DEVOPS INTRODUCTION

What is DevOps

Definition

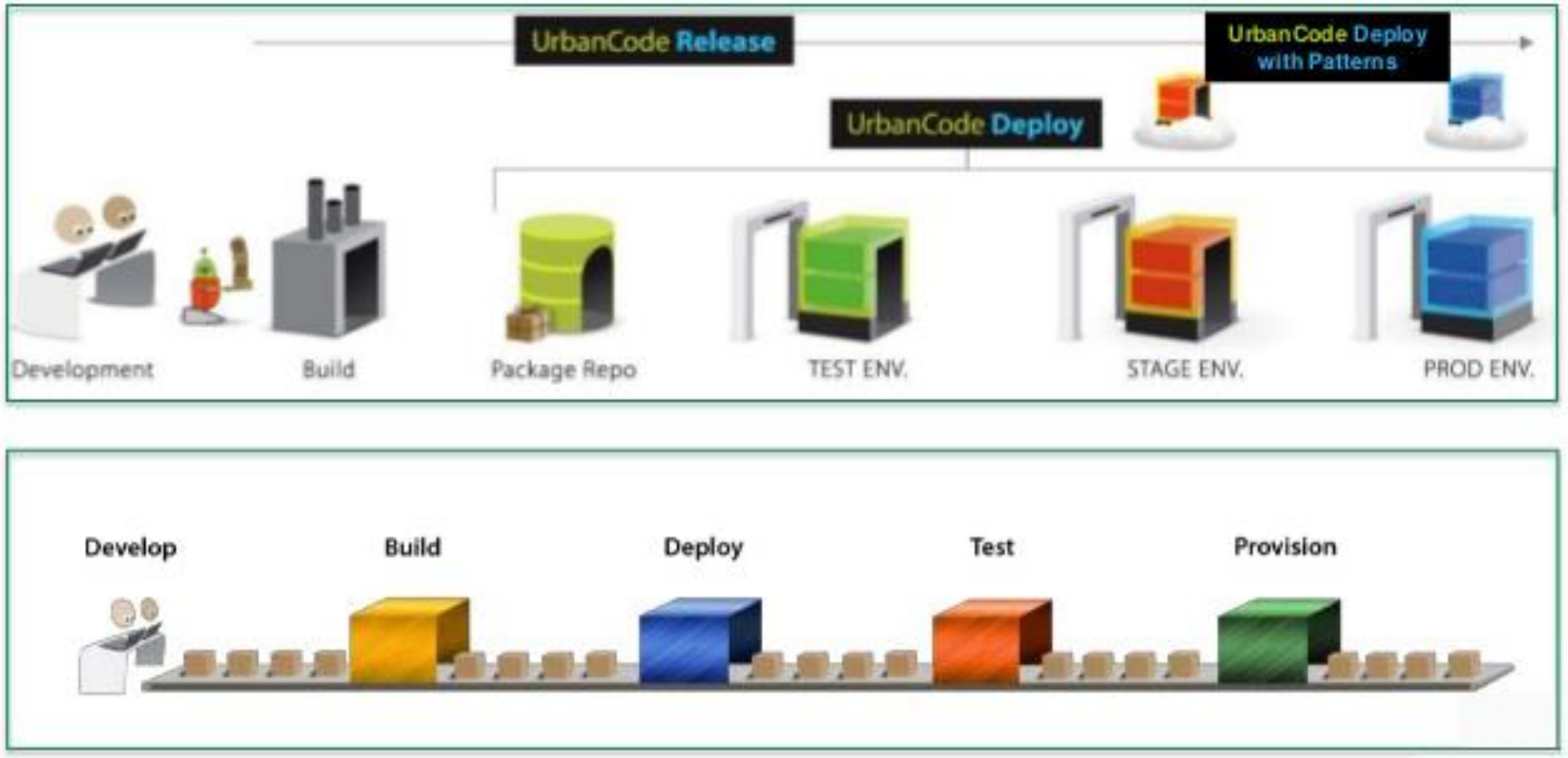
- Wikipedia: DevOps is a set of practices that combines software development (Dev) and IT operations (Ops)
- Microsoft: DevOps is the union of people, process, and products to enable continuous delivery of value to our end users. “Donovan Brown”
- Amazon: DevOps is the combination of cultural philosophies, practices, and tools that increases an organization’s ability to deliver applications and services at high velocity

Manufacturing Automation

- On December 1, 1913, Henry Ford installs the first moving assembly line for the mass production of an entire automobile.
- This reduced the time it took to build a car from more than 12 hours to two hours and 30 minutes.
- Car companies that resisted that change simply perished.

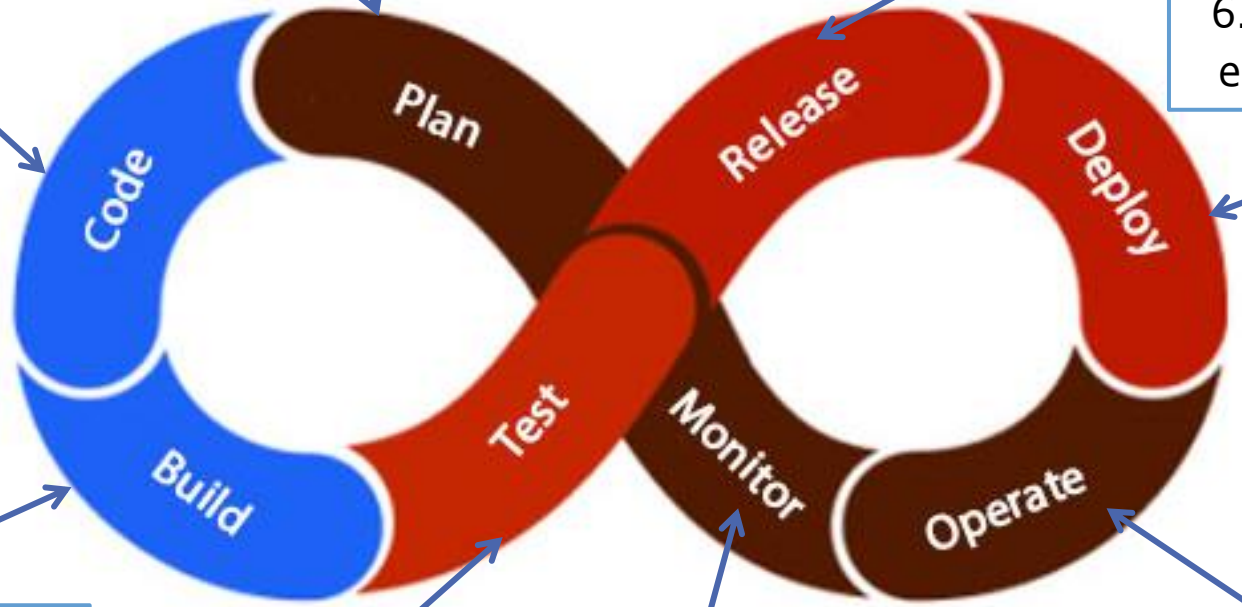


DevOps is the Assembly Line of software development



1. Plan the type of application you want to create

2. Write the code



3. Integrate various codes to create a build

4. Test built application with automated testing tools

5. Plan, schedule and control the built process in a different environment

6. Deploy the code into a cloud environment for further usage

7. Delivery to the customer for daily use

8. Monitored any issues, failures, errors…

Why DevOps?

Reduced deployment failures





Micro-services architectural

Shorter development cycles





Career boost

Faster Innovation Cycles





Widely adopted Tools

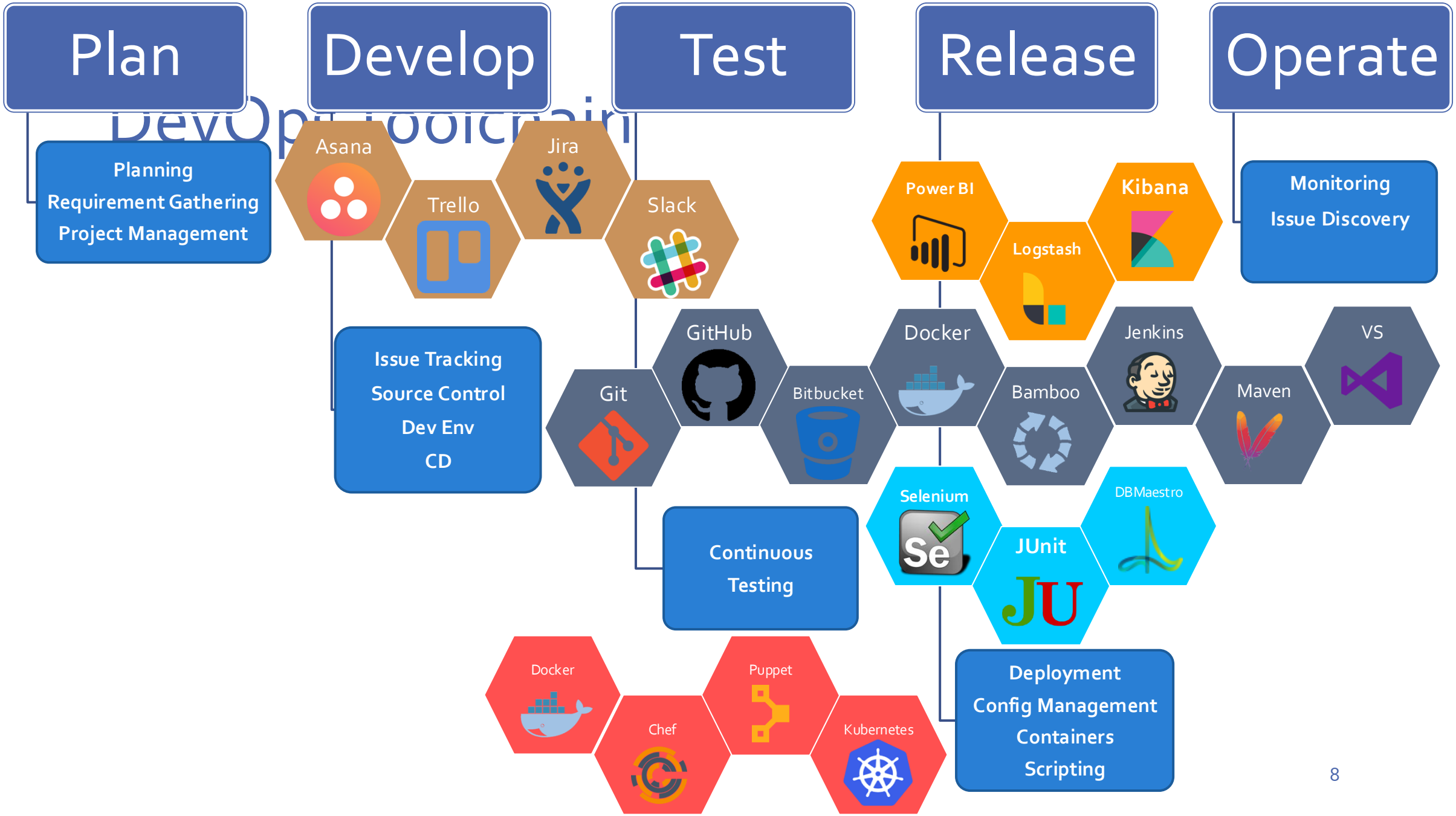
Scalability





Reduced costs

DevOps Toolchain



DevOps Engineer

- Understands the Software Development Lifecycle and has the full understanding of various automation tools for developing digital pipelines.
- Works with developers and the IT staff to oversee the code releases.
- DevOps engineers usually have three different positions:

1. **DevOps Engineers, with a Dev Bias:** Work in a software development role, building software applications.
2. **DevOps Engineers, with an Ops Bias:** Could be compared to systems engineers or systems administrators.
3. **SREs (Site Reliability Engineers):** focus on creating scalable, highly available and reliable software systems.

DevOps Engineer Skillset

1. OS: Most of the companies have their environment on Linux. DevOps engineer should be aware of Linux fundamentals and knowledge on at least one scripting language is a must (Python, Ruby, Pearl).
2. DevOps Tools: development, testing, deployment and monitoring tools.
3. Continuous Integration and Continuous Delivery (CI & CD): done through the utilization of automation tools in an Agile environment.
4. Infrastructure as Code: the ability to manage automated scripts that can provision infrastructure needed by developers.
5. Cloud Experience: experienced deploying applications in a cloud ecosystem (AWS, Google, Azure…) as well as a clear understanding of IaaS and PaaS.

DevOps Engineer Skillset

6. Developmet Skills: an understanding of software development, and writing scripts that will orchestrate the deployments of DEV and Production environments via tools such as Chef, Puppet, CFEngine or other tools of this kind
7. Soft Skills
   - Management Skills: DevOps engineer would be required to be a bridge between the silos and bring different teams together to work towards a common goal.
   - People Skills: a DevOps engineer would be working with multiple teams, made up of Humans!
   - Communication Skills
   - …

Measuring DevOps Success

- Deployment Frequency
  - Deployment: a software deployment to production or to an app store.
  - Tells you how often you're delivering something of value to end users.
- Change Lead Time
  - The time it takes to go from code committed to code successfully running in production.
  - Obtained by capturing the time at which each revision was initiated in the development pipeline, and the time at which that same revision successfully runs the last action of the deployment pipeline.
  - Averaging these numbers over time → mean time lead

Measuring DevOps Success

- Change Fail Rate
  - A measure of how often deployment failures occur in production that require immediate remedy (like rollbacks).
    1- Track each deployment and indicate whether it was successful or not.
    2- Track the ratio of successful to unsuccessful deployments to production over time.
- Mean Time to Detect & Repair (MTTR)
  - The average time between the start and resolution of an incident.