

Lecture 09-10: Software Integration (\neq DevOps)

EGCI341

Outline

- DevOps story
- What is DevOps?
- DevOps Framework
- Becoming a DevOps Engineer
- eDevOps
- References

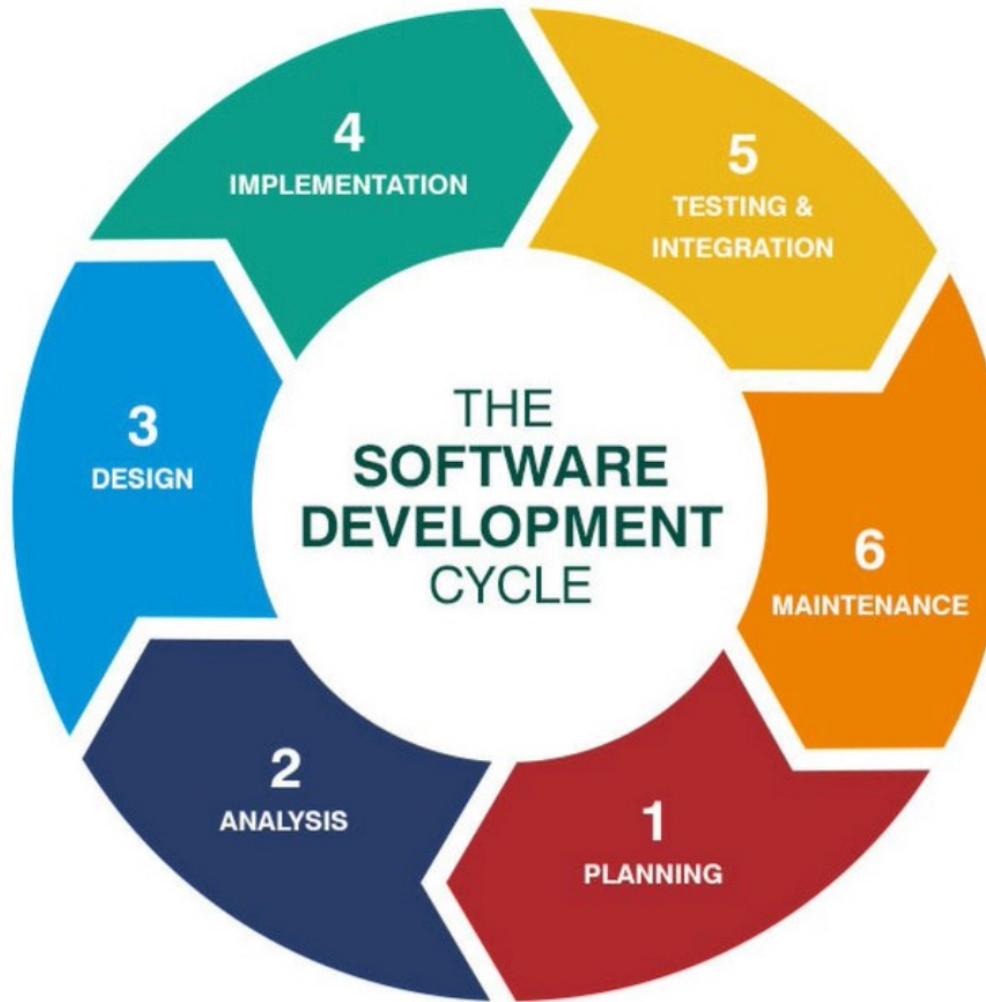
CI/CD: Code Integration & Code Deployment

Problem:

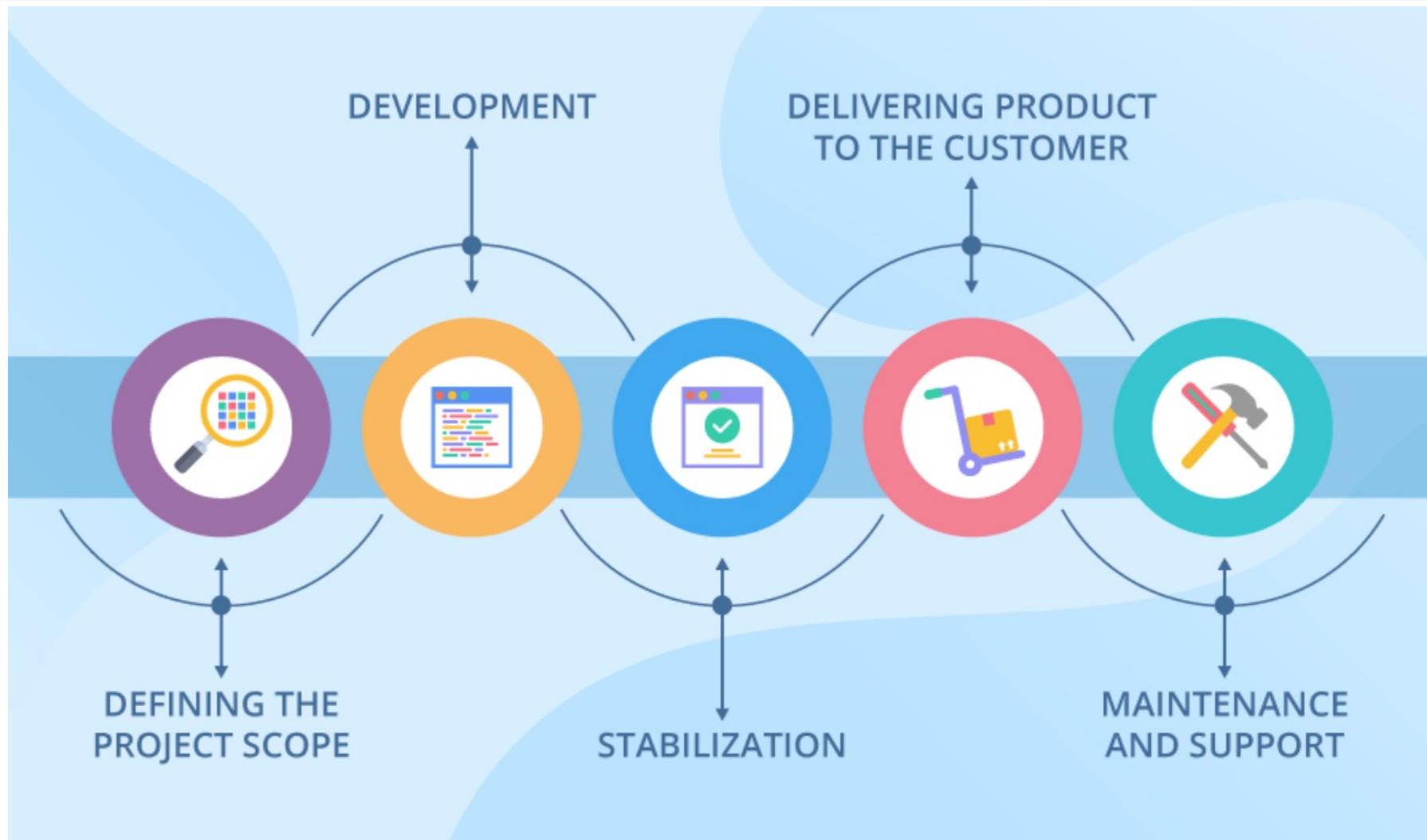
Company	Deploy Frequency	Deploy Lead Time	Reliability	Customer Responsiveness
Amazon	23,000 / day	Minutes	High	High
Google	5,500 / day	Minutes	High	High
Netflix	500 / day	Minutes	High	High
Facebook	1 / day	Hours	High	High
Twitter	3 / week	Hours	High	High
Typical Enterprise	Once every 9 months	Month or Quarters	Low/Med	Low/Med

ที่มา: หนังสือ The Phoenix Project: A Novel about IT, DevOps, and Helping Your Business Win

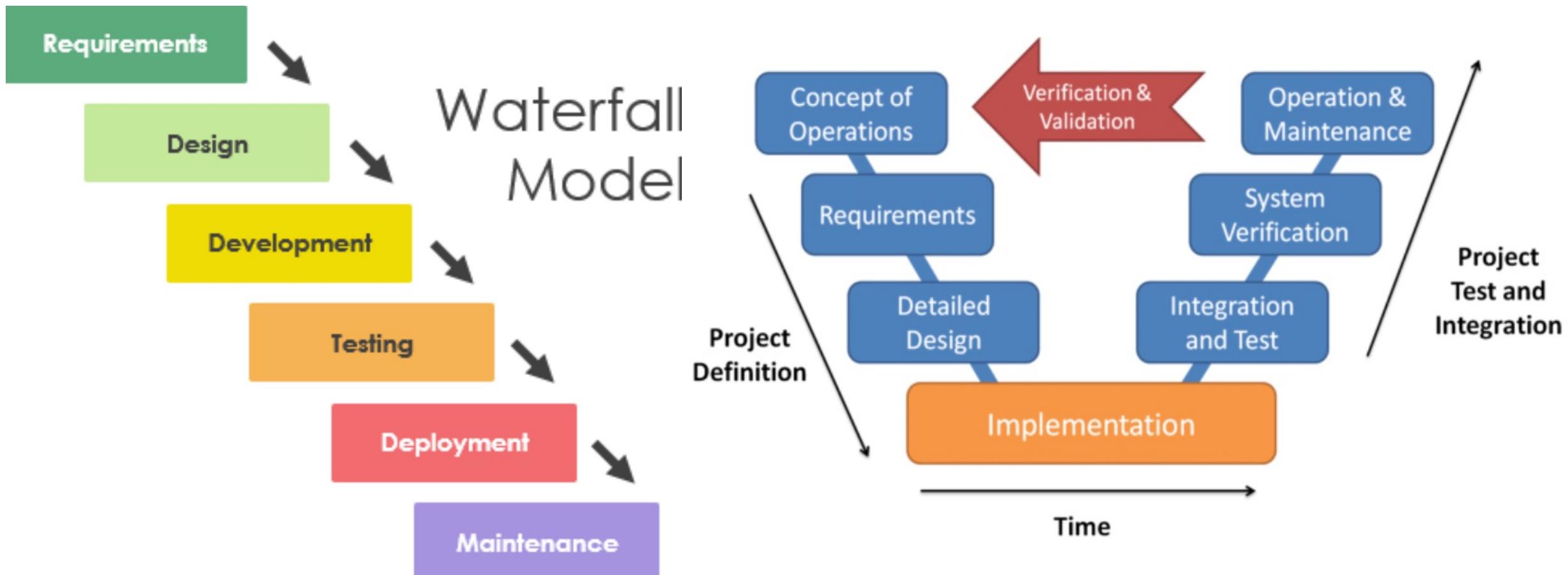
The World before DevOps



The World before DevOps (Cont.)

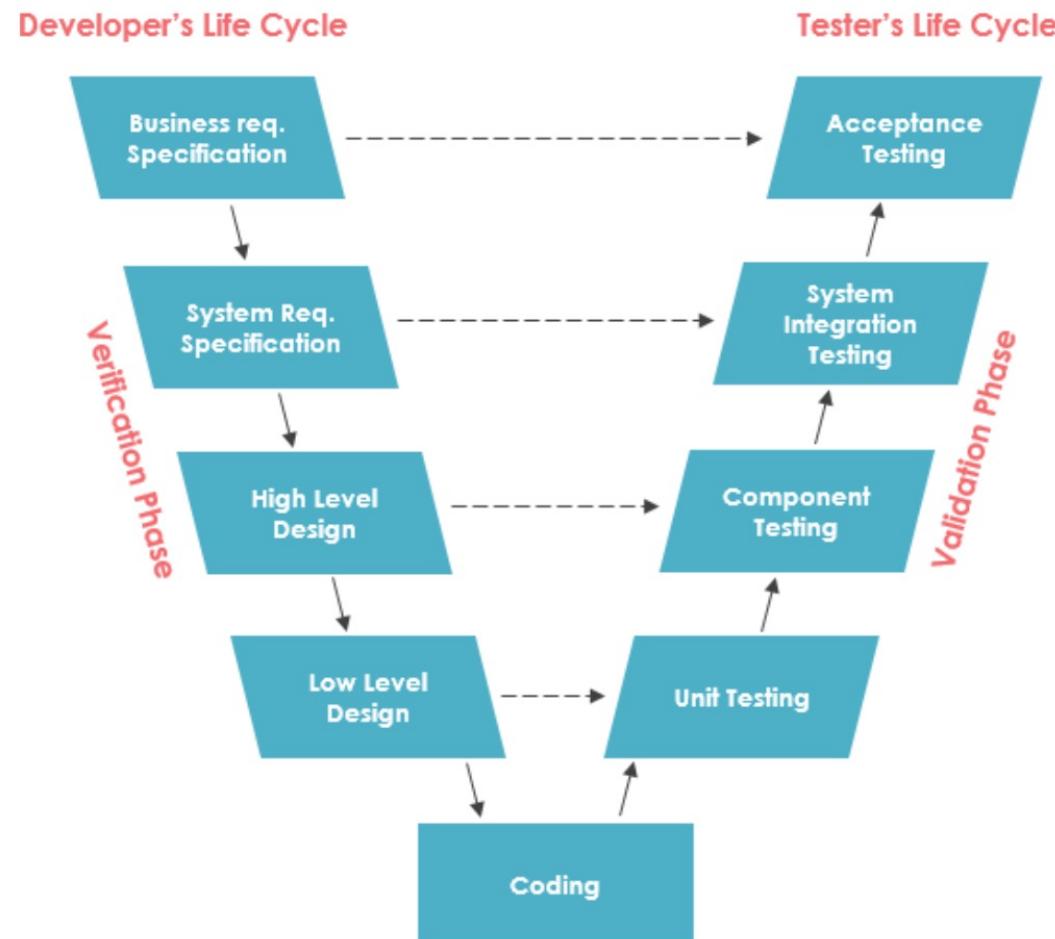


The World before DevOps – Waterfall Model

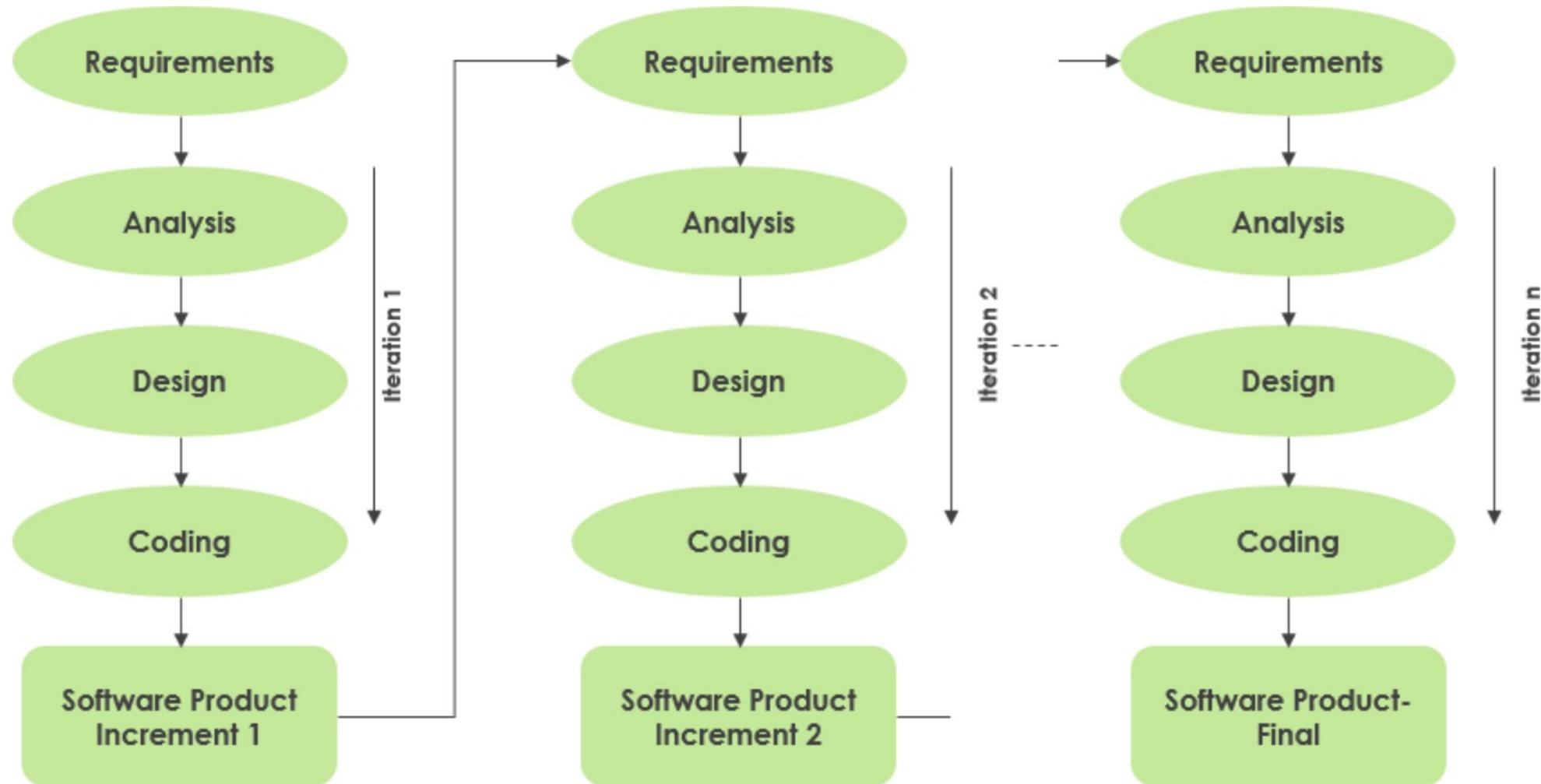


The World before DevOps – V-Model

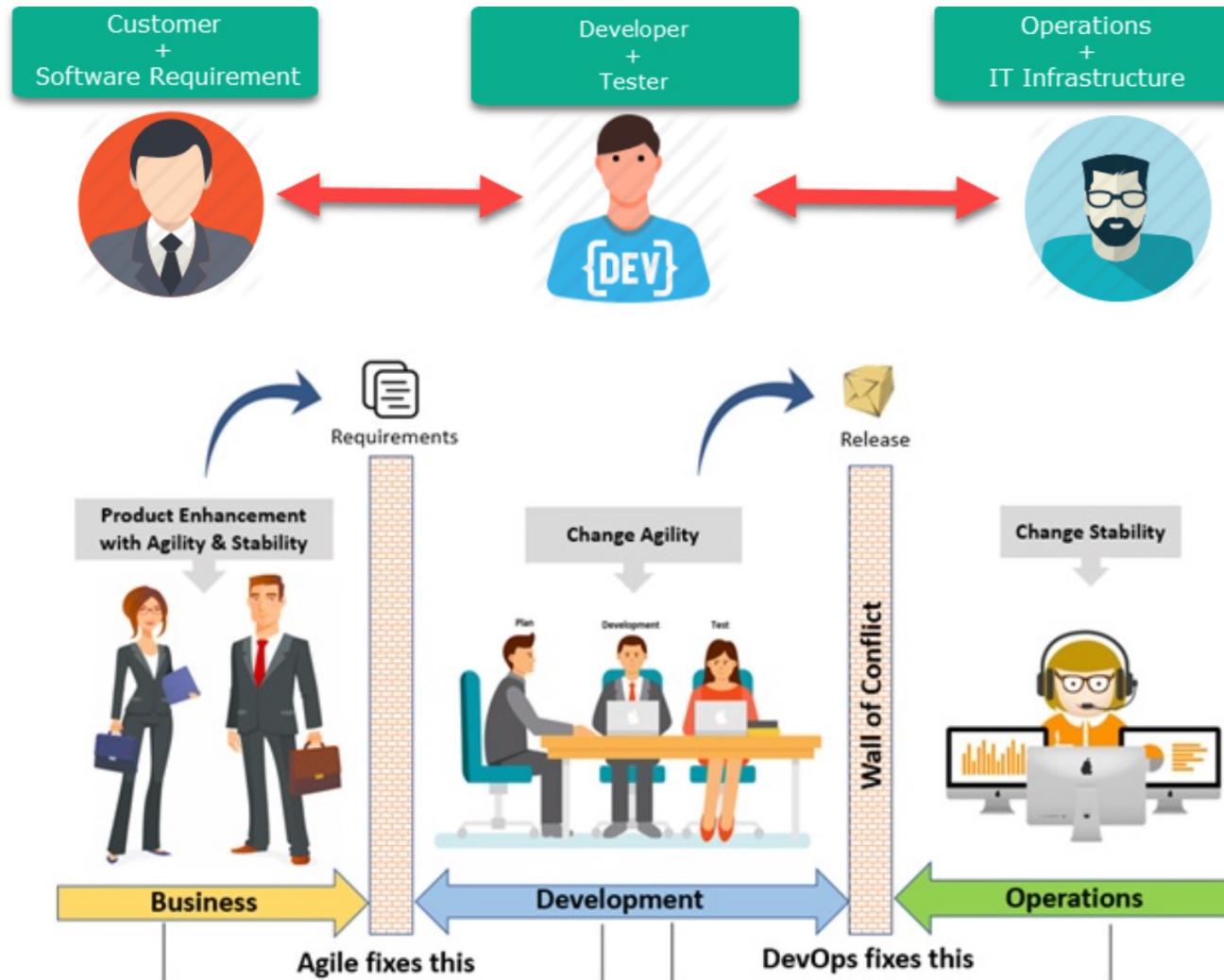
V-Model



The World before DevOps – Incremental Model



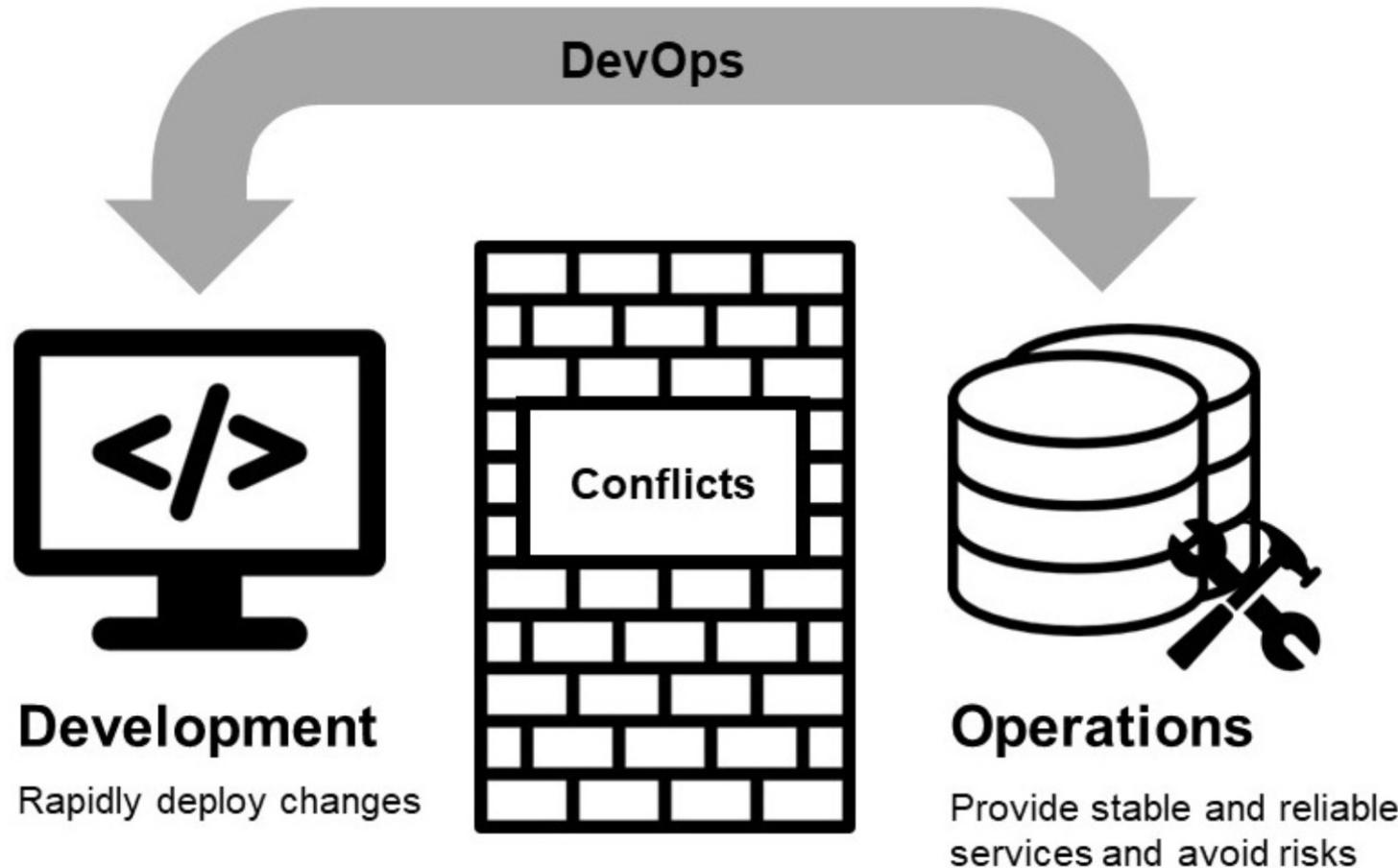
The World before DevOps



The World before DevOps (cont.)



The World before DevOps (cont.)



The World before DevOps (cont.)

Who is feeling the pain?



Developers

- Frequently disagree with system administrators
- Spending time on mundane manual tasks



Operations

- Challenging audits
- Hard-to-diagnose outages
- Receive code from developers with little communication



Product Managers

- Frequently disagree with system administrators
- Costs and inefficient projects



System Administrators

- Frequently disagree with developers and others
- Lack visibility in build, deploy, test, and release processes
- Considerable time spent on upkeep



Test Engineers

- Frequently disagree with system administrators
- Receive differing instructions from developers' code and operations' directions



QA

- Dealing with slow error-prone software delivery
- Spending copious amounts of time fixing costly errors



Customer Service

- Trying to placate disgruntled customers
- Stuck in the middle of a system with little communication



End Users

- Frequent and prolonged outages
- Hard-to-navigate systems

DevOps Story

Patrick Debois starts assessing IT Value Chain



Agile System Administrators Group is launched on Google



Inaugural "DevOps Days" are held in Ghent, Belgium



Industry leading software vendors increase market presence with "Enterprise" class DevOps tools



devX is born and Xceed launches the "12 days of DevOps"



2007

2008

2009

2010

2011

2012

2013

2014

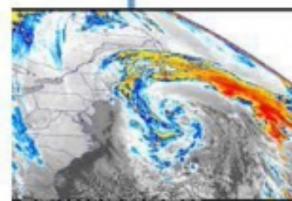
2015



Andrew Clay Shafer and Patrick Debois meet at Agile Conference 2008



Open Source toolsets rip up the legacy playbook



John Allspaw and Paul Hammond present "10 Deploys per day" at Velocity

The "Perfect Storm" of adjacent methodologies occurs



Cameron Haight predicts that DevOps will hit the big time in 2015 across Enterprise organisations



What does the future of DevOps hold for your organisation?

DevOps begins to provide positive impact to "Enterprise" IT and experiences seismic adoption rates

What is DevOps?

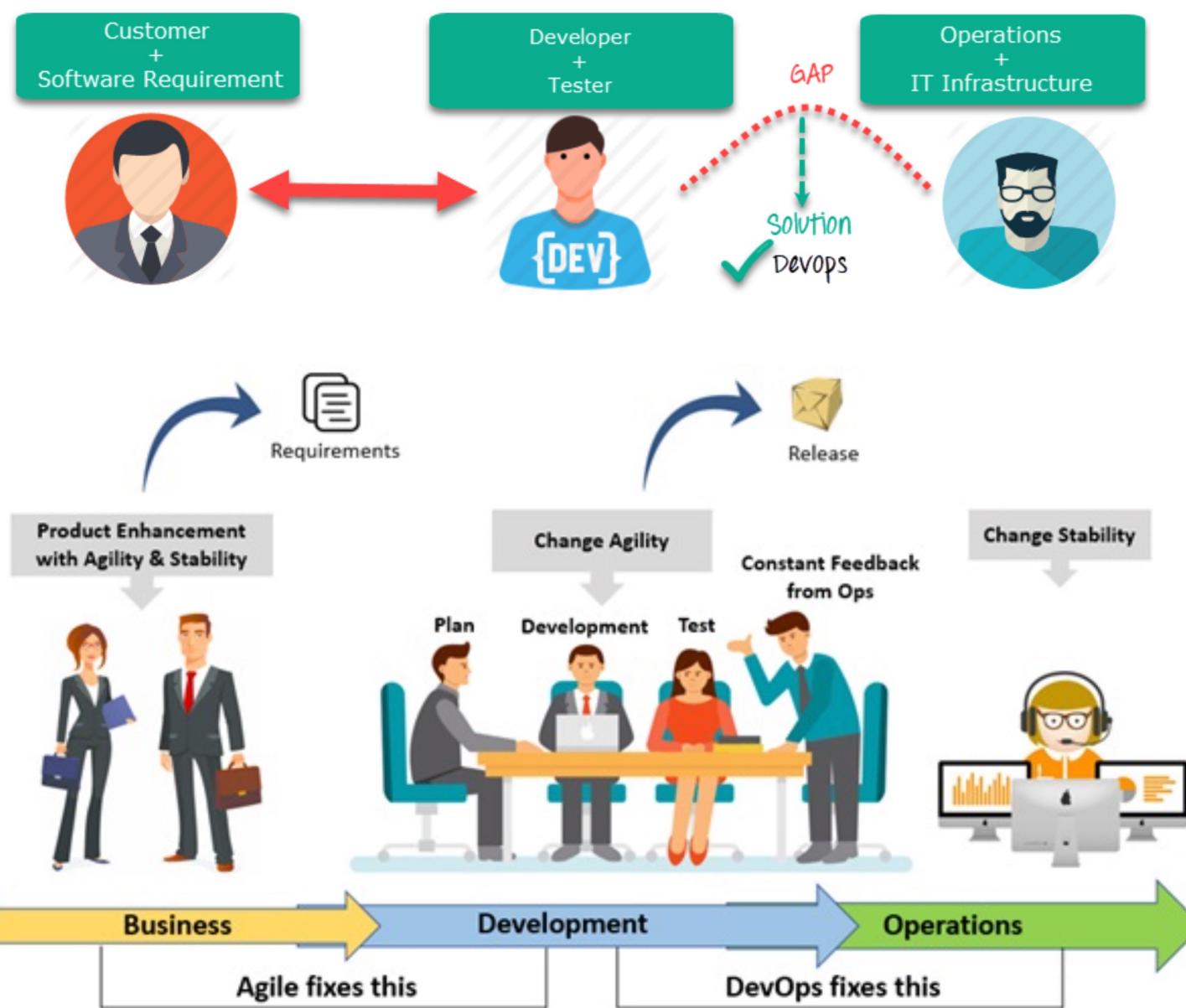
DevOps is a culture, movement or practice that emphasizes the collaboration and communication of both Software Developers and other Information-Technology professionals while automating the process of software delivery and infrastructure changes.

What is DevOps? (cont.)

DevOps in an extension of the agile mindset and methodologies that is focused on bringing customer value without compromising creativity (development) and stability (operations)

“DevOps Fundamentals Workshop” V1.0 - SECC

What is DevOps?



DevOps is not ...

- Simply combining Development and Operations in one team
- Simply a function or title
- Just tools and automation
- A separate team
- A one-size-fits-all solution

DevOps Framework Overview

People

- Mindset
- Roles and Responsibilities

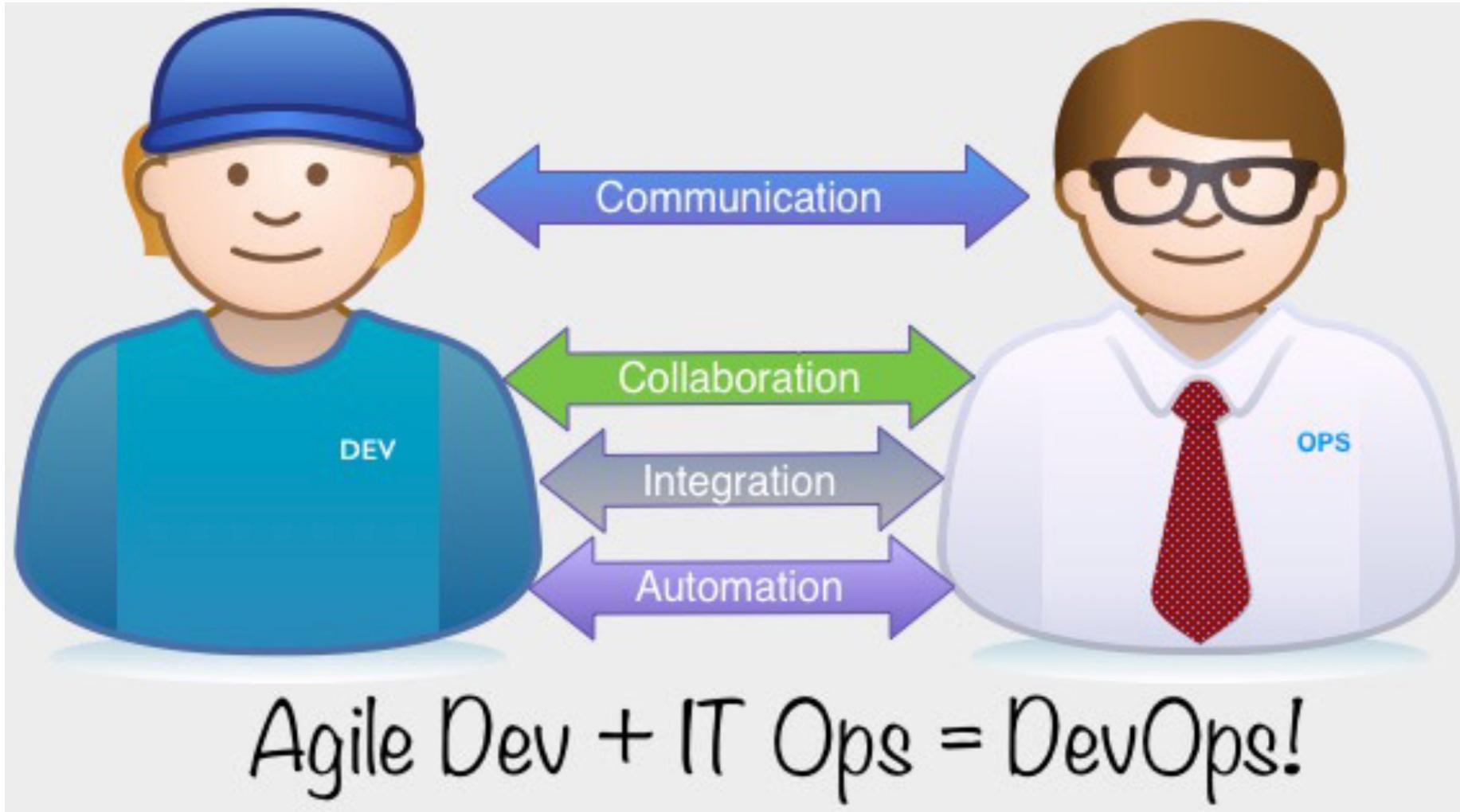
Process

- Practices
 - DevOps Pipeline
 - Configuration Measurement
 - 5Cs
 - Automation
 - IaC
 - Containization
 - Kpis

Technology

- DevOps Toolchain
- DevOps Periodic Table

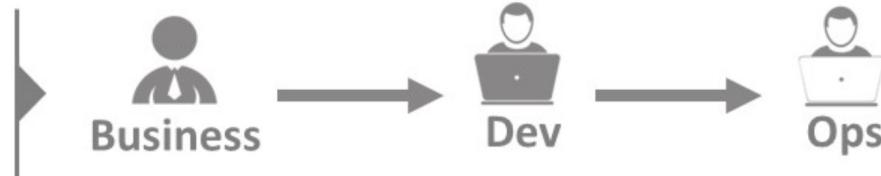
People: Mindset



Mindset

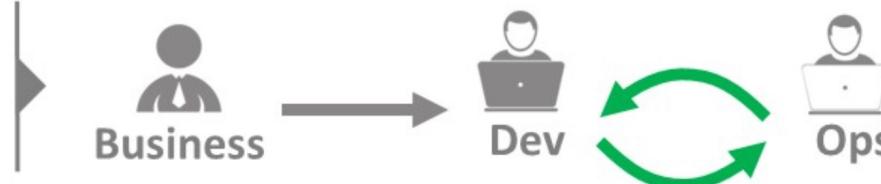
V Mode

- Relationships between actors are very segmented
- Long deployment delays with tunnel effect risk



V Mode + DevOps

- Improvement of time frame for testing and deployment
- Limitation of error risks



Agile

- Acceleration of time frame for upstream phases: design, developments, testing
- Focus on added value for the business



Agile + DevOps

- Acceleration of TTM via actions on the whole delivery cycle
- Improvement of quality via industrialization and automation of manual tasks



People: Mindset – Lean Thinking

Lean (S/W Dev.) is a systematic method in which the core idea is to maximize customer value while **minimizing waste**

- **Value** is everything that your customer is willing to pay money for
- **Waste** is any action or step in a process that does not add value to the customer. In other words, waste is any process that the customer does not want to pay for

People: Roles and Responsibilities

Who benefits from DevOps?



Developers

- Fewer mundane, repetitive tasks because of automation
- More time developing



Test Engineers

- Faster resolution to problems due to increased visibility and communication
- Stop bugs before they get too deep



Operations

- 22% less time spent on unplanned work and rework
- Better code from developers because of increased communication



QA

- Receive less problems caused by human error
- 50% less time remediating security issues



Product Managers

- 46x more frequent software deployments
- 440x faster lead time for changes



Customer Service

- Work with happier customers
- Part of a more collaborative environment



System Administrators

- 96x faster recovery from failures
- 3x lower failure change rate
- Less complex to manage



End Users

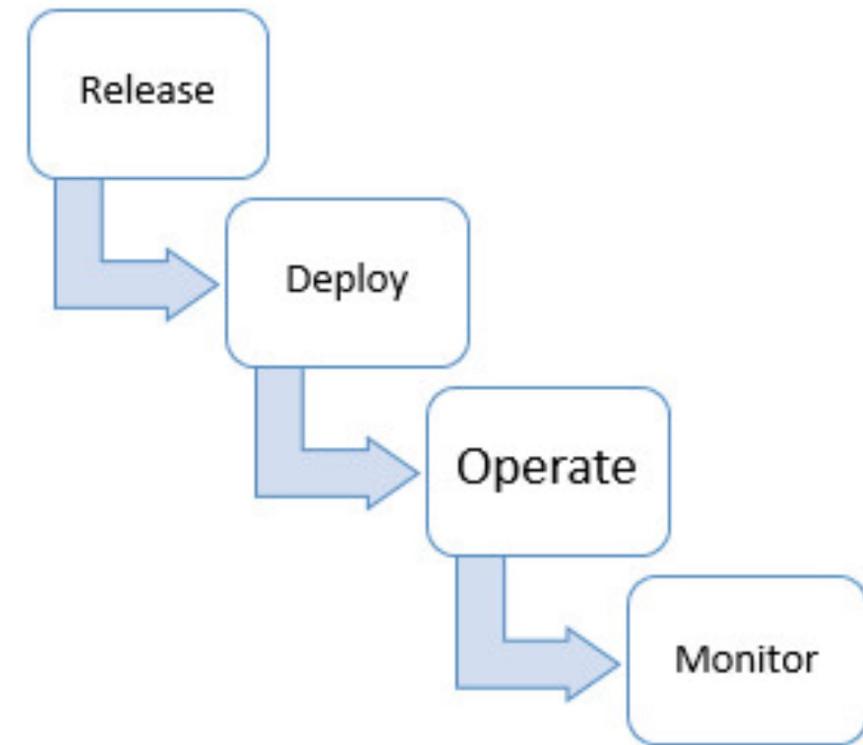
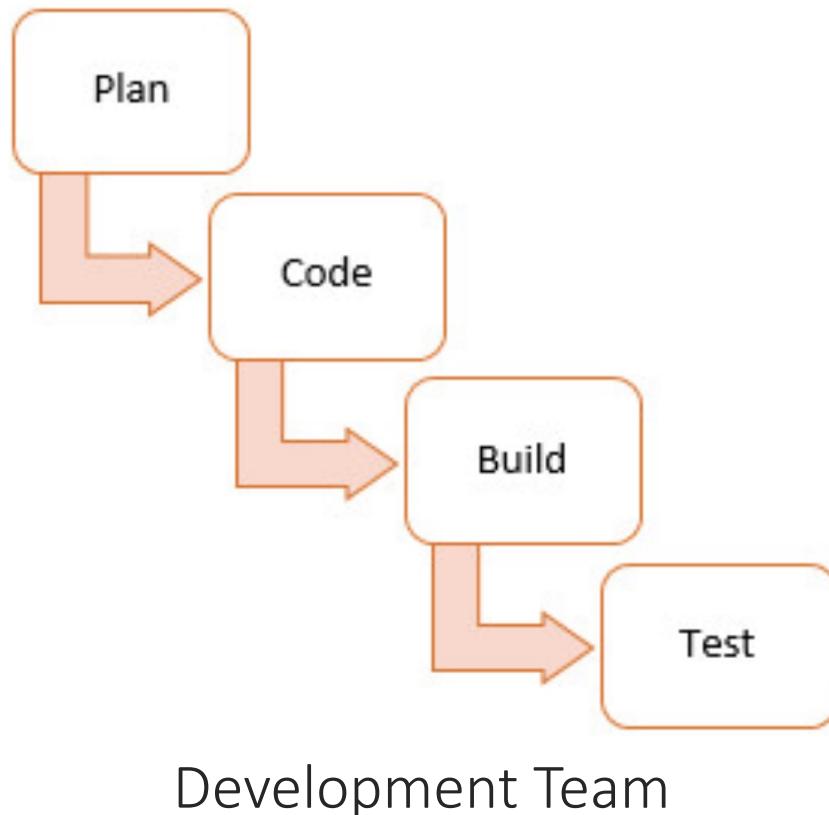
- Experience more consistent processes and applications
- Enjoy a pleasant experience

People: Mindset Conclusion

- Bridging gaps
- Delivering value to customers

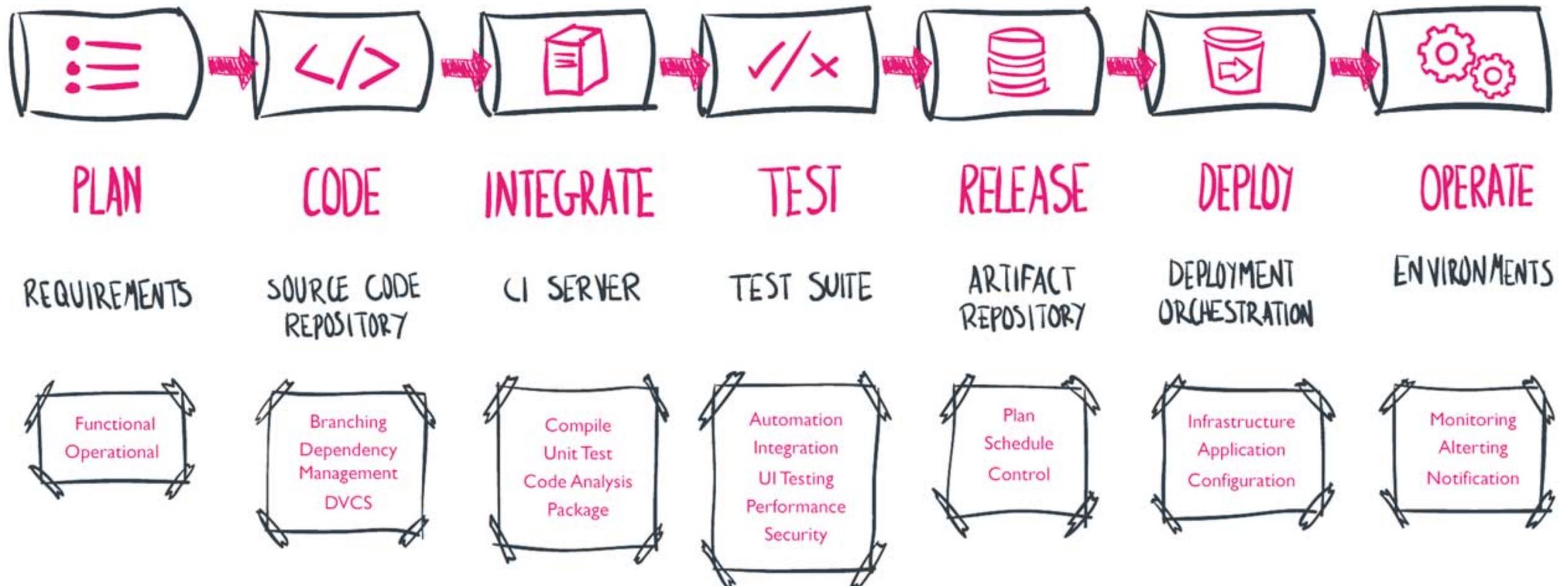
Process: DevOps Pipeline

A **DevOps pipeline** is a set of practices that the development (Dev) and operations (Ops) teams implement to build, test, and deploy software faster and easier. One of the primary purposes of a pipeline is to keep the software development process organized and focused.



Operations Team

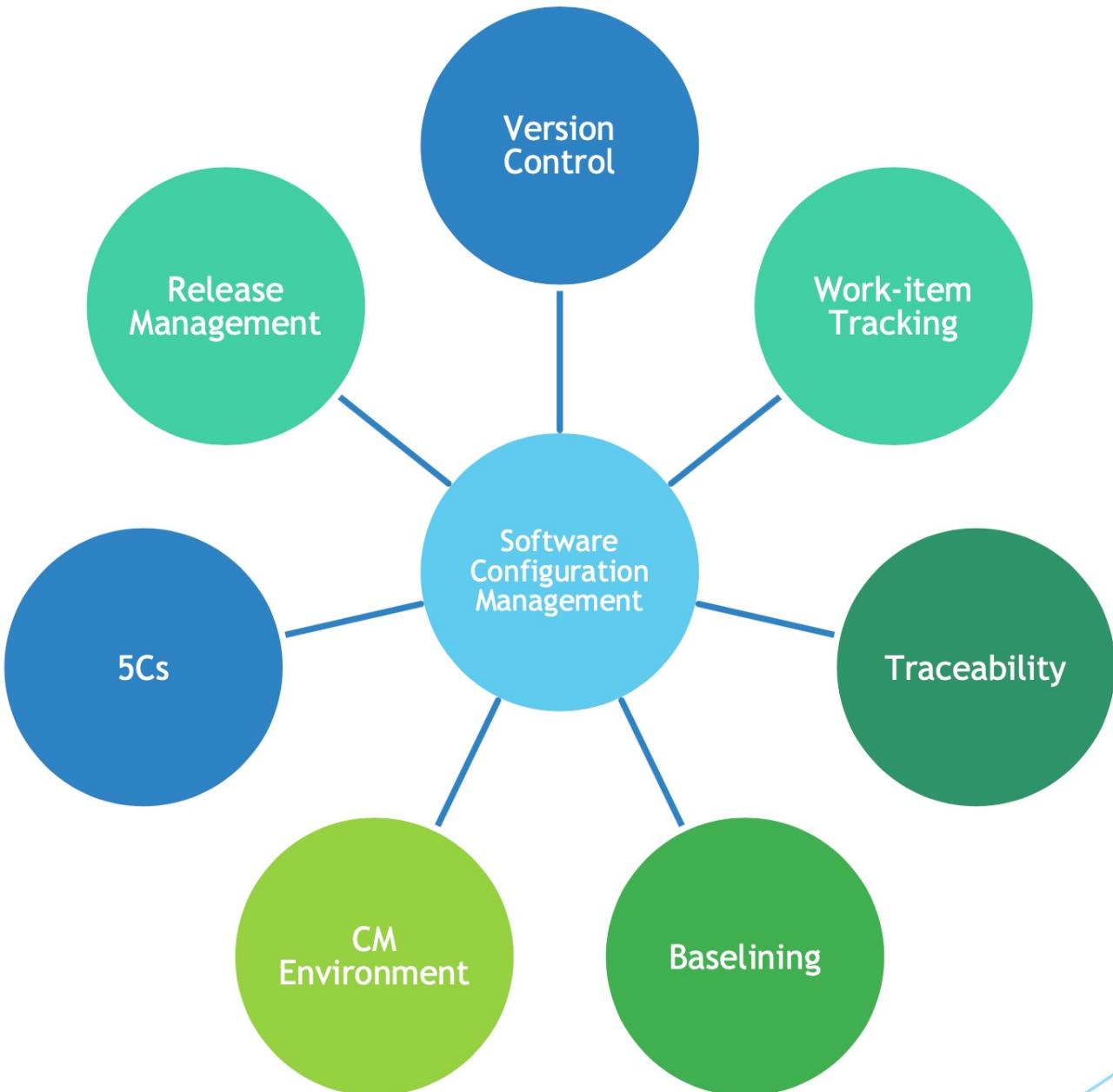
DevOps Pipeline



Software Configuration Management (SCM)

Software Configuration Management (SCM)

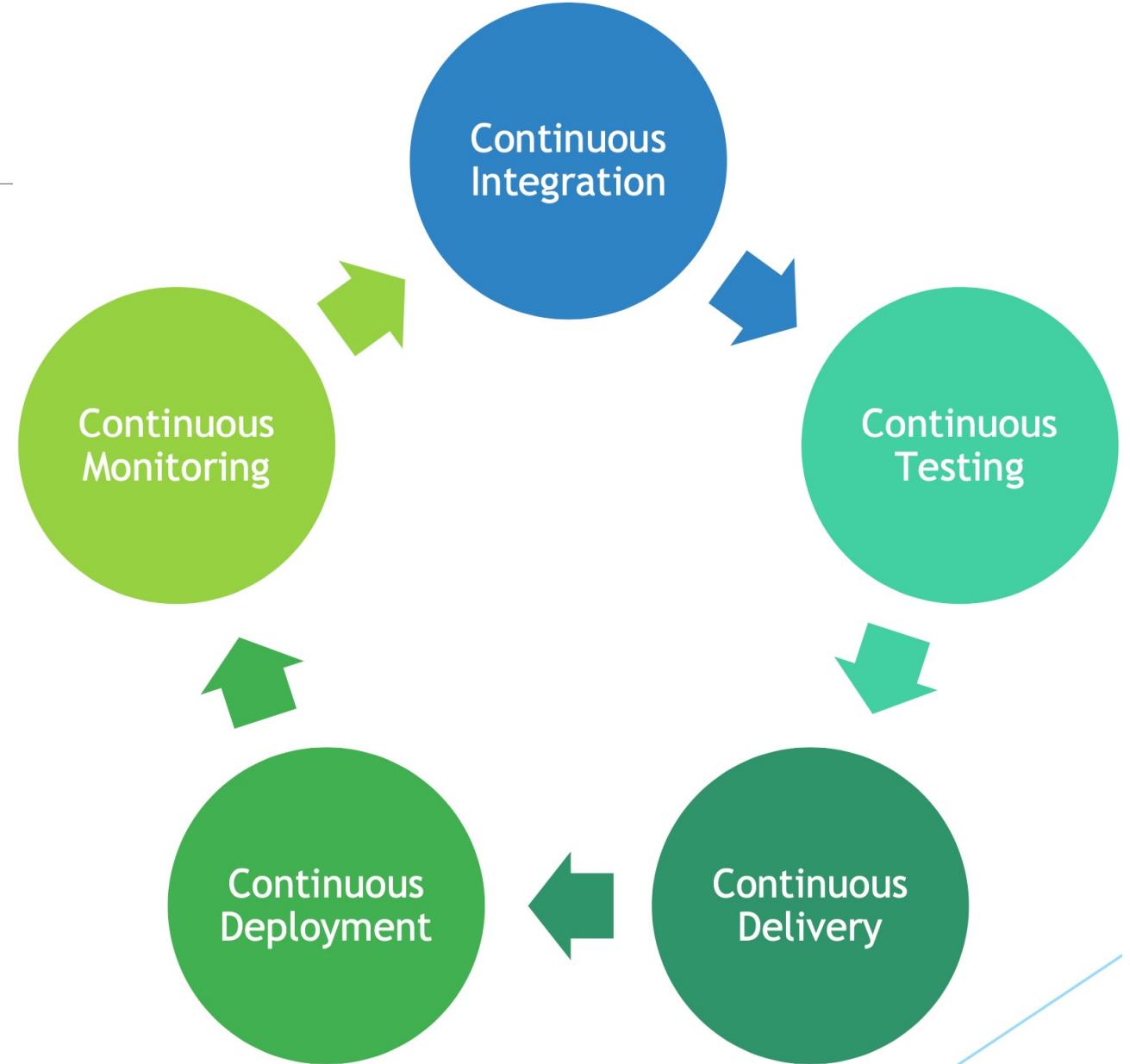
is a process to systematically manage, organize, and control the changes in the documents, codes, and other entities during the Software Development Life Cycle. The primary goal is to increase productivity with minimal mistakes.



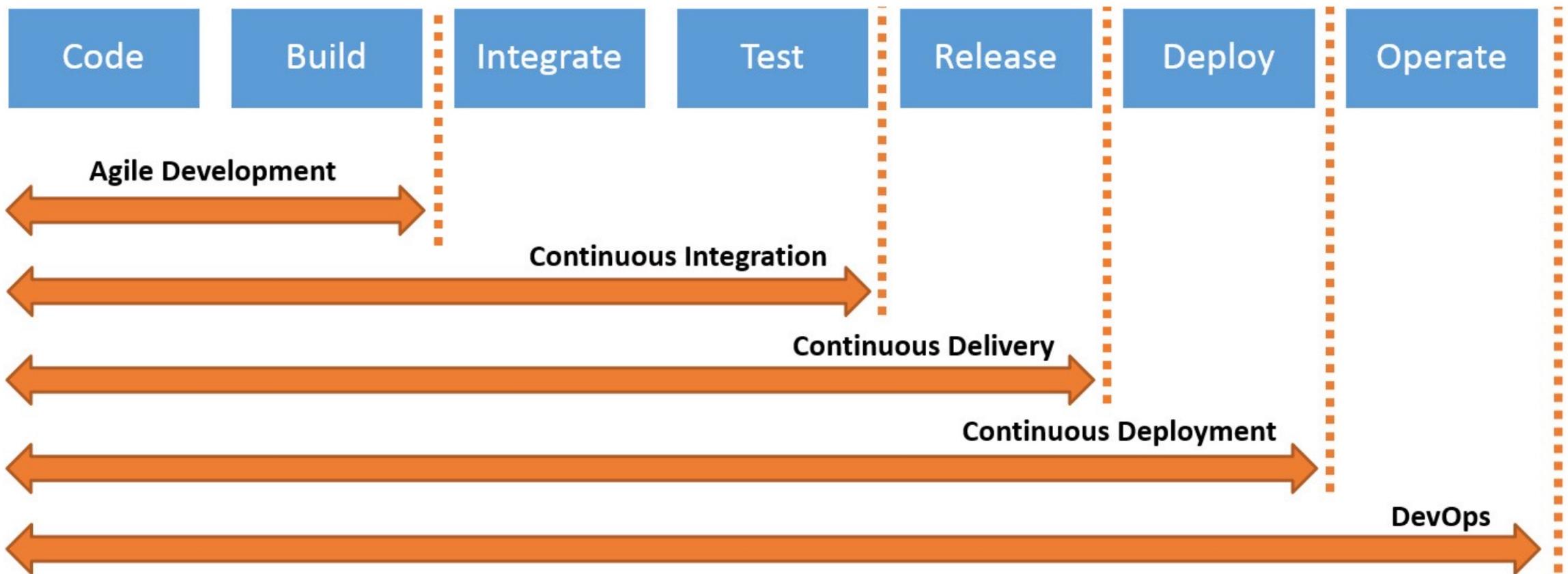
5Cs

5Cs of DevOps

are a set of practices in software development that are of repeatable nature and can be automated to increase quality and productivity.



5Cs and DevOps Pipeline



Automation

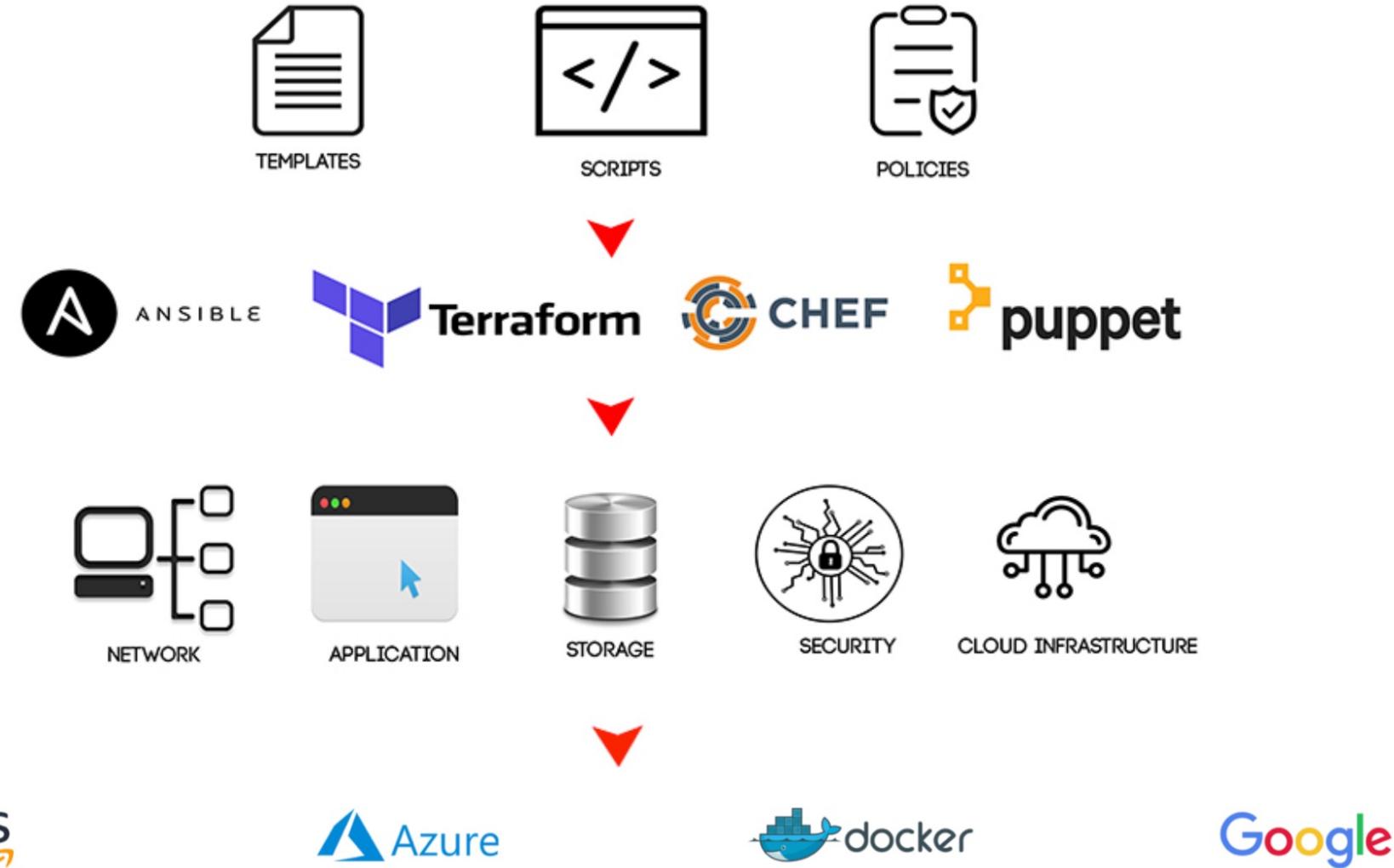
Automation is the process of creating software and systems to replace repeatable processes / activities and reduce manual intervention, thereby significantly increasing both **quality** and **productivity**.

Examples:

- Build (Ant, Maven, Gradle, Sbt, Grunt, Gulp)
- Unit tests (JUnit)
- Code Inspection (SonarQube)
- Packaging (JAR, Docker)
- Deployment (Jenkins, Ansible, Spinnaker)
- System/Functional tests (Cucumber, JBehave, Robot Framework, Selenium, Appium)
- Non-functional tests (Jmeter, Gatling, OWASP ZAP) u Provisioning (Ansible, Puppet, Chef, Terraform)
- Documentation (Swagger)

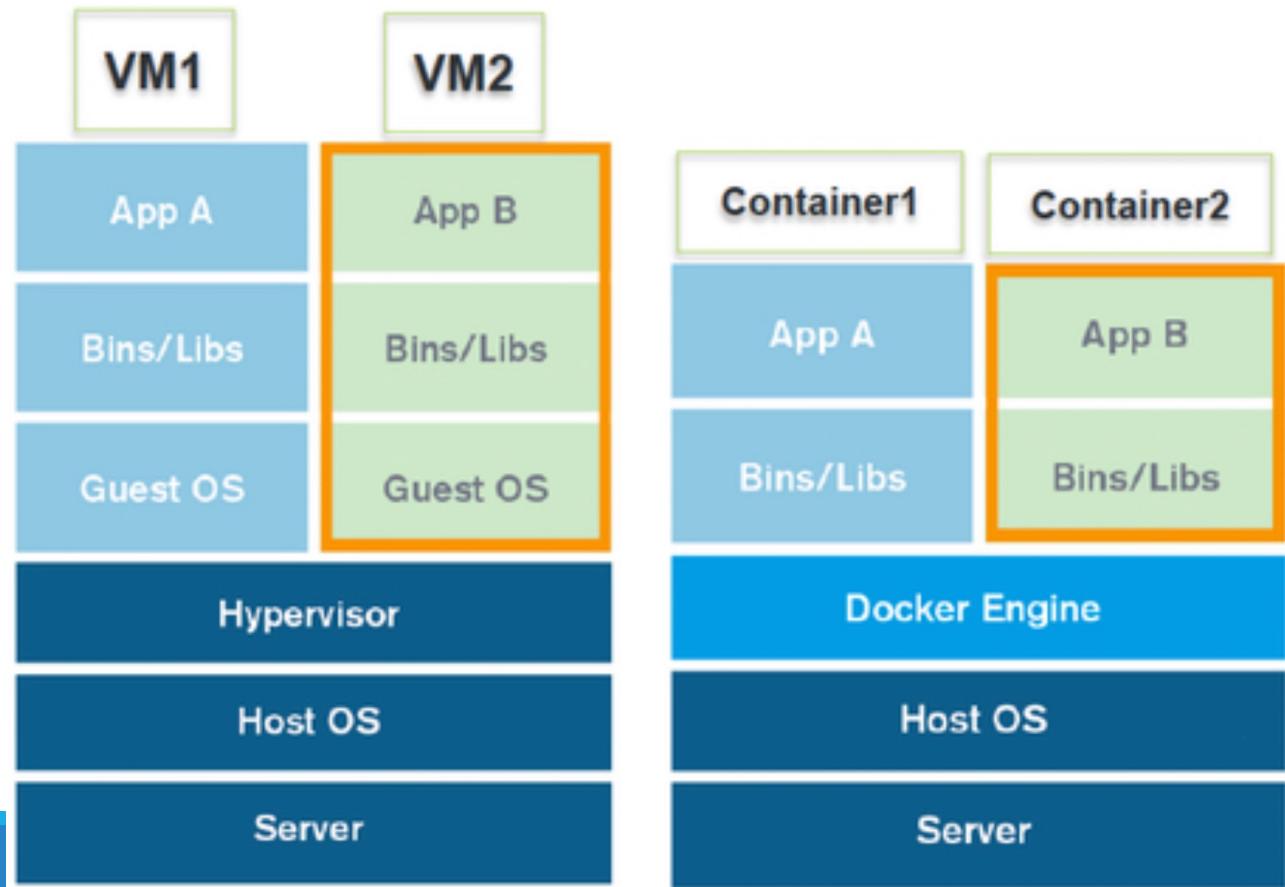
Infrastructure as Code (IaC)

Infrastructure as Code (IaC) is the process of managing and provisioning computer data centers through machine-readable definition files, rather than physical hardware configuration or interactive configuration tools.



Containerization

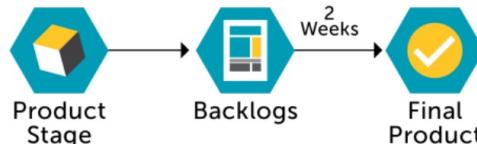
Containerization is the process of packaging an application along with its required libraries, frameworks, and configuration files together so that it can be run in various computing environments efficiently.



DevOps Practices Combined

AGILE DEVELOPMENT

Daily Standup



Process Flow Chart

User Inputs

Daily Scrum

Sprint Phrases

Continuous Delivery

Continuous Testing

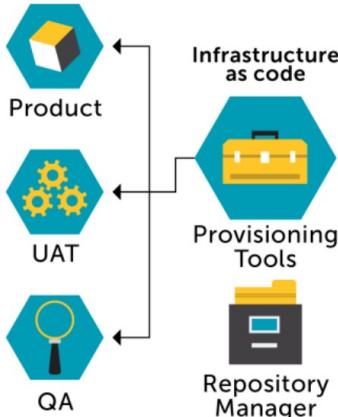
Continuous Integration

Sprint Review

Retrospection

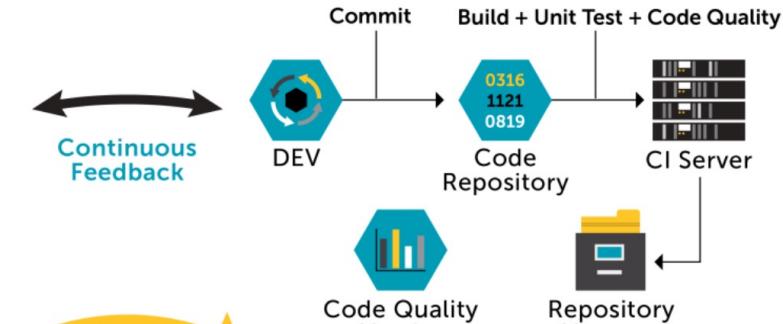
Continuous Feedback

CONTINUOUS DELIVERY



CONTINUOUS INTEGRATION

Commit



Code Quality Metrics

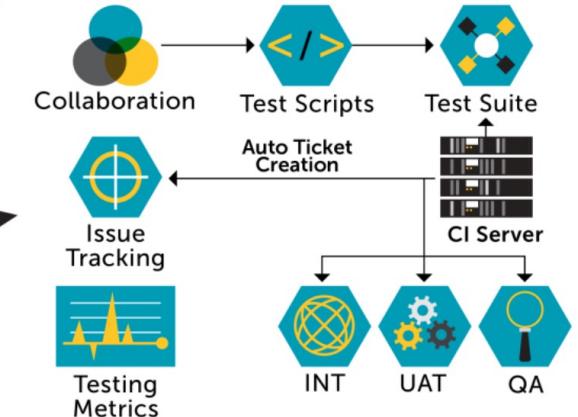
Repository Manager

Code Quality Metrics

Repository Manager

Continuous Feedback

CONTINUOUS TESTING



KPIs – CALMS Model

Culture

- People focus
- Mindset
- Embrace change

Automation

- CI/CD
- Infrastructure as Code (IaC)

Lean

- Focus on producing value for the end-user
- Productivity

Measurement

- Measure everything
- Show improvement

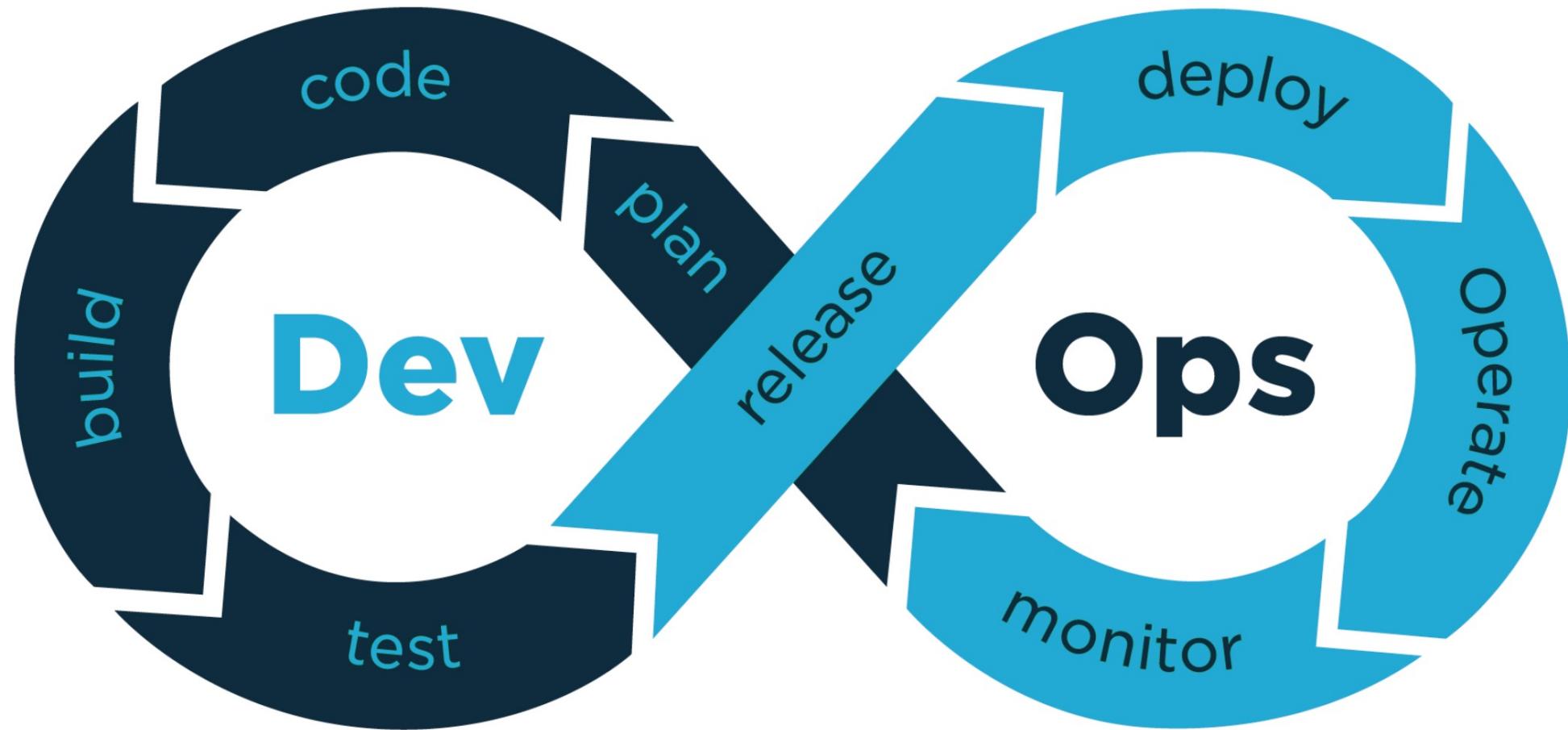
Sharing

- Collaboration
- Communication
- Open information sharing

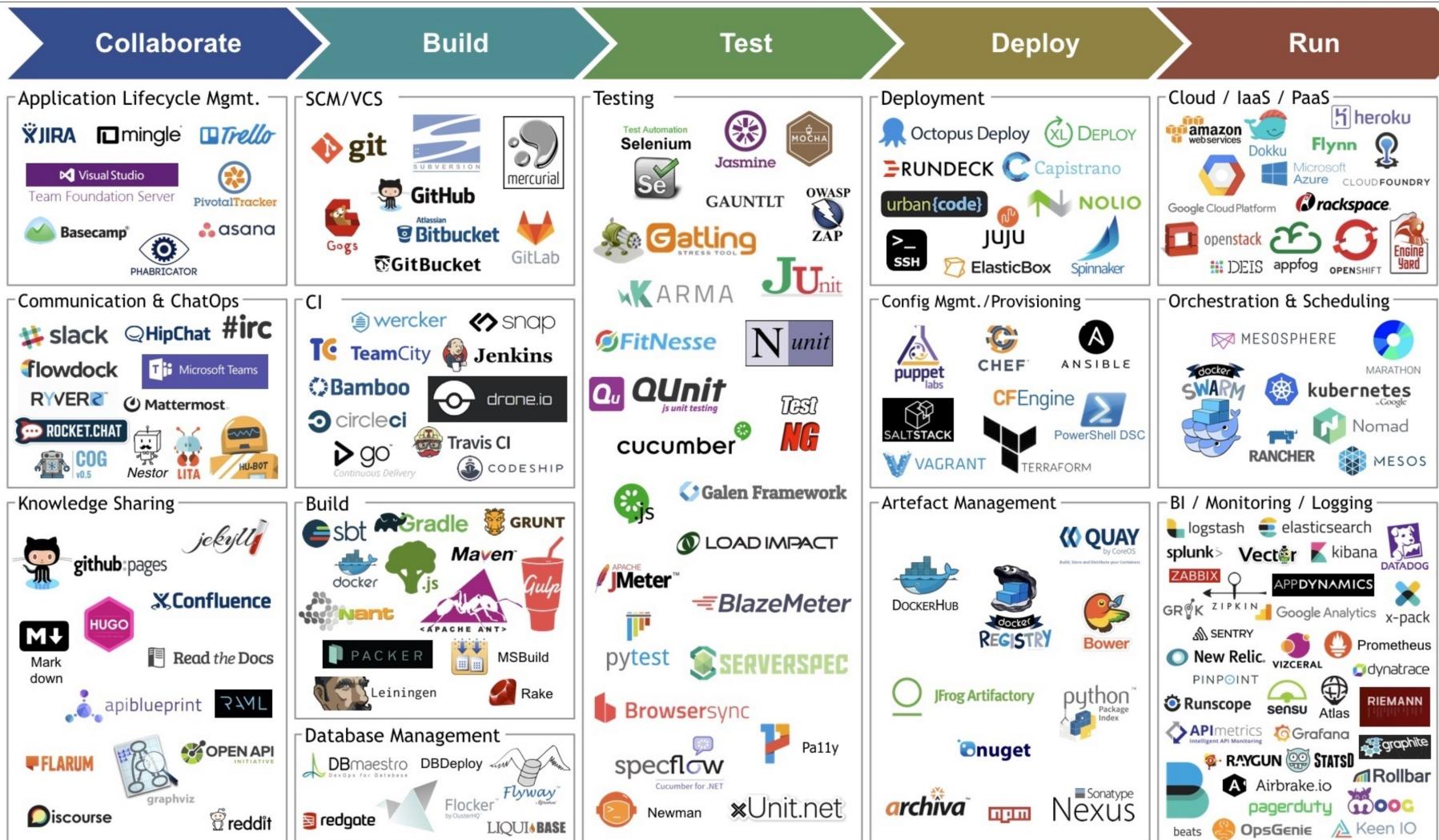
More KPIs

- **Mean Time To Production:** How long does it take for any newly committed source code to reach production?
- **Deployment Frequency:** How often are releases deployed into production?
- **Average Lead Time:** How long does it take for a new feature to be developed, built, tested, and deployed into production?
- **Deployment Speed:** How much time does it take to deploy a new release into production?
- **Production Failure Rate:** How often do failures occur in production?
- **Mean Time To Recover (MTTR):** How long does it take to recover from a failure?

DevOps Toolchain



DevOps Toolchain



<http://www.jamesbowman.me>

DevOps Periodic Table

The DevOps Periodic Table is a comprehensive chart mapping 118 DevOps tools and technologies across a grid of 18 columns and 10 rows. The columns represent periodic groups, and the rows represent atomic elements. The chart uses color coding to categorize tools based on their primary function or technology stack.

Legend:

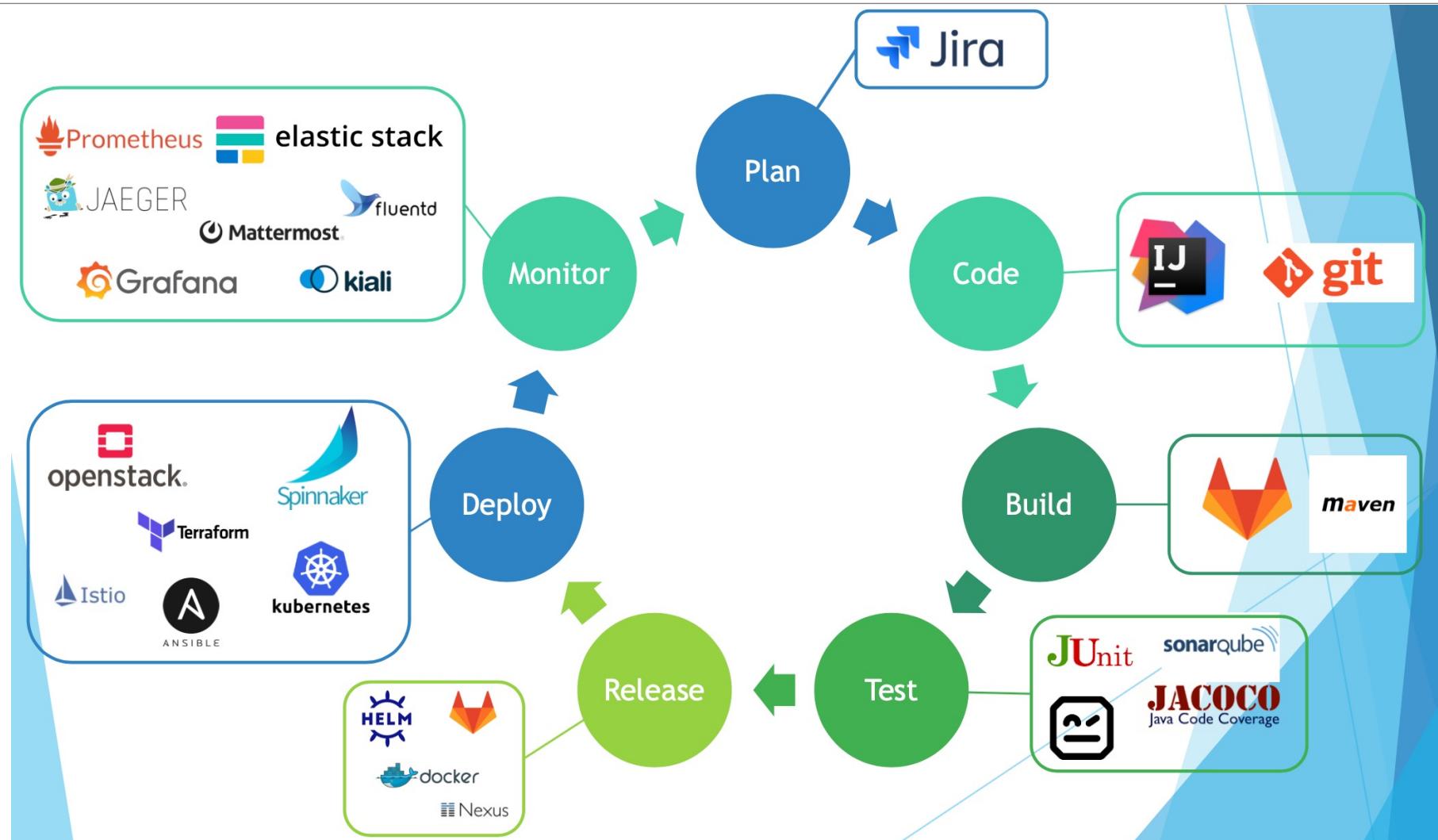
- En**: Enterprise
- Fr**: Free
- Fm**: Freemium
- Os**: Open-source
- Pd**: Paid

Tool Categories:

- AiOps/Analytics**: Aja, In, Sp, Dt, Gr
- Collaboration**: Daa, Tp, Pv, Br, Dd, Ja, Ad, Snx, Az, Gc, Ac, Ch, Dh, Np, Ic, So, Pu, Hc, Ae, Azk, Ra, Qt, Sk, Od, Cfd, Gf
- Continuous Integration**: Jenkins (Jn), Azure DevOps (Azc), GitLab CI (Glc), Travis CI (Tr), CircleCI (Cc), Maven (Mv), Atlassian Bamboo (Ab), Gradle (Gd), AWS CodeBuild (Acb), Atlassian Jira (Aj), BMC Helix ITSM (Bi), Atlassian Trello (At), ServiceNow (Sw), TOPdesk (Td), PagerDuty (Pd)
- Enterprise Agile Planning**: Targetprocess (Tp), Planview (Pv), Rally (Br)
- Security**: Digital.ai Agility (Daa), Security (Security), Testing (Testing)
- Value Stream Management**: ServerLess/Pass (ServerLess/Pass), Value Stream Management (Value Stream Management)
- Artifact Repository**: Artifactory (Ja), AWS (AWS), Slack (Sl), Microsoft Teams (Mt), Ansible (Rha), HashiCorp Terraform (Ht), Docker (Dk), Red Hat OpenShift (Rho), Liquibase (Lb), Delphix (Dp), UrbanCode Deploy (Ud), CyberArk Conjur (Ck), HashiCorp Vault (Hv), HashiCorp Release (Ur), AWS Lambda (AI), Atlassian Bitbucket (Abb)
- Configuration**: Red Hat Ansible (Rha), Chef (Ch), AWS CloudFormation (Acf), Kubernetes (Ku), Amazon EKS (Ak), Docker Enterprise (De), Idera (Id), Harness (Ha), Veracode (Vc), SonarQube (Sr), Micro Focus Fortify SCA (Ff), Azure Functions (Azf), Compuware ISPW (Ci)
- Database Automation**: Puppet (Pu), HashiCorp Consul (Hc), Amazon ECS (Ae), Azure AKS (Azk), Rancher (Ra), Quest Toad (Qt), Spinnaker (Sk), Octopus Deploy (Od), Synopsys Black Duck (Sb), Checkmarx SAST (Cx), Heroku (He), Subversion (Sv)
- ITSM/Issue Tracking**: Mattermost (Mm), Salt (Sa), HashiCorp Vagrant (Hg), HashiCorp Packer (Hp), Google GKE (Gk), Helm (Hm), DBMaestro (Db), Cloudberry CD (Cfd), AWS CodeDeploy (AcD), Snort (Sn), PortSwigger Burp Suite (Pbs), Google Firebase (Gf), Cloud Foundry (Cf)
- Containers**: Docker (Docker), Red Hat OpenShift (Red Hat OpenShift), HashiCorp Terraform (HashiCorp Terraform), Kubernetes (Kubernetes), Amazon EKS (Amazon EKS), Docker Enterprise (Docker Enterprise), Idera (Idera), Harness (Harness), Veracode (Veracode), SonarQube (SonarQube), Micro Focus Fortify SCA (Micro Focus Fortify SCA), Azure Functions (Azure Functions), Compuware ISPW (Compuware ISPW), Octopus Deploy (Octopus Deploy), Synopsys Black Duck (Synopsys Black Duck), Checkmarx SAST (Checkmarx SAST), Heroku (Heroku), Subversion (Subversion), Cloudberry CD (Cloudberry CD), AWS CodeDeploy (AWS CodeDeploy), Snort (Snort), PortSwigger Burp Suite (PortSwigger Burp Suite), Google Firebase (Google Firebase), Cloud Foundry (Cloud Foundry)
- Deployment**: Digital.ai Deploy (Dad), Nexus IQ (Sni), Aqua Security (Aq), CloudBees Flow (Cfr), BMC RLM (Brl), GitLab SCM (Gls)
- Release Orchestration**: OWASP ZAP (Ow), Digital.ai App Protection (Dap), Digital.ai Release (Dar), AWS CodePipeline (Acp)
- Source Control Management**: Git (Gi), GitHub (Gh), GitLab SCM (Gls)

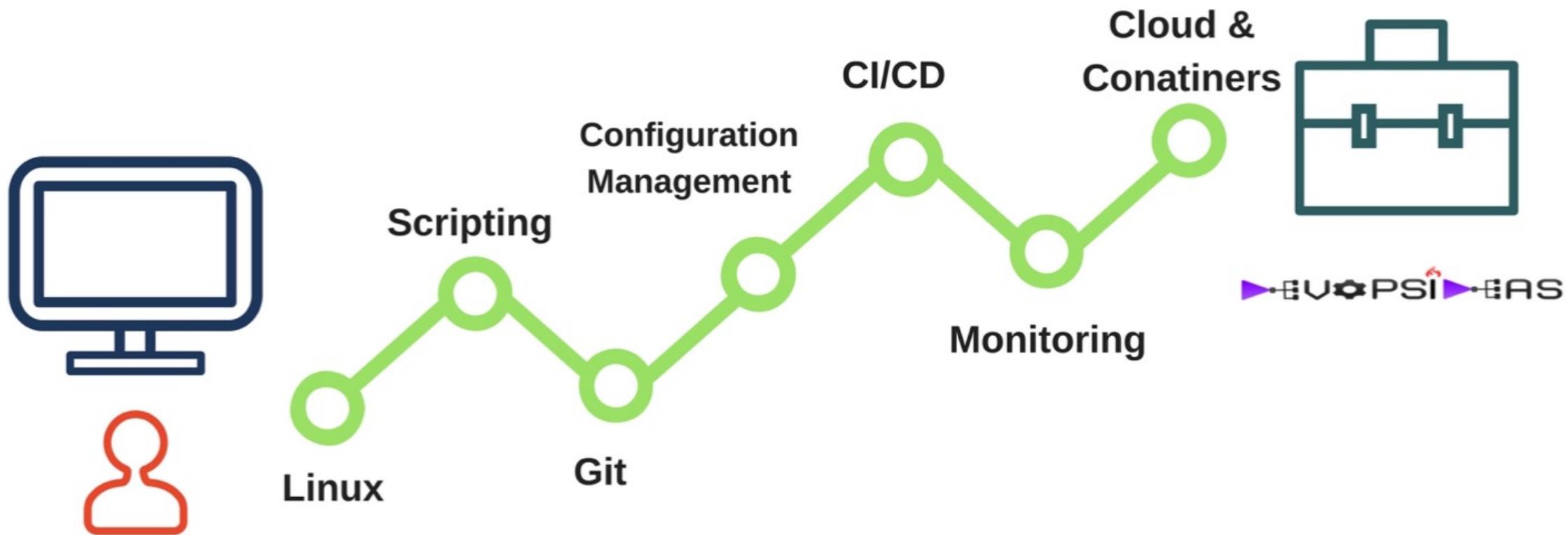
1 En	Aja Atlassian Jira Align	2 Os	Gi Git																																
3 En	Daa Digital.ai Agility	4 En	Tp Targetprocess	5 En	Azp Azure DevOps Pipelines	6 Os	Ow OWASP ZAP	7 En	Dap Digital.ai App Protection	8 En	Dar Digital.ai Release	9 En	Acp AWS CodePipeline	10 Os	Gh GitHub																				
11 En	Pv Planview	12 En	Br Broadcom Rally	13 En	Dad Digital.ai Deploy	14 En	Sni Sonatype Nexus IQ	15 En	Aq Aqua Security	16 En	Cfr CloudBees Flow	17 En	Brl BMC RLM	18 Os	Gls GitLab SCM																				
19 Pd	In Instana	20 En	Dd Datadog	21 En	Ja JFrog Artifactory	22 En	Aws AWS	23 En	Sl Slack	24 En	Mt Microsoft Teams	25 Os	Rha Red Hat Ansible	26 Os	Ht HashiCorp Terraform	27 Os	Dk Docker	28 En	Rho Red Hat OpenShift	29 Os	Lb Liquibase	30 Fm	Dp Delphix	31 En	Ud UrbanCode Deploy	32 En	Ck CyberArk Conjur	33 Os	Hv HashiCorp Vault	34 En	Ur UrbanCode Release	35 En	AI AWS Lambda	36 Fm	Abb Atlassian Bitbucket
37 En	Sp Splunk	38 En	Ad AppDynamics	39 Os	Snx Sonatype Nexus	40 En	Az Azure	41 En	Gc Google Cloud	42 En	Ac Atlassian Confluence	43 Os	Ch Chef	44 En	Acf AWS CloudFormation	45 Os	Ku Kubernetes	46 En	Ak Amazon EKS	47 En	De Docker Enterprise	48 En	Id Idera	49 En	Ha Harness	50 En	Vc Veracode	51 Os	Sr SonarQube	52 En	Ff Micro Focus Fortify SCA	53 En	Azf Azure Functions	54 En	Ci Compuware ISPW
55 En	Dt Dynatrace	56 En	Nr New Relic	57 Fm	Dh Docker Hub	58 En	Np npm	59 En	Ic IBM Cloud	60 En	So Stack Overflow	61 Fm	Pu Puppet	62 Os	Hc HashiCorp Consul	63 En	Ae Amazon ECS	64 En	Azk Azure AKS	65 Os	Ra Rancher	66 Fm	Qt Quest Toad	67 Os	Sk Spinnaker	68 En	Od Octopus Deploy	69 En	Sb Synopsys Black Duck	70 En	Cx Checkmarx SAST	71 Fm	He Heroku	72 Os	Sv Subversion
73 Os	Gr Grafana	74 Os	EI Elastic ELK Stack	75 Os	Yn Yarn	76 Os	Nu NuGet	77 Os	Os OpenStack	78 Os	Mm Mattermost	79 Os	Sa Salt	80 Os	Hg HashiCorp Vagrant	81 Os	Hp HashiCorp Packer	82 En	Gk Google GKE	83 Os	Hm Helm	84 En	Db DBMaestro	85 En	Cfd Cloudberry CD	86 En	AcD AWS CodeDeploy	87 Os	Sn Snort	88 Fm	Pbs PortSwigger Burp Suite	89 En	Gf Google Firebase	90 Os	Cf Cloud Foundry
91 Os	Jn Jenkins	92 En	Azc Azure DevOps Code	93 Os	Glc GitLab CI	94 Os	Tr Travis CI	95 Fm	Cc CircleCI	96 Os	Mv Maven	97 Pd	Ab Atlassian Bamboo	98 Os	Gd Gradle	99 En	Acb AWS CodeBuild	100 Os	Aj Atlassian Jira	101 En	Bi BMC Helix ITSM	102 En	At Atlassian Trello	103 En	Sw ServiceNow	104 En	Td TOPdesk	105 Os	Pd PagerDuty						
106 Fr	Tt Tricentis Tosca	107 Pd	Nn Neotys NeoLoad	108 Fr	Se Selenium	109 Fr	Ju JUnit	110 Pd	Sl Sauce Labs	111 Pd	Ct Compuware Topaz	112 En	Ap Appium	113 Os	Sq Squash TM	114 Fr	Cu Cucumber	115 Fr	Jm JMeter	116 Pd	Pa Parasoft	117 En	Dai Digital.ai	118 En	Tp Tasktop	119 En	Pr Plutora	120 Os	Gl GitLab						

DevOps Toolchain Example



Becoming a DevOps Engineer

Skills for all functions

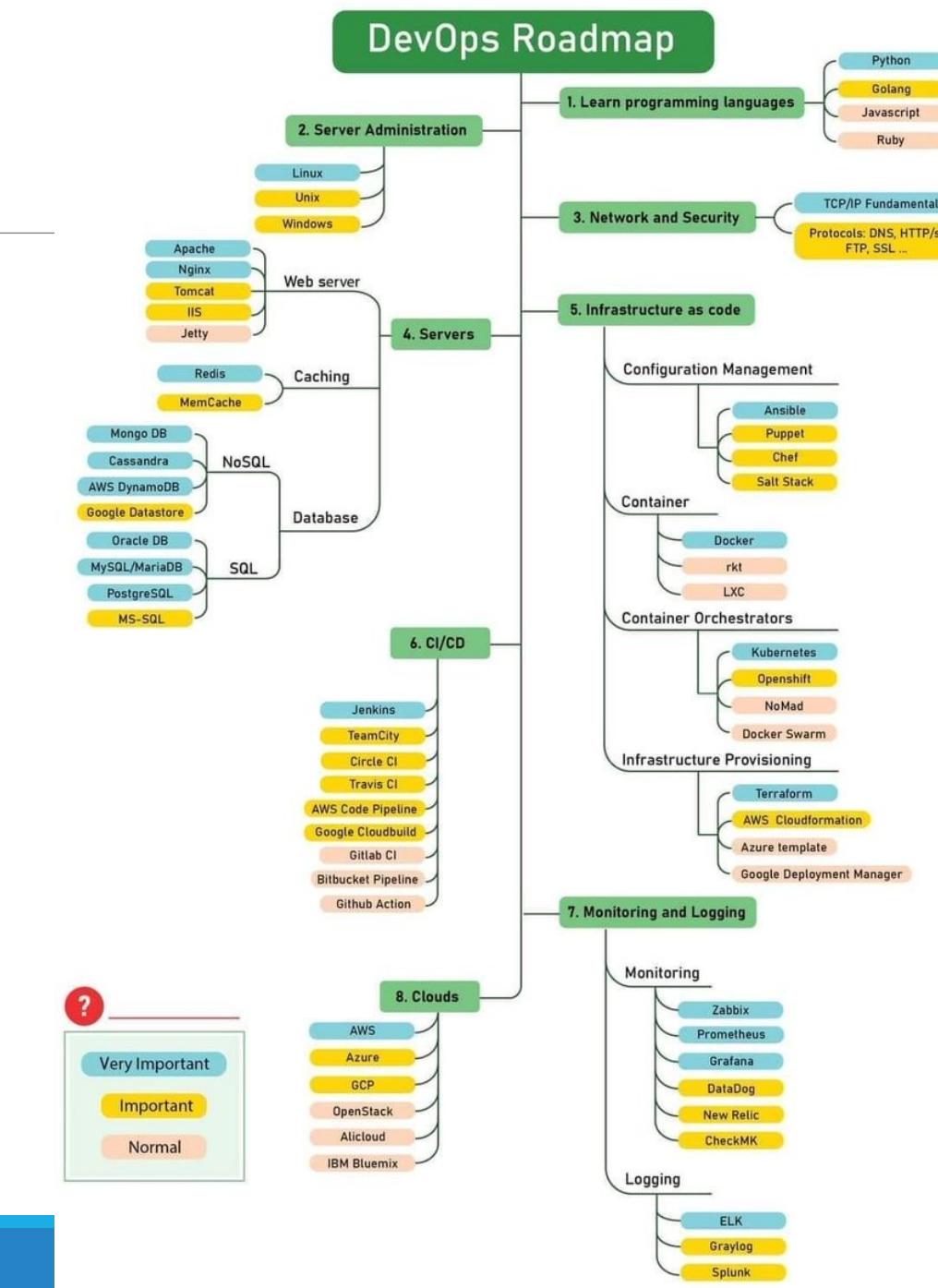


Specific Skills per Function/Role

Developer	Tester	Cloud Architect / Engineer	DevOps Engineer
<ul style="list-style-type: none">• TDD• BDD• Static Code Analysis• Serverless Architectures• SMED (e.g. Vagrant)	<ul style="list-style-type: none">• BDD• Test Automation at all levels and of all types• Chaos Engineering	<ul style="list-style-type: none">• Solutions Architecture for major cloud vendors• Cost optimization	<ul style="list-style-type: none">• IaC• Configuration Management Tools• Chaos Engineering• Infrastructure Testing Tools (Server Spec.)

DevOps Roadmap

DevOps Roadmap



Apply DevOps

1. Build the DevOps team
2. Design the DevOps methodology
3. Design the software architecture
4. Design the DevOps Pipeline
5. Start building the DevOps pipeline
6. Start building the software solution
7. Deliver, measure and iterate with improvement

Example of Software Architecture

- A Cloud-native application
- Serverless Architectures using Google Cloud Functions or classic monolithic or microservice architecture

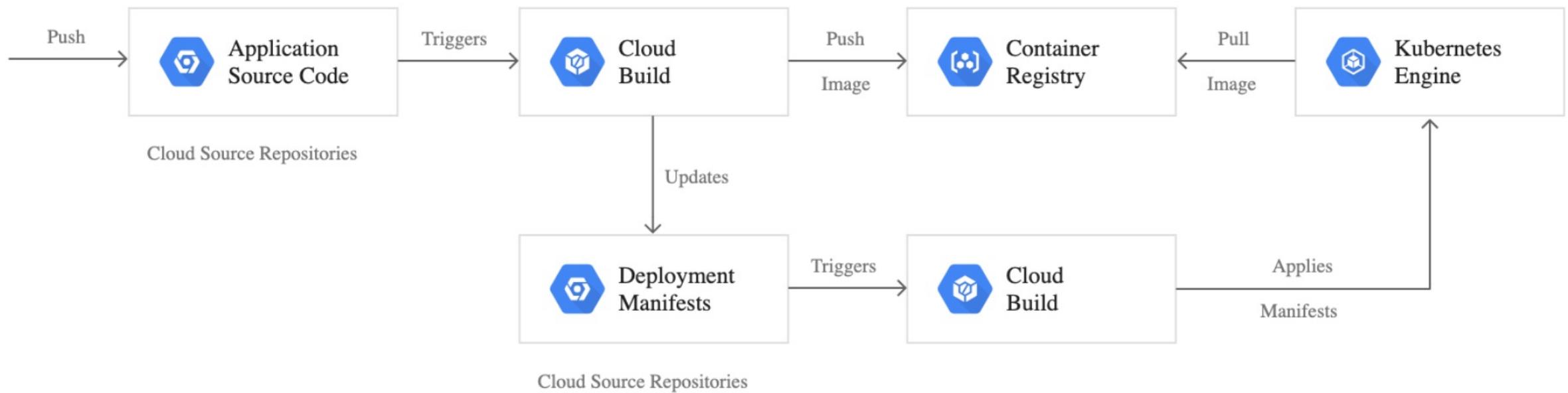
Example of DevOps Pipeline

- An "All Google" pipeline

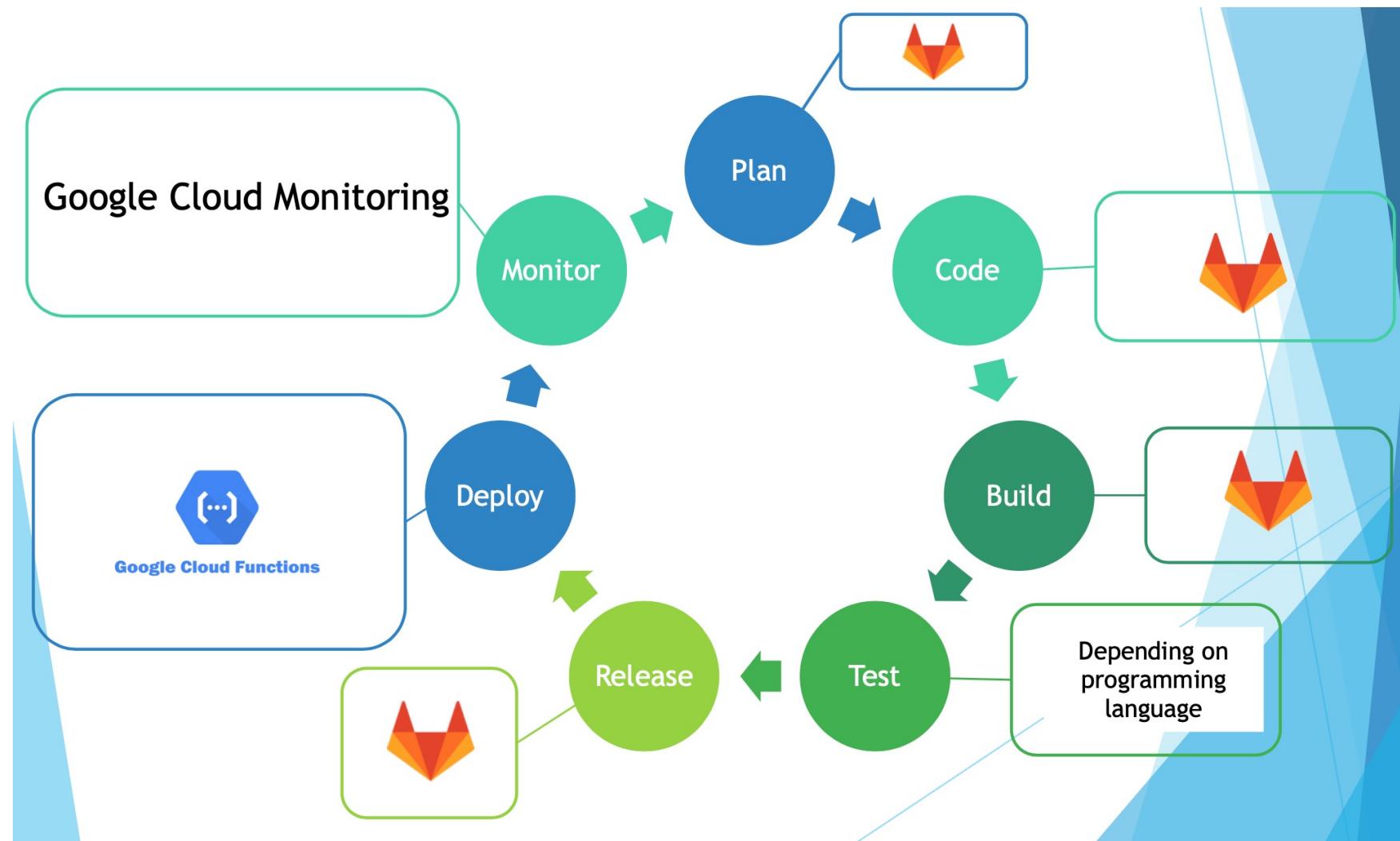
OR

- A heterogenous pipeline

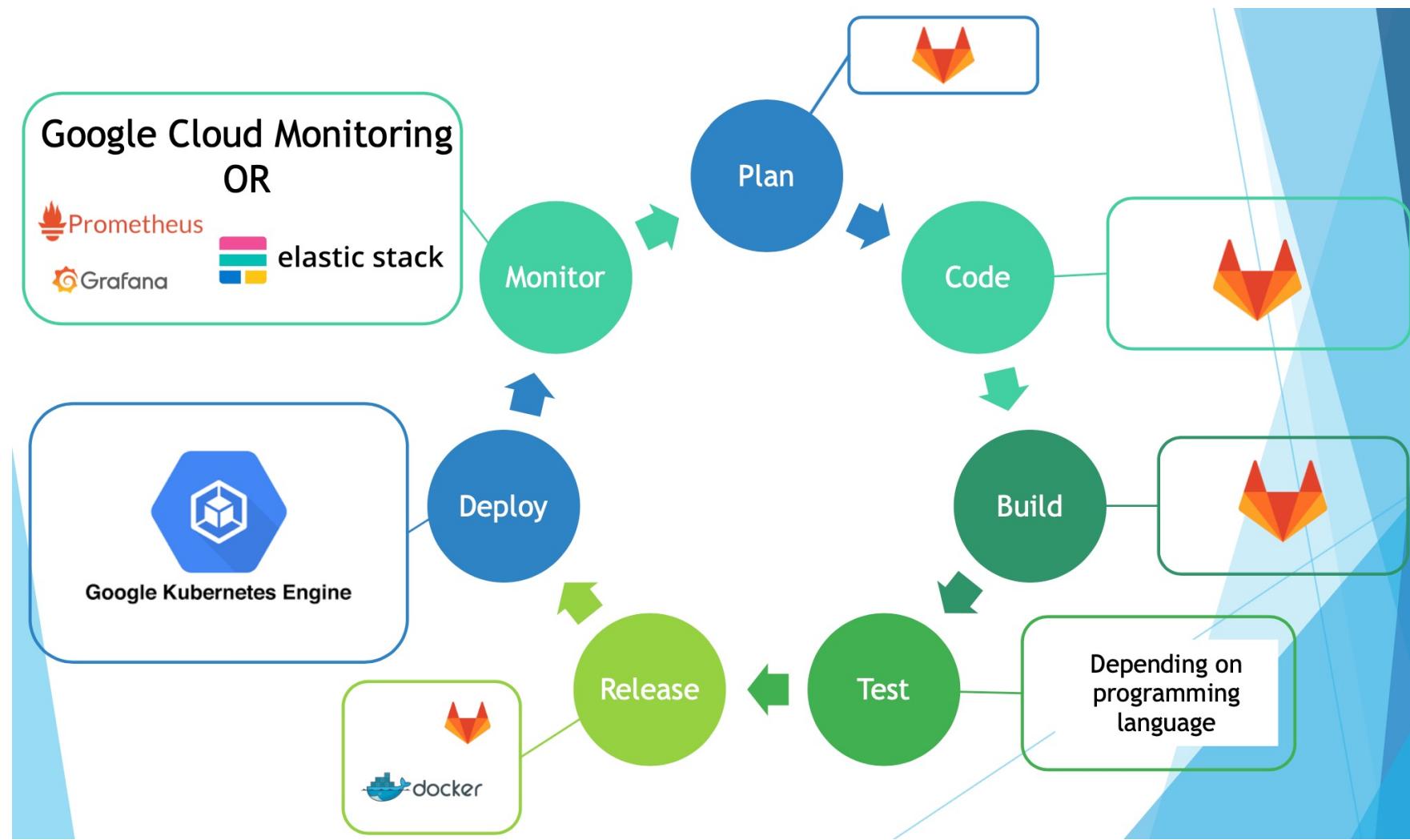
Example of Google Pipeline



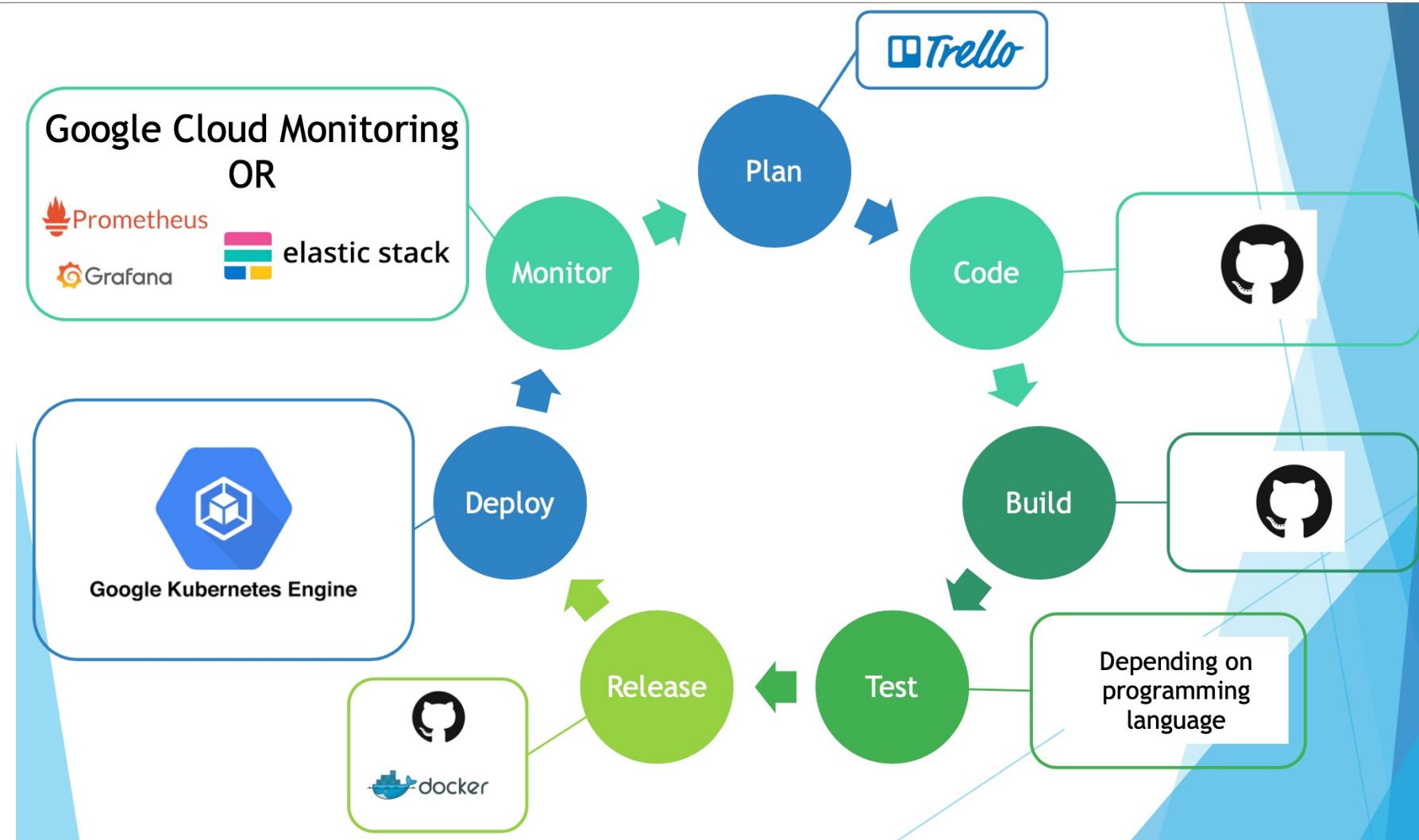
Hetrogenous Pipeline (1)



Hetrogenous Pipeline (2)



Hetrogenous Pipeline (3)

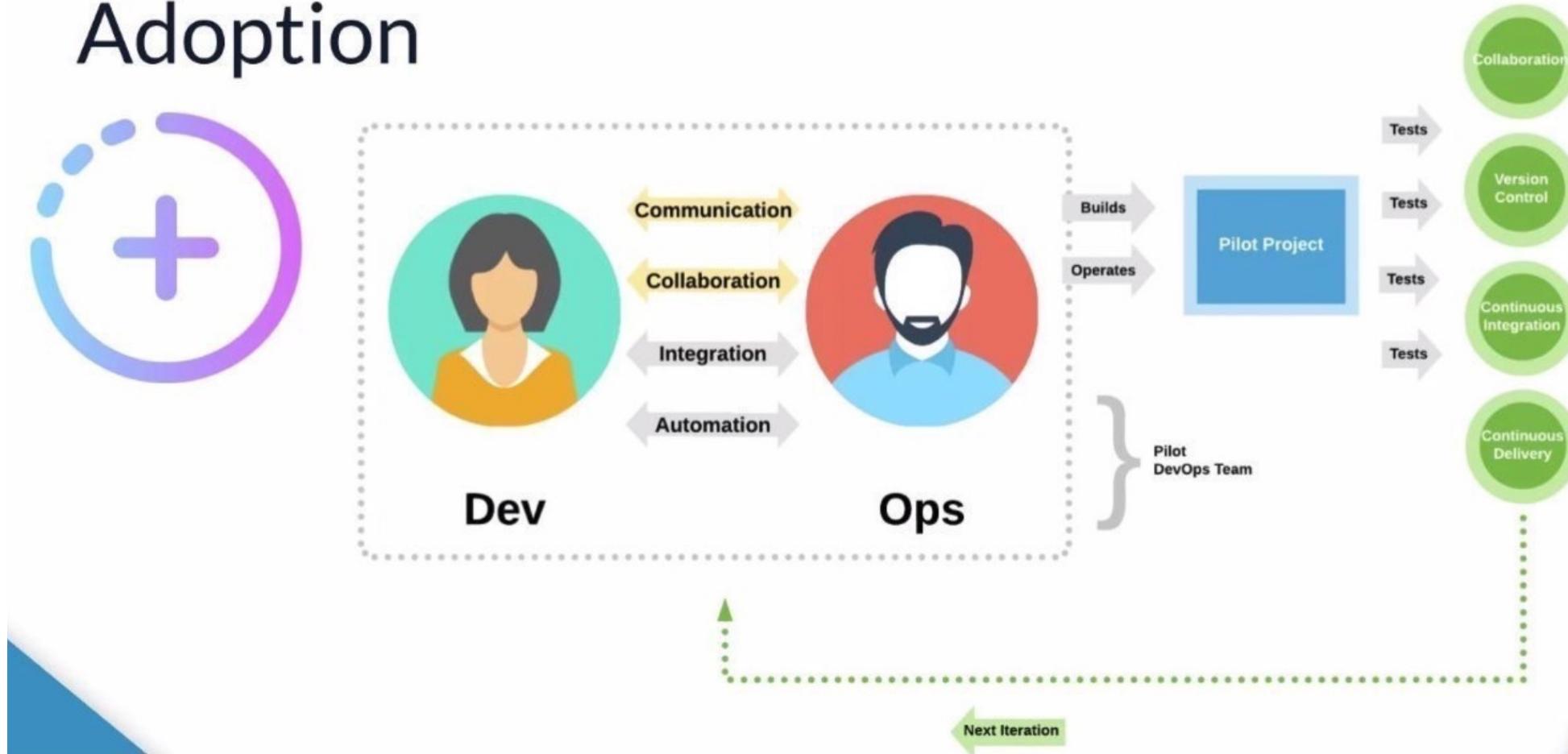


Measure and Improve

- Conduct Agile retrospectives
- Measure efficiency of DevOps framework and pipeline using CALMS and metrics
- Continuously improve!

DevOps Adoption Overview

Adoption



References

- DevOps for beginners, Ahmed MISBAH, December 7th, 2020
- Getting Started with DevOps, Ahmed MISBAH, December 19th, 2020

