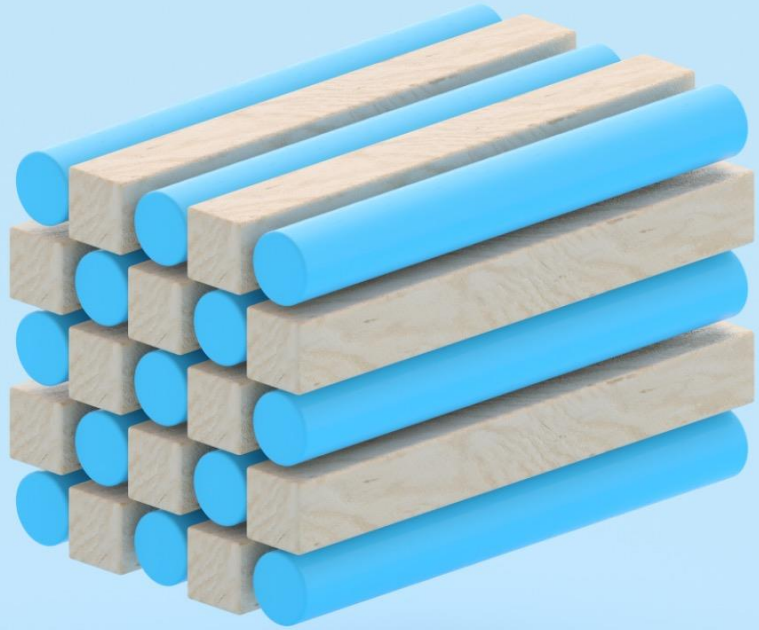
DevOps Primer

Evolution in software supply chain

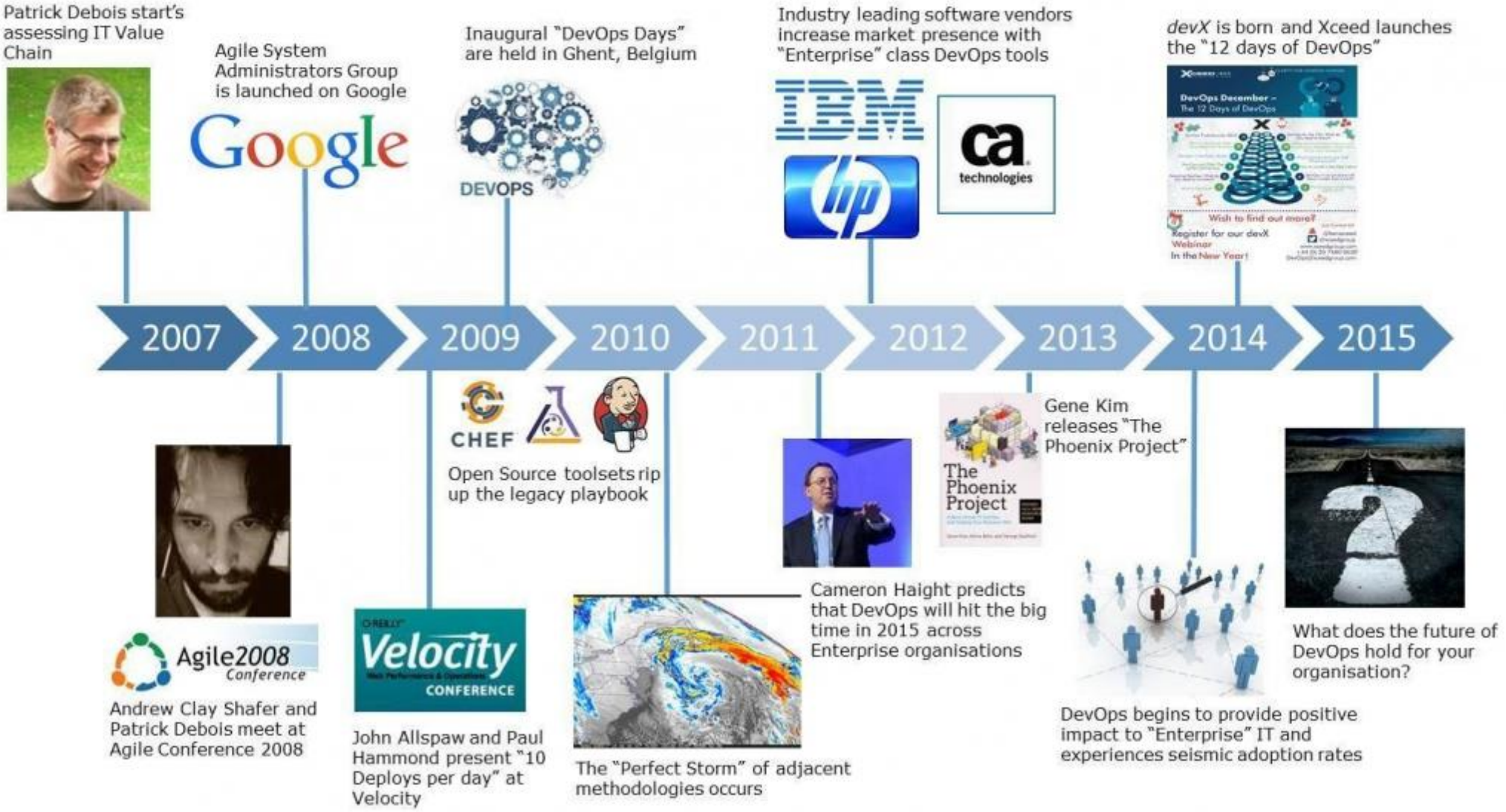
Mekka Williams
Principal Engineer, Office of the CTO (Cloud Heroes)
04/2023



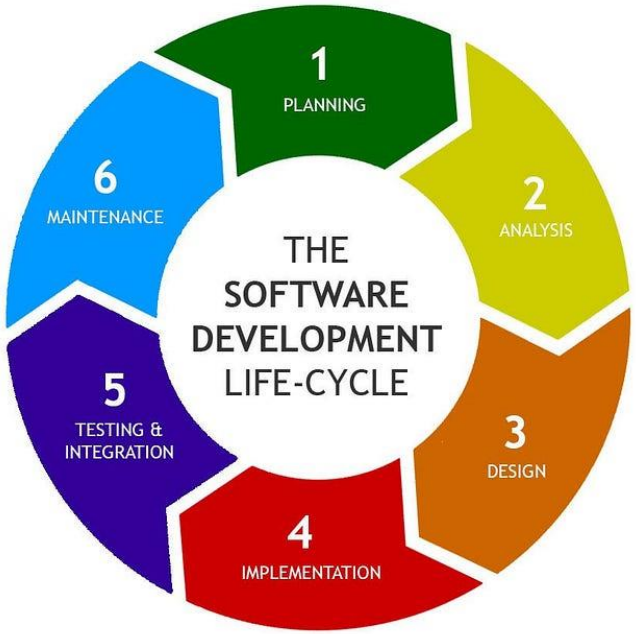
Agenda

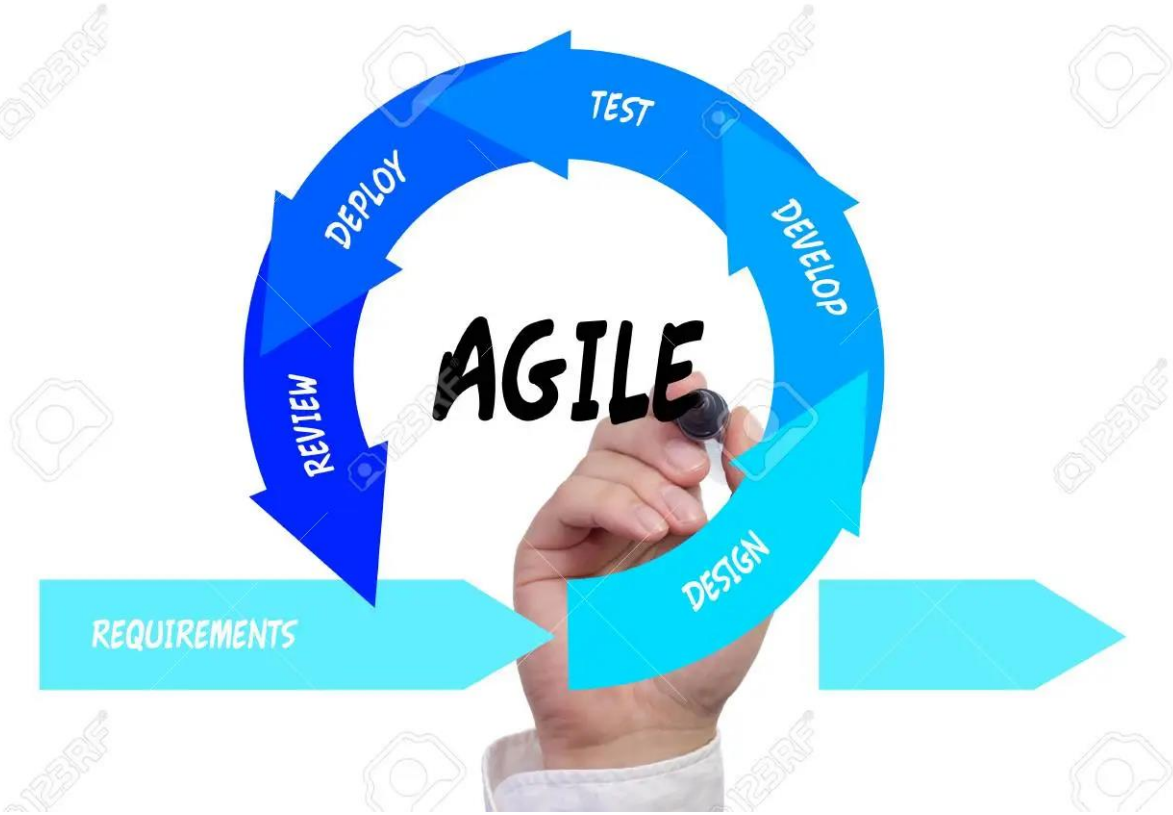
- Origins
- DevOps Basics
- State of DevOps
- Measurement/Analysis
- Future

The History of DevOps



Origins: Software Development Life Cycle





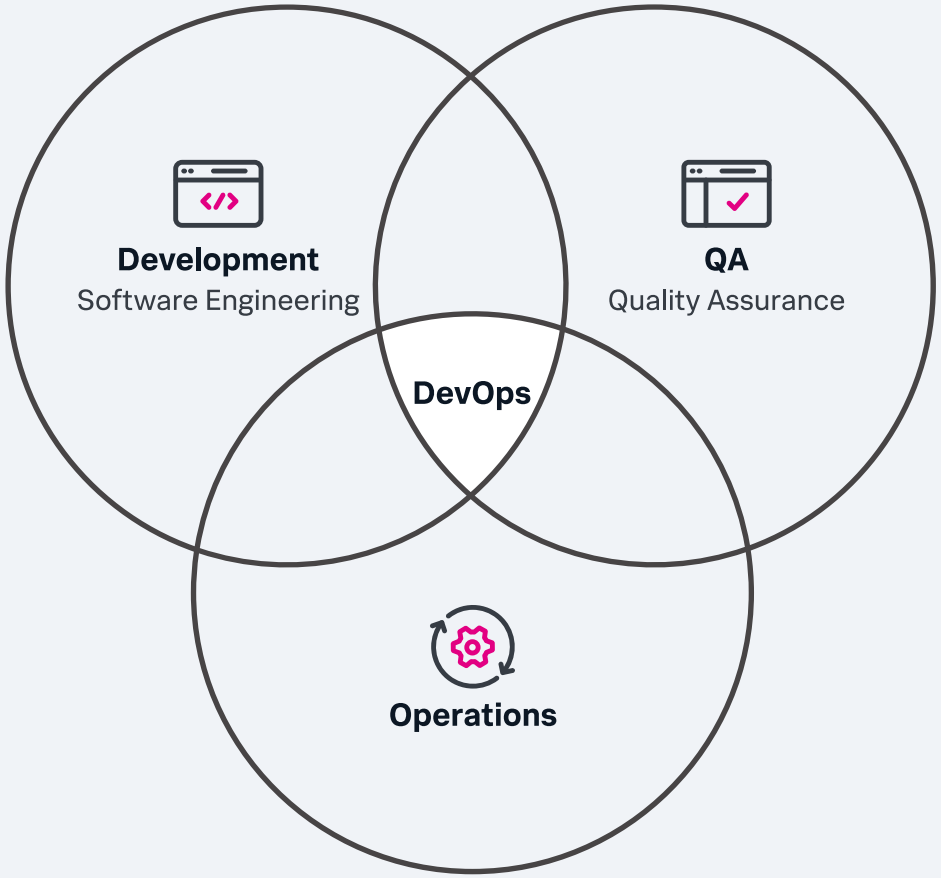
DevOps Basics

	1970s-1980s	1990s	2000s-Present
Era	Mainframes	Client/Server	Commoditization and Cloud
Representative technology of era	COBOL, DB2 on MVS, etc	C++, Oracle, Solaris, etc.	Java, MySQL, RedHat, Ruby on Rails, PHP, etc.
Cycle time	1-5 years	3-12 months	2-12 weeks
Cost	$1M-$100M	$100k-$10M	$10k-$1M
At risk	The whole company	A product line or division	A product feature
Cost of failure	Bankruptcy, sell the company, massive layoffs	Revenue miss, CIOs job	Negligible

How do we deliver quality software in the most efficient way, continuously improving our utilization of resources while continuing to focus on increasing value to end users?

DevOps Basics

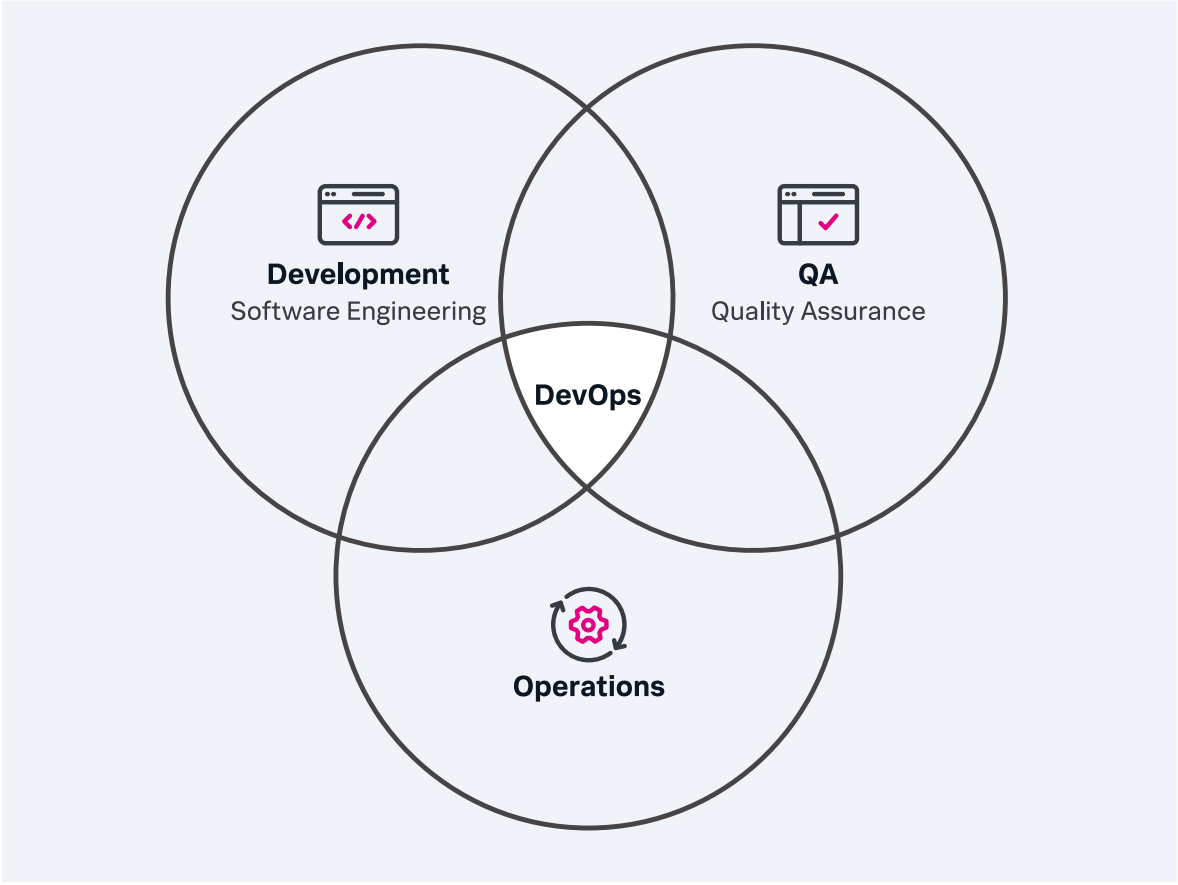
DevOps is a Culture!



DevOps Basics: What is DevOps?

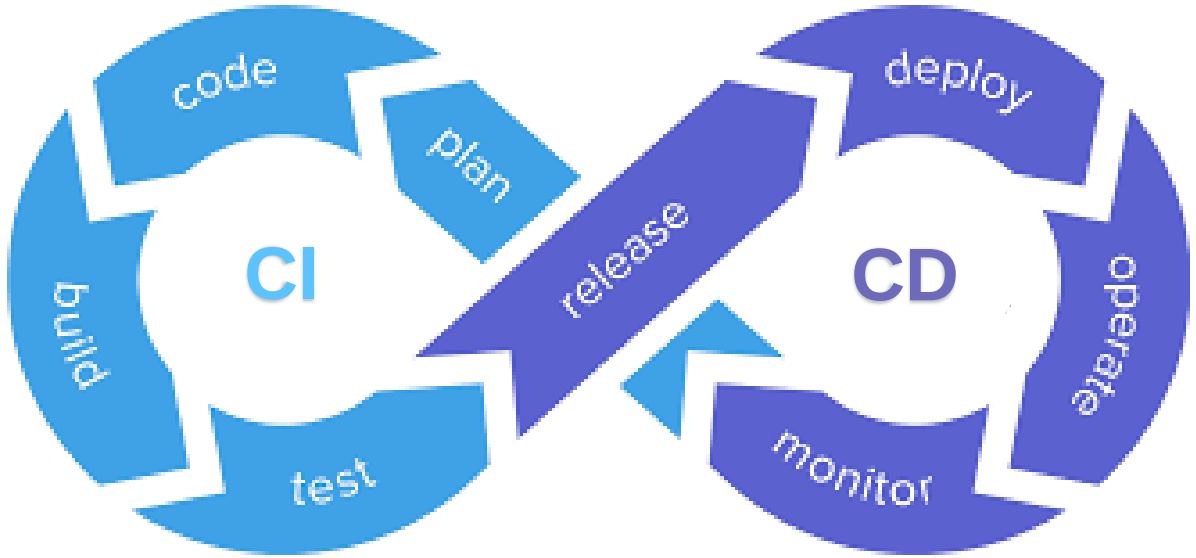
DevOps is a Culture!

A culture of collaboration between software development and operational teams that fosters a shared responsibility model for efficiently executing the tasks required to propagate through the software development lifecycle, through to production, with a focus on delivering increased value to customers.



DevOps Basics: DevOps in Practice

- Development
  - Plan
  - Code
  - Build
  - Test
- Operations
  - Release
  - Deploy
  - Operate
  - Monitor
- Repeat!
  - Continuous Feedback
  - Continuous Improvement



DevOps Basics: CI/CD Explained

CI/CD Explained

CI/CD –

A method to frequently deliver verified software functionality to end users by automating the stages of software development. The main concepts of CI/CD are Continuous Integration, Continuous Delivery and Continuous Deployment

Continuous Integration –

A software development practice where developers integrate code into a shared repository frequently (up to several times a day). Those integrations are then *automatically* verified by build and test automation.

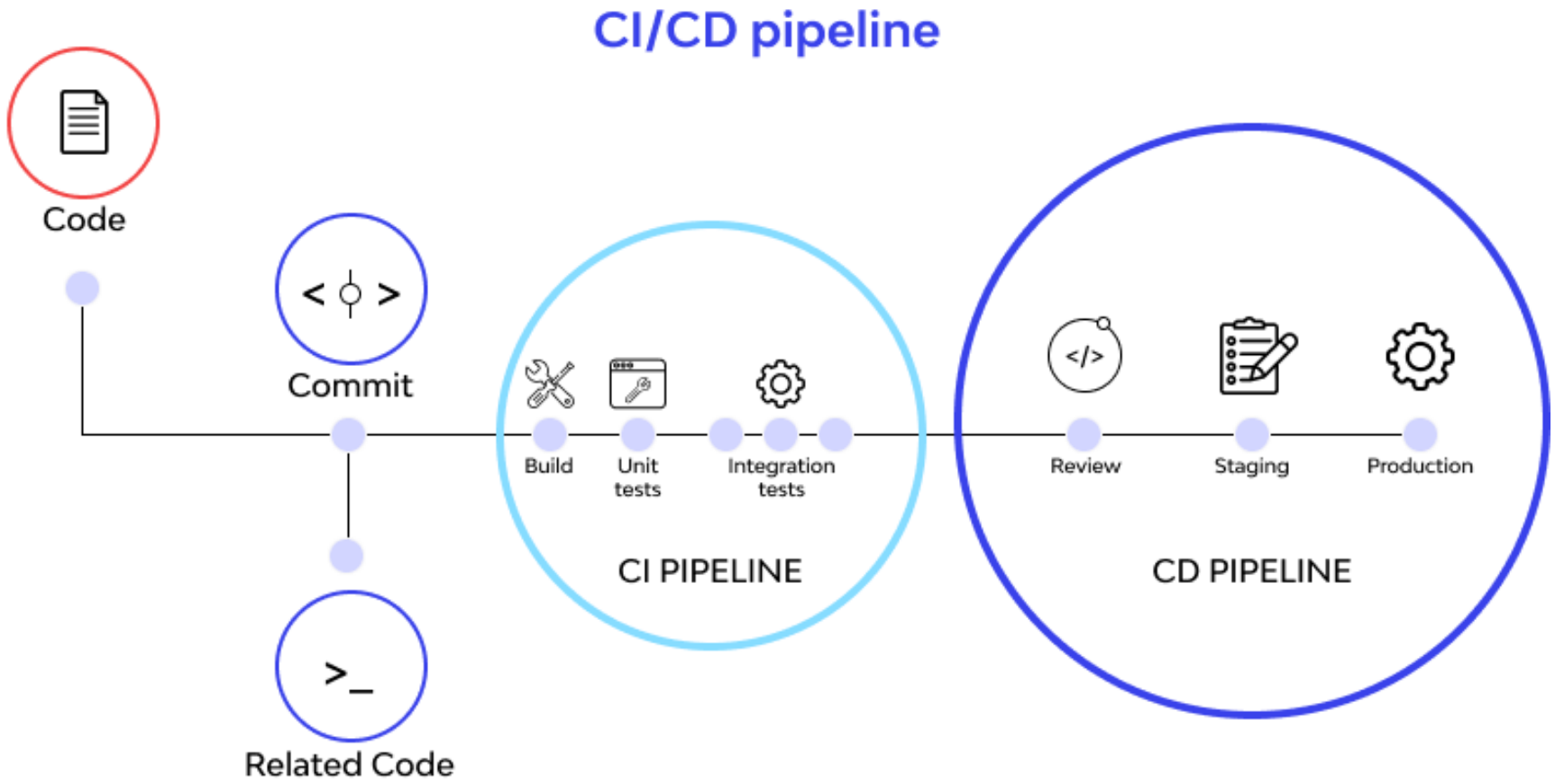
Continuous Delivery –

A software development practice that *automatically* prepares newly integrated software for release to production.

Continuous Deployment –

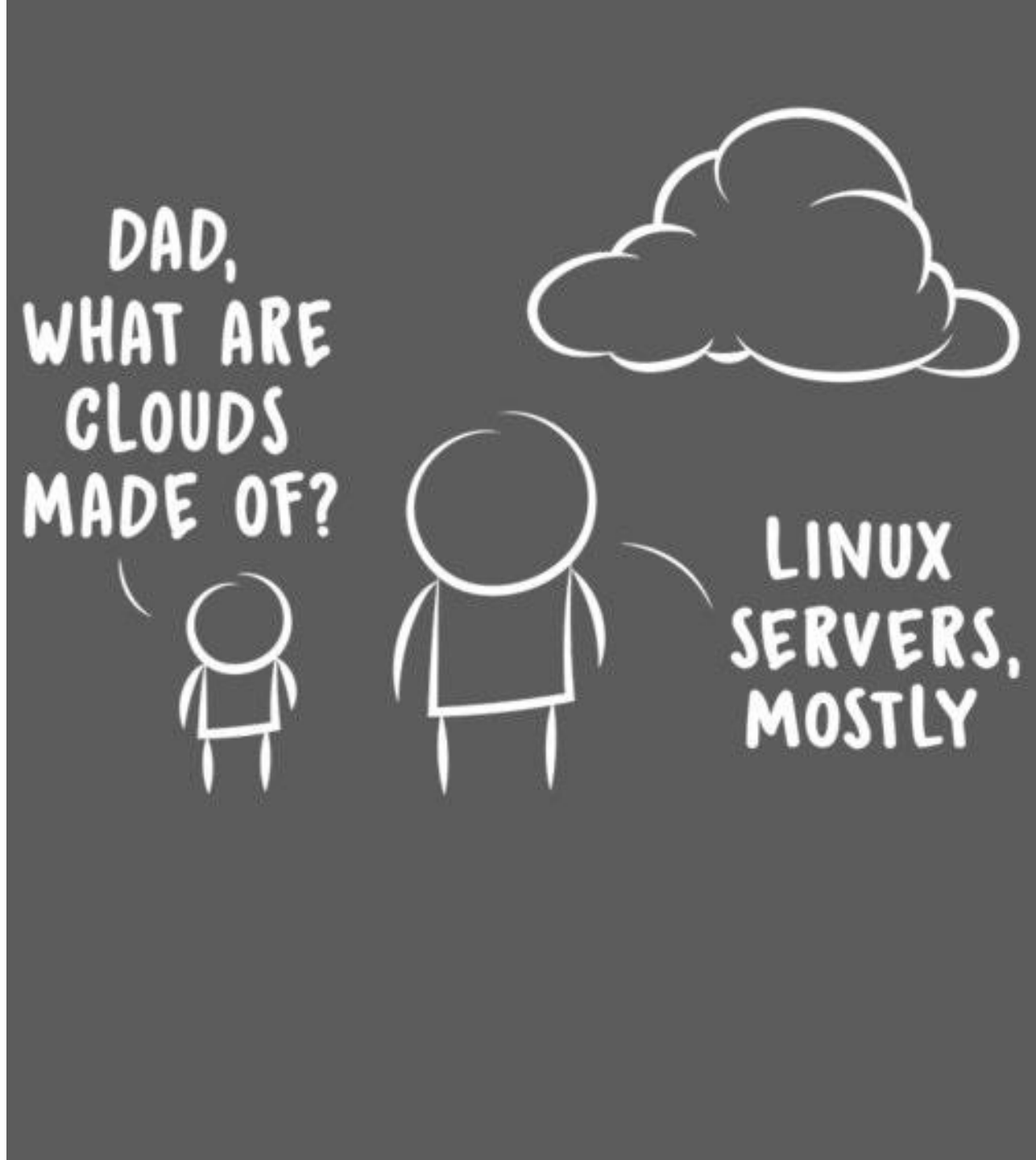
A software release strategy that *automatically* pushes appropriately verified developer changes through any required pipeline stages, into the production environment and made available to end users.

DevOps Basics: CI/CD Explained



Real Quick: What Is The Cloud

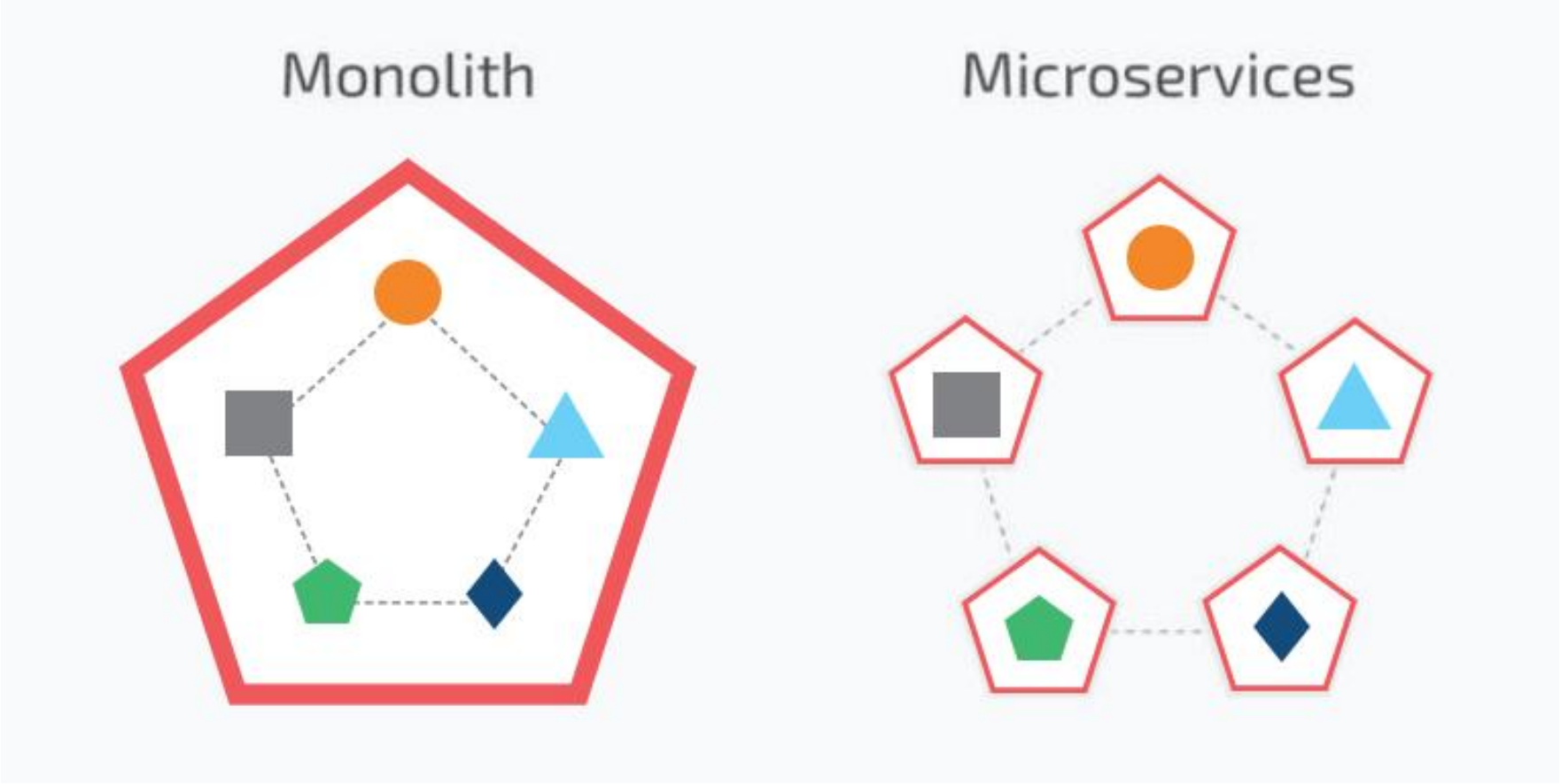
- Computing service that is provided by third-party companies over the internet, where customers can access and use computing resources like storage, processing power, and applications without having to invest in and maintain their own infrastructure.
- Offered on a pay-per-use basis, allowing customers to scale up or down their usage as needed, and providing them with flexibility and cost-efficiency.
- Offers a variety of architecture models to suit a variety of software products. Enables rapid development, delivery and deployment due to the “outsourcing” of operations.



Cloud and DevOps

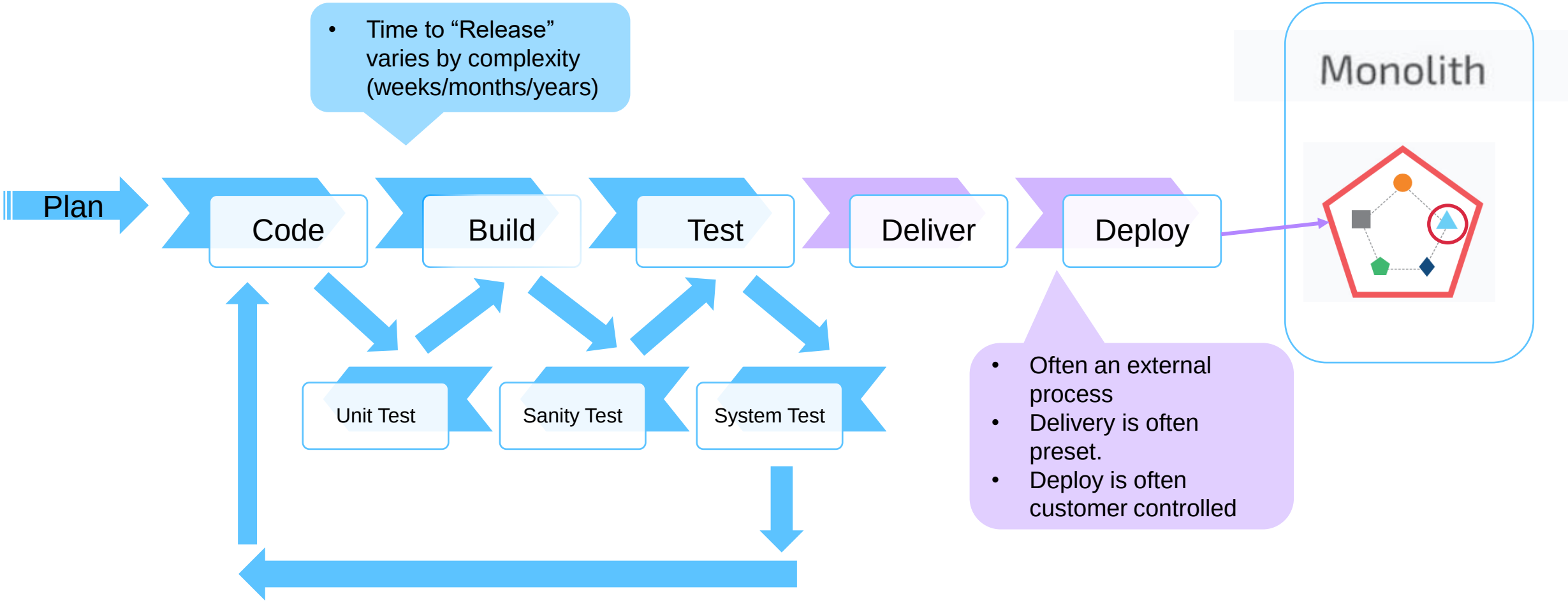
Pipelines: Monoliths vs Microservices

Software Development Modernization



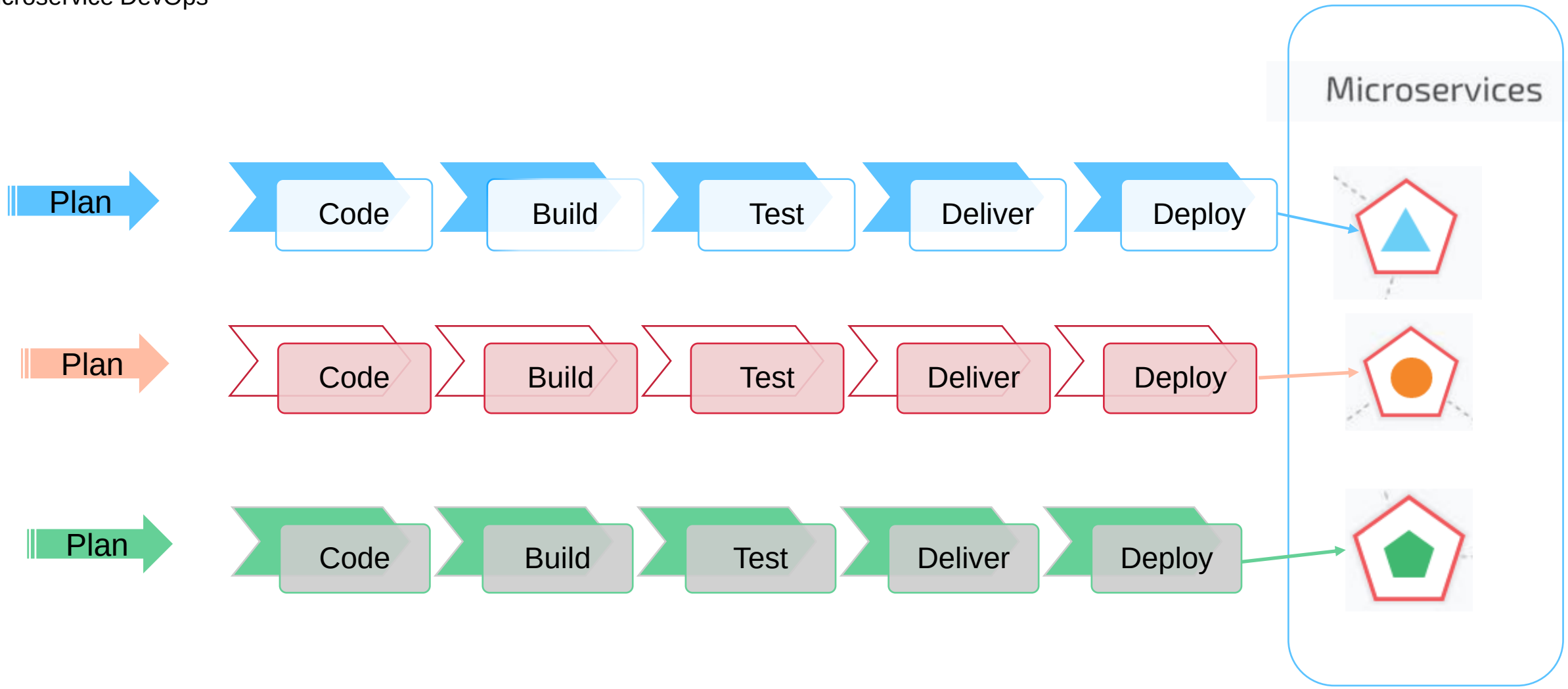
Pipelines: Monoliths vs Microservices

Monolith DevOps



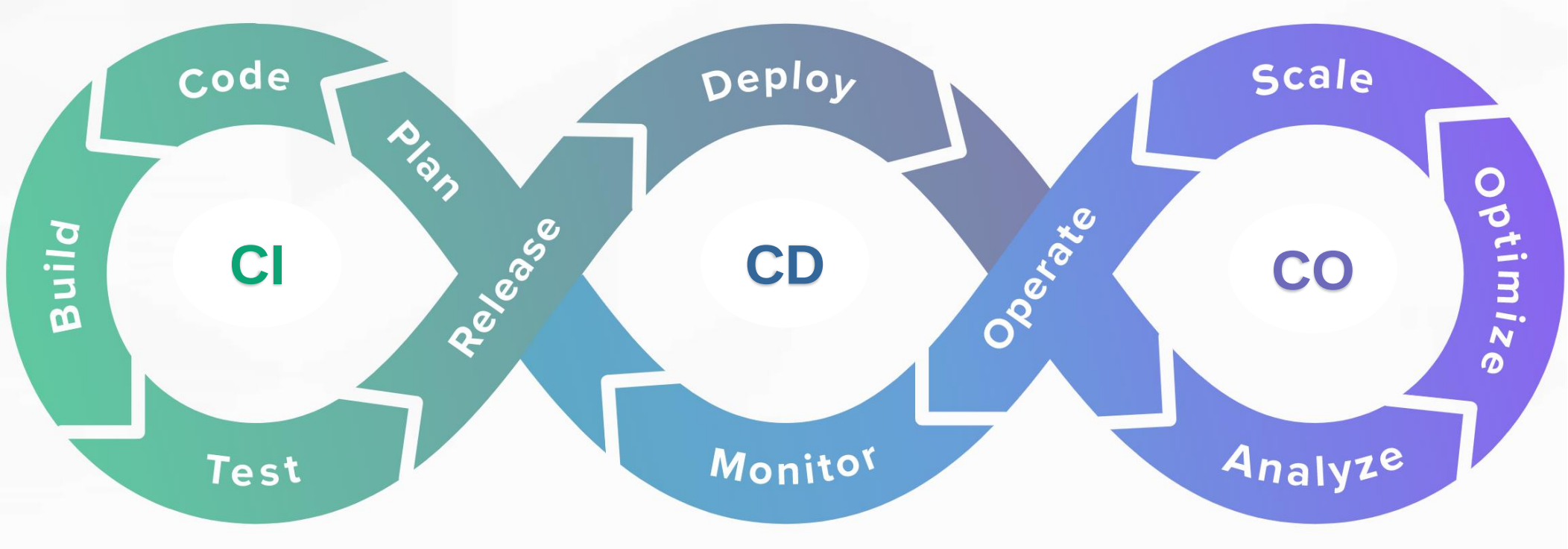
Pipelines: Monoliths vs Microservices

Microservice DevOps



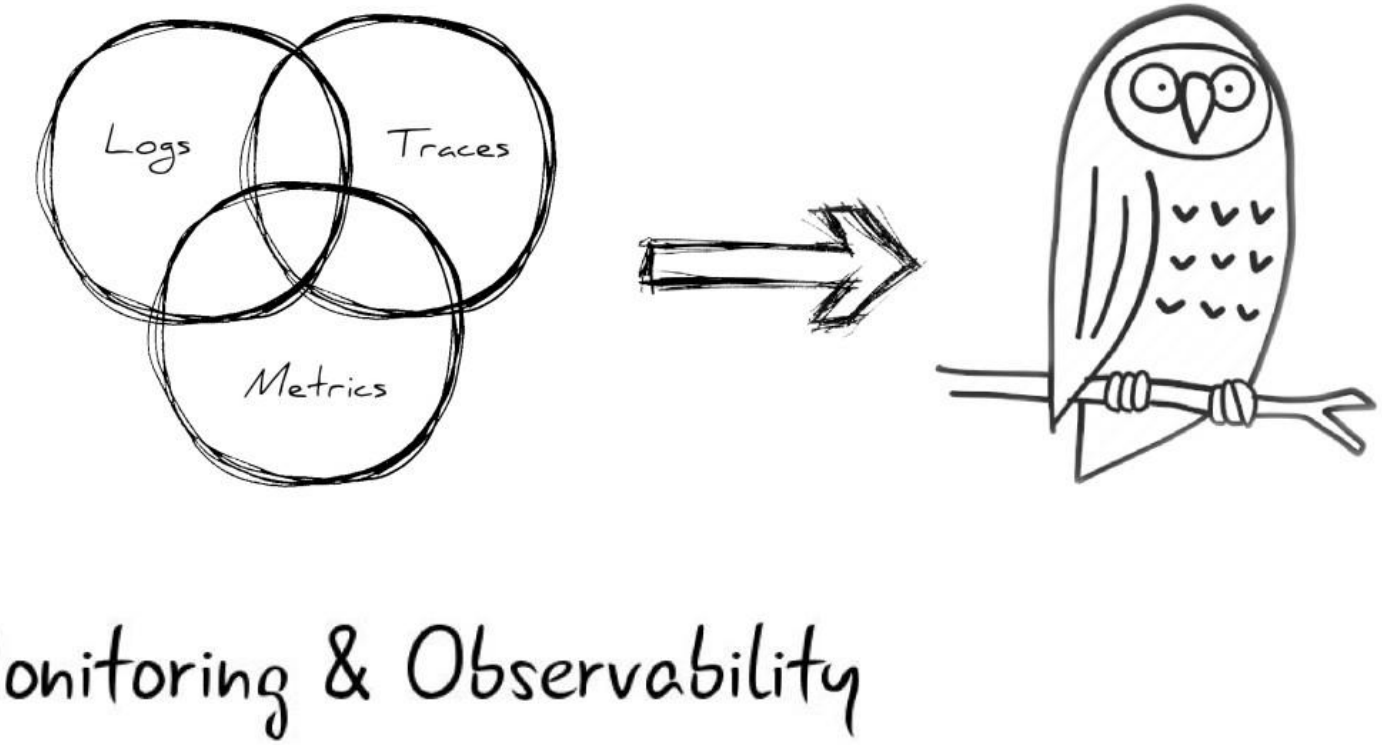
Continuous Optimization

The Goal



Continuous Optimization

Monitoring and Observability



How Do We Measure?

Traditional Metrics

- Deployment Frequency
  - Rate of production deployments
- Lead Time
  - Time it takes to implement, test and deliver code
- Mean Time To Restore (MTTR)
  - Time it takes to restore service after failure incident
  - Total Maintenance Time/Number of Repairs
- Change Failure Rate
  - The rate at which faulty code makes it into production (per deployment)
  - Total Number of Failed Deployments/Total Number of Deployments
- Monitoring and Observability

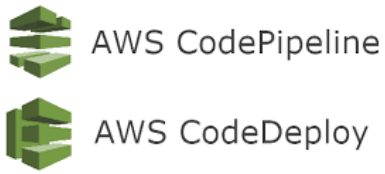
Automation

Automate all of the things

- Infrastructure/Configuration
- Development Environment
- Build
- Test
- Delivery
- Deployment
- Feedback
- Improvement
- Cost Optimization

“*Automation applied to an inefficient operation will magnify the inefficiency*”
- Bill Gates

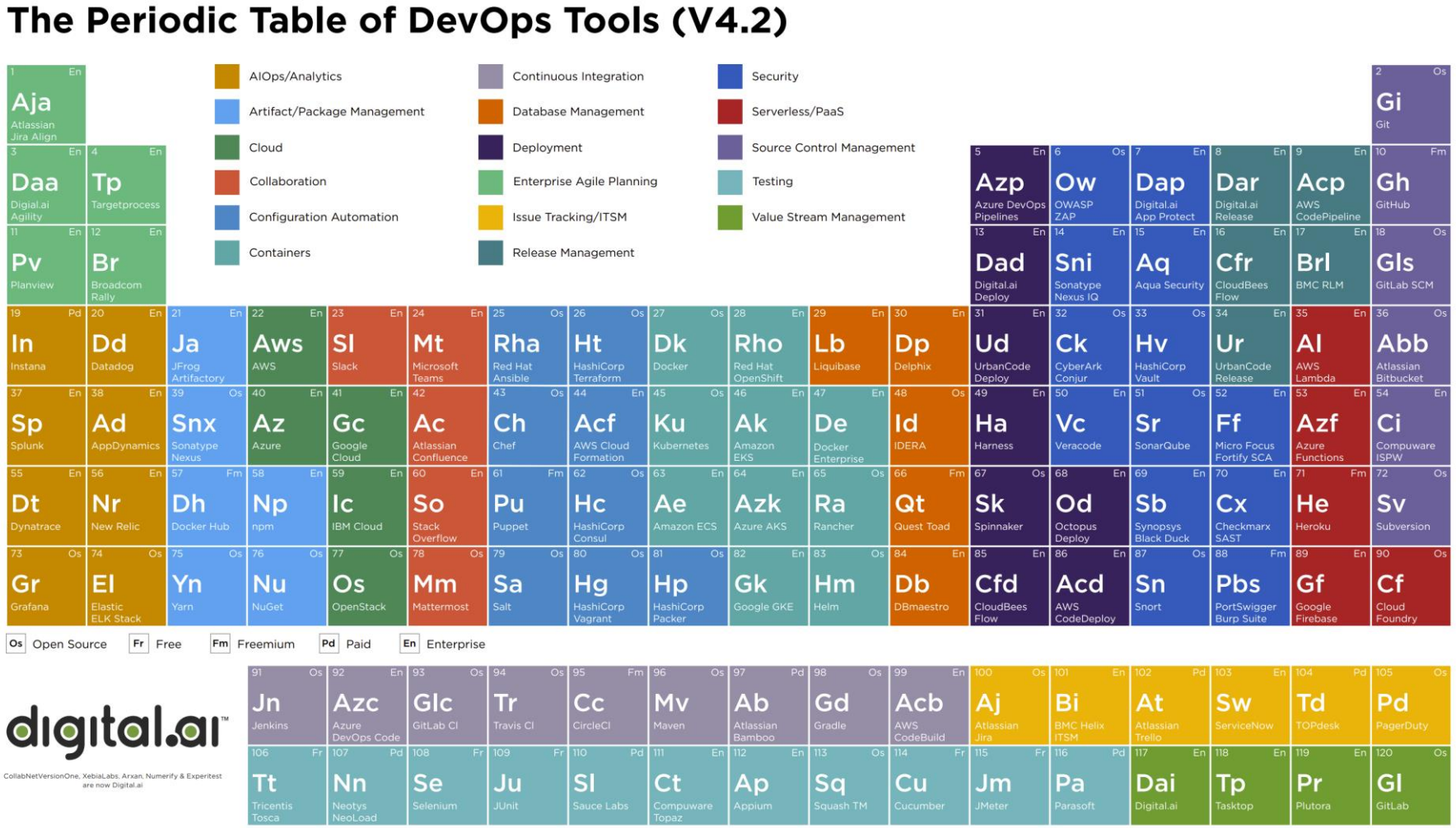
Tools Landscape



GitLab

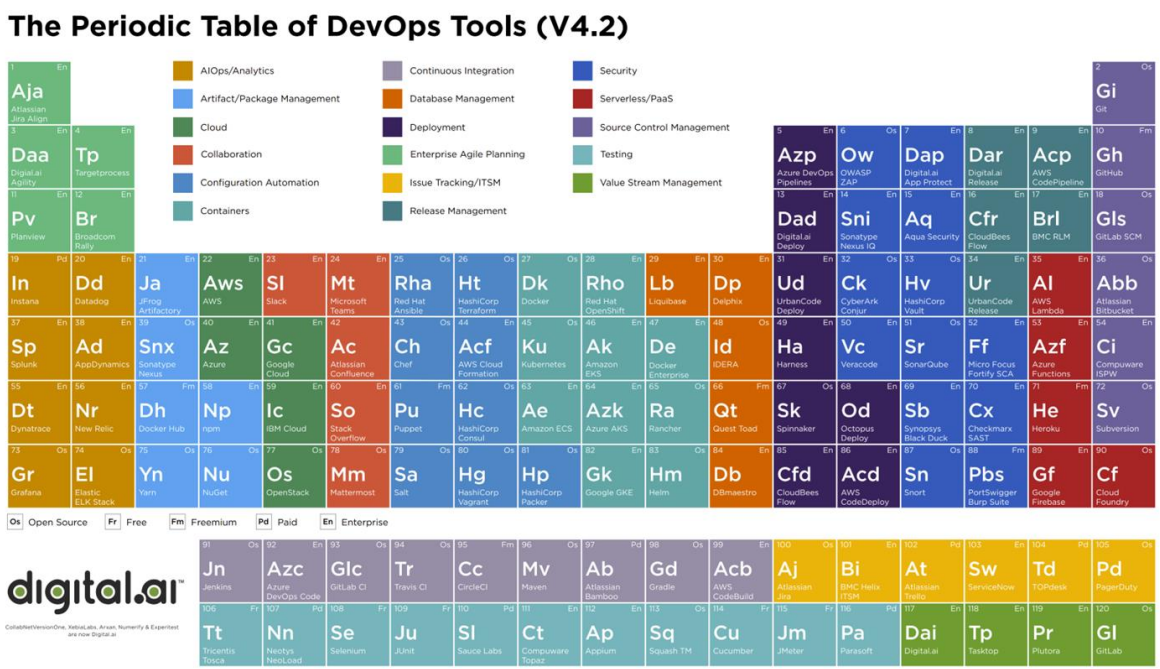
GitHub Actions

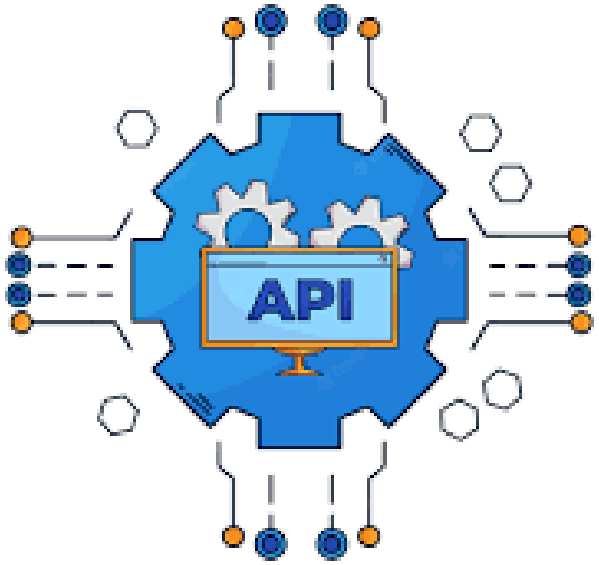
Tools Landscape



DevOps Futures

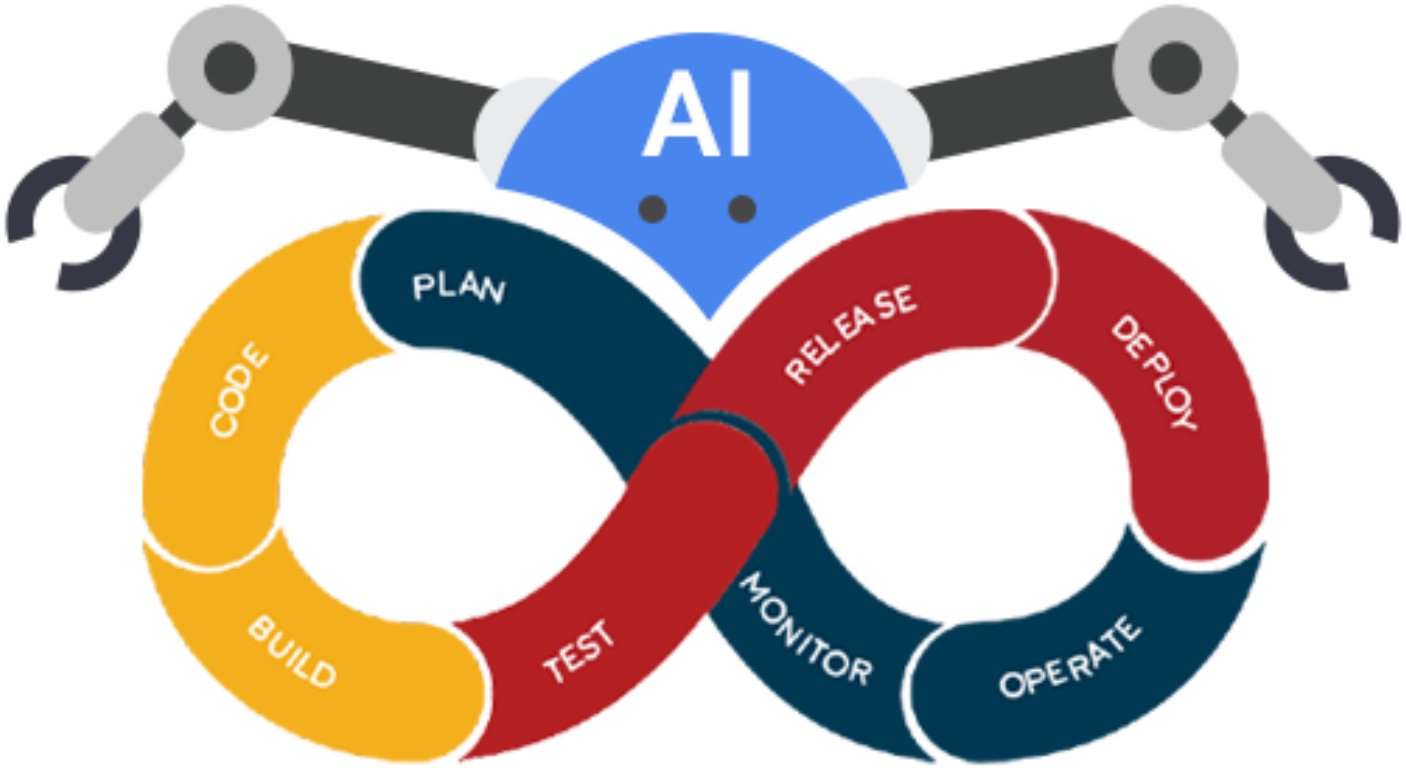
DevOps-as-a-Service, Platform Engineering





DevOps Futures: AI

Low Code, No Code, Generative AI



References

- DORA – DevOps Research and Assessment
- Monolith to Microservices – Sam Newman
- DevOps Benchmark Studies (Humanitec 2021)
- Periodic Table of DevOps Tools (Digital.ai)
- The Phoenix Project
- The DevOps Handbook

Thank You!