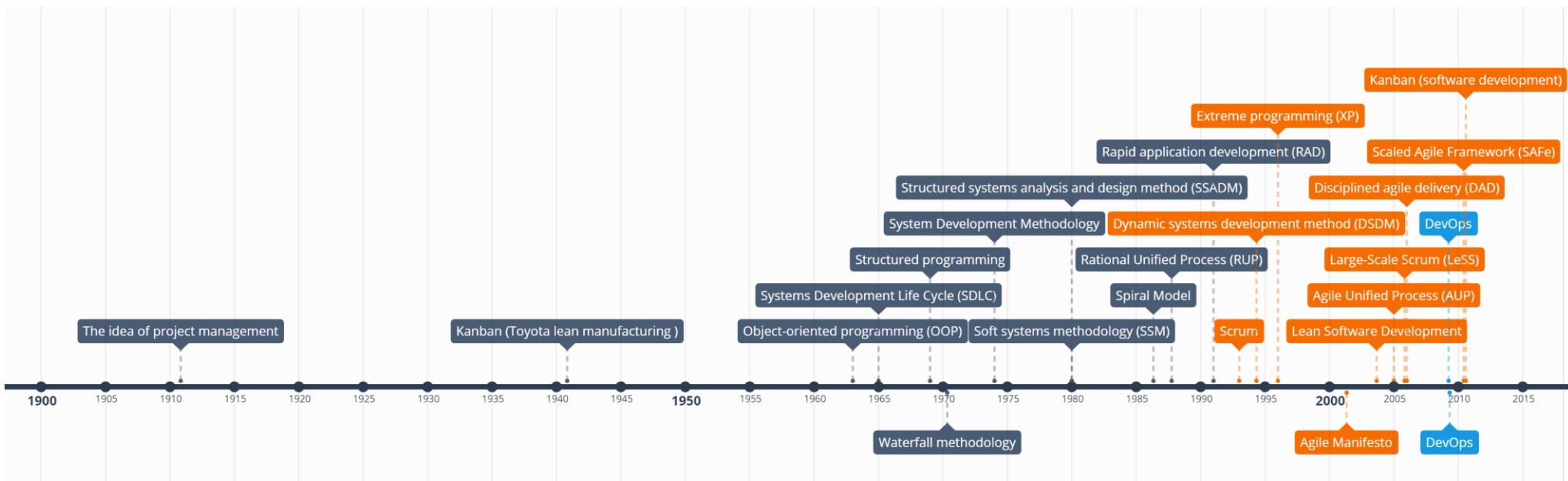


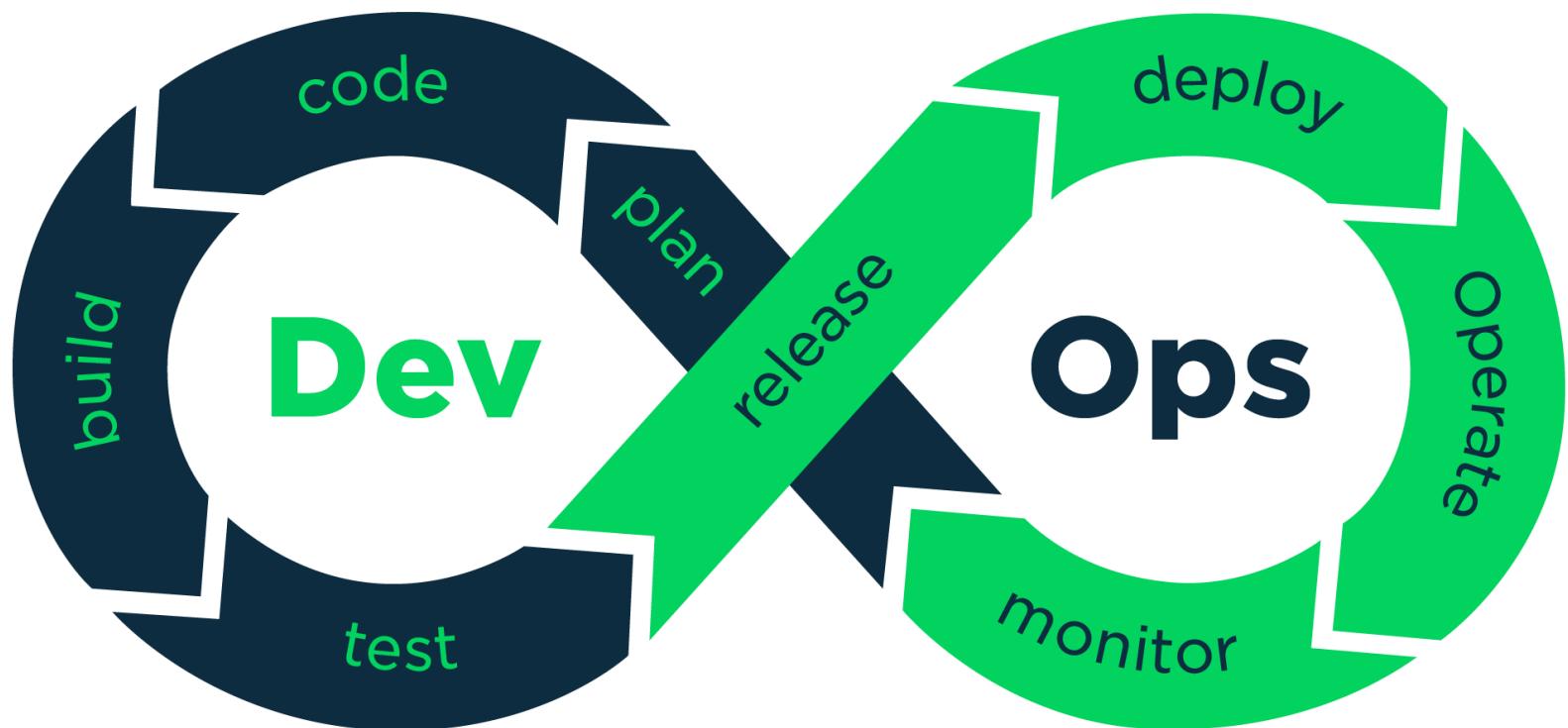
AGILE? METHODOLOGIES?: DEVOPS

Lecturer:
Ehsan Shoja

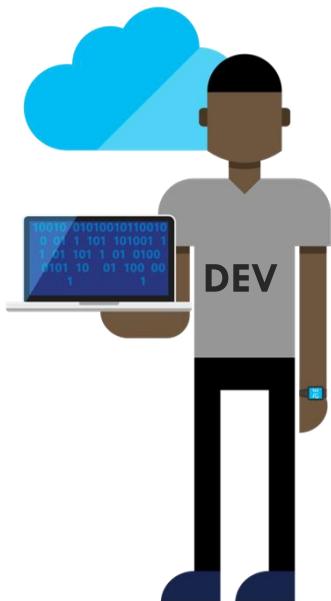
SOFTWARE DEVELOPMENT PROCESS HISTORY



DEVOPS



*“A software development method that emphasizes **communication**, **collaboration**, **integration**, **automation**, and **measurement** of cooperation between software developers and other IT professionals”*



DEVOPS

- ❑ DevOps is more than just development and operations teams working together. It's more than tools and practices. DevOps is a mindset, a cultural shift, where teams adopt new ways of working.
- ❑ A DevOps culture means developers get closer to the user by gaining a better understanding of user requirements and needs. Operations teams get involved in the development process and add maintenance requirements and customer needs.
- ❑ It means adhering key principles that help DevOps teams deliver applications and services at a faster pace and higher quality than organizations using the traditional software development model.

DEVOPS PRINCIPLES: COLLABORATION

- ❑ The key premise behind DevOps is collaboration. Development and operations teams coalesce into a functional team that communicates, shares feedback, and collaborates throughout the entire development and deployment cycle. Often, this means development and operations teams merge into a single team that works across the entire application lifecycle.
- ❑ The members of a DevOps team are responsible for ensuring quality deliverables across each facet of the product. This leads to more ‘full stack’ development, where teams own the complete backend-to-frontend responsibilities of a feature or product. Teams will own a feature or project throughout the complete lifecycle from idea to delivery. This enhanced level of investment and attachment from the team leads to higher quality output.

DEVOPS PRINCIPLES: AUTOMATION

- ❑ An essential practice of DevOps is to automate as much of the software development lifecycle as possible. This gives developers more time to write code and develop new features. Automation is a key element of a CI/CD pipeline and helps to reduce human errors and increase team productivity. With automated processes, teams achieve continuous improvement with short iteration times, which allows them to quickly respond to customer feedback.

DEVOPS PRINCIPLES: CONTINUOUS IMPROVEMENT

❑ Continuous improvement was established as a staple of agile practices, as well as lean manufacturing and Improvement Kata. It's the practice of focusing on experimentation, minimizing waste, and optimizing for speed, cost, and ease of delivery. Continuous improvement is also tied to continuous delivery, allowing DevOps teams to continuously push updates that improve the efficiency of software systems. The constant pipeline of new releases means teams consistently push code changes that eliminate waste, improve development efficiency, and bring more customer value.

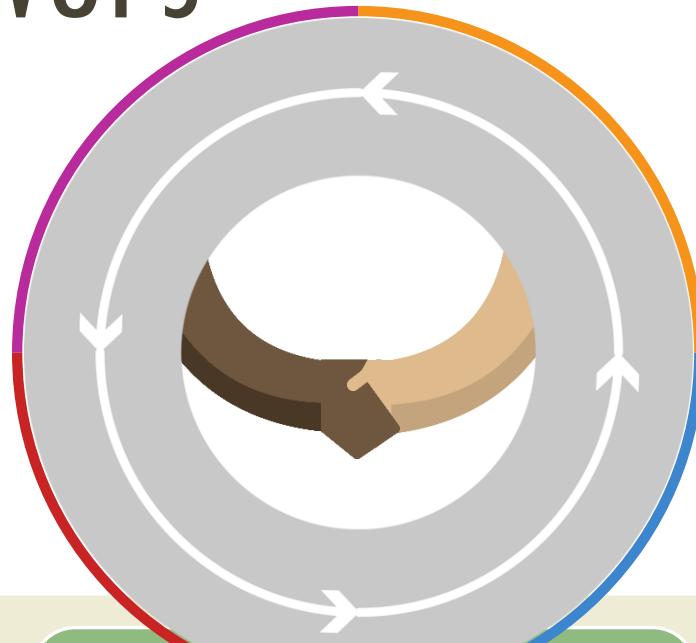
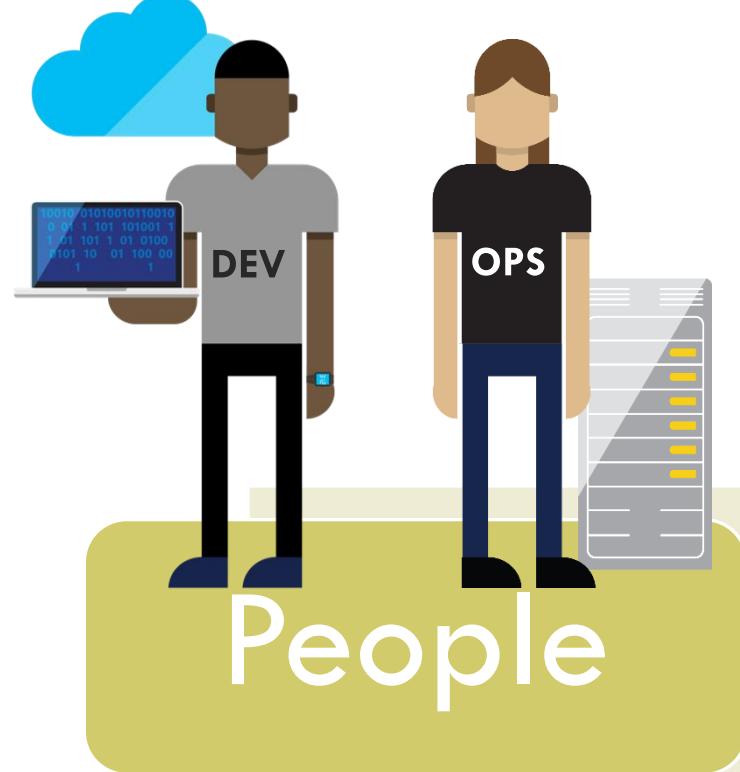
DEVOPS PRINCIPLES: CUSTOMER-CENTRIC ACTION

- DevOps teams use short feedback loops with customers and end users to develop products and services centered around user needs. DevOps practices enable rapid collection and response to user feedback through use of real-time live monitoring and rapid deployment. Teams get immediate visibility into how live users interact with a software system and use that insight to develop further improvements.

DEVOPS PRINCIPLES: CUSTOMER-CENTRIC ACTION

□ DevOps teams use short feedback loops with customers and end users to develop products and services centered around user needs. DevOps practices enable rapid collection and response to user feedback through use of real-time live monitoring and rapid deployment. Teams get immediate visibility into how live users interact with a software system and use that insight to develop further improvements.

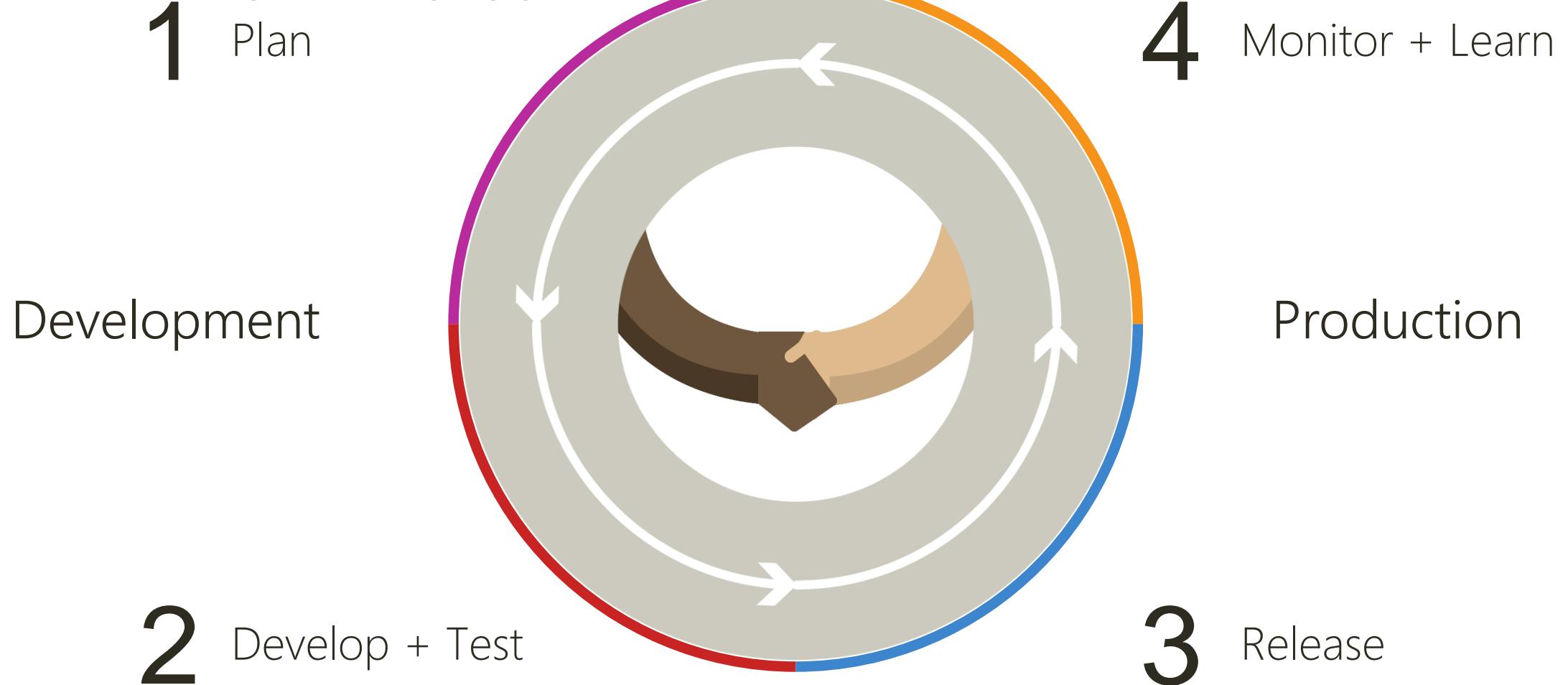
LIFECYCLE OF DEVOPS



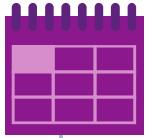
Process



DEVOPS PROCESS

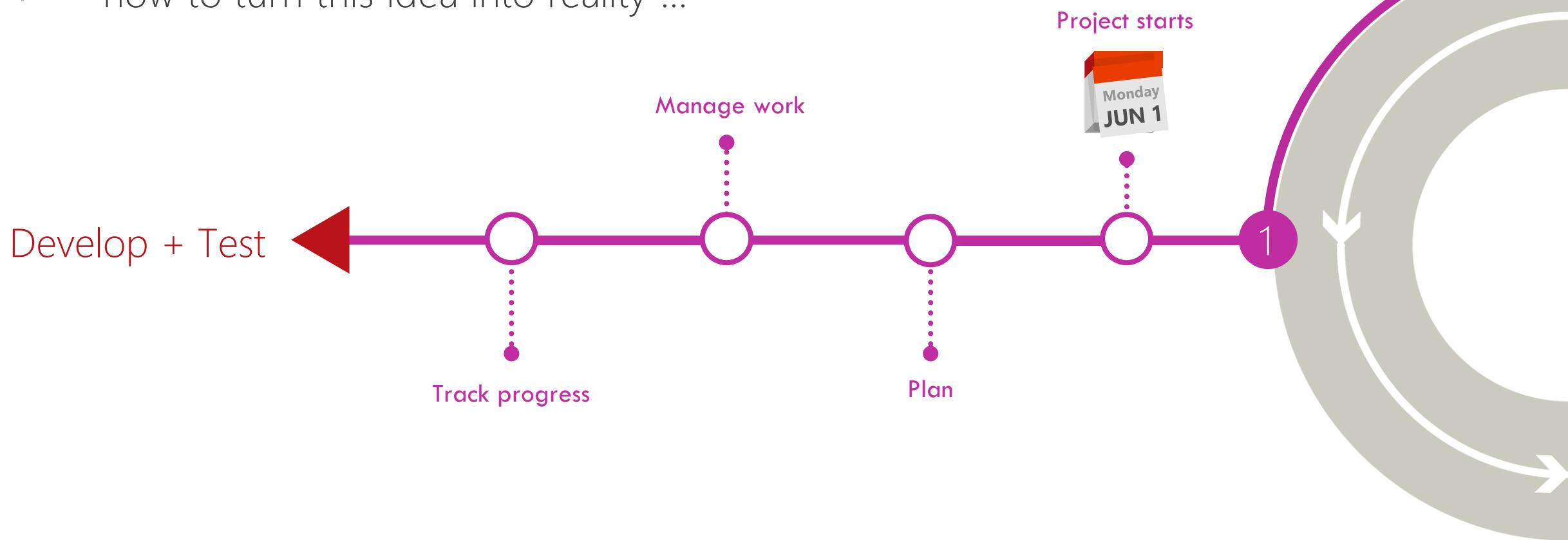


Thanks to Donovan Brown for the amazing graphics!



Plan

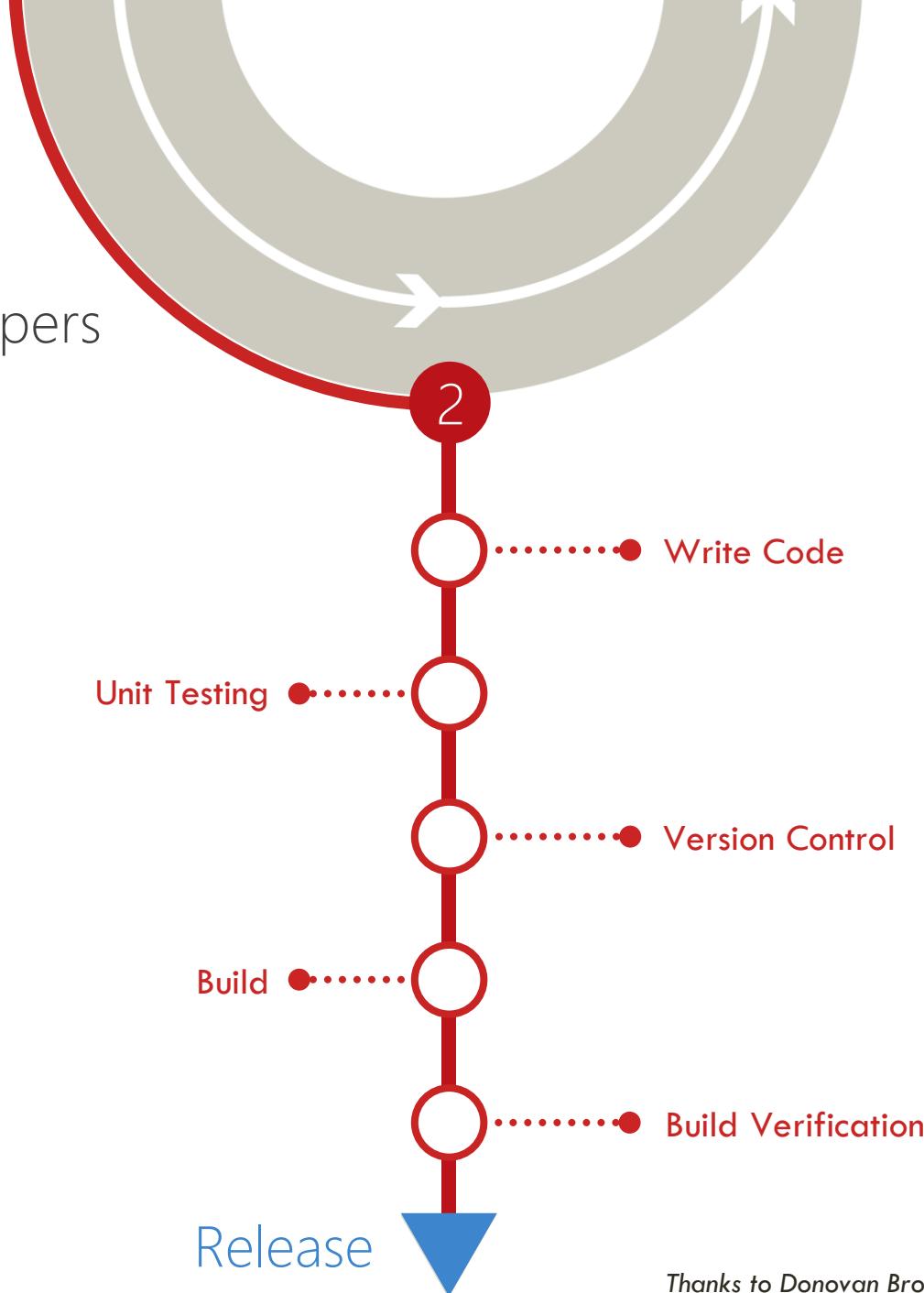
It starts with an idea – and a plan
how to turn this idea into reality ...





Develop + Test

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

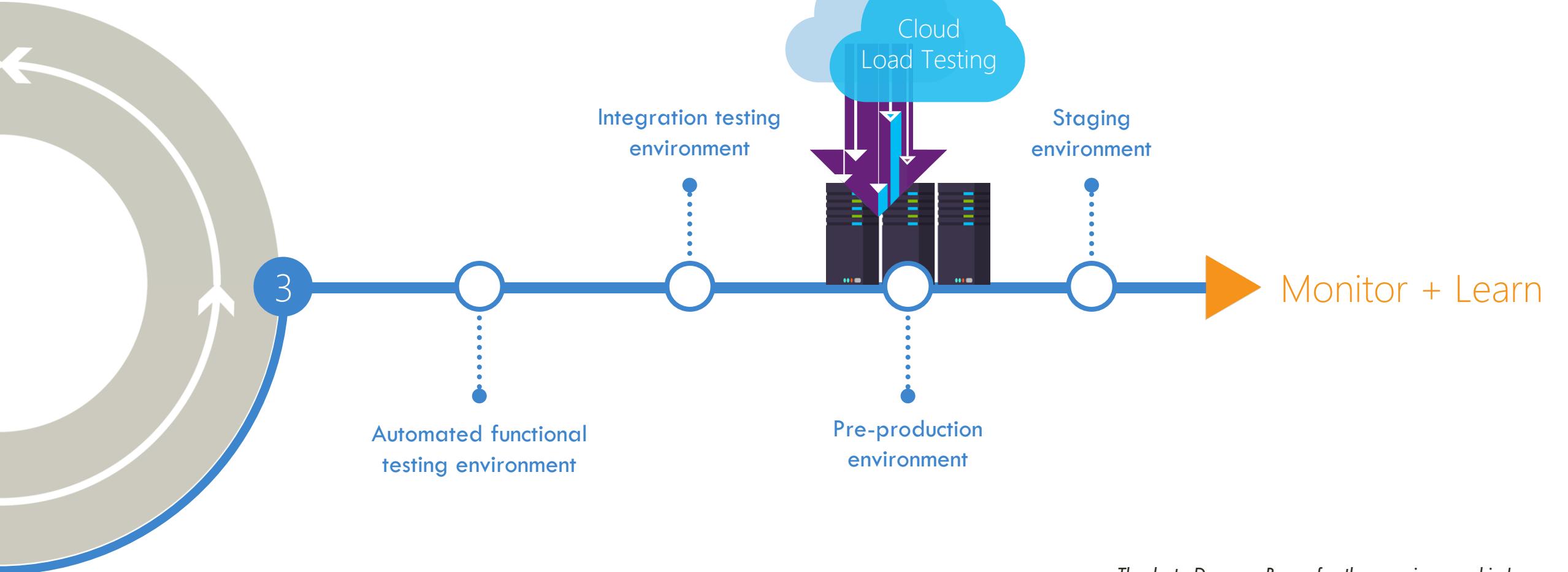


Thanks to Donovan Brown for the amazing graphics!



Release

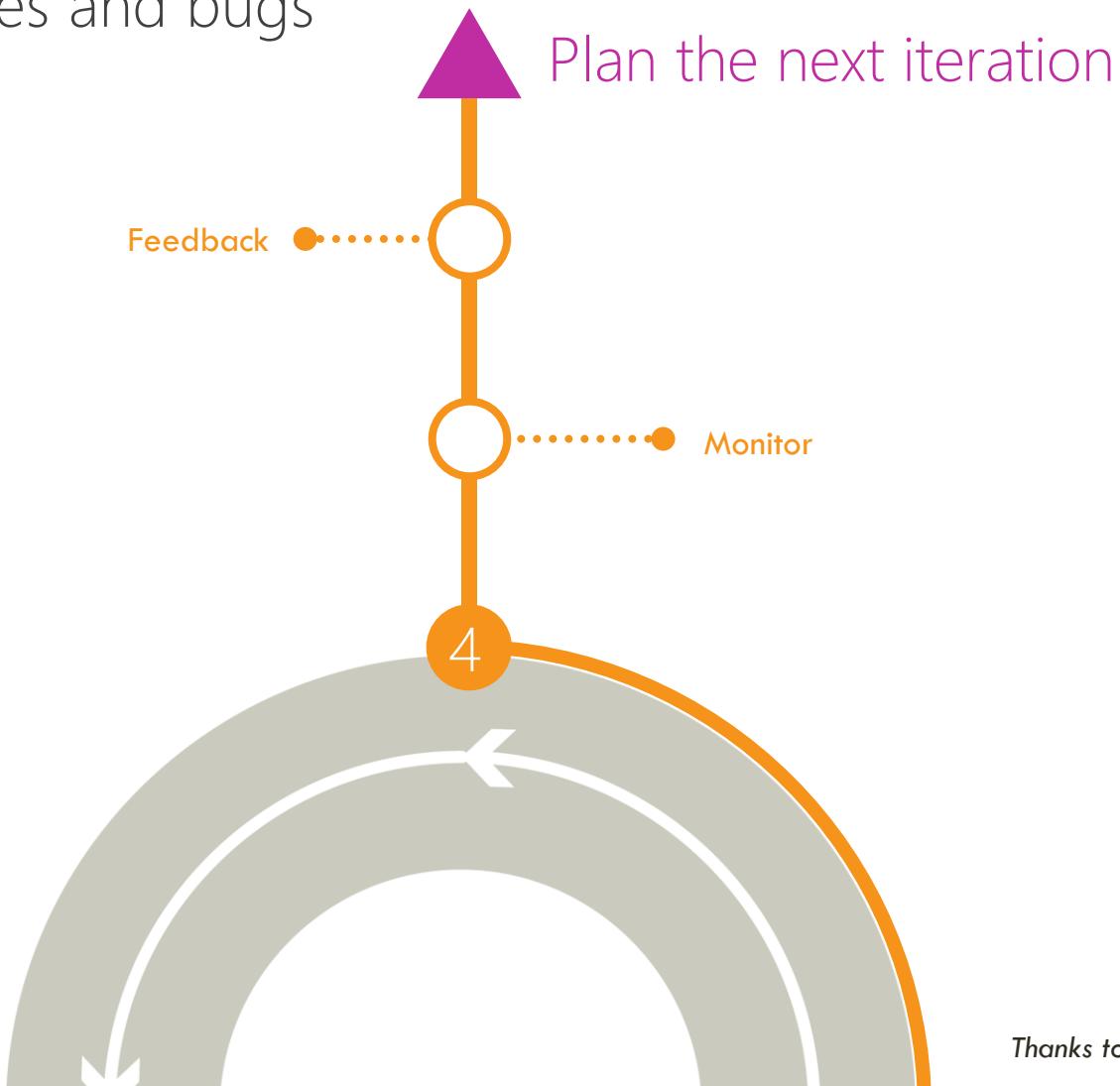
When all tests pass, the build is deployed to testing environments for each stage in the release process





Monitor + Learn

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



Thanks to Donovan Brown for the amazing graphics!

DEVOPS PRACTICES

Continuous integration

Continuous Deployment / Delivery

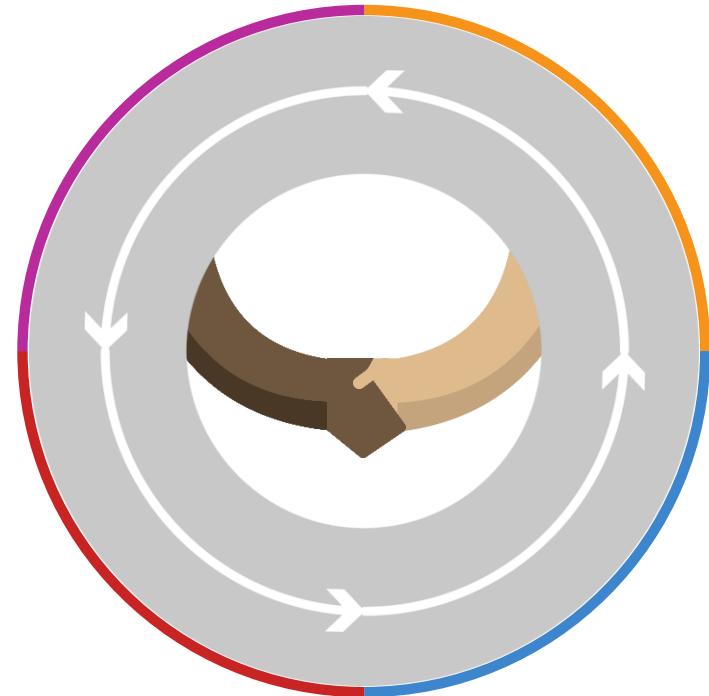
Infrastructure as Code

Automated testing

Application Performance Monitoring / Management

Release Management

Configuration Management



Doing one of these practices doesn't mean you are doing DevOps!

Source: <http://www.itproguy.com/devops-practices>

DEVOPS PRACTICES: CONTINUOUS INTEGRATION & CONTINUOUS DELIVERY

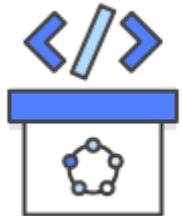
Continuous Integration

Continuous integration is a software development practice where developers regularly merge their code changes into a central repository, after which automated builds and tests are run. The key goals of continuous integration are to find and address bugs quicker, improve software quality, and reduce the time it takes to validate and release new software updates.



Continuous Delivery

Continuous delivery is a software development practice where code changes are automatically built, tested, and prepared for a release to production. It expands upon continuous integration by deploying all code changes to a testing environment and/or a production environment after the build stage. When continuous delivery is implemented properly, developers will always have a deployment-ready build artifact that has passed through a standardized test process.



DEVOPS PRACTICES: INFRASTRUCTURE AS CODE

- ❑ Infrastructure as code is a practice in which infrastructure is provisioned and managed using code and software development techniques, such as version control and continuous integration.
- ❑ The cloud's API-driven model enables developers and system administrators to interact with infrastructure programmatically, and at scale, instead of needing to manually set up and configure resources.
- ❑ Engineers can interface with infrastructure using code-based tools and treat infrastructure in a manner similar to how they treat application code. Because they are defined by code, infrastructure and servers can quickly be deployed using standardized patterns, updated with the latest patches and versions, or duplicated in repeatable ways.



DEVOPS PRACTICES: INFRASTRUCTURE AS CODE

- ❑ IaC defines system topologies and resources to allow your team to manage the resources during the coding phase reliably in a controlled manner.
- ❑ You can store and version those definitions in a version control system, so your team can review and revert quickly.
- ❑ It helps reduce human errors in complex production environments.

1	En	PERIODIC TABLE OF DEVOPS TOOLS (V1)												XebiaLabs	2	Fm	
O														Deliver Faster	Aws		
120																	
3	Ox	4	Ox														
My MySQL		Gt	Git														
Mq MSSQL		Sv	Subversion														
Pq PostgreSQL		Gh	Github	Mv Maven	Gr Gradle	Mr Meister	Jn Jenkins	Ba Bamboo	Tr Travis CI	Ar Archiva	Fn FitNesse	Se Selenium	Gn Gating	Gd Deployment Manager	Sf SmartProg	Cb Cobbler	Kb Kubernetes
Mg MongoDB		Bb Bitbucket	Br Buildr	At ANT	Bm BuildMaster	Cs Codeship	Sn Snap CI	Cs Cucumber	Nv NUnit	Cu Cucumber	Cj Cucumber.js	Qu Qunit	Cp Capistrano	Ju Juju	Rd Rundeck	Cf CFEngine	Pk Packer
Db DB2		Hg Mercurial	Qb QuickBuild	Ub UrbanCode Build	Ta Visual Build	Tc TeamCity	Sh snippet	Cc CruiseControl	Jt JUnit	Jm JMeter	Tn TestNG	Ry RapidDeploy	Cy CodeDeploy	Oc Octopus Deploy	No Nolio	Eb ElasticBox	Ad Apprenda
Cs Cassandra		Hx Helix	Msb MSBuild	Rk Rake	Lb LuntBuild	Co Continuum	Ca Continua CI	Gu Gump	Ng NuGet	Ap Appium	Xltv XL TestView	Tc TestComplete	Go Go	Ef ElectricFlow	Xld XL Deploy	Mo Mesos	Cf Cloud



Share



Embed



Become Excellent!

 [Subscribe here!](#)

Contact us!