Chapter 2 – DevOps Overview

Learning Topics

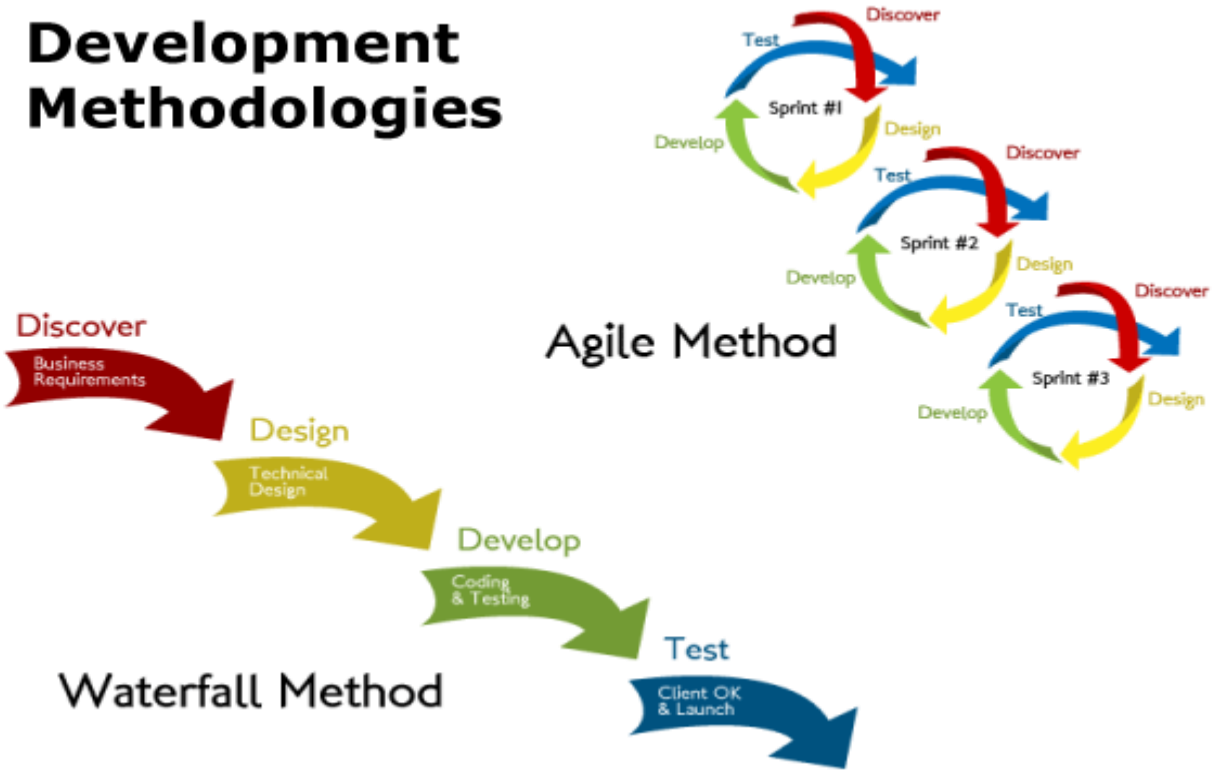
- Overview of Agile
- Agile and DevOps
- What is DevOps?
- DevOps Tool Chain
- DevOps Lifecycle
- DevOps & Automation

Software Development Methodologies

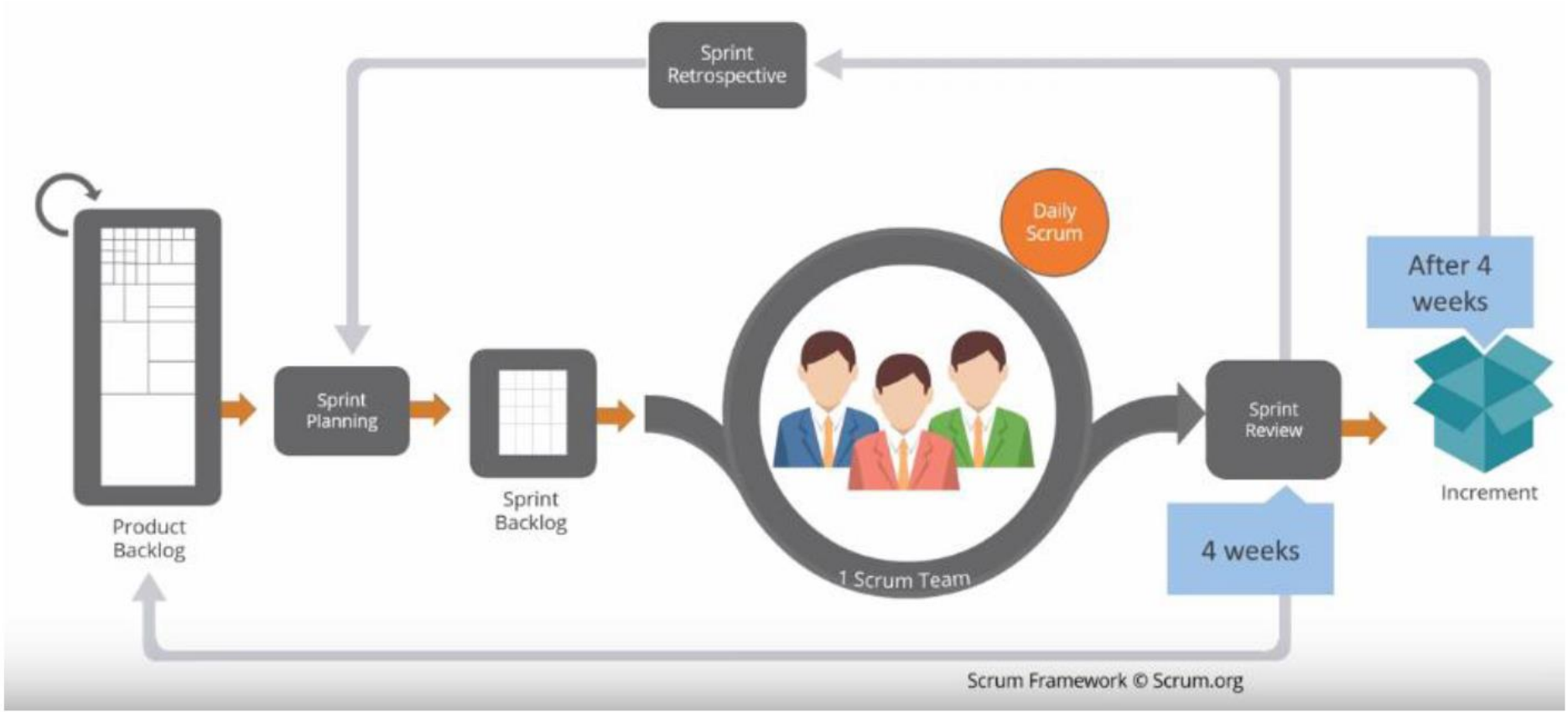
Waterfall believes a team should only start making its software ready for release when all of the functionality for the release has been developed.

Agile introduces the idea that the team should get their software ready for release throughout development. Many variations of agile believe this should be done at periodic intervals.

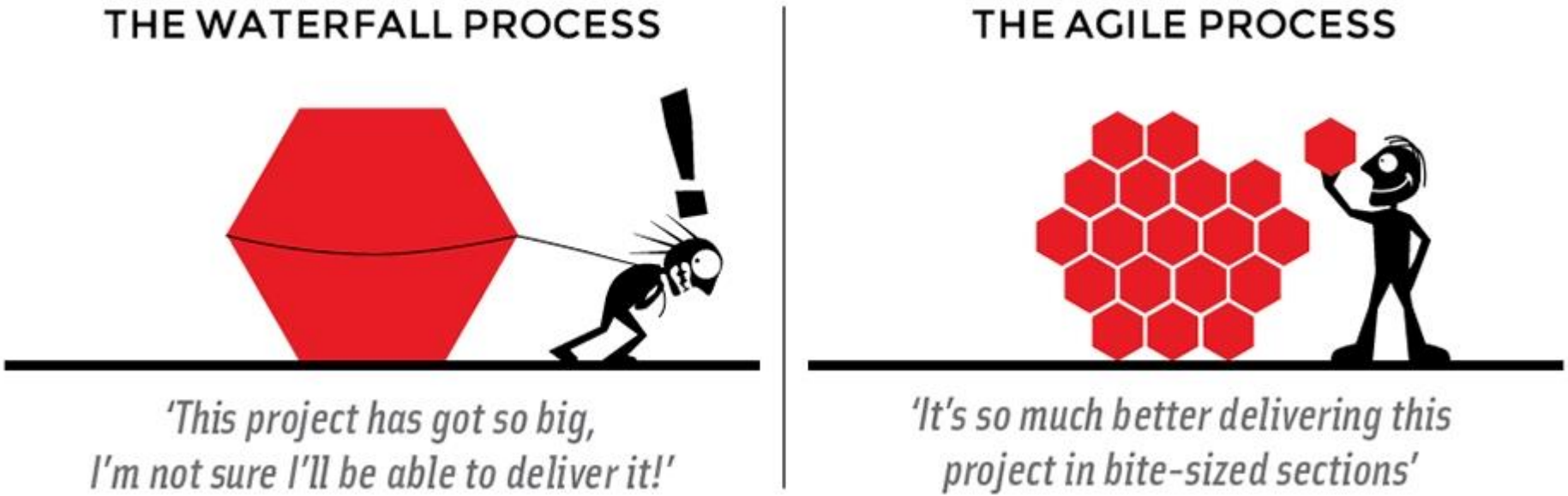
Waterfall Vs Agile



Agile Workflow



Waterfall Vs Agile

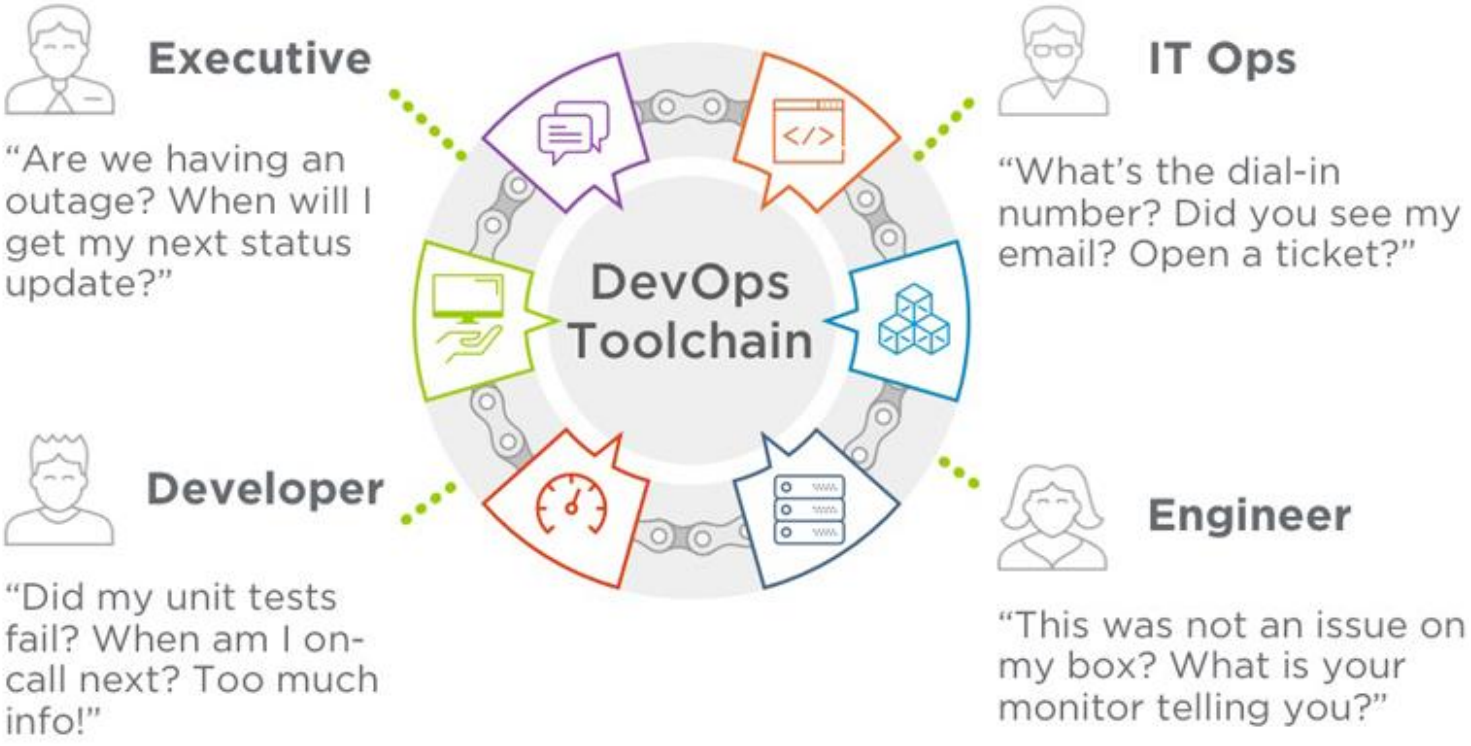


Reduce Blast Radius

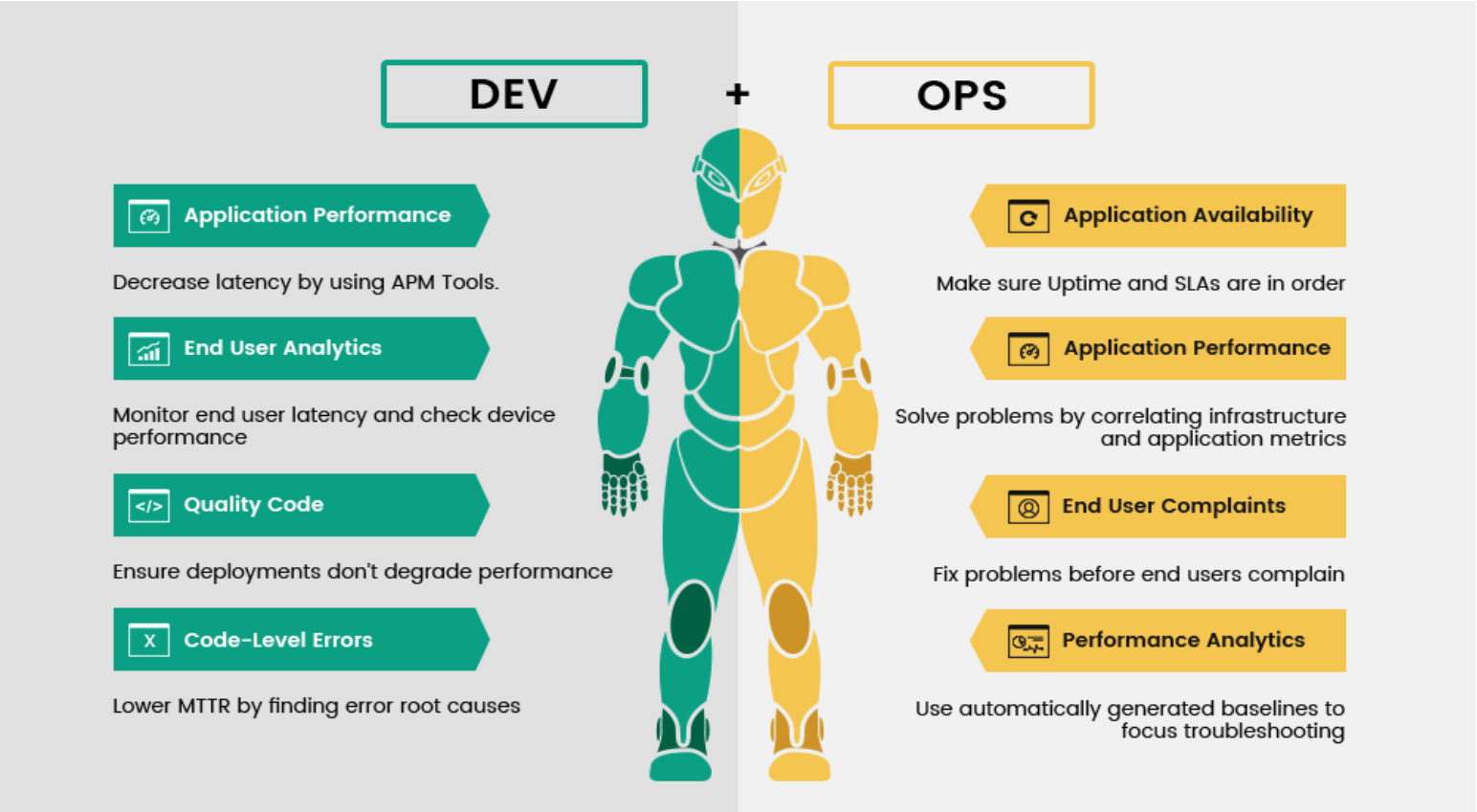
Agile & DevOps – How They Inter-Relate?

- Some of the principles of Agile manifesto that lead to a DevOps environment
  - Our highest priority is to satisfy the customer through early and continuous delivery of valuable software
  - Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale
  - Business people and developers must work together daily throughout the project
- DevOps - talks about collaboration between all teams including business
- DevOps - focuses on automation to complete the many tasks and steps that go into making a deliverable potentially shippable increment
- Using tools to replace the non-human steps like build creation, environment setup, trigger test cases, automated deployments multiple times in a day

Project Stakeholders Issues



Project Stakeholders Responsibilities



What DevOps Is Not?

A title

A team

A tool

Only culture

Only automation

NoOps

What is DevOps?

DevOps is a software engineering culture and practice that aims at unifying software Development and Software Operations.

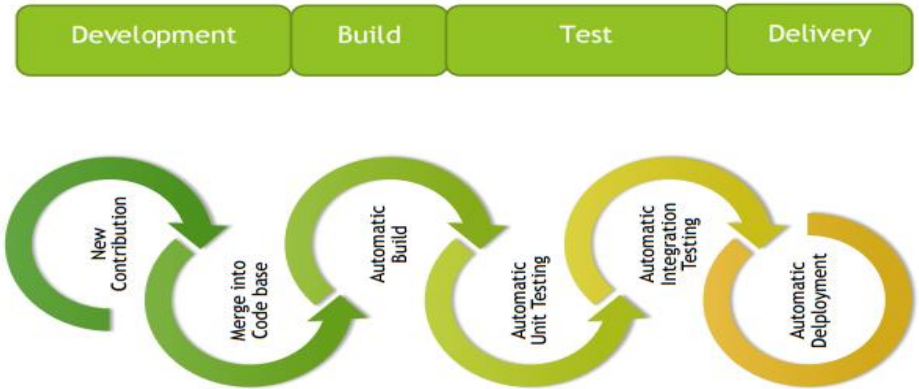
DevOps aims at shorten the development cycles, and increase deployment frequency, and have more dependable releases.

DevOps is predominantly about culture and tools are secondary.

What is DevOps?

DevOps culture is all about collaboration between developers and other parties involved in end to end software development life cycle.

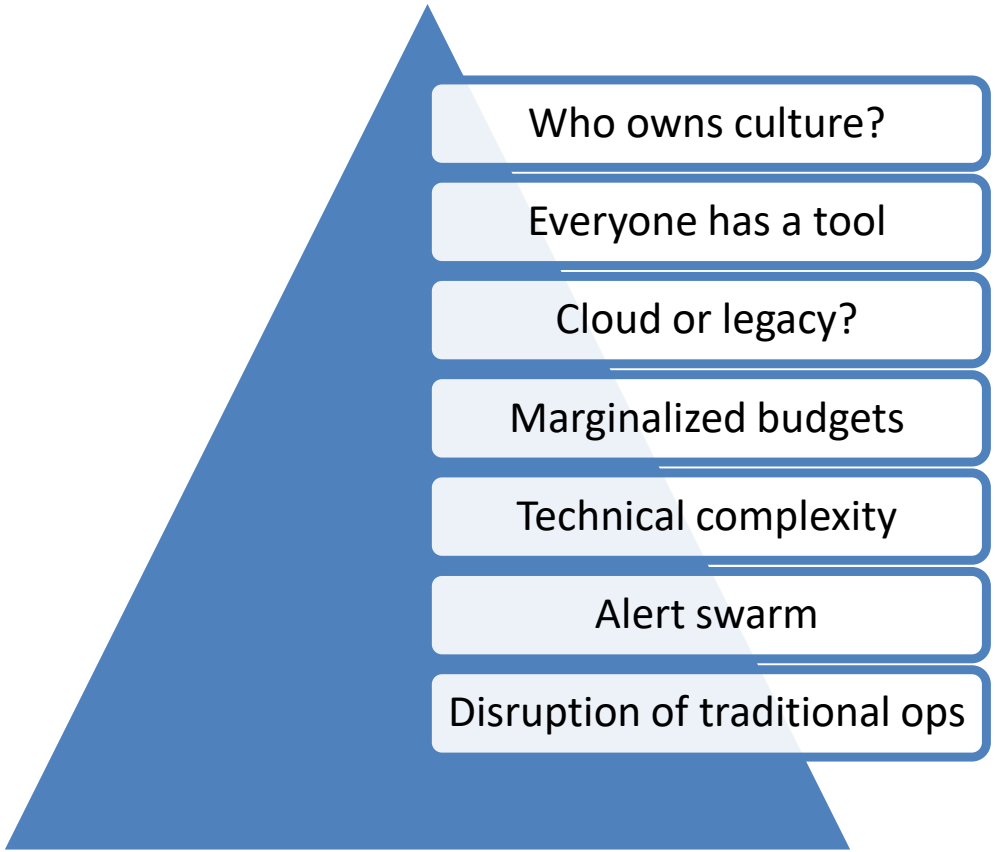
Since one of the goal of DevOps is to have frequent releases - Its emphasis is on automating processes required to release and change software



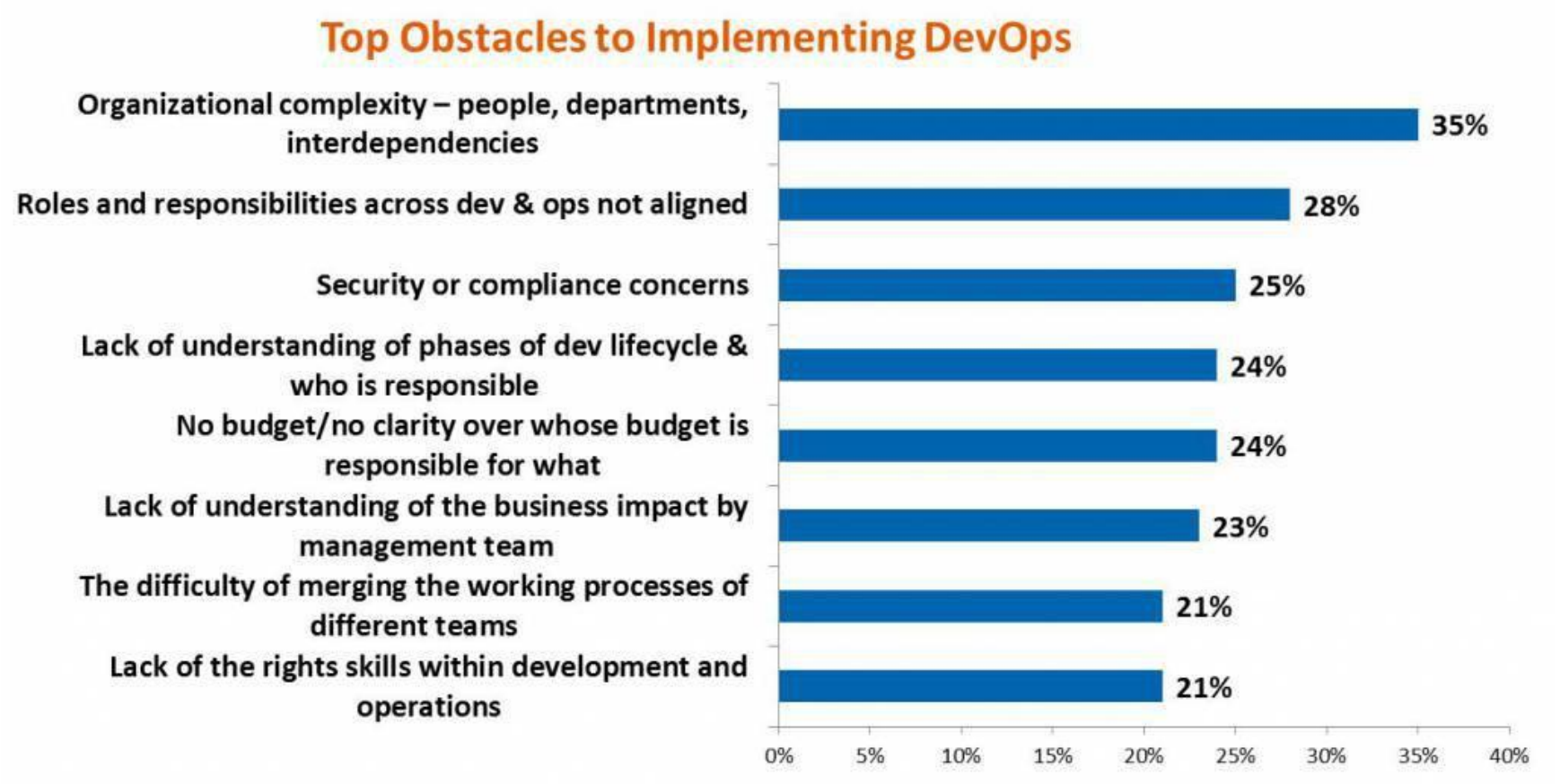
What is DevOps?

- Set of principles and practices
- Helps to deploy software better, faster
- Advocates automation and monitoring
- Recognizes interdependence of development and operations
- Using shared set of tools
- Culture
- Automation
- Measurement
- Sharing

Challenges for DevOps?



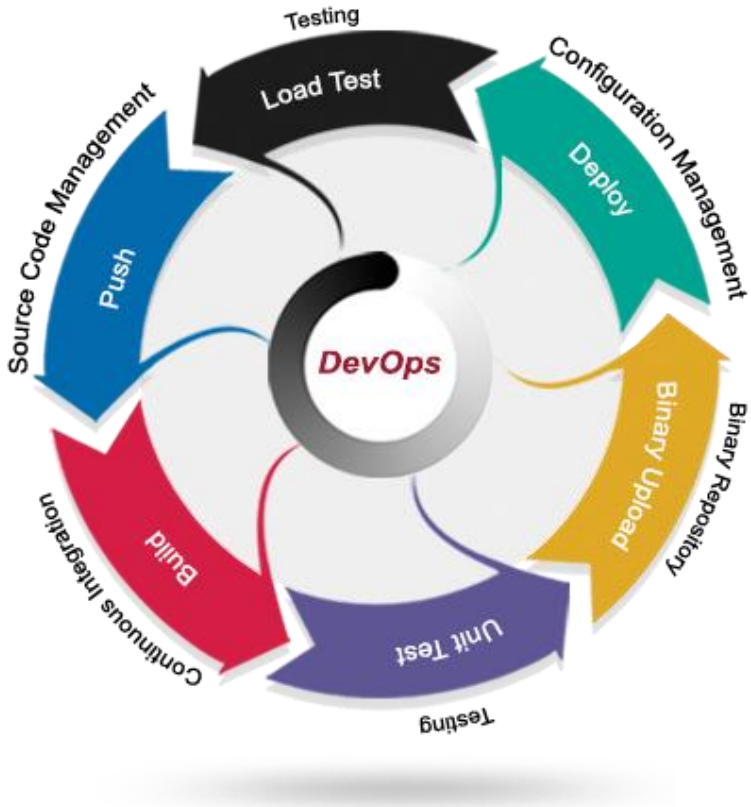
Challenges for DevOps?



Software Development Life Cycle

DEVOPS
LIFE CYCLE

- ✓ Push Code
- ✓ Fetch Changes
- ✓ Run Unit Tests
- ✓ Build Artifacts
- ✓ Store Artifacts
- ✓ Provision environment
- ✓ Deploy Your Build
- ✓ Run Load & Functional Tests
- ✓ Dev -> QA -> Staging -> Production



DevOps - Spanning across entire delivery pipeline

Continuous Integration | Continuous Delivery

DevOps Key Behviors

- Strive for clarity, challenge uncertainties in requirements
- Challenge the validity of assumptions
- Seek simplicity and reliability; do not strive to produce the optimal design
- Seek the least risky design that meets requirements
- Avoid adding extra code or features beyond the bare minimum
- Fix the root cause of each issue instead of patching to fix the symptoms

DevOps Goals

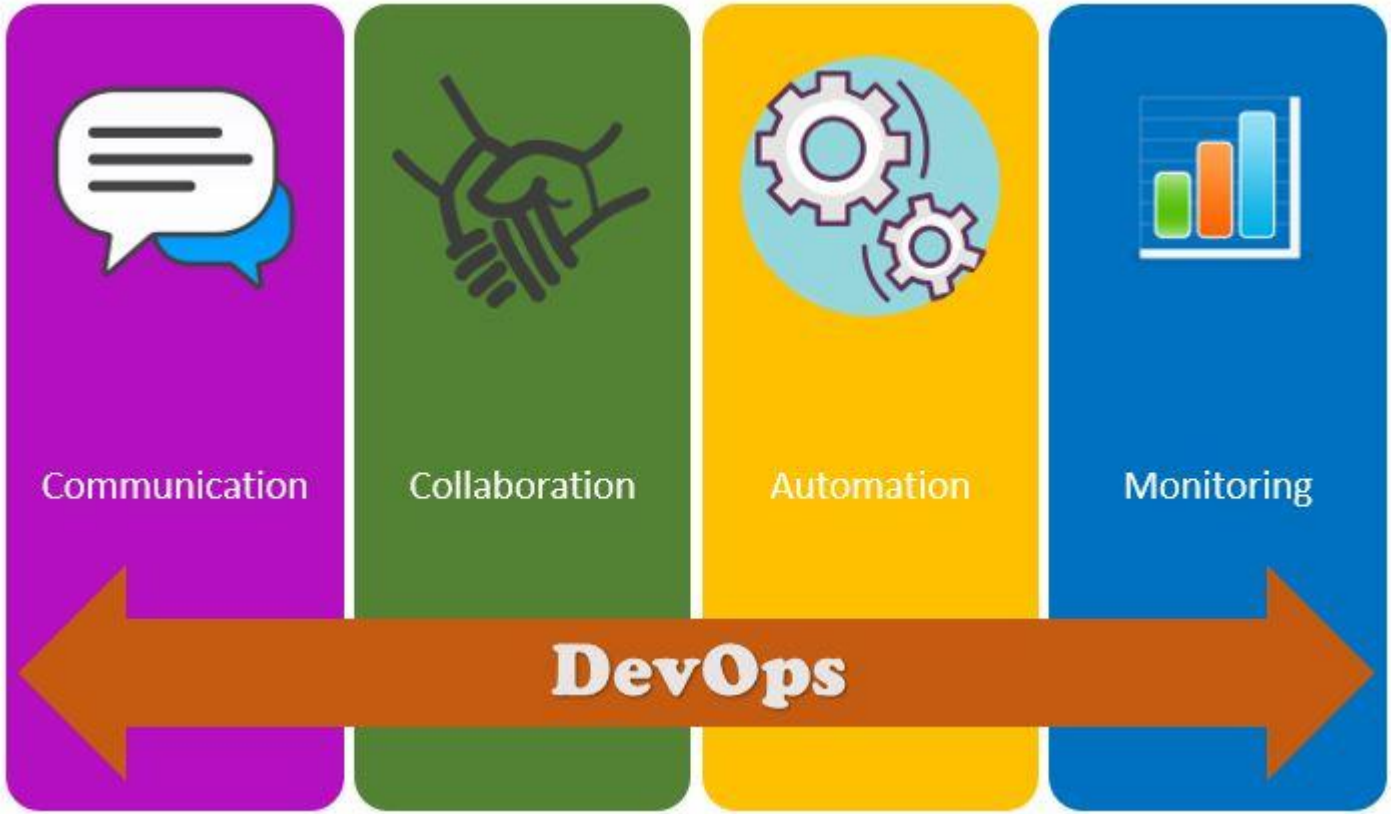
- Increased frequency of software deployment
- Increased collaboration
- Reduction in cost of software development
- Reduce testing time
- Leverage standardized integrated software tools
- Improve performance of individuals and teams
- Increased agility
- Reduced outages

Top 5 Predictors of IT Performance

- Peer-reviewed change approval process
- Version control for all production artifacts
- Win-win relationship between dev and ops
- Proactive monitoring
- High trust organizational culture

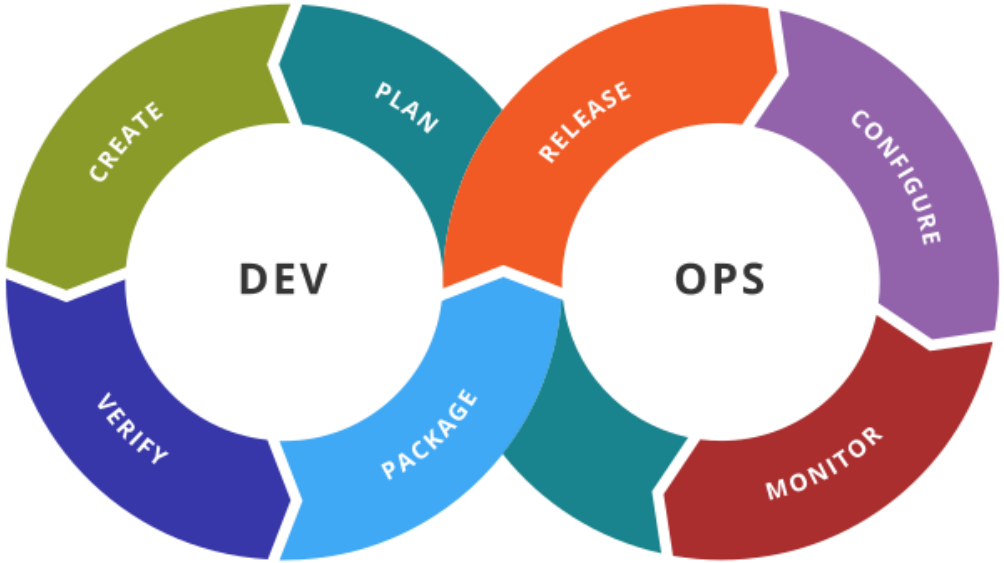
**Source: State of DevOps Report – Puppet Labs*

4 Pillars of DevOps



DevOps Toolchain

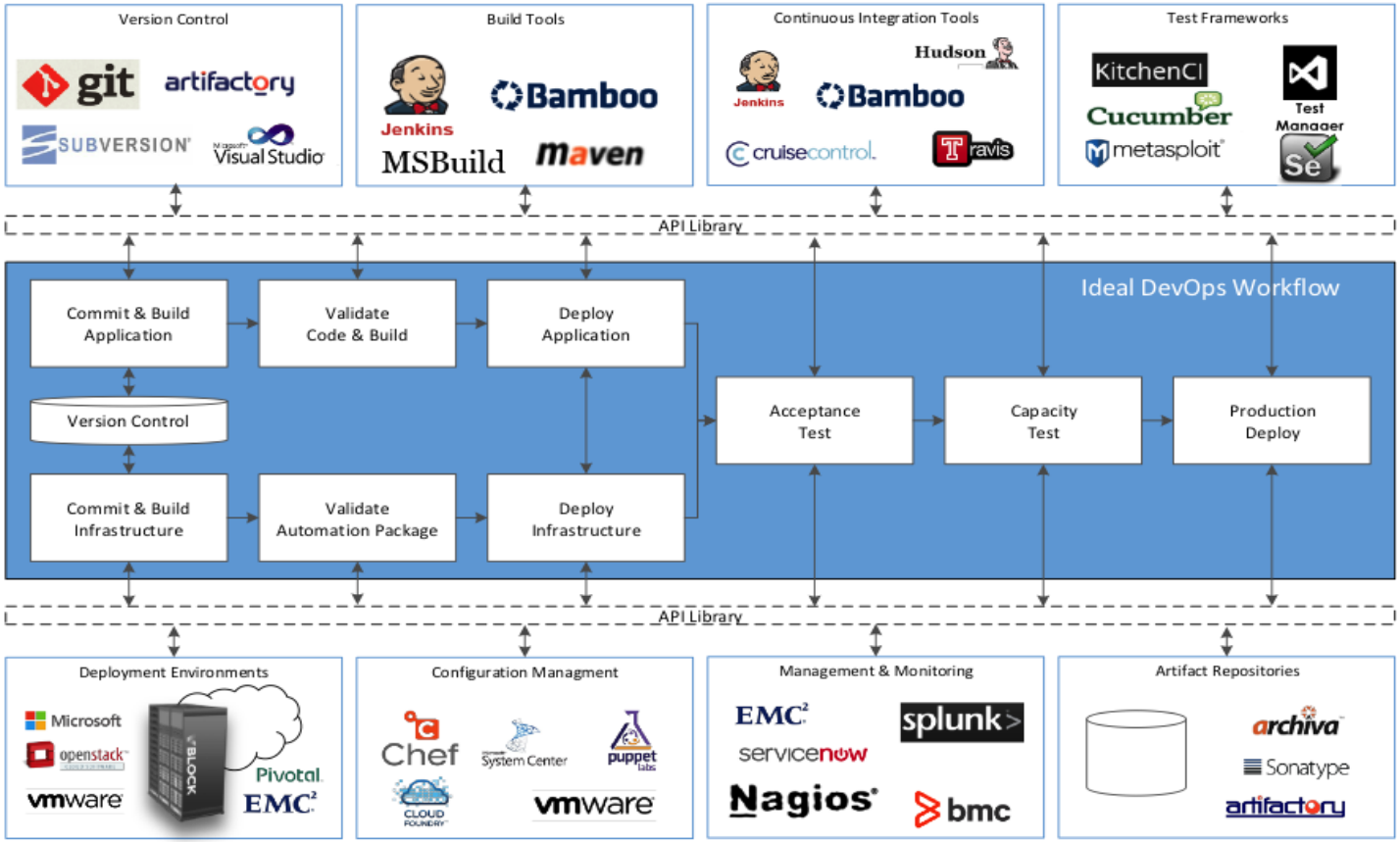
A **DevOps toolchain** is a set or combination of tools that aid in the delivery, development, and management of applications throughout the software development lifecycle, as coordinated by an organisation that uses DevOps practices.



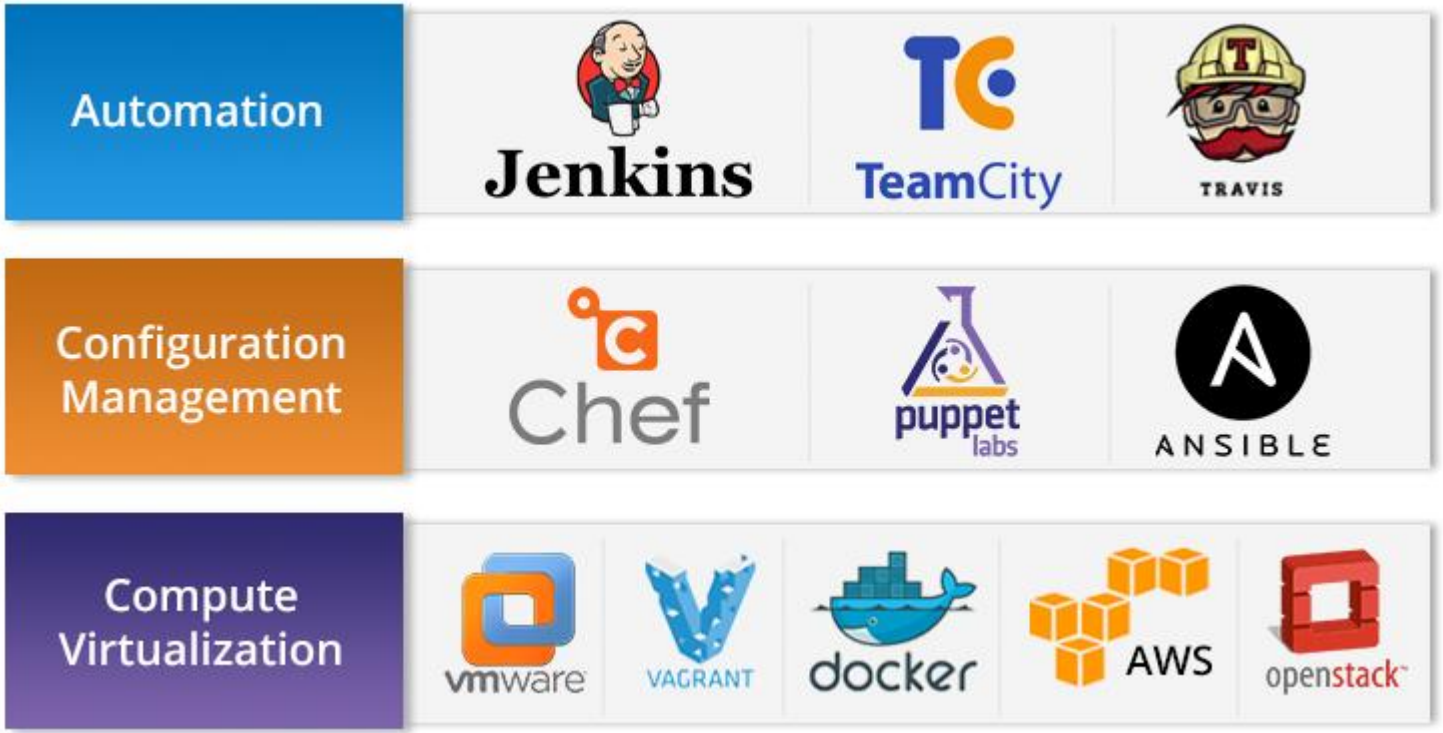
Generally, DevOps tools fit into one or more activities, which supports specific DevOps initiatives:

- ➔ Plan/Requirement
- ➔ Create/Development
- ➔ Verify/Test
- ➔ Package/Artifact
- ➔ Release
- ➔ Configure, and Monitor.

DevOps Tool Chain



DevOps Tools For Training

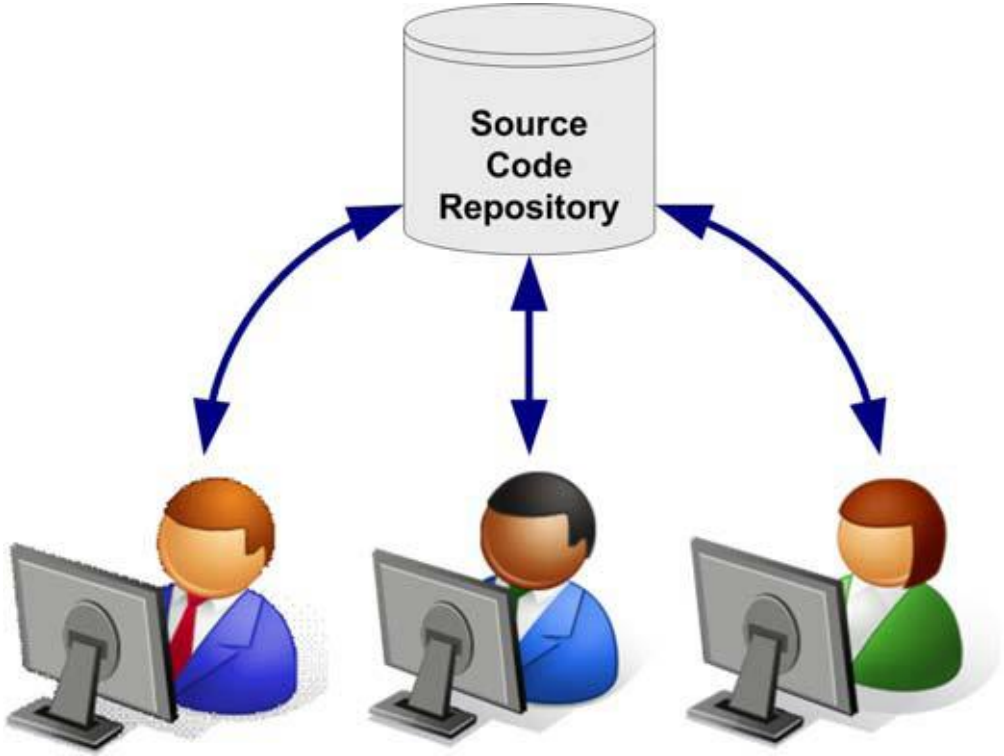


Source Version Control

What is Version Control?

Version control is a system that records changes to a file or set of files over time so that you can recall specific versions later.

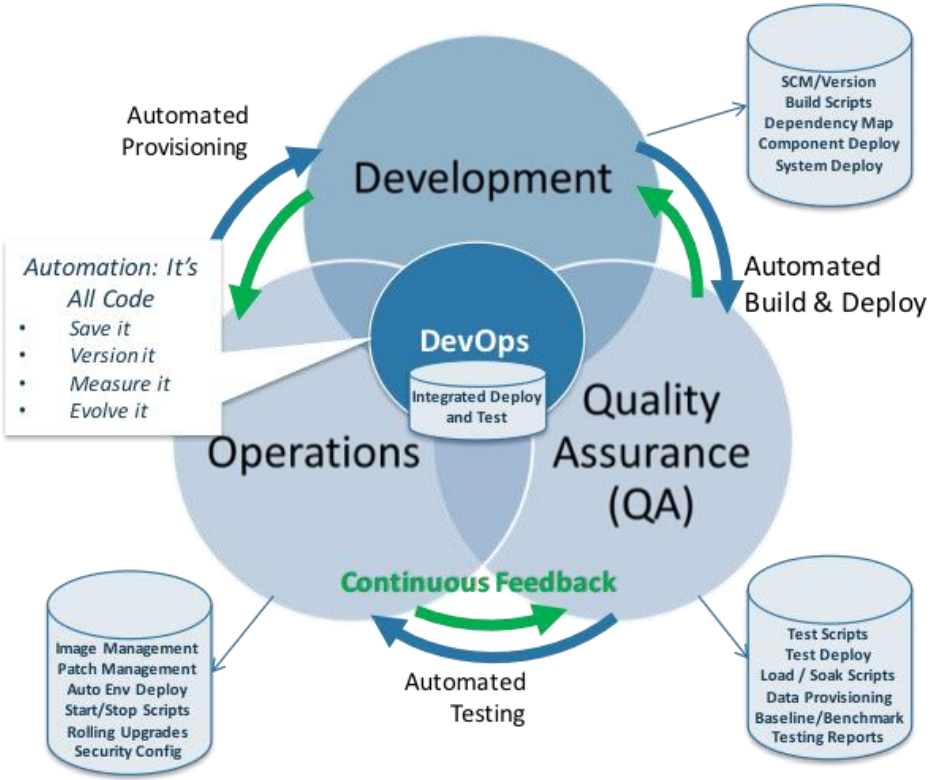
It help with tracking changes made to each and every part of your source code.



Version Control in DevOps

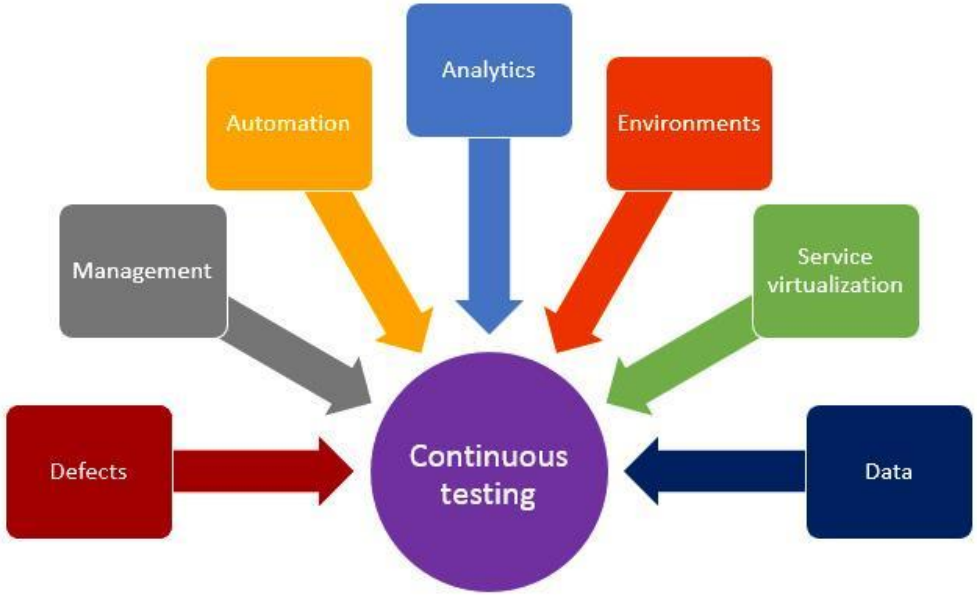
Versioning becomes critical to business success because corporate value is often contained in the software assets created by the company.

Long long ago, there used to be “The Build Team”. Build Team lived in a world of Cron, Bash, Perl, and nightly builds. It’s hard to believe in today’s world that 24-hour turnaround was once considered standard practice.



Continuous Testing (CT) and DevOps

- “Continuous testing enables a project team to execute tests when needed, not when possible.”
- Informed customers demand seamless user experience.
- Applications need an extraordinary connectivity and dependence between processes, systems, and infrastructure.
- Testing various complex applications, products, and services can pose a big challenge as testing needs to make sure that high-quality software is delivered at fast-track speed.

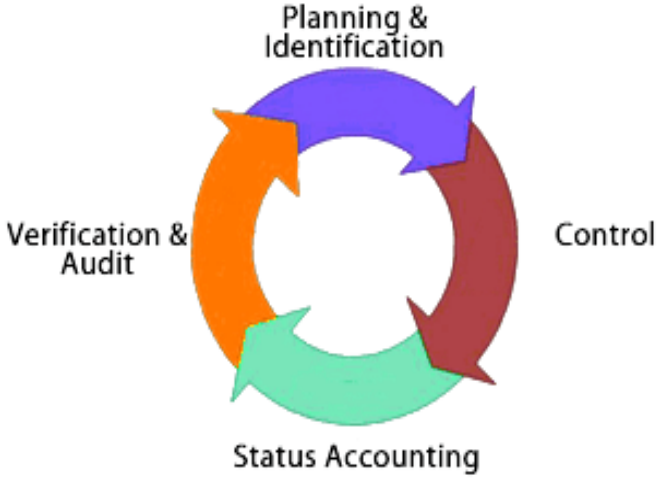


Testable User Stories

- Agile development often specify requirements in terms of user stories
- Effective user stories need to be testable
- Features of effective user stories are **SIT** - **S**mall **I**ndependent **T**estable :
  - It needs to describe an action which has value to a specific user
  - It needs to target a specific user or role
  - It needs to have clearly stated acceptance criteria which can easily be tested
  - It needs to be small enough to implement in a few days
  - It needs to be short and precise

Configuration Management

- Configuration management is the process of standardizing resource configurations and enforcing their state across IT infrastructure in an automated yet agile manner.
- Configuration management represents the one true source of the configuration items. A configuration item is anything that can be configured and that is absolutely necessary for the success of your project.
- For example, source codes, property files, binaries, servers, and tools can all be configuration items for a software firm



Configuration Management Process

Configuration Management in DevOps

- DevOps starts and ends with configuration management
- Configuration management in DevOps is made up of:
  - **Source Code Repository** — Used primarily during the development phase.
  - **Artifact Repository** — Used during the development and operations phases.
  - **Configuration Management Database** — Used during the development and operations phases for following activities
    - Change management
    - Provisioning environments
    - Incident management
    - Infrastructure as Code

Continuous Integration

- Requires developers to integrate code into a shared repository
- Each check-in is verified by an automated build
- Allowing teams to detect and solve problems quickly
- Integrate at least daily
- Spend less time debugging and more time adding features

"Continuous Integration doesn't get rid of bugs, but it does make them dramatically easier to find and remove."

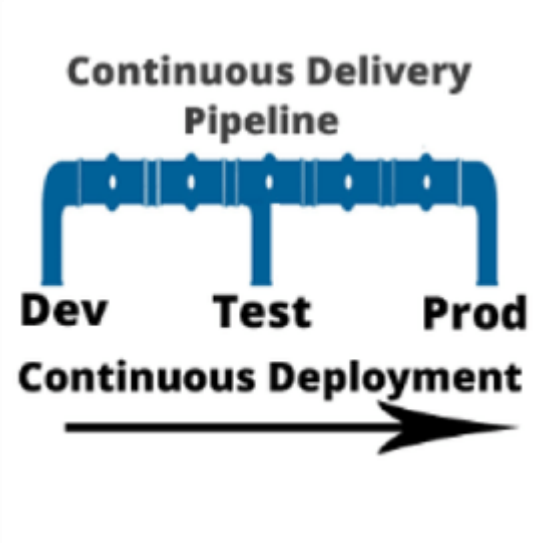
Continuous Delivery

- Extends continuous integration
- Provides fast, automated feedback on the production readiness of system
- It can help large organizations become as lean, agile and innovative as start-ups
- Reliable, low risk releases
- Test, support, development and operations work together as one delivery team to automate and streamline the build-test-release process
- Enables push button deployments on demand

- Continuous delivery does not mean that you are deploying every day. It means that you COULD release when necessary

Continuous Deployment

- Set of practices that enable every change that passes automated tests to be automatically deployed to production
- Removes the manual step in continuous delivery pipeline
- Results in multiple deployments per day



DevOps and Automation

- Automation is an essential element
- Automation enables agility, consistency, speed and reliability
- Treating infrastructure as code
- Repeatable and reliable deployment processes
- Development and automation testing performed against production-like systems
- On-demand creation of development, testing etc. environments
- Proactive monitoring
- Shared access to automated testing, deployment and monitoring tools
- Streamlines software delivery and prepares Ops for the long run

This concludes Chapter 2 – DevOps Overview

Let us move to Chapter 3 – SCM with GitHub