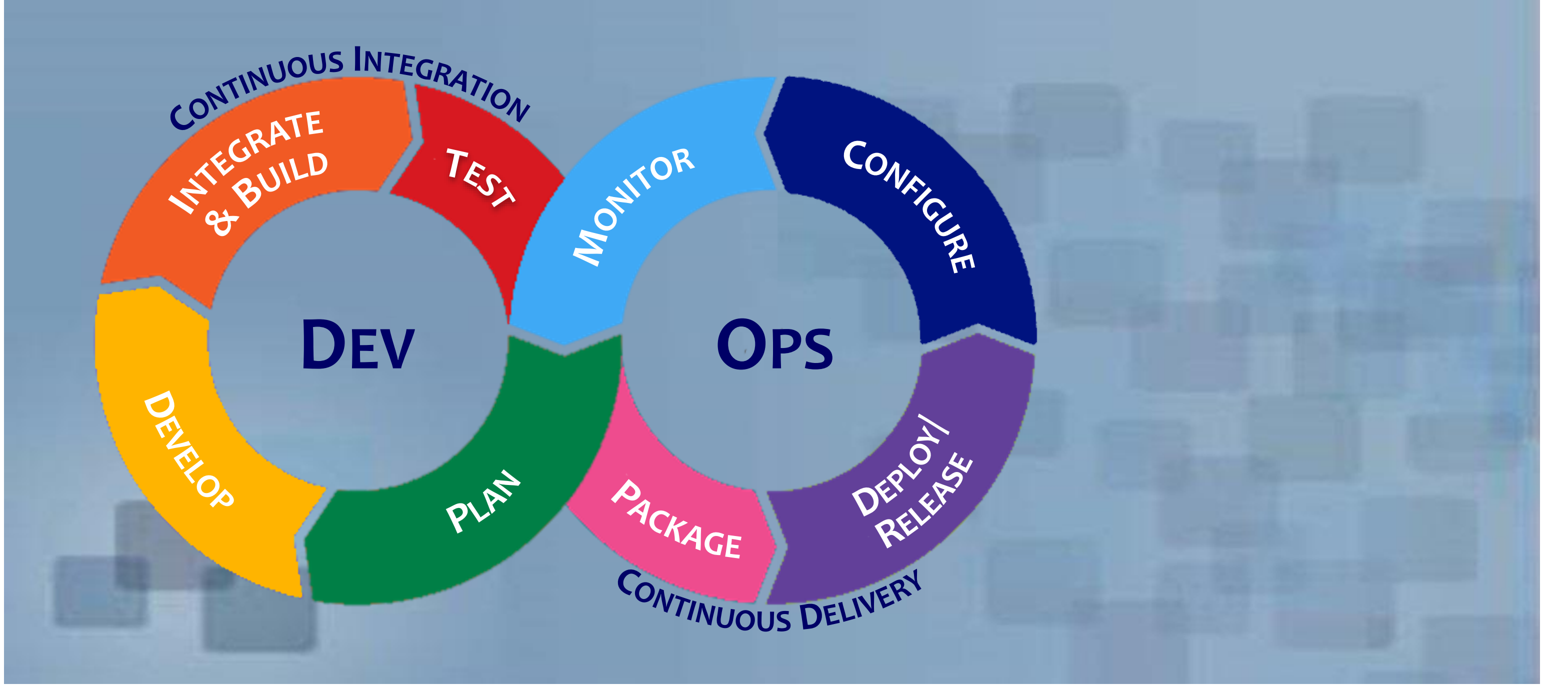
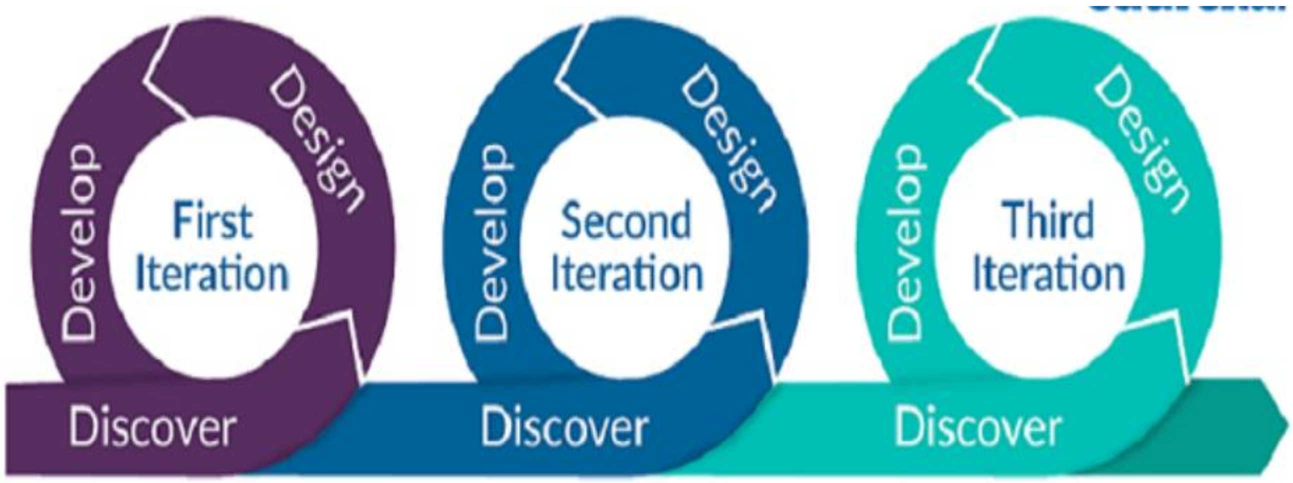

Continuous Integration
Integrate & Build
Test
Develop
Dev
Plan
Monitor
Configure
Ops
Package
Deploy| Release
Continuous Delivery

DevOps

Waterfall model

- A sequential process in the development of a system or software that follows a top-down approach
- Unless you complete a particular phase, you could not proceed to the next phase, i.e., after Req analysis, Design, then Development and testing followed by Deployment.
- Also, the working software was delivered only after the final phase of this model.
- The major drawback is that requirements keep on changing from time to time and this model suffers a lot for that

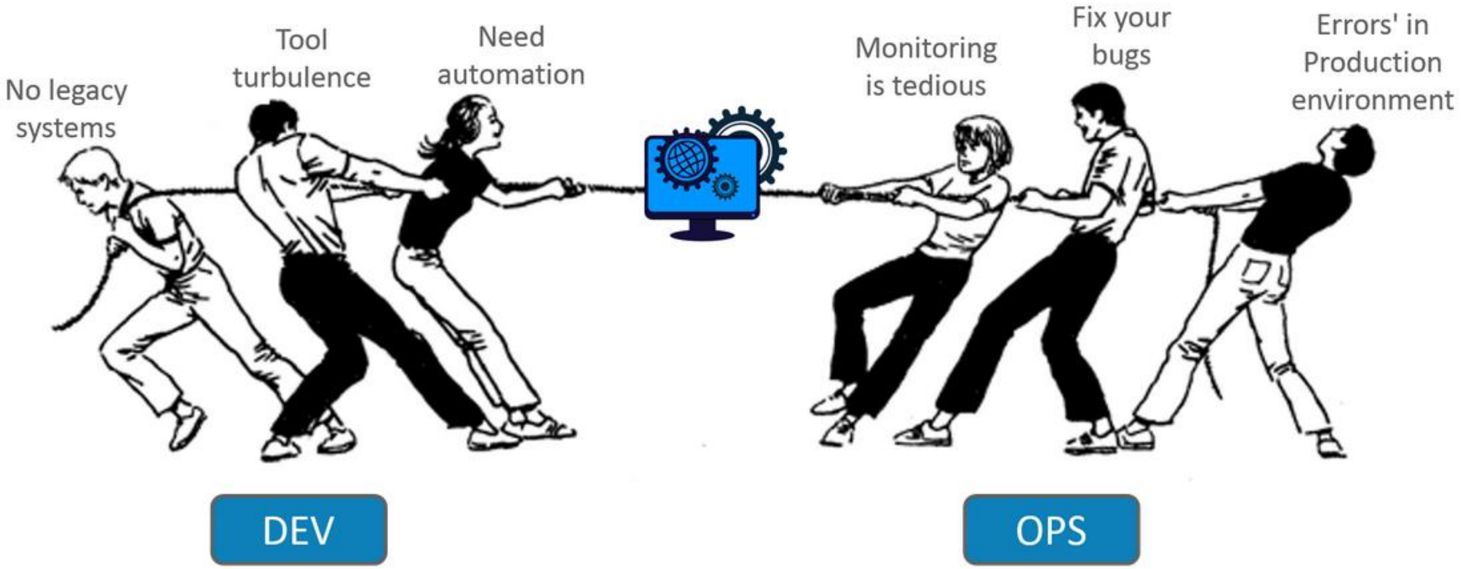
Agile methodlogy



It encourages continuous iteration of development and testing throughout the software development life cycle of the project.

Issues evolved with Agile

- There was a lack of collaboration between Developers and Operation Engineers and this slowed down the development process and releases.
- Software companies had begun to realize the need for better collaboration between the teams and faster delivery of software.
- This gave birth to the DevOps approach.
- DevOps enabled continuous software delivery with less complex problems to fix and faster resolution of problems.


No legacy systems
Tool turbulence
Need automation
Monitoring is tedious
Fix your bugs
Errors' in Production environment
DEV
OPS

What is DevOps?

A cross-disciplinary community of practice dedicated to the study of building, evolving and operating rapidly-changing resilient systems at scale. –***Jez Humble***

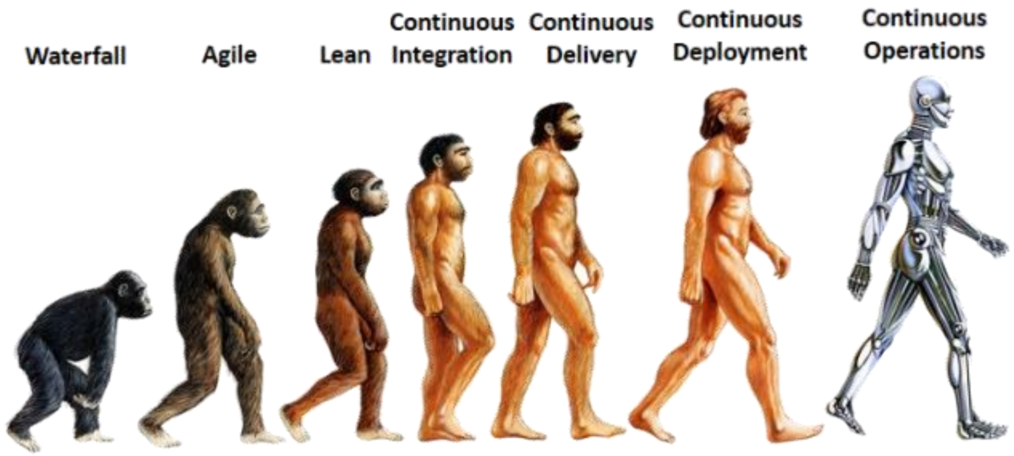
An IT mindset encouraging communication, collaboration, integration and automation among software developers and IT operations to improve the speed and quality of delivering software. –***VersionOne***

A set of practices and cultural changes — supported by the right tools — that creates an automated software delivery pipeline, enabling organizations to win, serve, and retain customers. ***–Forrester***

A professional movement advocating a collaborative working relationship between Development & IT Operations, resulting in the fast flow of planned work, while simultaneously increasing the reliability, stability, resilience & security of the production environment. ***– Gene Kim***

History of DevOps

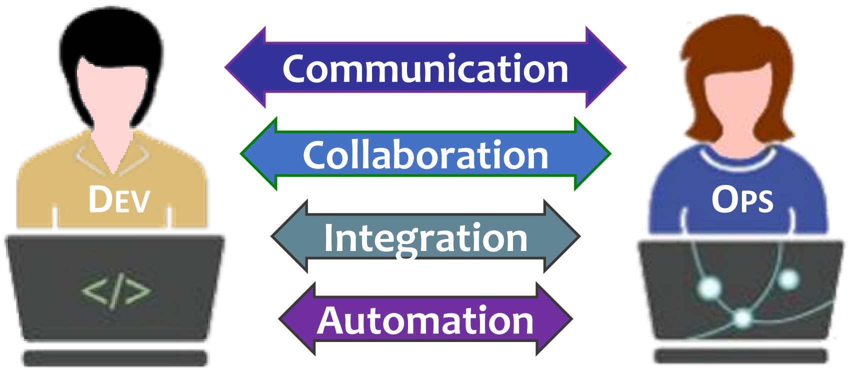
- Patrick Debois, a Belgian consultant, project manager, and agile practitioner is one among the initiators of DevOps.
- A presentation on "10+ Deploys per Day: Dev and Ops Cooperation at Flickr" helped in bring out the ideas for DevOps and resolve the conflict of " It's not my code, it's your machines! "
- DevOps blends lean thinking with agile philosophy.


Waterfall
Agile
Lean
Continuous Integration
Continuous Delivery
Continuous Deployment
Continuous Operations

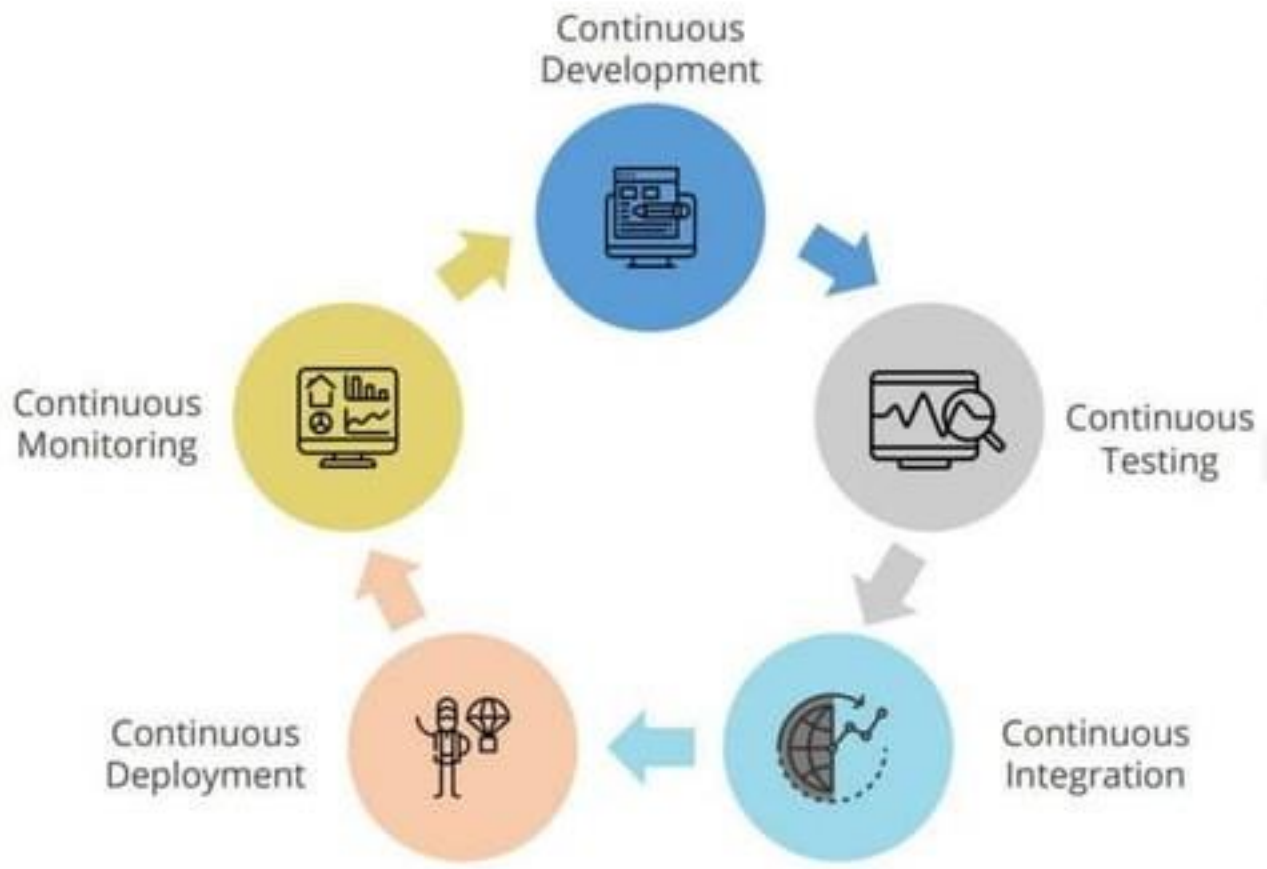
DevOps is …

A professional cultural movement/philosophy/mindset emphasizing …

- Continuous collaboration between development & operations
- Automated CI/CD pipelines, working in small-batches, with shorter lead-times (frequent deployment), and low failure-rates.
- Agile (coding & automation) practices applied to infrastructure, configuration, deploying/releasing, and monitoring.



DevOps Architecture



Continuous Integration

- This stage is the heart of the entire DevOps life cycle. Since there is continuous development of software, the updated code needs to be integrated continuously as well as smoothly with the systems to reflect changes to the end-users.
- It is a software development practice in which the developers require to commit changes to the source code more frequently. This may be on a daily or a weekly basis.
- Every commit is then built and this allows early detection of problems if they are present.
- Building code not only involves compilation but it also includes code review, unit testing, integration testing, and packaging.

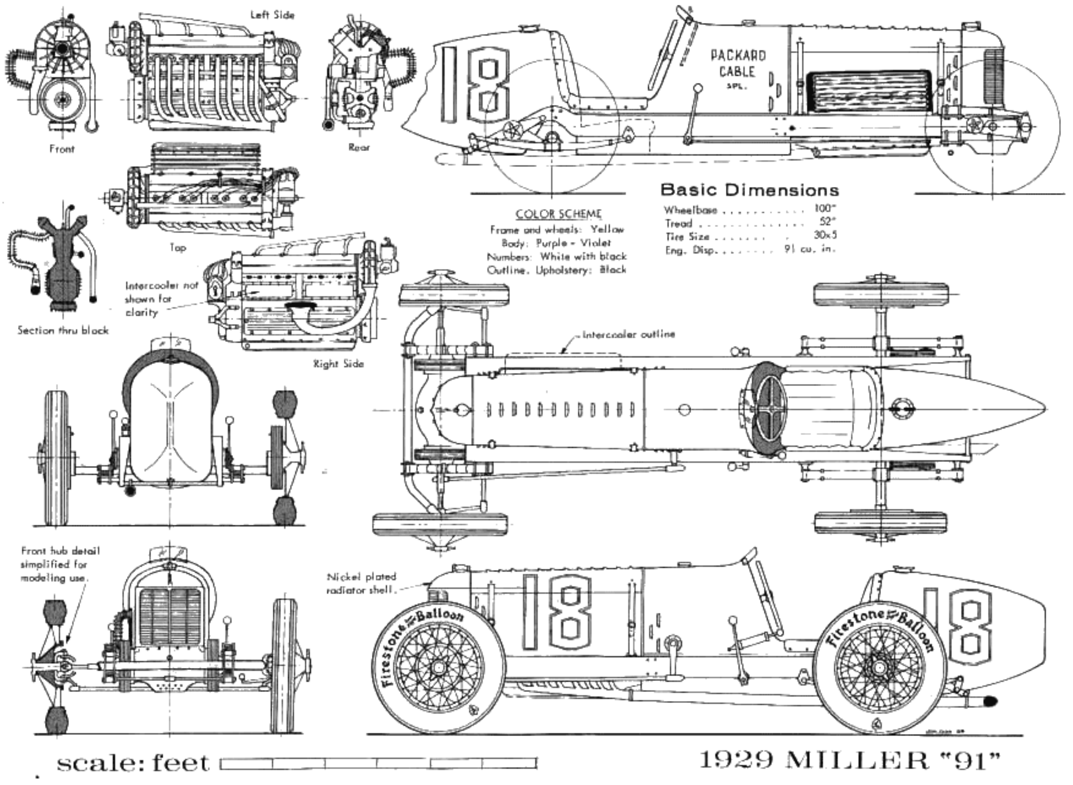
Continuous Deployment (CD)

CD is the stage where the code is deployed to the production servers.

Traditional Development

The Inventors

- Create new features and functionality in “dev” environment
- Occasionally deliver new product to operators, along with instructions
- May incorporate feedback from operators in future deliveries
- Rewarded for delivering new features



The inventors are responsible for changing the system

Traditional Operations

The Mechanics

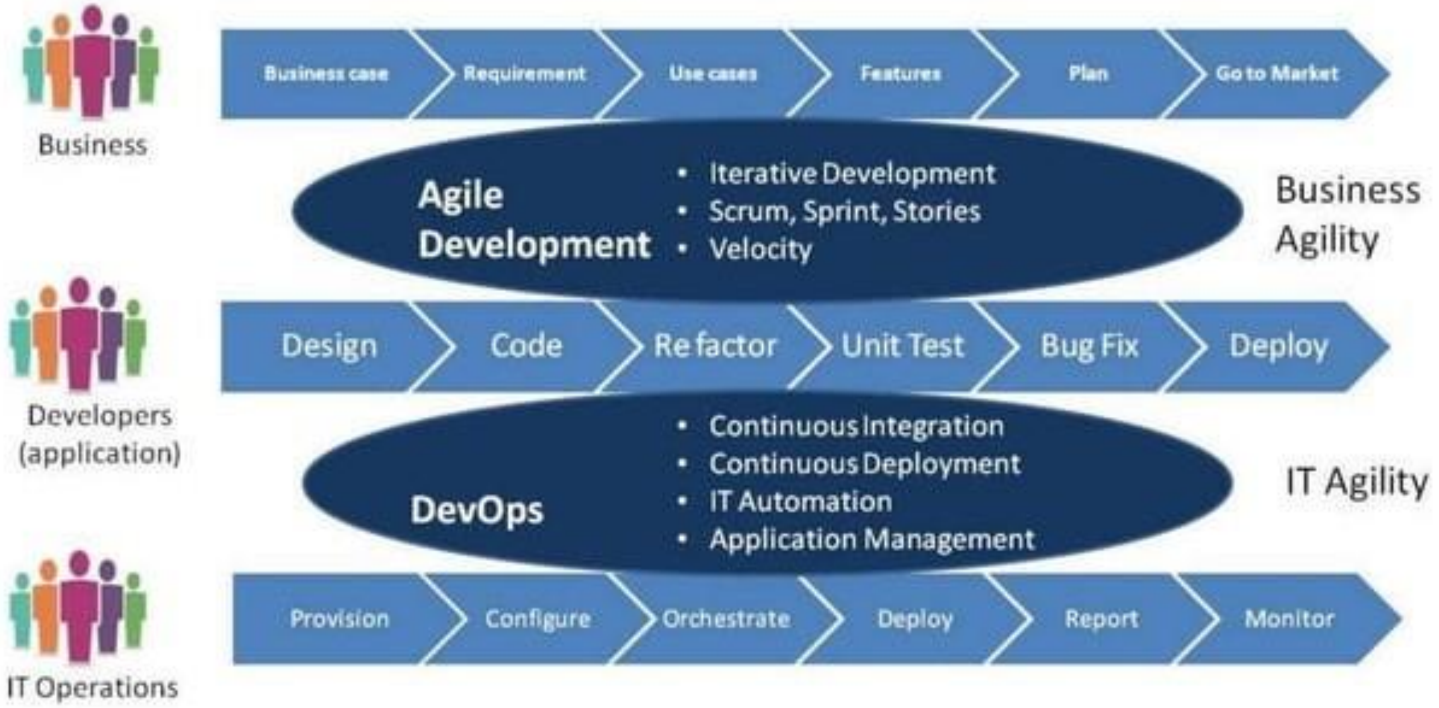
- Receive new product from developers to be installed and operated
- Expected to keep production systems up and running
- Track problems, deployment failures, and system outages
- May provide feedback to the inventors for future consideration
- Penalized for downtime

The mechanics are responsible for keeping the system in operation

Traditional Development and Operations

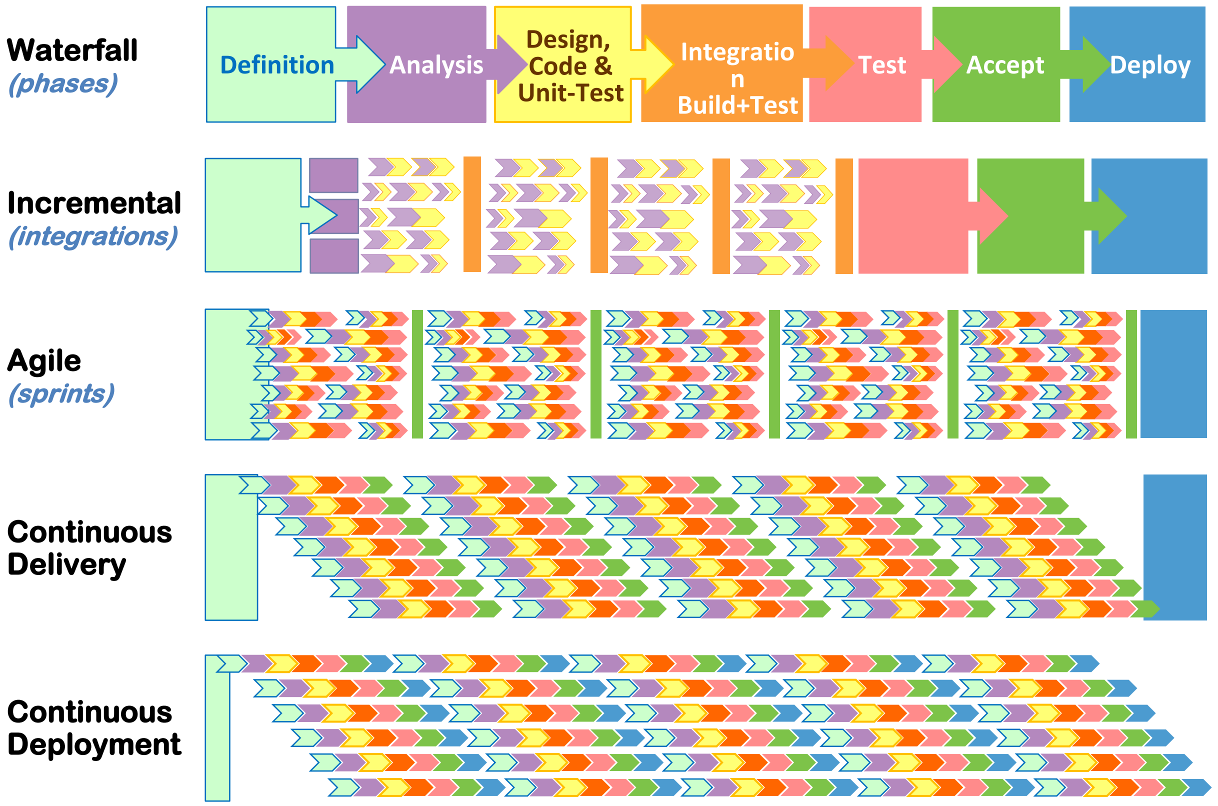
Dev
WALL OF CONFUSION
ops

Relationship between Agile & DevOps



Source: http://www.effectivepmc.com/devops

Comparison of Lifecycles

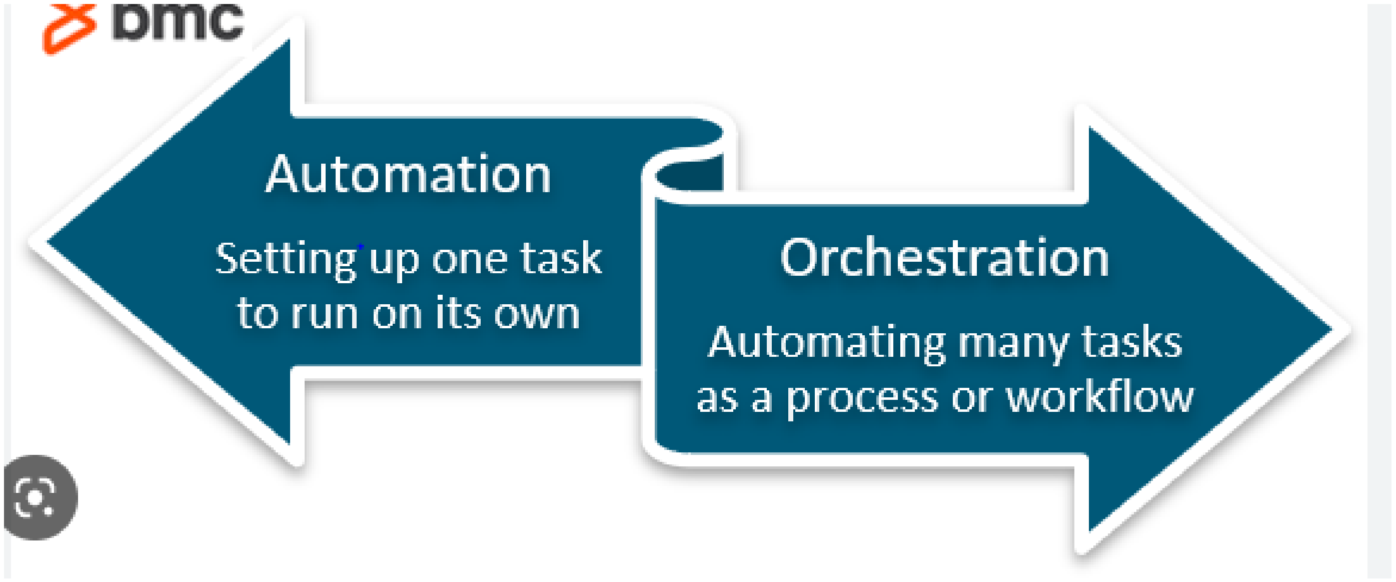


DevOps Orchestration

- DevOps orchestration is the automation of numerous processes that run concurrently in order to reduce production issues and time to market, while automation is the capacity to do a job or a series of procedures to finish an individual task repeatedly.

- Many people believe that DevOps orchestration is just merging several jobs into a larger script. DevOps orchestration services include such jobs into a process or workflow, which may involve many automated tasks and stages, and resources to streamline the entire workflow or process.

DevOps Orchestration


bmc
Automation
Setting up one task to run on its own
Orchestration
Automating many tasks as a process or workflow

CALMS Model of DevOps

Culture
- Focus on People
- Embrace Change & Experiment

Automation
- Continuous Delivery
- Infrastructure as Code

Lean
- Focus on Producing Value for the User
- Small Batch-sizes

Measurement
- Measure Everything
- Show the Improvement

Sharing
- Open Information Sharing
- Collaboration & Communication

Deployment Strategy in DevOps

© 2003–2008 Mountain Goat Software®

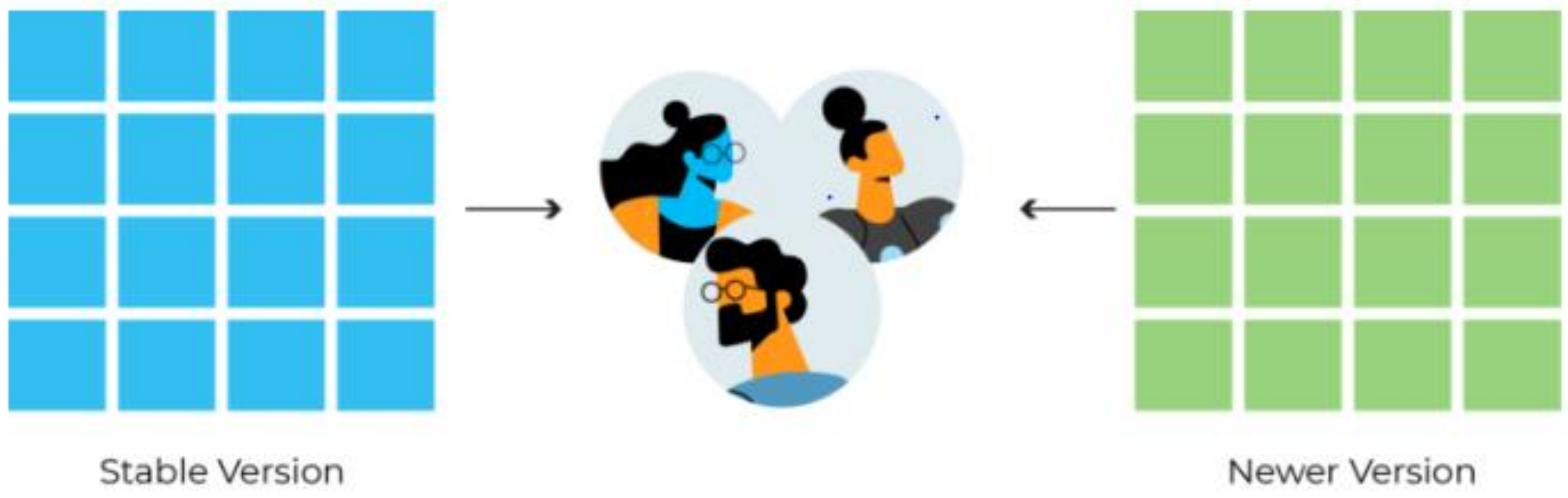
What Is a Deployment Strategy ?

- A deployment strategy is any technique employed by DevOps teams to successfully launch a new version of the software solution they provide.

Blue/Green Deployment

- In this type of deployment strategy, the new version of the software runs alongside the old version. Note that you can also refer to this as red/black deployment strategy in some cases.
- Here, the stable or the older version of the application is always blue or red, while the newer version is green or black.
- After the new version has been tested and certified to meet all the requirements, the load balancer automatically switches the traffic from the older version to the newer version.

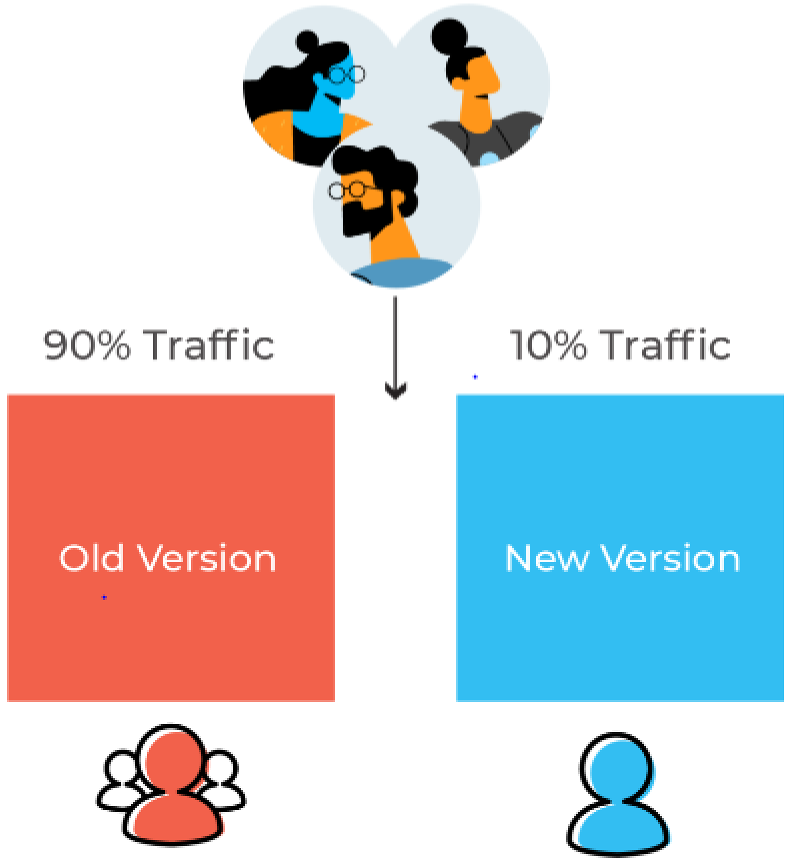
Blue/Green Deployment



Canary Deployment

- The deployment team sets up the new version and then gradually shifts the production traffic from the older version to the newer version.
- For example, at a point in time during the deployment process, the older version might retain 90% of all traffic for the software while the newer version hosts 10% of the traffic.
- This deployment technique helps the DevOps engineers test the stability of the new version.

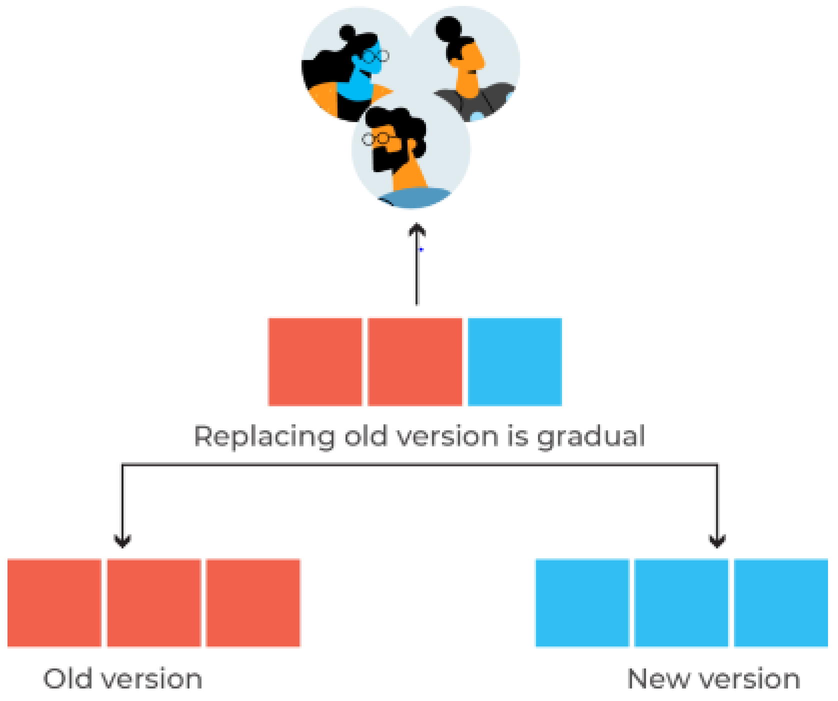
Canary Deployment



Ramped Deployment

- The ramped deployment strategy gradually changes the older version to the new version. Unlike canary deployment, the ramped deployment strategy makes its switch by replacing instances of the old application version with the instances from the new application version one instance at a time. You can also call this method the rolling upgrade deployment strategy.
- When developers replace all instances of the older version, they shut down the older version. The new version then controls the whole production traffic.

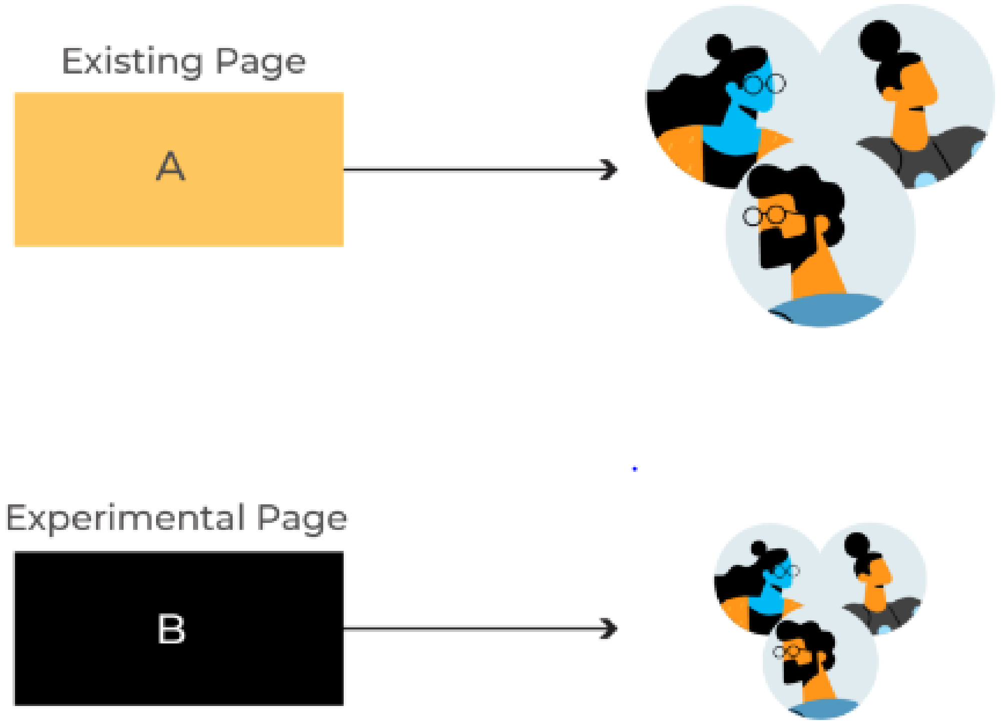
Ramped Deployment



A/B Testing Deployment

- In A/B testing deployment, developers deploy the new version alongside the older version. However, the new version is only available to a subset of users.
- These users are selected based on specific conditions and parameters the engineers choose.
- These parameters can be the user’s location, type of device, UI language, and operating system.

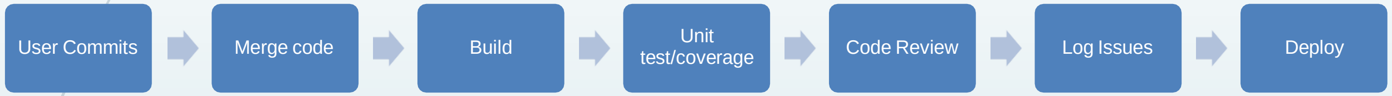
A/B Testing Deployment



DevOps Delivery Pipeline

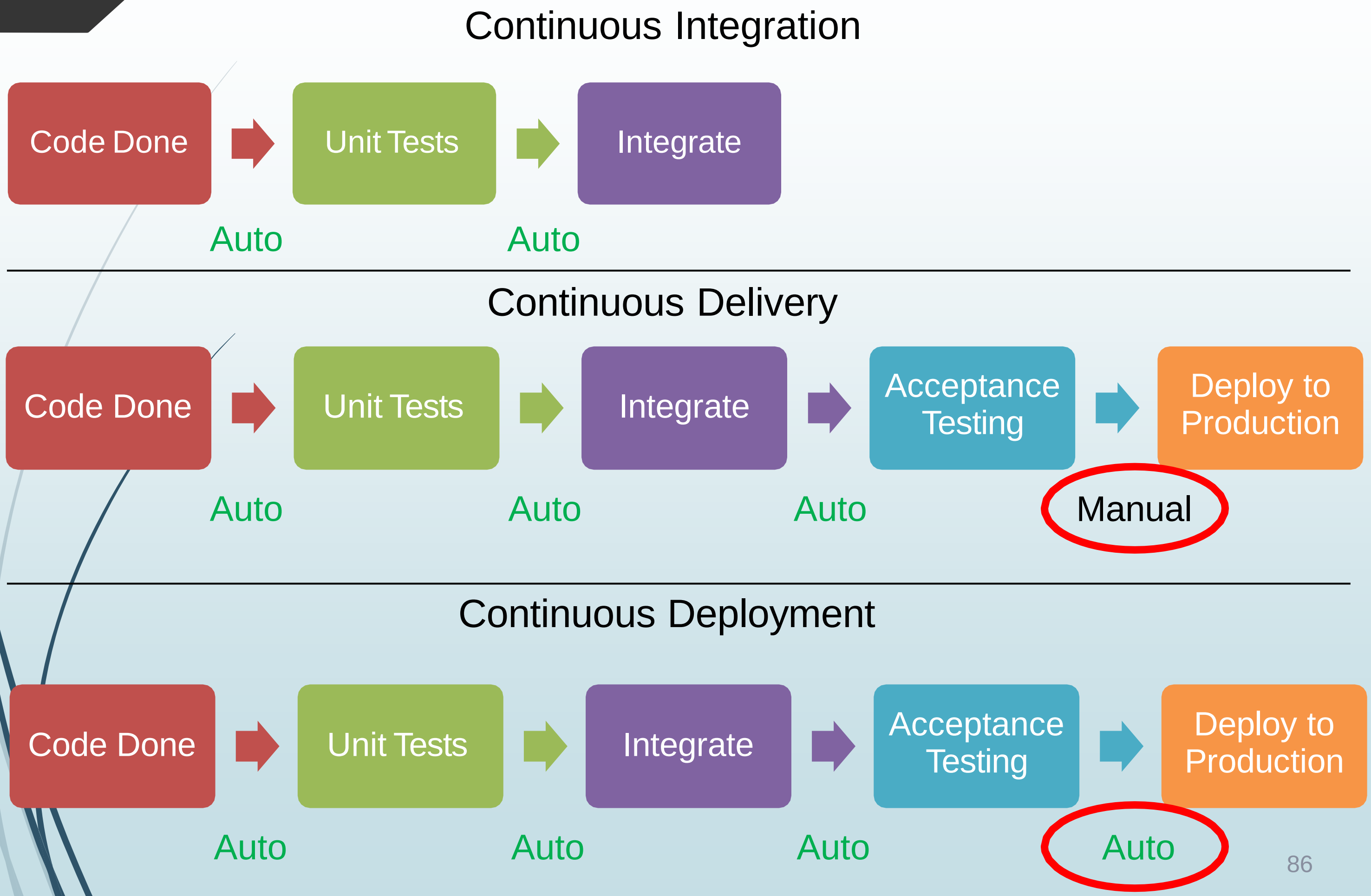
© 2003–2008 Mountain Goat Software®

A Pipeline is a chain of tasks that can be automated

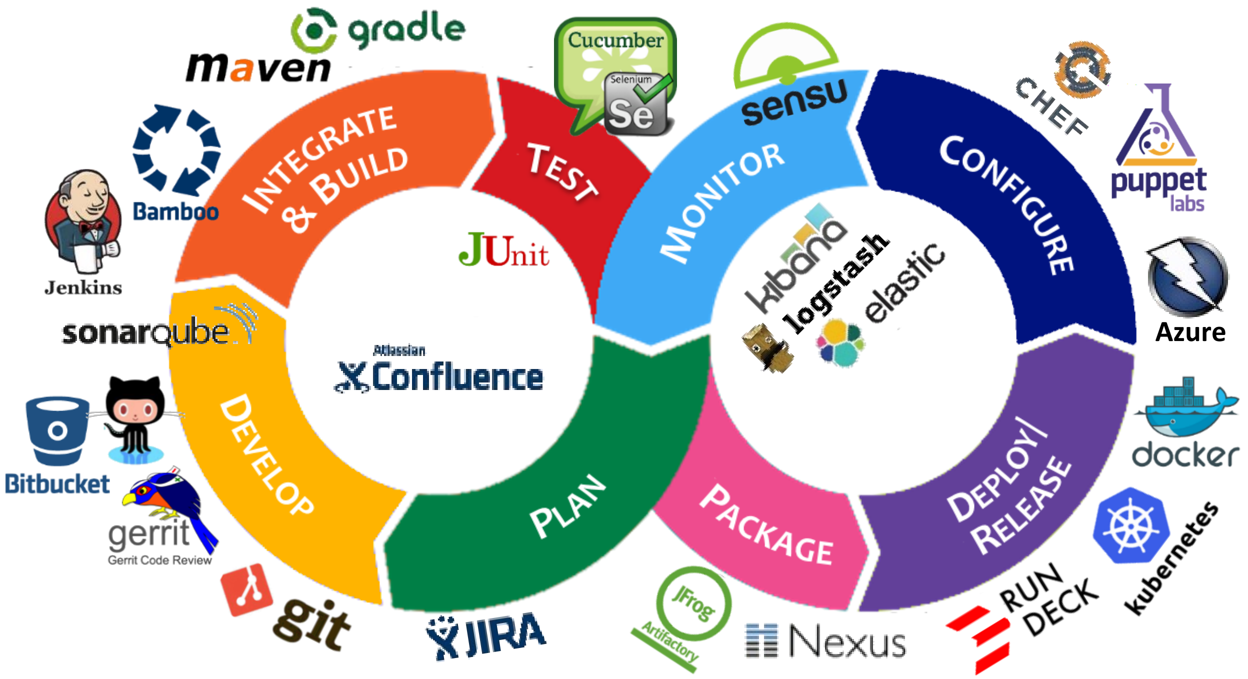


- **Integration tools use pipelines to perform tasks repetitively and continuously**
- **The process is called Continuous Integration (CI)**

Pipeline Stages



DevOps Toolchain / Pipeline



A **DevOps toolchain** is a set or combination of tools that aid in the delivery, development, and management of applications throughout the software development lifecycle, as coordinated by an organization that uses DevOps practices.

Tool Stack Implementation in DevOps

© 2003–2008 Mountain Goat Software®

DevOps Tools

SCM tools

For Source-Code Management (SCM) ,version control tools such as Git, GitHub, Subversion, TFS, and Mercurial are used.

Software build tools

For automating the build process of an executable application from source code, software build tools such as Maven, Gradle, Ant, and Grunt are used.

DevOps Tools

ANSIBLE

CMT and Deployment tools

For deployment and operations phase, CMT and automation tools such as Jenkins, AWS CodeDeploy, Chef, Puppet, Ansible, and Terraform are used.

Nagios

Monitoring tools

For monitoring system performance and productivity, to reduce (or even eliminate) downtime, monitoring tools such as Nagios are used.

Containerization tools

For packaging an application with its required libraries, frameworks, and configuration files to efficiently run it in various computing environments, containerization tools such as Docker and Kubernetes are used.

DevOps Tools

Testing tools

In continuous testing phase, the built software is continuously tested for bugs using testing tools such as Selenium, TestNG, and JUnit.

Integration tools

CI/CD pipelines are created for procuring updated source code and constructing the build into *.exe* format using tools such as Jenkins.

DevOps Tools Selection

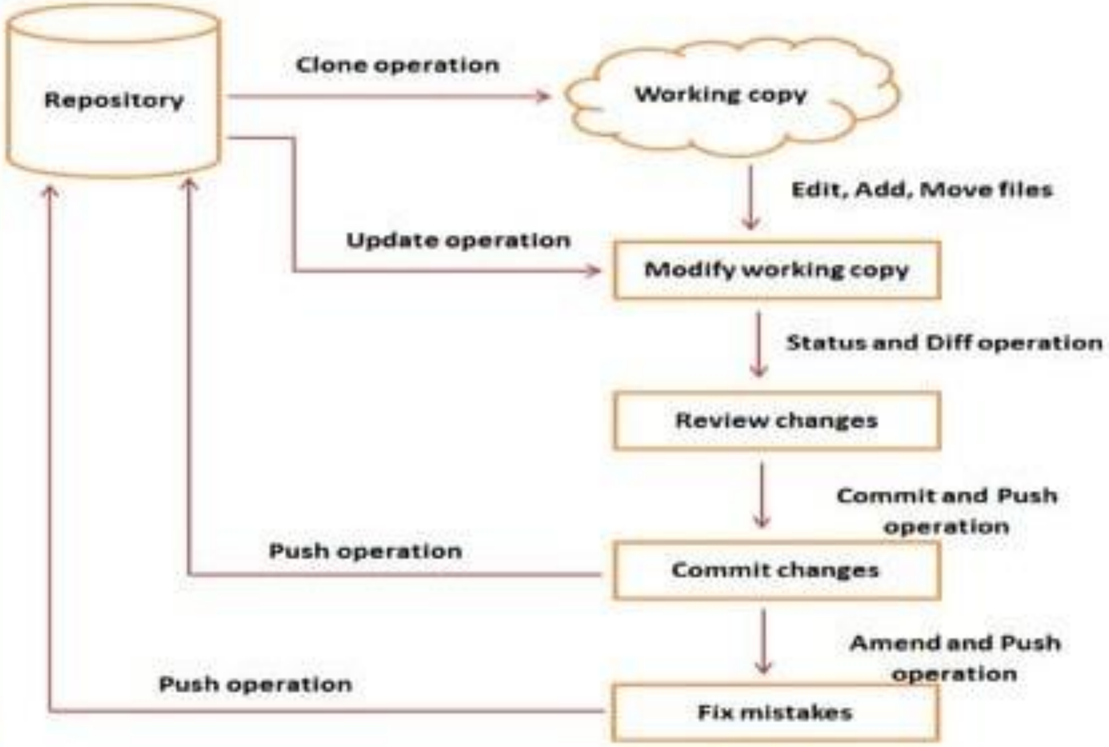
- Open Source
- Licensed
- Compatible (Tools that work together)

DevOps Tools

Git

- In recent years, Git has become incredibly popular for source code management, particularly as the site GitHub has become more popular for hosting open source projects.
- It stands out from other version control management for the ease with which it handles branching and merging.
- It's also very easy to use with distributed development teams, and it offers fast performance.
- Many DevOps teams use it to manage the source code for their applications.
- Its list of well-known users includes many of the biggest firms in the technology industry, such as Google, Facebook, Microsoft, Twitter, LinkedIn, Netflix, the Linux kernel and many others.

DevOps Tools

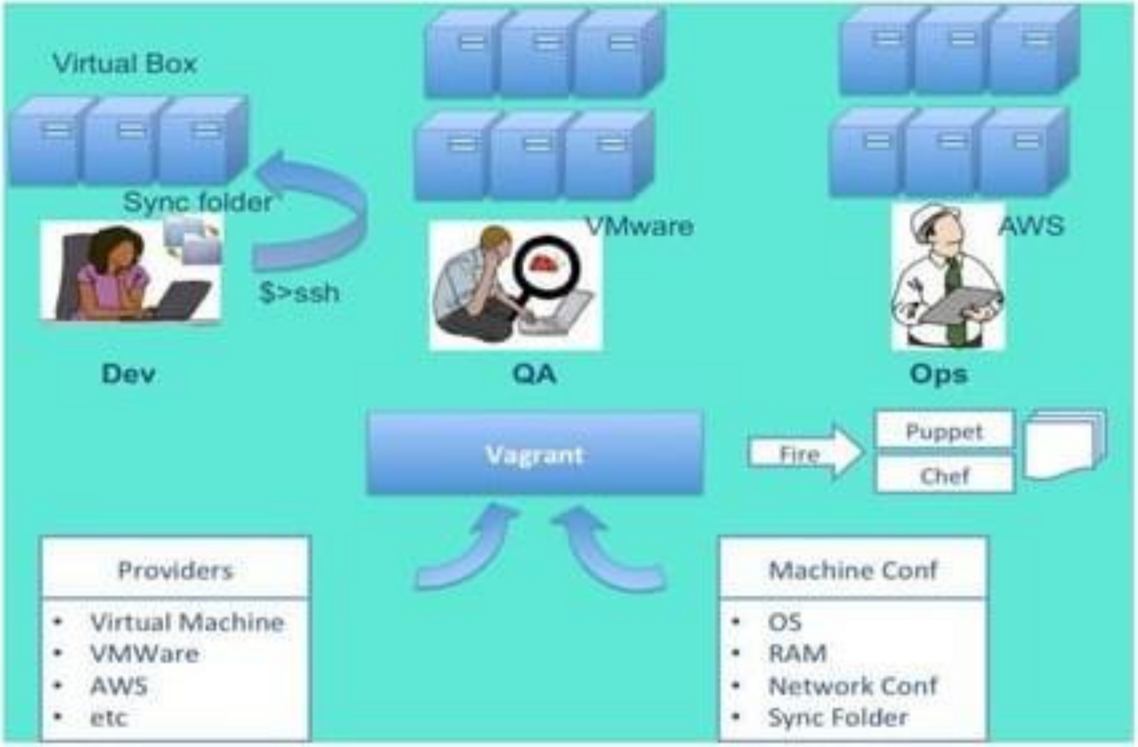


DevOps Tools

Vagrant

- Owned by DevOps tool vendor HashiCorp, Vagrant aims to make it easy to set up development environments that are lightweight, portable and reproducible.
- It's a command-line utility for managing virtual machines. Its users include the BBC, Expedia, Yammer, Mozilla, Nokia and others.
- It integrates with Chef, Puppet, VMware, Amazon Web Services and many other DevOps tools and cloud services.
- Paid VMware plug-ins are available through partners, and HashiCorp offers related paid tools for managing DevOps environments.
- Vagrant manages all the necessary configurations for the developers in order to avoid the unnecessary maintenance and setup time, and increases development productivity.
- Vagrant uses "Provisioners" and "Providers" as building blocks to manage the development environments.

DevOps Tools

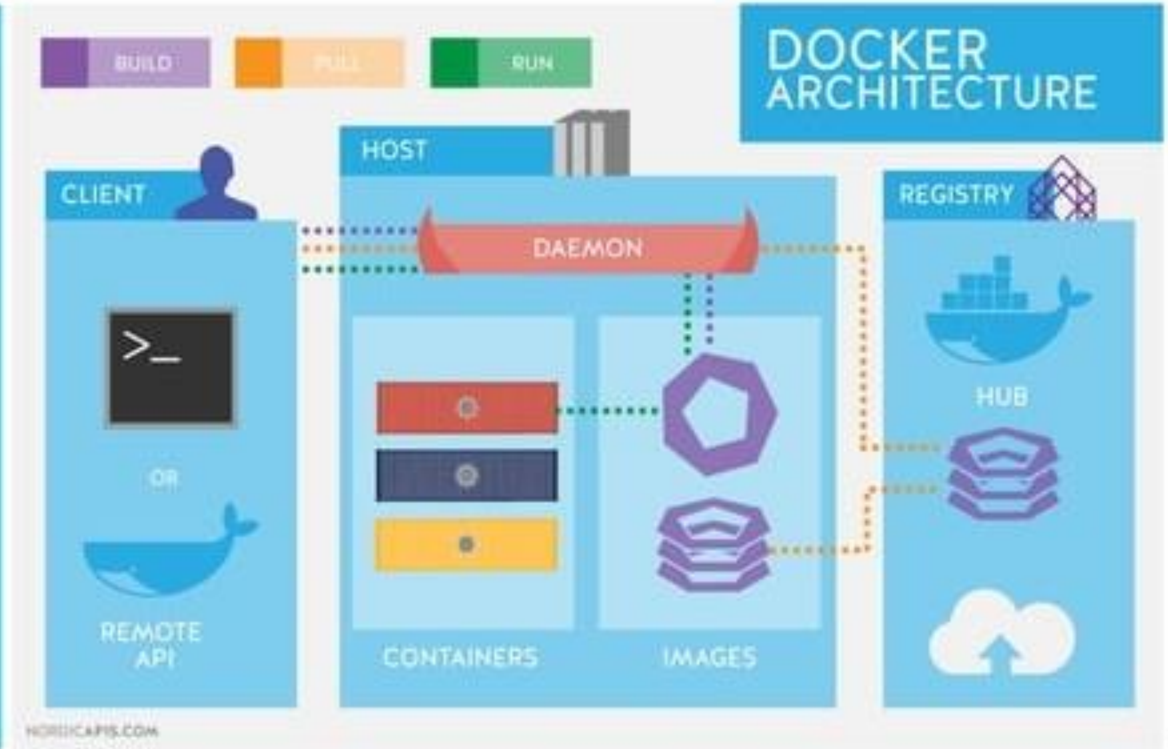


DevOps Tools

Docker

- Docker is at the forefront of the new trend toward containerization.
- It packages together everything that an application needs to run—the code, the runtime, system tools, libraries, etc.—so that applications will operate the same way no matter where they are deployed.
- Containers are more lightweight than virtual machines, and they also offer some security benefits.
- A recent survey conducted by Docker found that 80 percent of enterprises surveyed plan their DevOps implementations around Docker.
- Docker implements a high-level API to provide lightweight containers that run processes in isolation.

DevOps Tools



Docker Demo

app.py

```
from flask import Flask
app = Flask(__name__)

@app.route('/')
def hello():
    return "Hello, World!"
```

requirements.txt

```
Flask==2.2.3
```

Dockerfile

```
FROM python:3.9-slim
WORKDIR /app
COPY requirements.txt .
RUN pip install -r requirements.txt
COPY . .
CMD ["python", "app.py"]
```

```
docker build -t flask-app .
docker run -p 5000:5000 flask-app
```

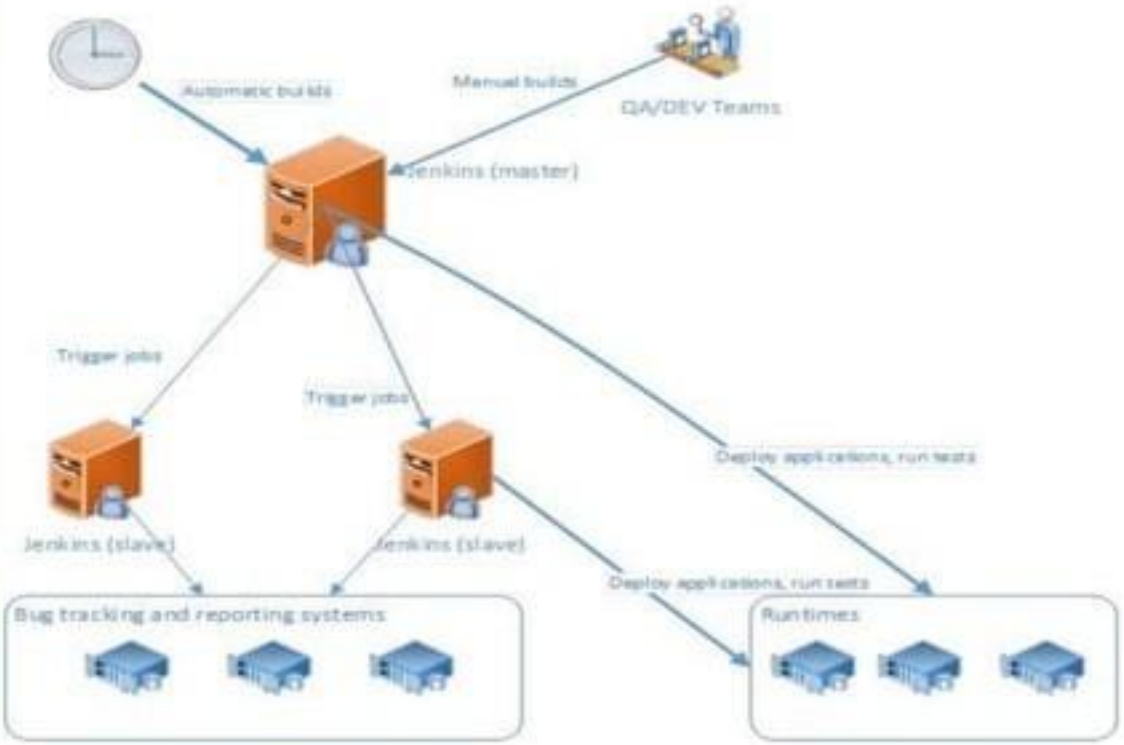
The app is now accessible at http://localhost:5000

DevOps Tools

Jenkins

- The "leading open source automation server," Jenkins was forked from Hudson and offers many of the same capabilities.
- It boasts easy installation and configuration, hundreds of plugins, extensibility and a distributed architecture that allows it to speed the process of testing.
- It has a very active user community with lots of scheduled events that offer opportunities to learn more about the software.
- There is also plenty of documentation on the website, including a blog that is updated regularly.
- Jenkins Pipeline is a suite of plugins which supports implementing and integrating continuous delivery pipelines into Jenkins.
- Pipeline provides an extensible set of tools for modeling simple-to-complex delivery pipelines "as code".

DevOps Tools

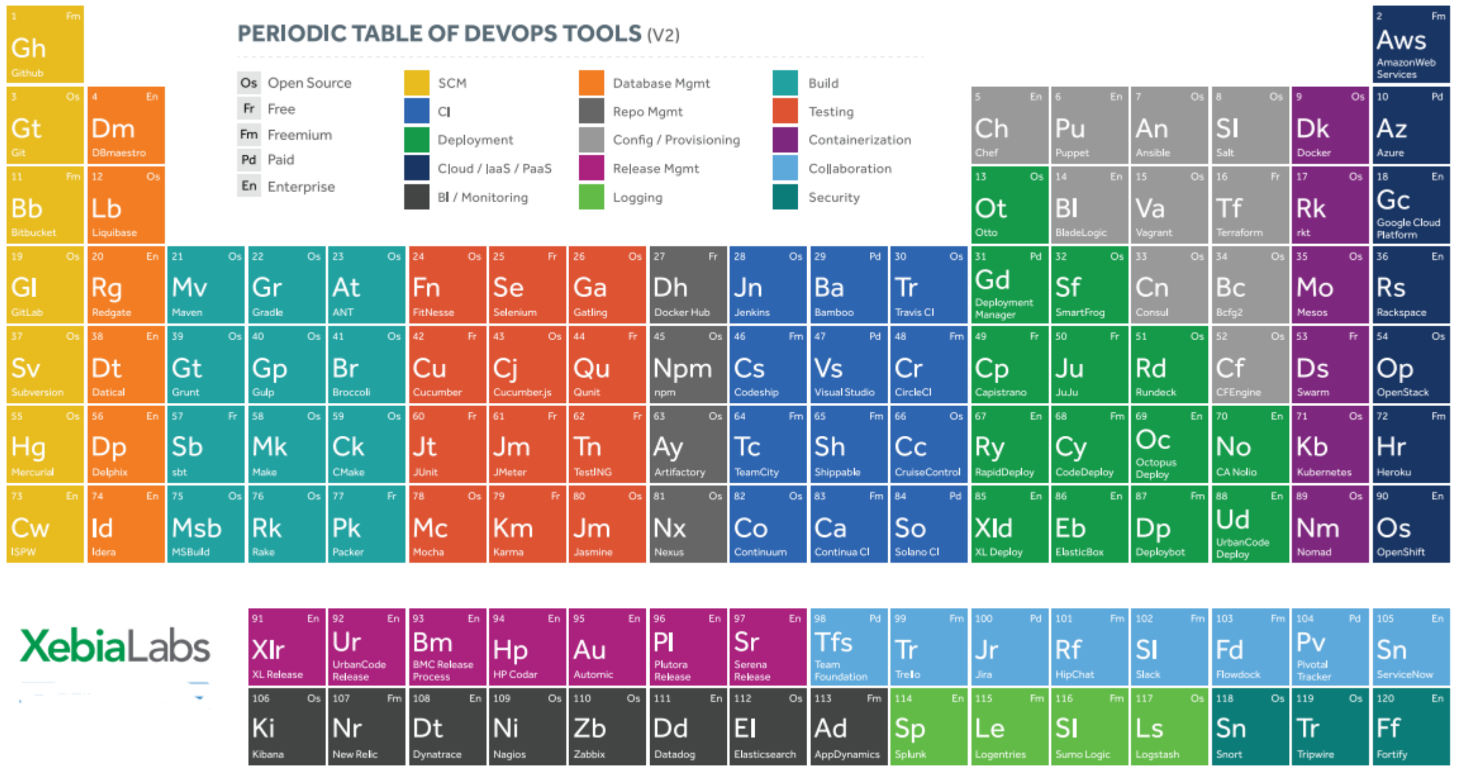


Jenkins Demo

```
pipeline {
   agent any  // Run on any available Jenkins agent

   stages {
      stage('Checkout Code') {
         steps {
            git 'https://github.com/user/java-maven-app.git'
         }
      }

      stage('Build') {
         steps {
            sh 'mvn clean package'
         }
      }

      stage('Test') {
         steps {
            sh 'mvn test'
         }
      }
```

```
stage('Deploy') {
         steps {
            sshagent(['deployment-ssh-credentials']) {
               sh 'scp target/myapp.jar user@remote-
server:/path/to/deploy'
               sh 'ssh user@remote-server "java -jar
/path/to/deploy/myapp.jar"'
            }
         }
      }
   }

   post {
      success {
         echo 'Pipeline executed successfully!'
      }
      failure {
         echo 'Pipeline failed. Check logs for details.'
      }
   }
}
```

DevOps Tools Landscape



PERIODIC TABLE OF DEVOPS TOOLS (V2)

Os	Open Source	🟨	SCM	🟧	Database Mgmt
Fr	Free	🟦	CI	⬛	Repo Mgmt
Fm	Freemium	🟩	Deployment	⬜	Config / Provisioning
Pd	Paid	🟦	Cloud / IaaS / PaaS	🟪	Release Mgmt
En	Enterprise	⬛	BI / Monitoring	🟩	Logging

🟩	Build
🟥	Testing
🟪	Containerization
🟦	Collaboration
🟩	Security

1 Fm Gh Github																	2 Fm Aws AmazonWeb Services
3 Os Gt Git	4 En Dm DBmaestro											5 En Ch Chef	6 En Pu Puppet	7 Os An Ansible	8 Os Sl Salt	9 Os Dk Docker	10 Pd Az Azure
11 Fm Bb Bitbucket	12 Os Lb Liquibase											13 Os Ot Otto	14 En Bl BladeLogic	15 Os Va Vagrant	16 Fr Tf Terraform	17 Os Rk rkt	18 En Gc Google Cloud Platform
19 Os Gl GitLab	20 En Rg Redgate	21 Os Mv Maven	22 Os Gr Gradle	23 Os At ANT	24 Os Fn FitNesse	25 Fr Se Selenium	26 Os Ga Gatling	27 Fr Dh Docker Hub	28 Os Jn Jenkins	29 Pd Ba Bamboo	30 Os Tr Travis CI	31 Pd Gd Deployment Manager	32 Os Sf SmartFrog	33 Os Cn Consul	34 Os Bc Bcfg2	35 Os Mo Mesos	36 En Rs Rackspace
37 Os Sv Subversion	38 En Dt Datical	39 Os Gt Grunt	40 Os Gp Gulp	41 Os Br Broccoli	42 Fr Cu Cucumber	43 Os Cj Cucumber.js	44 Fr Qu Qunit	45 Os Npm npm	46 Fm Cs Codeship	47 Pd Vs Visual Studio	48 Fm Cr CircleCI	49 Fr Cp Capistrano	50 Fr Ju JuJu	51 Os Rd Rundeck	52 Os Cf CFEngine	53 Fr Ds Swarm	54 Os Op OpenStack
55 Os Hg Mercurial	56 En Dp Delphix	57 Fr Sb sbt	58 Os Mk Make	59 Os Ck CMake	60 Fr Jt JUnit	61 Fr Jm JMeter	62 Fr Tn TestNG	63 Os Ay Artifactory	64 Fm Tc TeamCity	65 Fm Sh Shippable	66 Os Cc CruiseControl	67 En Ry RapidDeploy	68 Fm Cy CodeDeploy	69 En Oc Octopus Deploy	70 En No CA Nolio	71 Os Kb Kubernetes	72 Fm Hr Heroku
73 En Cw ISPW	74 En Id Idera	75 Os Msb MSBuild	76 Os Rk Rake	77 Fr Pk Packer	78 Os Mc Mocha	79 Fr Km Karma	80 Os Jm Jasmine	81 Os Nx Nexus	82 Os Co Continuum	83 Fm Ca Continua CI	84 Pd So Solano CI	85 En Xld XL Deploy	86 En Eb ElasticBox	87 Fm Dp Deploybot	88 En Ud UrbanCode Deploy	89 Os Nm Nomad	90 En Os OpenShift
			91 En Xlr XL Release	92 En Ur UrbanCode Release	93 En Bm BMC Release Process	94 En Hp HP Codar	95 En Au Automic	96 En Pl Plutora Release	97 En Sr Serena Release	98 Pd Tfs Team Foundation	99 Fm Tr Trello	100 Pd Jr Jira	101 Fm Rf HipChat	102 Fm Sl Slack	103 Fm Fd Flowdock	104 Pd Pv Pivotal Tracker	105 En Sn ServiceNow
			106 Os Ki Kibana	107 Fm Nr New Relic	108 En Dt Dynatrace	109 Os Ni Nagios	110 Os Zb Zabbix	111 En Dd Datadog	112 Os El Elasticsearch	113 Fm Ad AppDynamics	114 En Sp Splunk	115 Fm Le Logentries	116 Fm Sl Sumo Logic	117 Os Ls Logstash	118 Os Sn Snort	119 Os Tr Tripwire	120 En Ff Fortify

XebiaLabs

Source: https://xebialabs.com/periodic-table-of-devops-tools/

People, Process and Products

© 2003–2008 Mountain Goat Software®

DevOps process

1 Plan

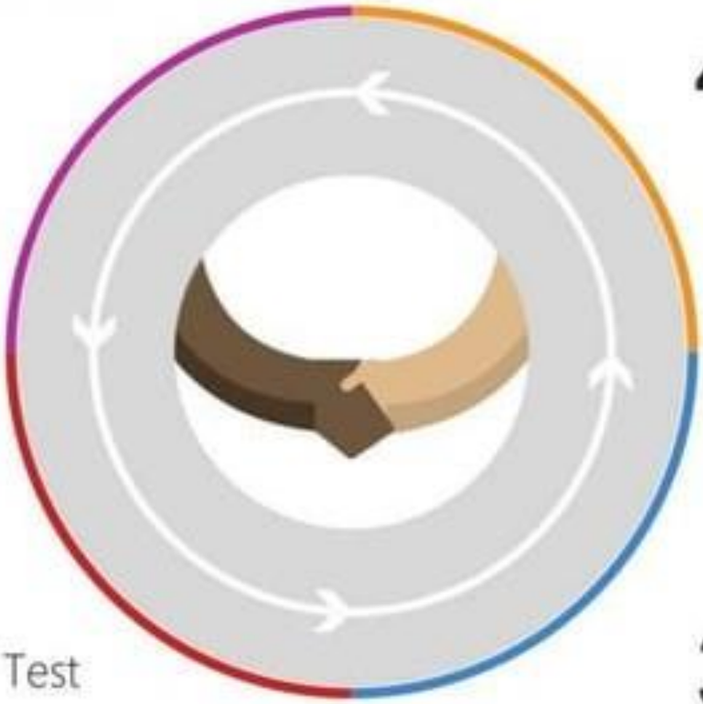
4 Monitor + Learn

Development

Production

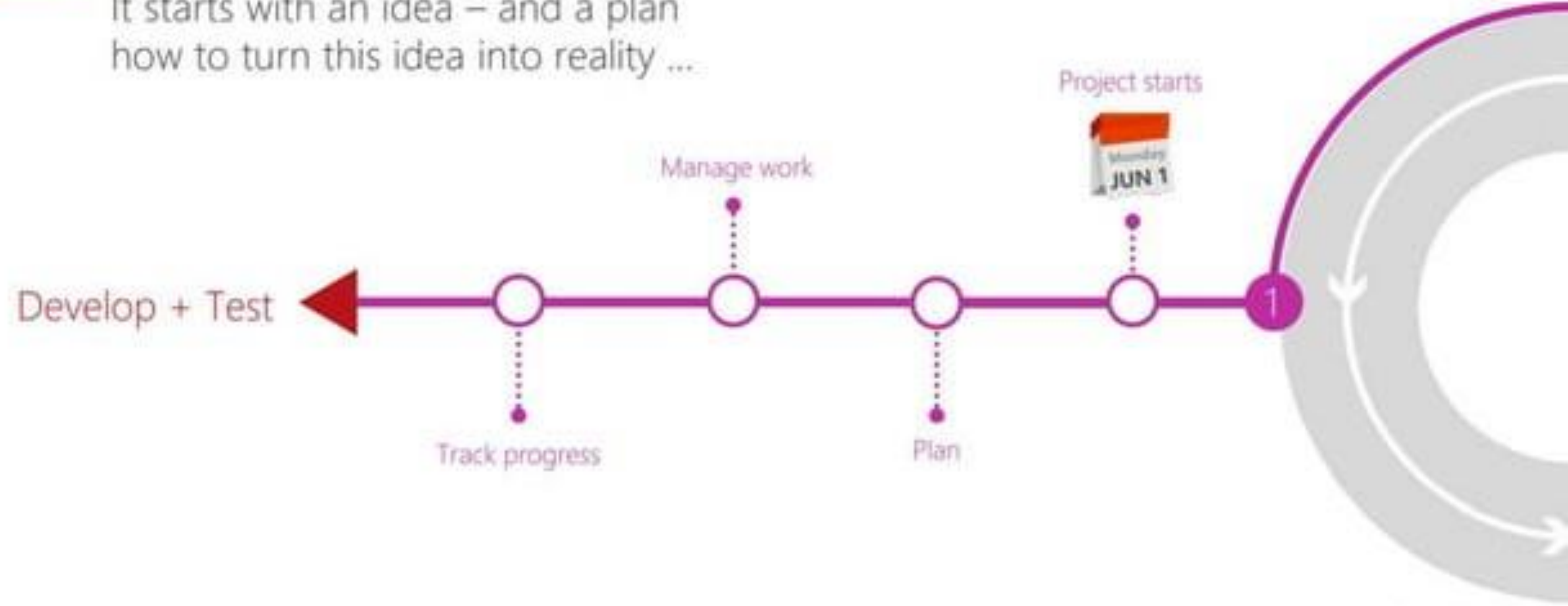
2 Develop + Test

3 Release



Plan

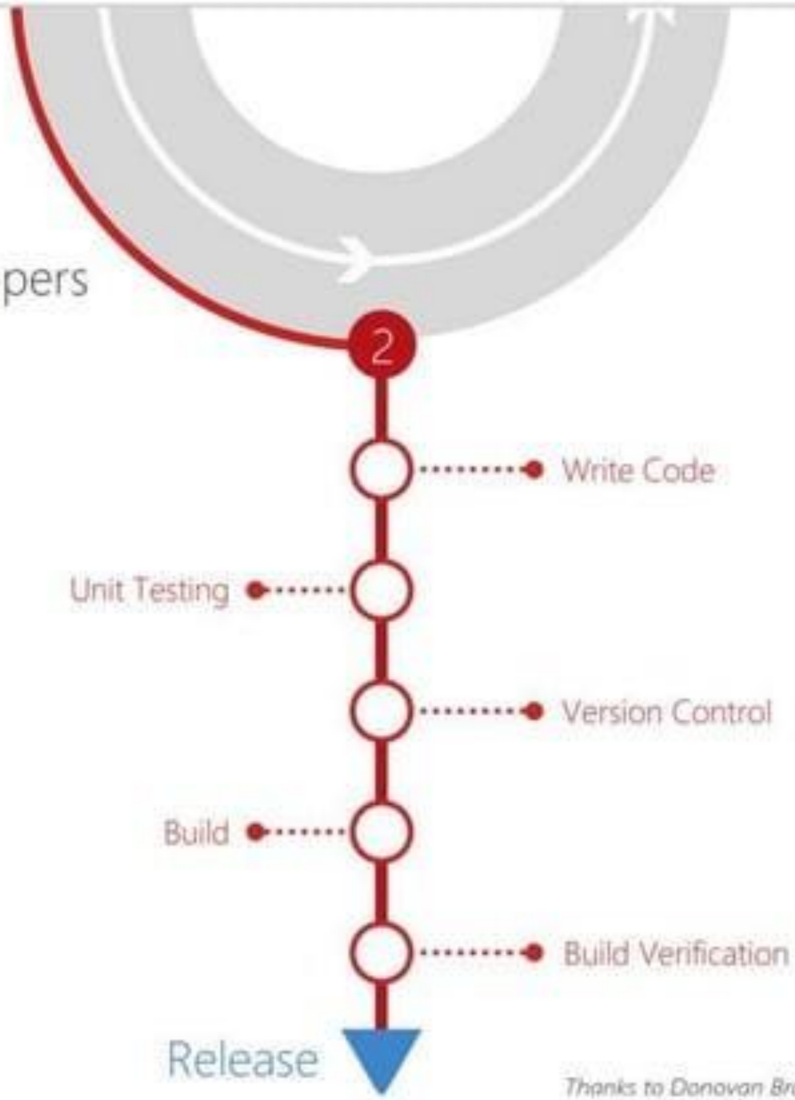
It starts with an idea – and a plan how to turn this idea into reality ...



Activate Window
Go to Settings to activ

Develop + Test

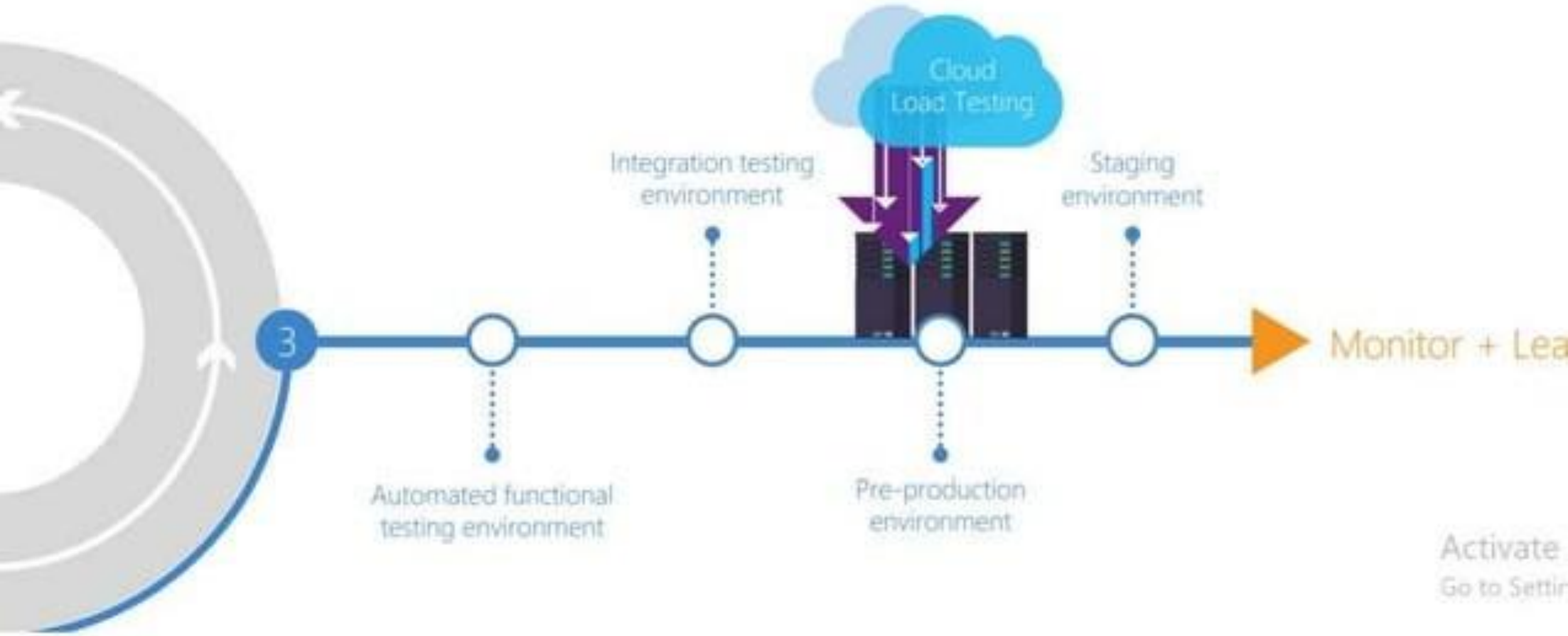
Once the iteration starts, developers turn great ideas into features ...



Thanks to Donovan Brown fo

Release

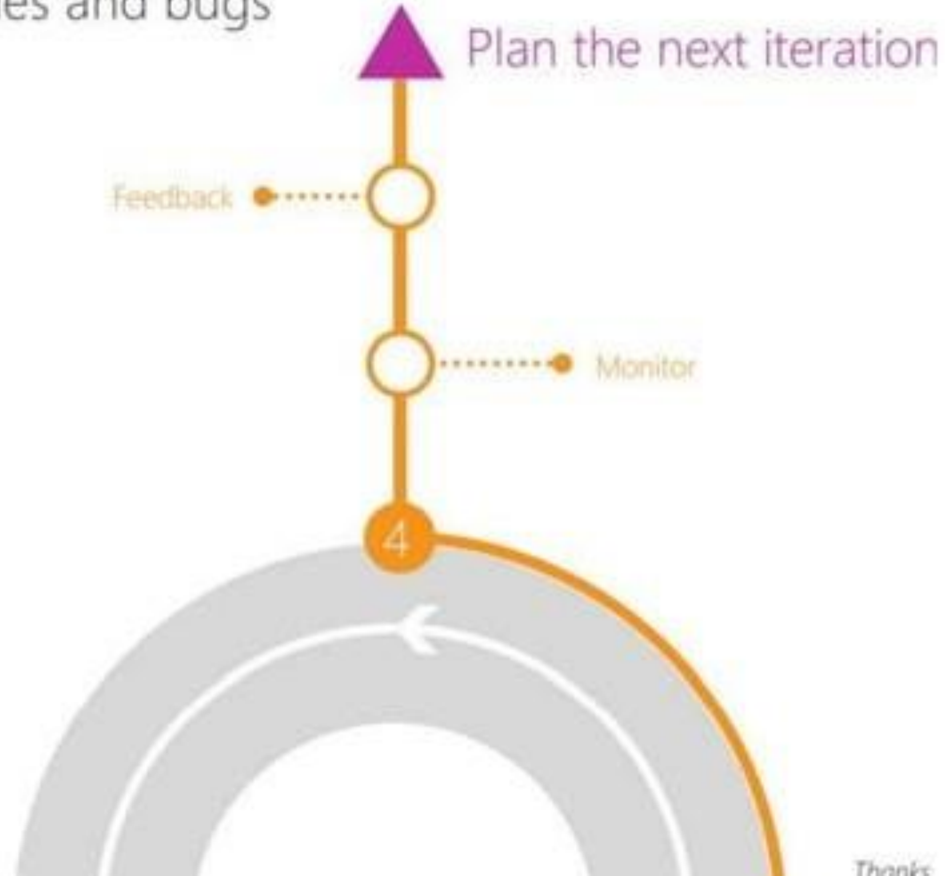
When all tests pass, the build is deployed to testing environments for each stage in the release process


Cloud Load Testing
Integration testing environment
Staging environment
3
Monitor + Lea
Automated functional testing environment
Pre-production environment

Activate
Go to Settir

Monitor + Learn

Learn and understand how users use your app, how it reacts
and quickly fix issues and bugs



Keys to DevOps

- Plan small/ fail fast/ deliver quickly
- Everything is under SCM
- Test & Automation
- People (Kaizen/ Quality Culture)
- Infrastructure under CM

DevOps Outcomes

- Improved deployment frequency;
- Faster time to market;
- Lower failure rate of new releases;
- Shortened lead time between fixes;
- Faster mean time to recovery
- Better employee engagement & leadership

Investment in DevOps and associated mindset is one of the top predictors of IT organizational performance!

Thank You!