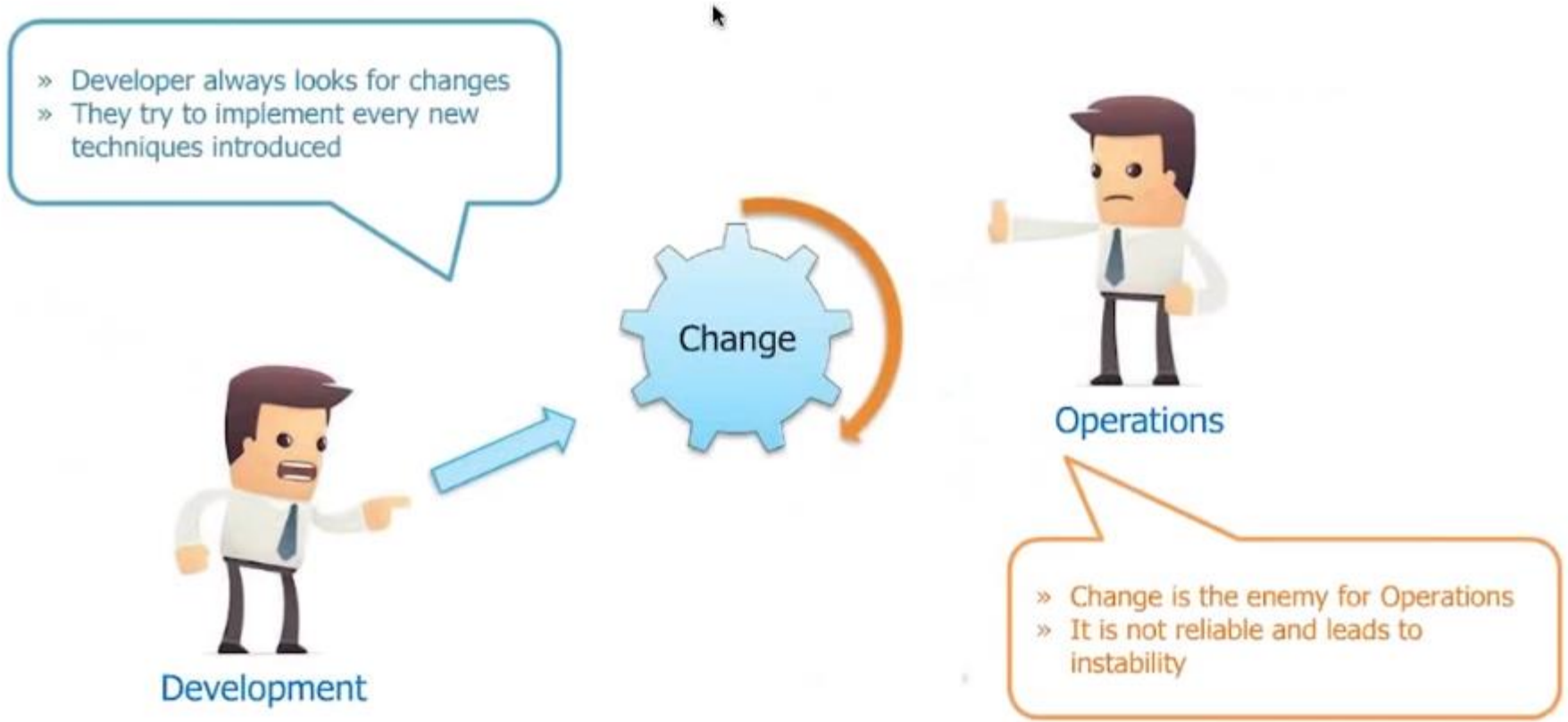
DevOps Overview

Agenda

- DevOps Principles
  - DevOps vs Traditional IT
  - DevOps vs Agile
- Tenants of DevOps
- DevOps Automation Tools
  - Source Control Tools
  - Continuous Integration Tools
    - Build and Test
    - Integration
  - Continuous Delivery and Deployment Tools
    - Test Tools
  - Change Management Tools

DevOps

Dev-Ops Problem



What is Dev-Ops

DevOps is the practice of operations & development engineers participating together in the entire service lifecycle.

Developers	Operations
• Create change • Add/Modify features • Don’t deploy consistent software	• Create stability • Create or enhance services • Resist change

Why DevOps

1. Predictability:	Offers significantly lower failure rate of new releases
2. Reproducibility:	Version everything so that earlier version can be restored anytime.
3. Maintainability:	Effortless process of recovery in the event of a new release crashing or disabling the current system.
4. Time to market:	Reduces the time to market up to 50% through streamlined software delivery. This is particularly the case for digital and mobile applications.

Why DevOps

5. Greater Quality:

DevOps helps the team to provide improved quality of application development as it incorporates infrastructure issues.

6. Reduced Risk:

DevOps incorporates security aspects in the software delivery lifecycle. It helps in reduction of defects across the lifecycle.

7. Resiliency:

The Operational state of the software system is more stable, secure, and changes are auditable.

8. Cost Efficiency:

DevOps offers cost efficiency in the software development process which is always an aspiration of IT companies' management.

9. Breaks larger code base into small pieces:

DevOps is based on the agile programming method. Therefore, it allows breaking larger code bases into smaller and manageable chunks.

The converged DevOps lifecycle

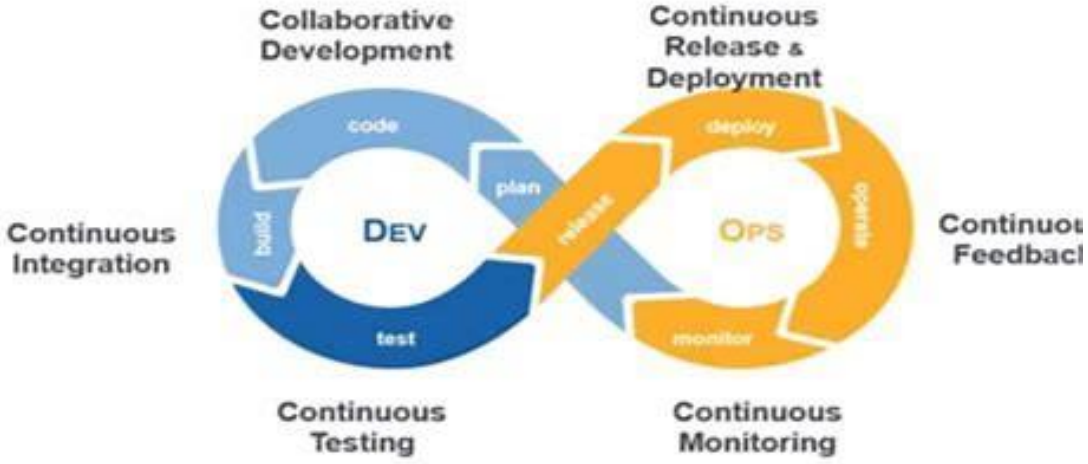
What is DevOps

- DevOps is not a goal, but a never ending process of continual improvement.
- DevOps = Tools + best practices
- Bridge the gap between the development and operations teams
- Continuous integration and deployment
- Integrated set of tools to automate the software delivery.

- Goal of DevOps
  - Speed of delivering applications to the end-users
  - Enabling faster end-user feedback

Aspects of DevOps

- People:
  - Highly motivated team of trained people
  - Effectively communicate and collaborate through entire journey of cultural change.
- Process:
  - Practices and strategies which provide value to the customer.
  - Look at gaps and propose a roadmap for implementation of giving appropriate recommendations.
- Tools:
  - Using the accelerators by automating the process
  - Open-Source (Jenkins, Git etc.)
  - Commercial (Microsoft Azure DevOps, IBM Rational, Jira etc.) or a mix of both.



Agile and DevOps - Complement each other

Agile

Develop a software in smaller sprints or iteration

Help the development teams to work on the end-user feedback and incorporate the changes in the newer releases.

Processes like XP, SCRUM etc.

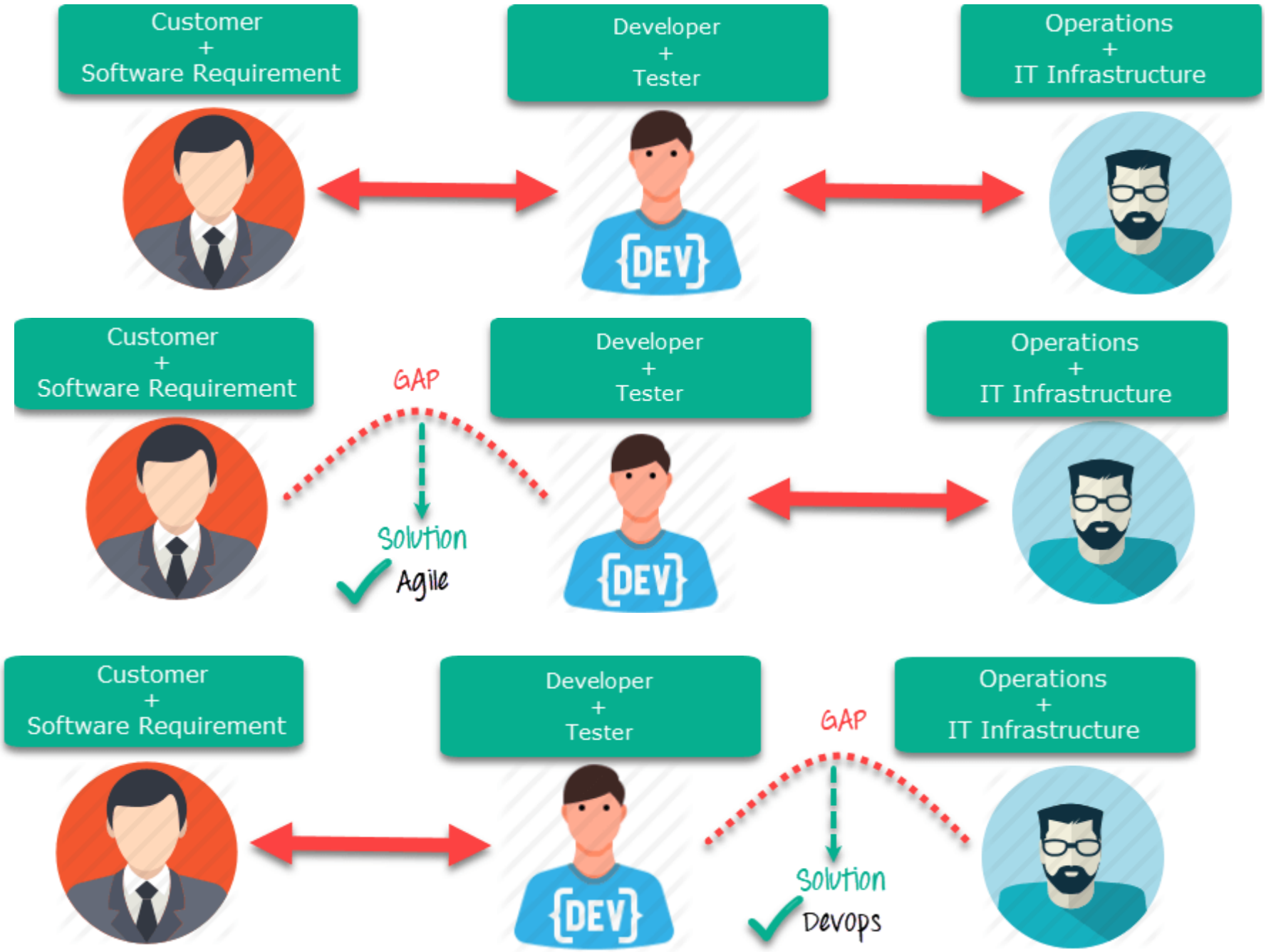
DevOps

Development and operation teams must have agile in their areas of work

Enable better collaboration between them.

Practices like Continuous Integration (CI), Continuous Delivery (CD), Continuous Testing (CT) and Continuous Monitoring (CM)

Agile vs DevOps



Agile	DevOps
Emphasize breaking down barriers between developers and management.	DevOps is about software deployment and operation teams.
Addresses gap between customer requirements and development teams.	Addresses the gap between development and Operation team
Focuses more on functional and non-functional readiness	It focuses operational and business readiness.
Agile development pertains mainly to the way development is thought out by the company.	DevOps emphases on deploying software in the most reliable and safest ways which aren't necessarily always the fastest.
Agile development puts a huge emphasis on training all team members to have varieties of similar and equal skills. So that, when something goes wrong, any team member can get assistance from any member in the absence of the team leader.	DevOps, likes to divide and conquer, spreading the skill set between the development and operation teams. It also maintains consistent communication.
Agile development manages on "sprints. It means that the time table is much shorter (less than a month) and several features are to be produced and released in that period.	DevOps strives for consolidated deadlines and benchmarks with major releases, rather than smaller and more frequent ones.

Agile Development with DevOps Culture

- Streamlined Deliveries
- Team work with Collaboration
- Continuous monitoring and Feedback

Components of DevOps

DevOps Adoption Journey

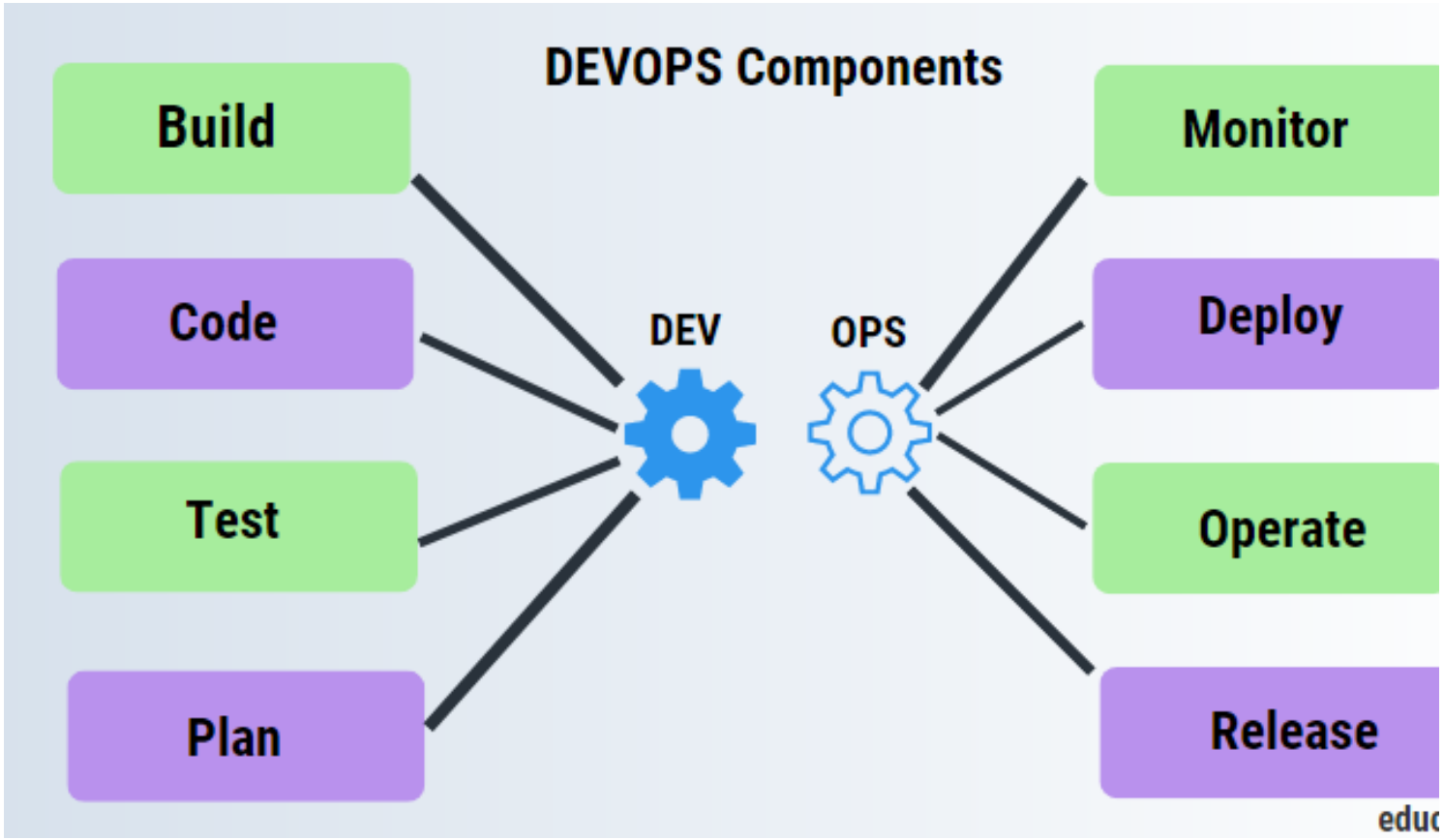
Cultural Transformation

Process Transformation

Technology Transformation

Automate everything

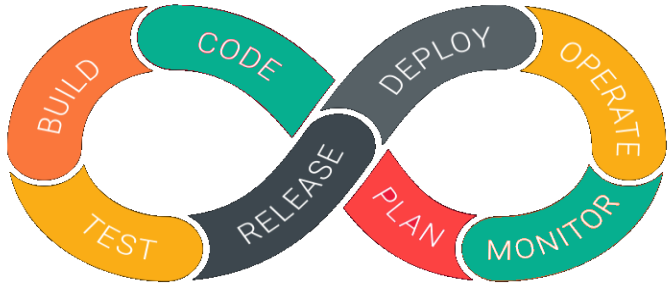
Adoption of Tools



Components of DevOps

- Continuous Integration
- Continuous Testing
- Continuous Delivery
- Continuous Monitoring

DevOps Lifecycle



1. Development	Development of software takes place constantly in small development cycles
2. Testing	QA team use tools like Selenium to identify and fix bugs in the new piece of code.
3. Integration	New functionality is integrated with the prevailing code, and testing takes place. Continuous development is only possible due to continuous integration and testing.
4. Deployment	Deployment process takes place continuously. Any changes made any time in the code, should not affect the functioning of high traffic website.
5. Monitoring	Operation team will take care of the inappropriate system behavior or bugs which are found in production.

Microsoft Ecosystem

Developer Workstations

Source

Visual Studio Online

Team Foundation Server

Develop

Visual Studio Online

Team Foundation Server

Microsoft Test Manager

Microsoft Monitoring Agent

Test

Visual Studio Online

Team Foundation Server

Release Management for Visual Studio

Build

Microsoft System Center

Release Management for Visual Studio

Automation Service

PowerShell

MAML

Microsoft Azure

xPlat CLI

Azure Resource Management

Deploy

Microsoft .NET

Windows Server

Microsoft SQL Server

Production / Stage

Dev/Test

Environments

Microsoft System Center

Visual Studio Online

Application Insights

Monitor and Learn

Team Foundation Server

Visual Studio Online

Release Management for Visual Studio

Microsoft System Center

Processes

Heterogeneous Ecosystem

Developer Workstations

Source

GitHub
Codeplex

Vagrant

docker

Develop

Test

Grunt

Gradle

Build

Chef

Puppet Labs

Hudson

Saltstack

Deploy

nodeJS

Java

php

python

Vagrant

docker

Production / Stage

Dev / Test

Environments

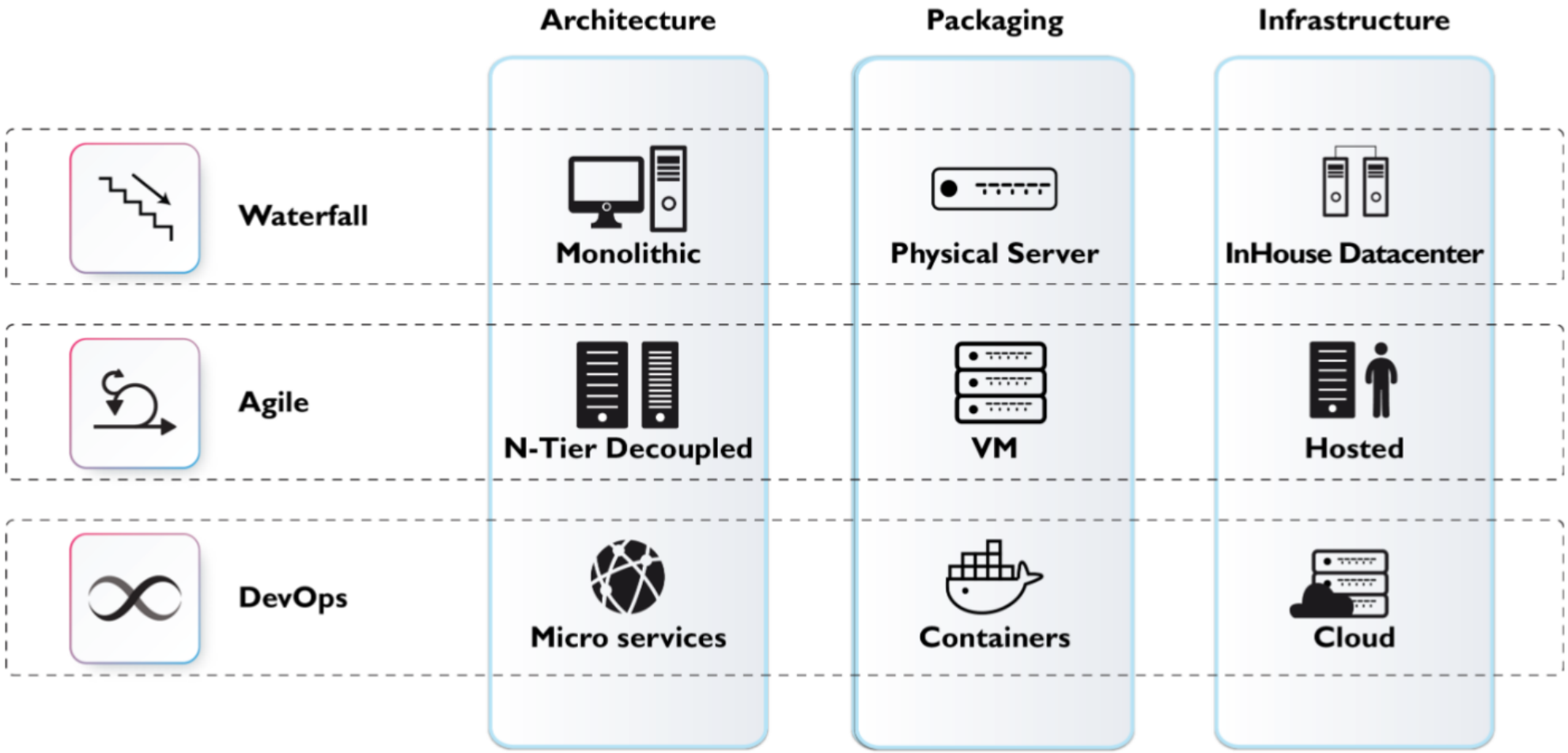
Nagios

Zabbix

Monitor and Learn

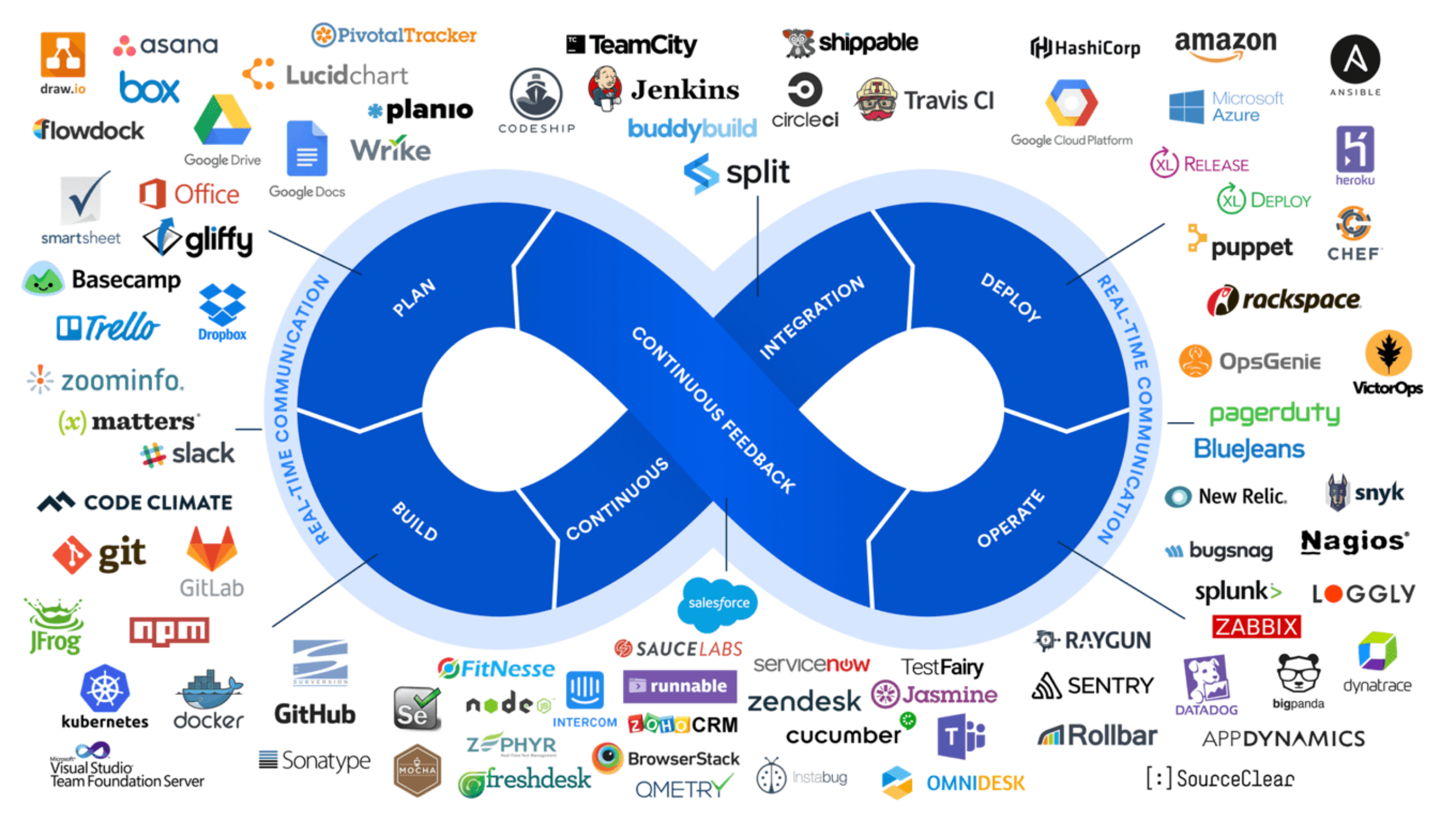
Processes

Reactive And Scalable Infrastructure With DevOps



Skills Of A DevOps Engineer

Skills	Description
Tools	• Version Control - GIT • Continuous Integration - Jenkins • Virtualization/ Containerization – Docker/Kubernetes • Configuration Management – Puppet/Ansible • Monitoring – Nagios
Networking Skills	• General networking skills – Establishing connection between the containers/Port Forwarding/ Container Orchestration
Other Skills	• People Skills • Process Skill • Customer Skill and Empathy • Cloud Awareness


draw.io
asana
PivotalTracker
box
Lucidchart
planio
flowdock
Google Drive
Google Docs
Wrike
TeamCity
CODESHIP
Jenkins
buddybuild
shippable
circleci
Travis CI
HashiCorp
amazon
Google Cloud Platform
Microsoft Azure
ANSIBLE
RELEASE
heroku
DEPLOY
CHEF
puppet
rackspace
split
Office
smartsheet
gliffy
Basecamp
Trello
Dropbox
zoominfo.
(x) matters
slack
CODE CLIMATE
git
GitLab
JFrog
npm
kubernetes
docker
GitHub
Sonatype
Visual Studio Team Foundation Server
PLAN
BUILD
CONTINUOUS
CONTINUOUS FEEDBACK
INTEGRATION
DEPLOY
OPERATE
REAL-TIME COMMUNICATION
REAL-TIME COMMUNICATION
OpsGenie
VictorOps
pagerduty
BlueJeans
New Relic.
snyk
bugsnag
Nagios
splunk>
LOGGLY
ZABBIX
RAYGUN
SENTRY
DATADOG
bigpanda
dynatrace
Rollbar
APPDYNAMICS
[:] SourceClear
salesforce
SAUCELABS
FitNesse
runnable
servicenow
TestFairy
zendesk
Jasmine
cucumber
INTERCOM
ZOHO CRM
ZEPHYR
BrowserStack
freshdesk
QMETRY
Instabug
OMNIDESK
MOCHA

DevOps Tools

- **Software Containers** – JBoss, Tomcat, Jenkins
- **Build Tools** – Ant, Rake, Maven
- **Code Review & Insight tools** – Crucible, Fisheye
- **Code Insight** – Fisheye, Sonar
- **Continuous Integration** – Jenkins, Bamboo
- **Cloud IaaS & PaaS tools** – Microsoft Azure, Google App Engine, AWS
- **Database & DB management tool** – Oracle, MySQL, Liquibase, SQL Server
- **Infrastructure Automation** – Puppet, Chef
- **Dependency Management** – Nexus, Maven
- **Deployment Automation** – Java Secure Channel, Fabric
- **Integrated Development Environment (IDE)** – Eclipse, Visual Studio
- **Issue Tracking** – JIRA, Greenhopper
- **Provisioning tools** – Eucalyptus
- **Monitoring** – CloudKick, Zabbix, Nagios
- **Testing** – AntUnit, Cucumber, JMeter, SoapUI, Selenium
- **Version-Control System** – GIT, SVN/Subversion, Perforce