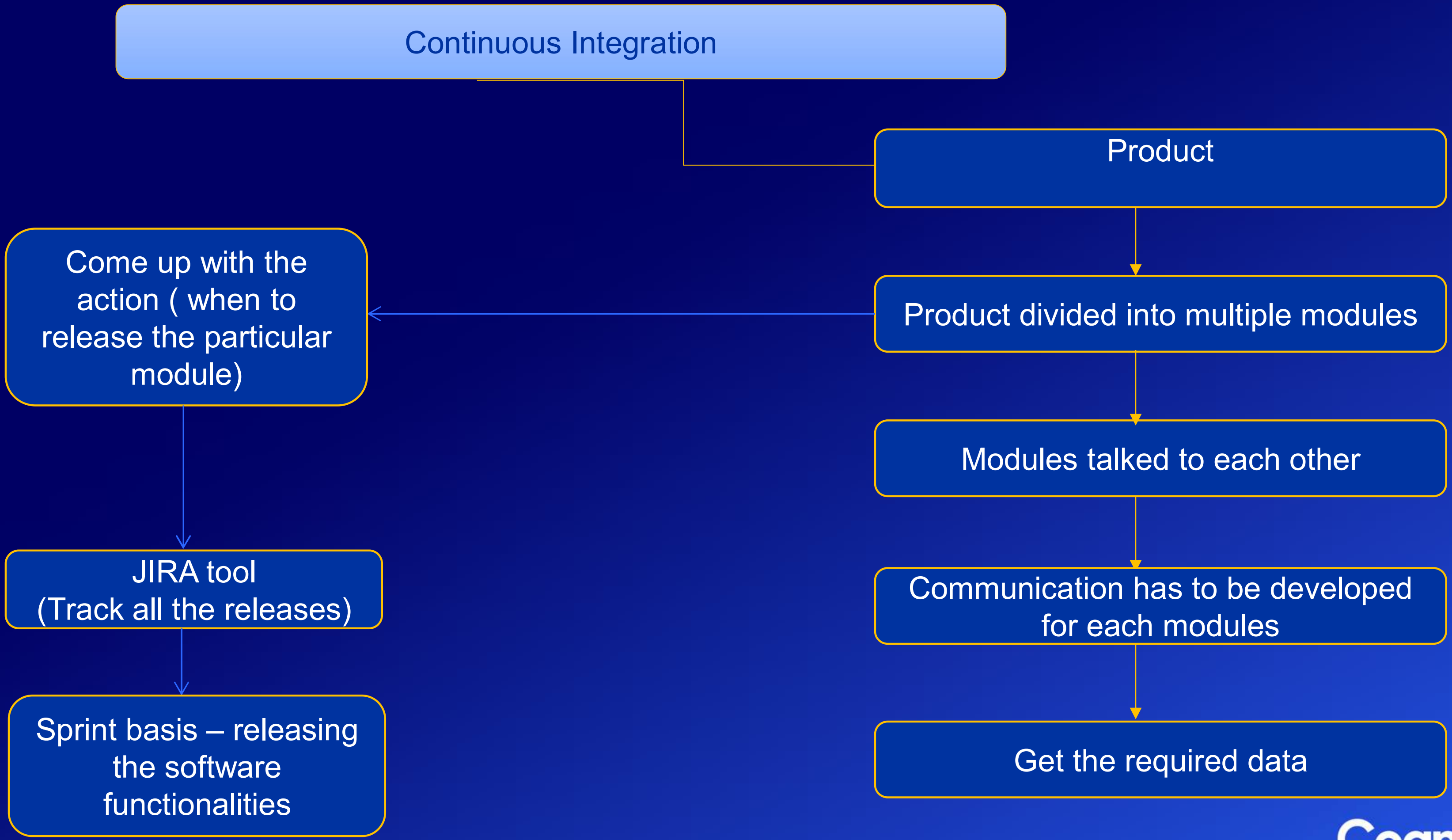
Cognizant

Devops Lifecycle – Development vs Operation



Cognizant

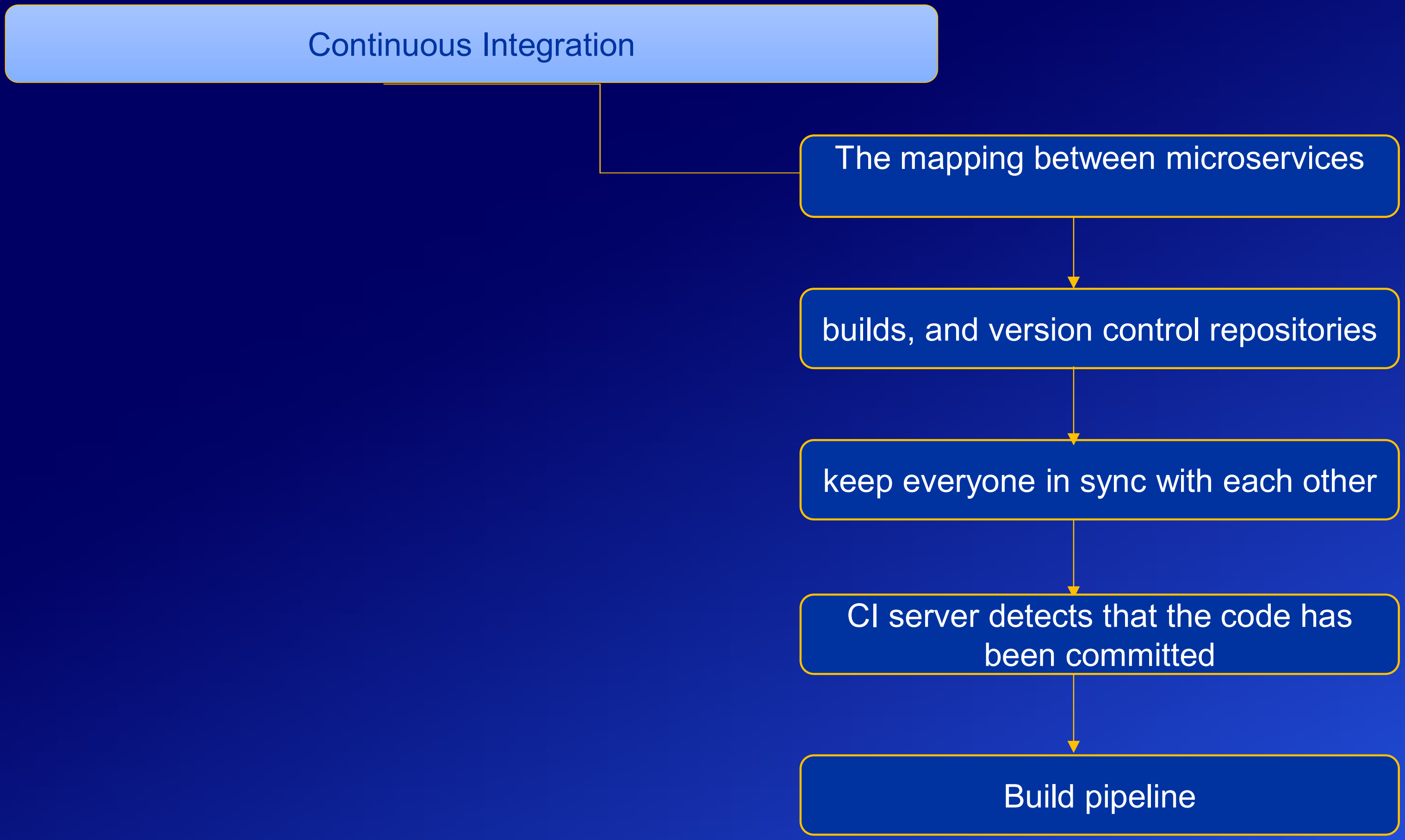
Introduction Continuous Integration

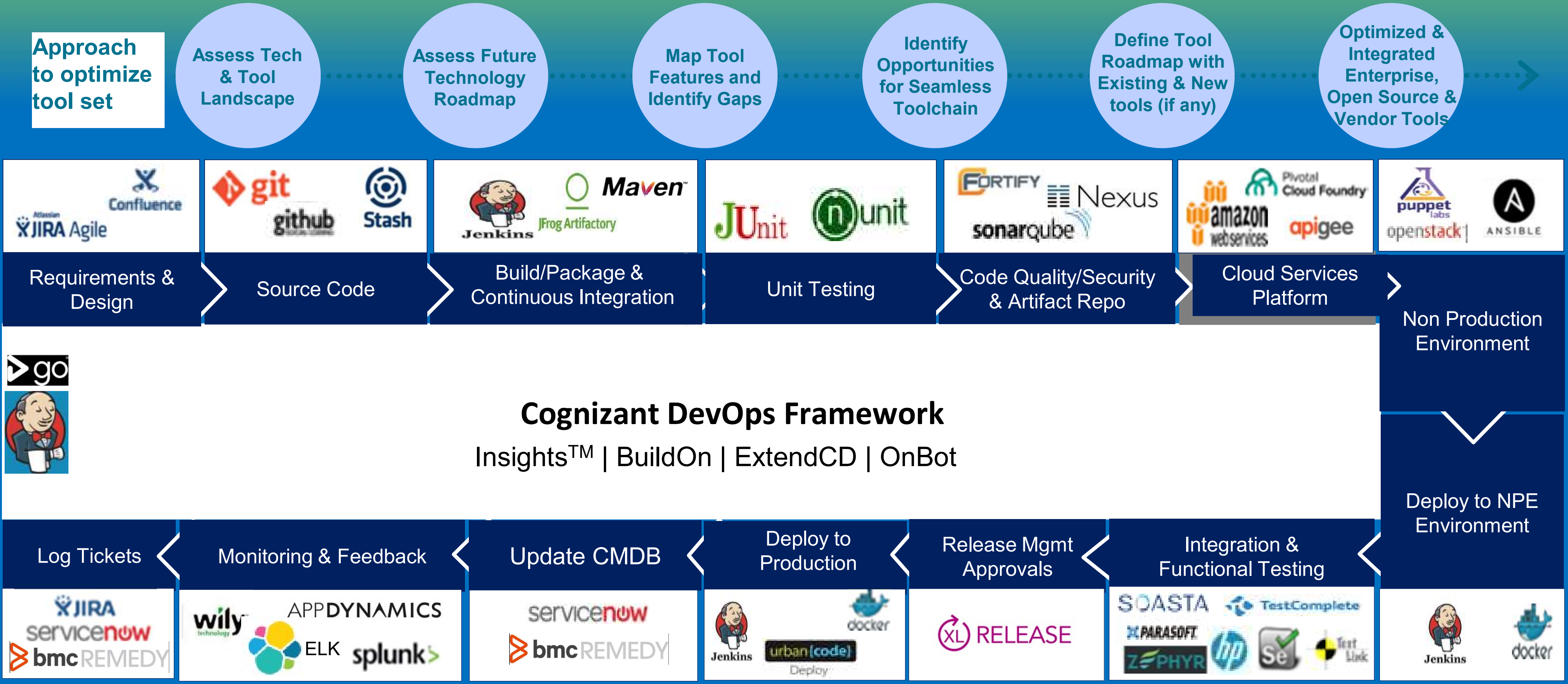


Continuous Integration
Product
Product divided into multiple modules
Modules talked to each other
Communication has to be developed for each modules
Get the required data
Come up with the action (when to release the particular module)
JIRA tool (Track all the releases)
Sprint basis – releasing the software functionalities

Cognizant Academy

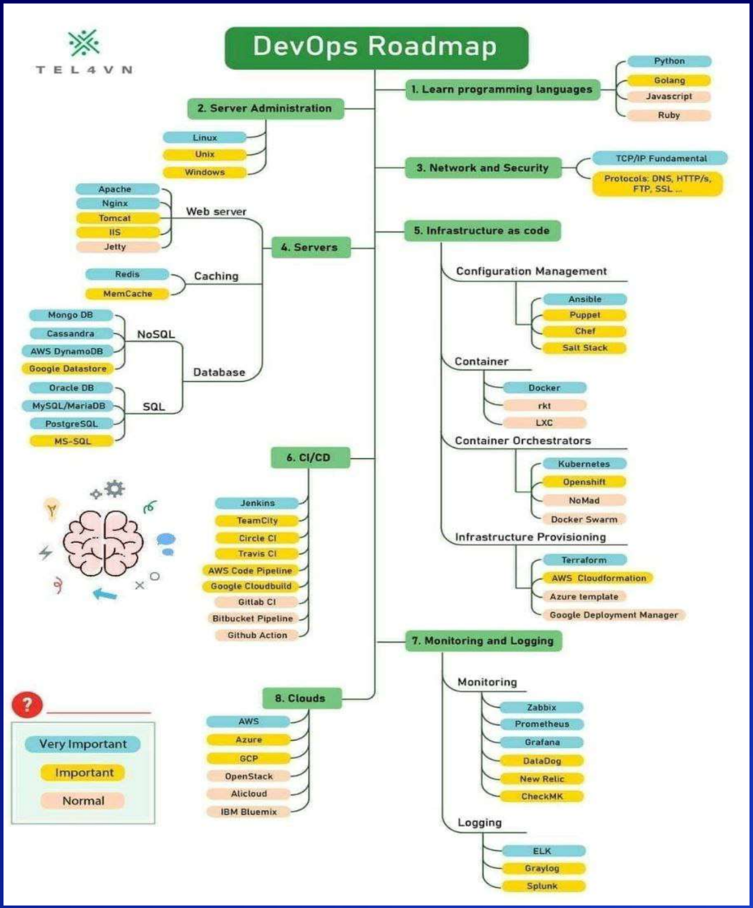
Introduction Continuous Integration


Continuous Integration
The mapping between microservices
builds, and version control repositories
keep everyone in sync with each other
CI server detects that the code has been committed
Build pipeline


Approach to optimize tool set
Assess Tech & Tool Landscape
Assess Future Technology Roadmap
Map Tool Features and Identify Gaps
Identify Opportunities for Seamless Toolchain
Define Tool Roadmap with Existing & New tools (if any)
Optimized & Integrated Enterprise, Open Source & Vendor Tools
Requirements & Design
Source Code
Build/Package & Continuous Integration
Unit Testing
Code Quality/Security & Artifact Repo
Cloud Services Platform
Non Production Environment
Deploy to NPE Environment
Integration & Functional Testing
Release Mgmt Approvals
Deploy to Production
Update CMDB
Monitoring & Feedback
Log Tickets
Cognizant DevOps Framework
Insights™ | BuildOn | ExtendCD | OnBot

Cognizant

Devops Life Cycle

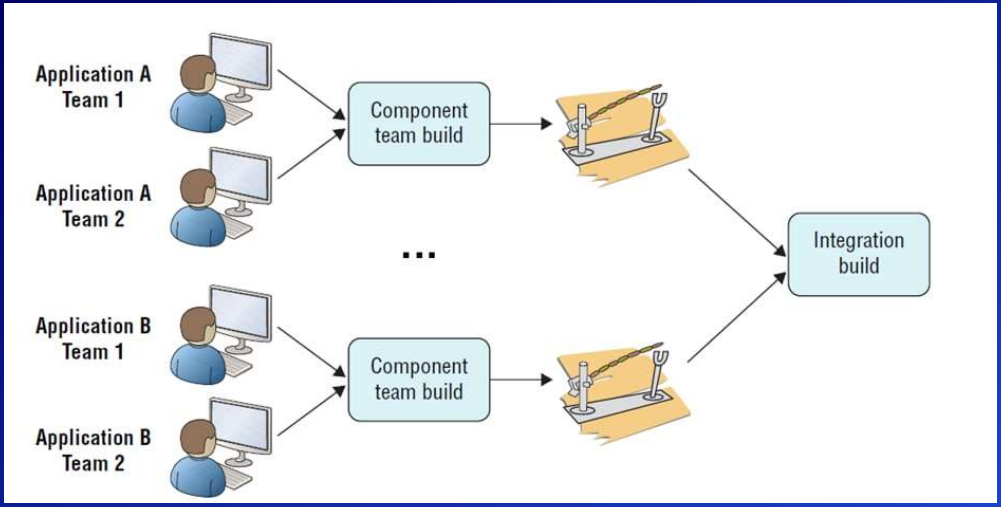


Cognizant Academy

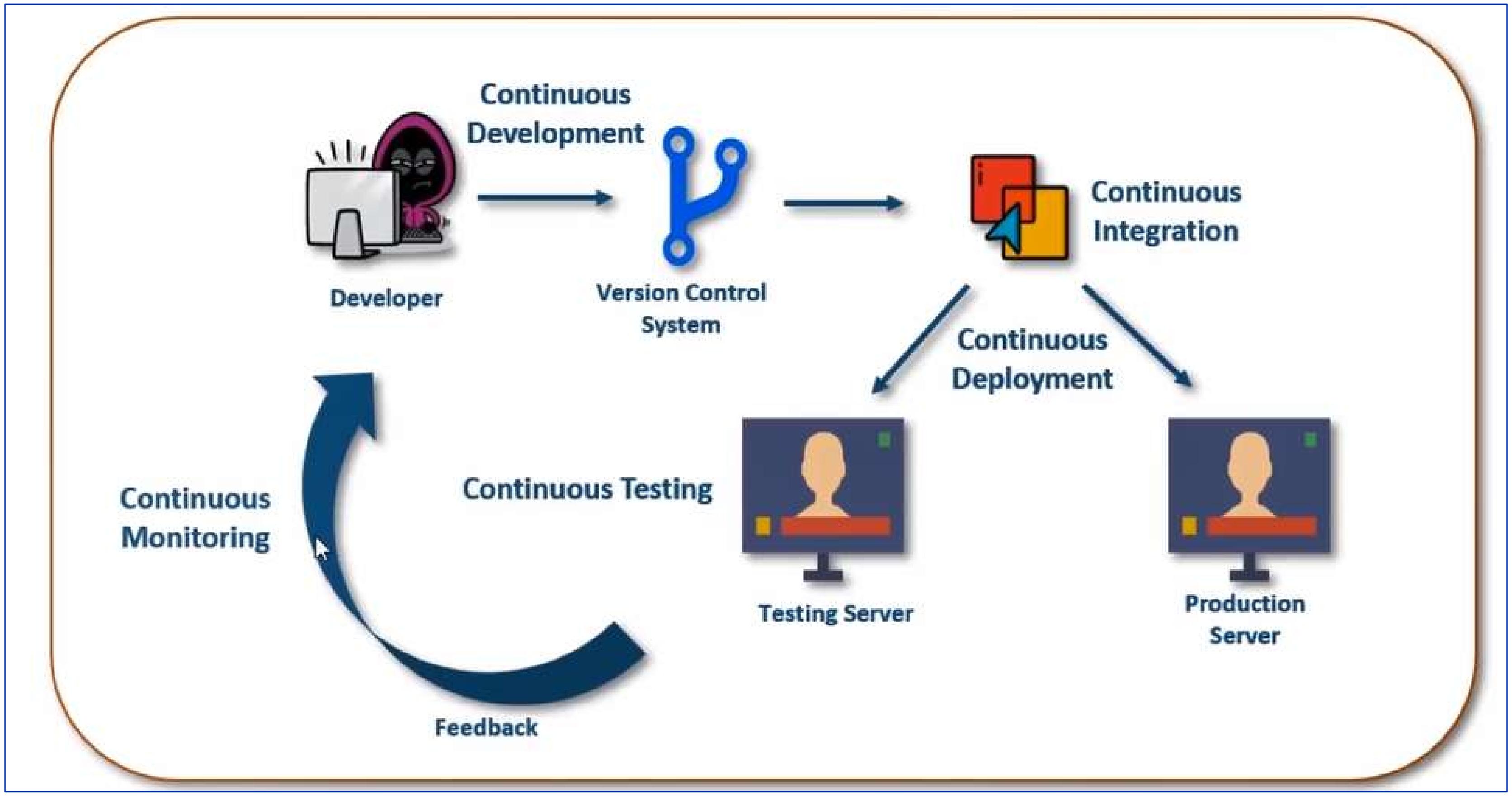
Devops- CI

Successful daily builds are the heartbeat of a software project.

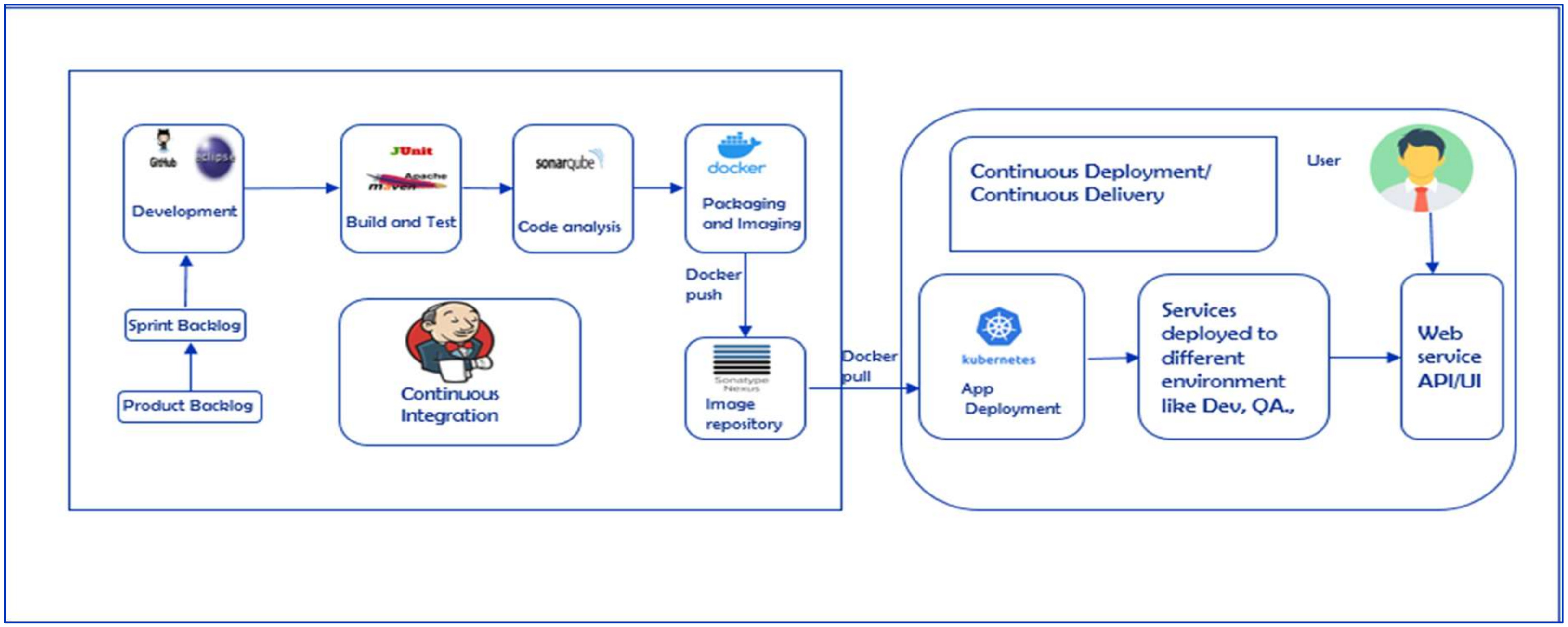
If you do not have successful daily builds, then you have no heartbeat, and your project is dead!



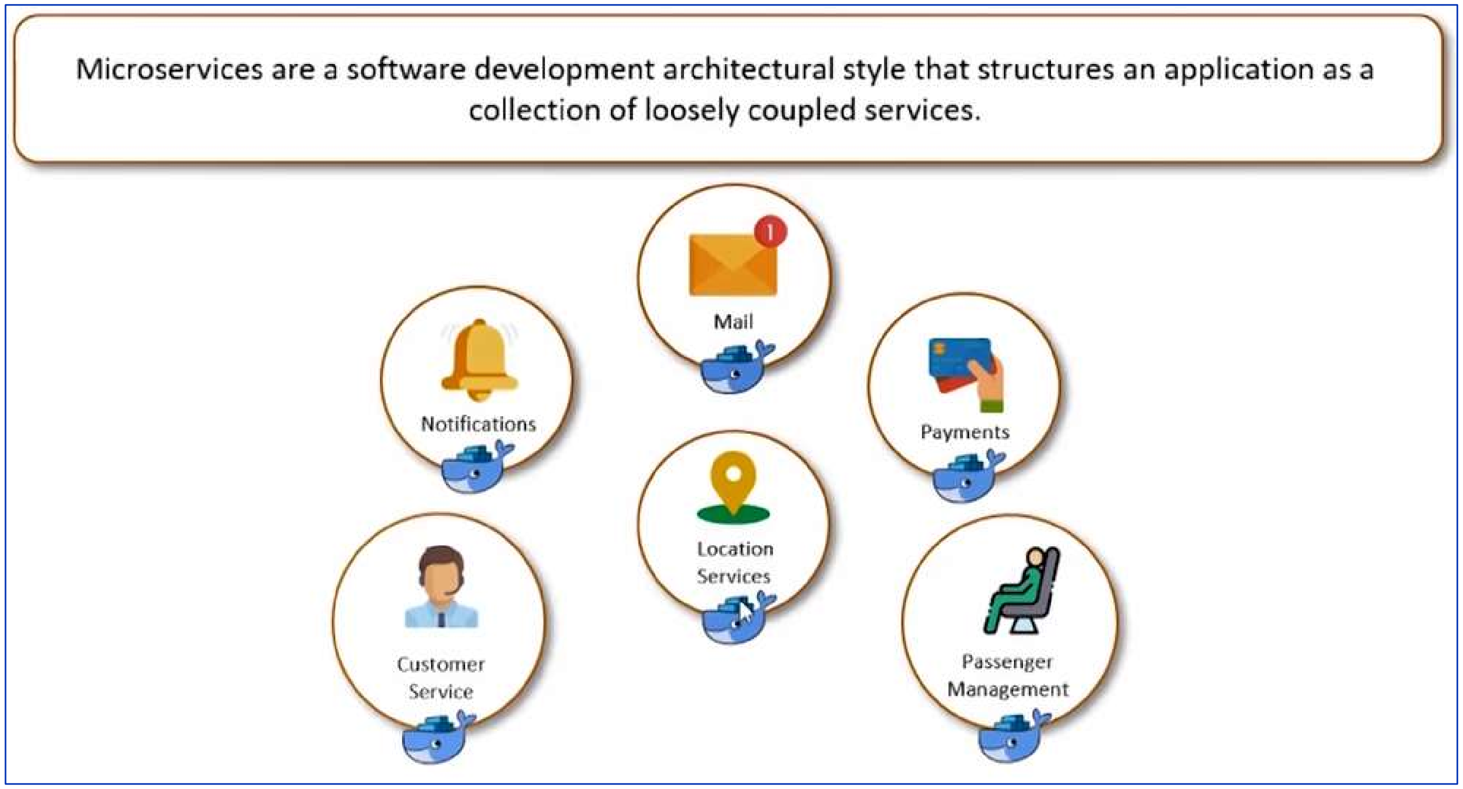
How Devops works



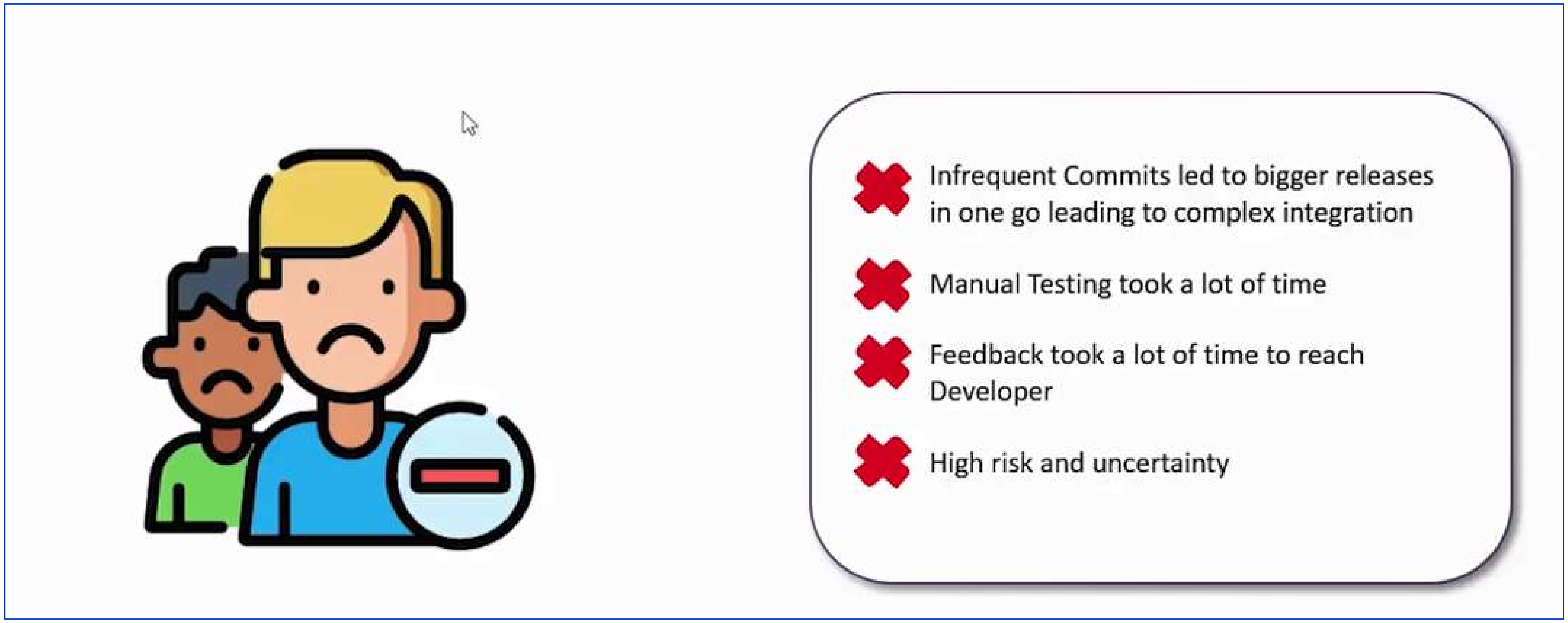
CI/CD Lifecycle



Microservices


Microservices are a software development architectural style that structures an application as a collection of loosely coupled services.

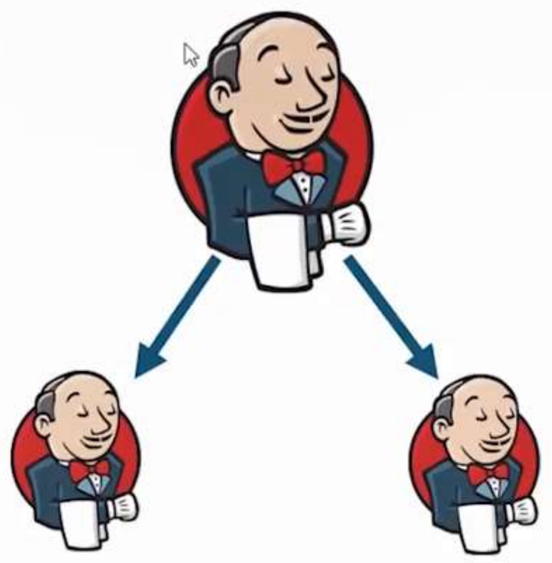
Before Continuous integration



- Infrequent Commits led to bigger releases in one go leading to complex integration
- Manual Testing took a lot of time
- Feedback took a lot of time to reach Developer
- High risk and uncertainty

Cognizant

After Continuous integration



- Jenkins Master-Slave architecture is important to accomplish Continuous Deployment
- Jenkins Slave need to have Jenkins installed on them
- Useful in scenarios where it distributes the load when we have parallel and complex builds

✓ Deploy 2 Servers Slave 1 and Slave 2

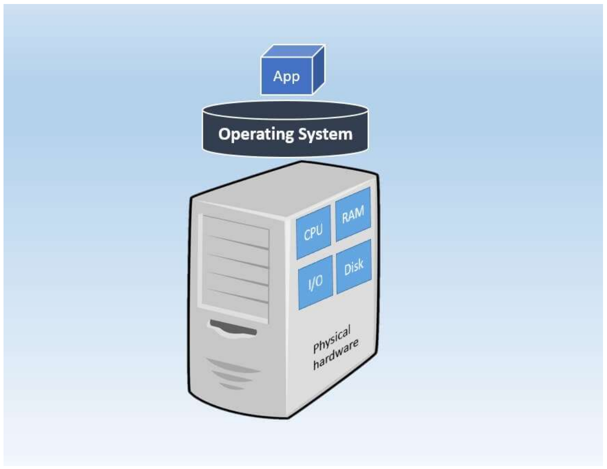
✓ Connect the Slaves using JNLP connection

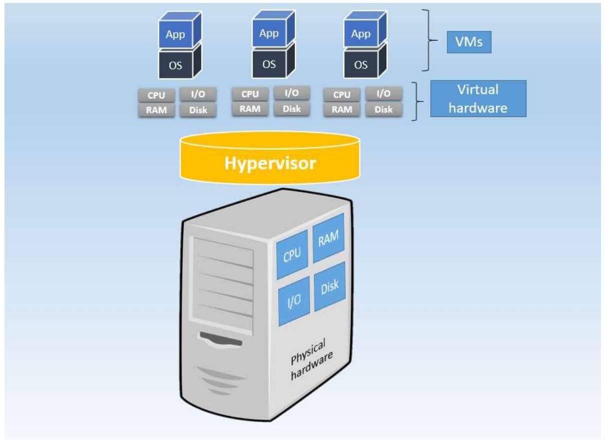
DevOps skills – DevOps engineer

- **Infrastructure knowledge**
- Integration skills
- Automations
- Logical thinking and skills
- Exploring multiple tools

Cognizant

Physical environment





- **Web server**
- **DB server**
- **Back-up server**

- Cost
- Resource wastage
- OS installation time
- Space (Handle lot of customers)

- **Virtualization model**
- **Hyper-V , Linux KVM , VMWARE – ESXi**

- Single Hardware management

- Cost reduced. (Instead of web,db,backup servers will include it in single server and manage it)

- Template concept
- Ex: 50 machines we need
- In this 50 machines , we install specific software

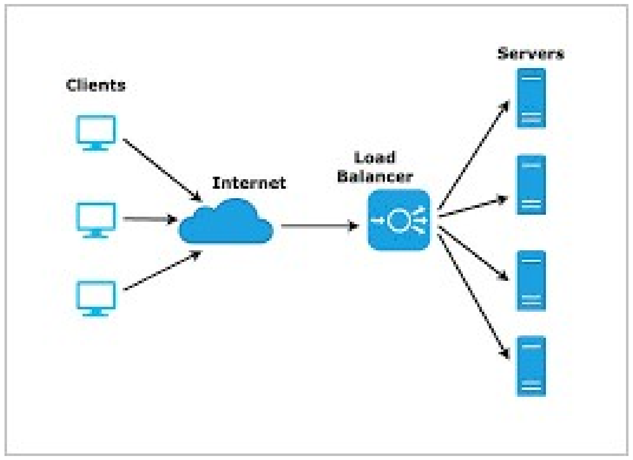
- In cloud we say AWS AMI

- Cloud utilized the internet as medium

- To get rid of data centre

- Autoscaling

Maintain applications



- **Load balancers**
- Without LB we cannot handle multiple connections
- Client's access say an example xyz website and connects with Load balancer and serve the request to end user.
- Autoscaling concept adds the machine on the fly.

- **Release and deployment**
- Version control – GIT
- Build tool – Maven
- Testing tool – Selenium
- CI/CD tool – Jenkins
- Deployed into AWS or docker or virtual server
- Configuration are maintained in Chef/Puppet/Ansible
- Monitoring the apps
- Nagios/Prometheus

Cognizant

Thank You!