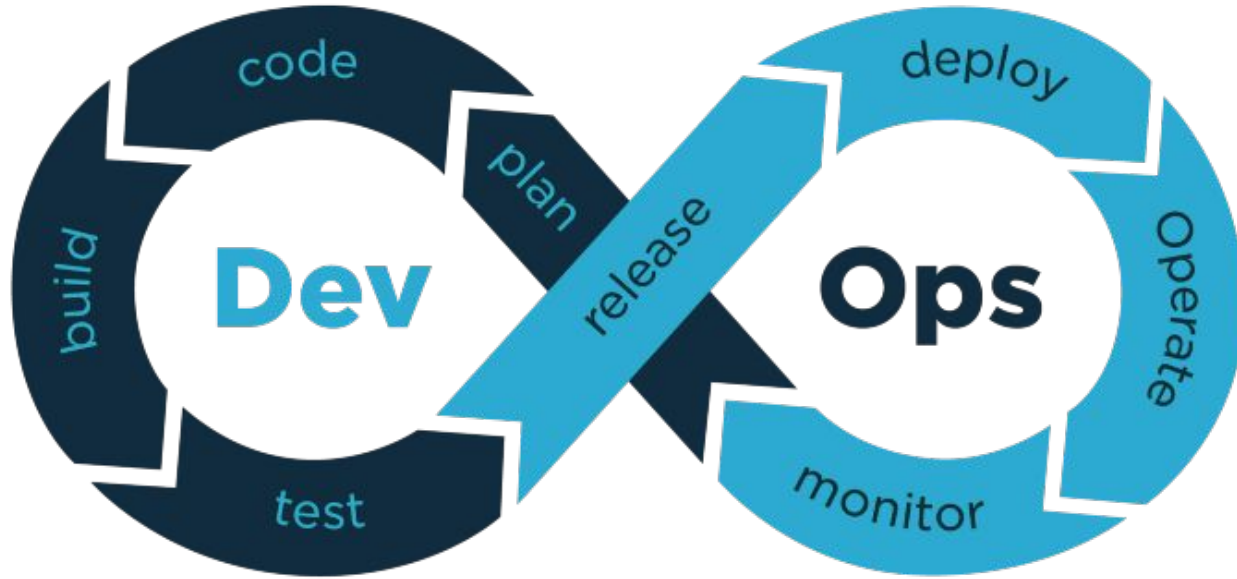


Introduction to DevOps @ Scale

Scaling the DevOps with Automation

TIC 4302 - Information Security Practicum II

Scaling the DevOps



Devops in a scaling environment is defined as **automation driven** environment for the establishment of devops culture and choosing of appropriate tools.

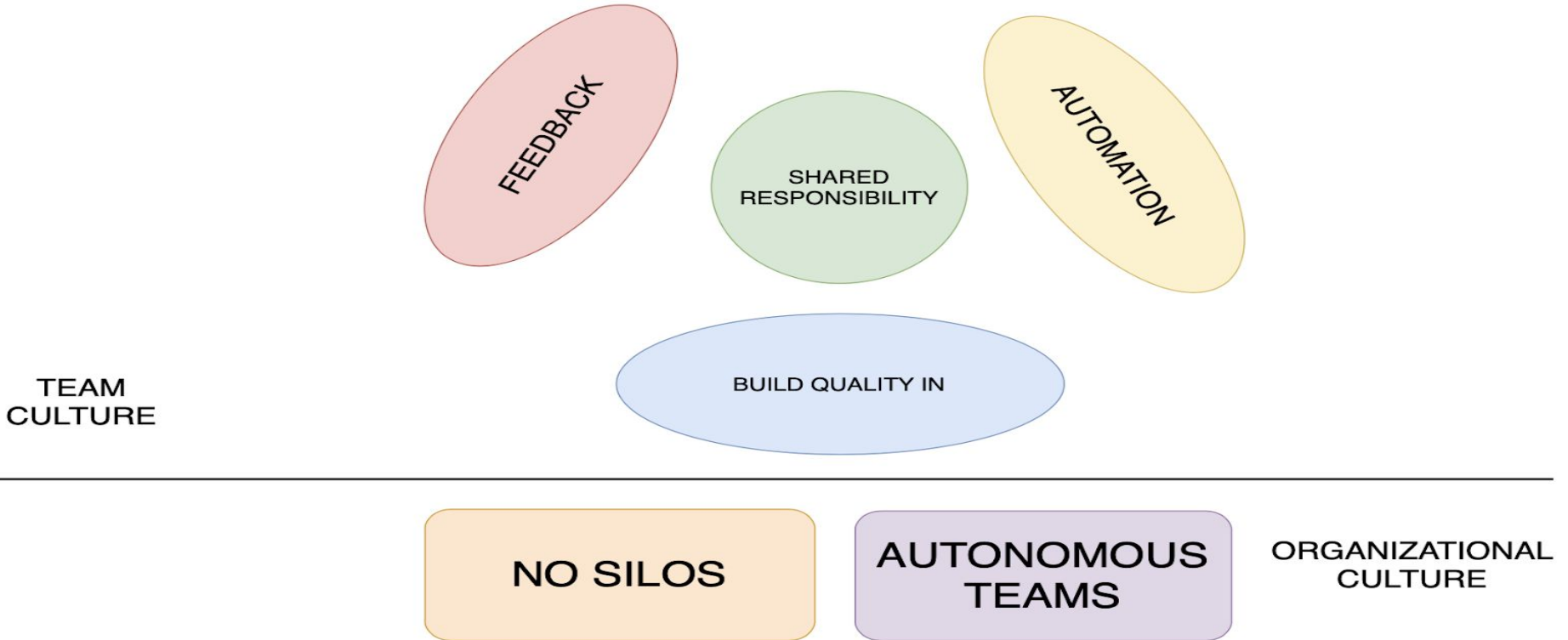
Key factors to Devops in a scaling environment

- Culture
- Tools
- Automation
- Communication and Collaboration

Culture

- Devops culture is shared responsibilities toward communication and collaboration across development, IT/ops and the business.
- DevOps **is not** a problem that you can solve with tools by modernizing the development and operations teams without joining them.
- DevOps culture is characterized by increased collaboration, decreasing silos, shared responsibilities, autonomous teams, improving quality, valuing feedback and **increasing automation**.

Culture



Tools

- DevOps model relies on effective tooling to help teams rapidly and reliably deliver/deploy.
- The tools automate manual tasks, help teams manage complex environments at scale, and keep engineers in control of the high velocity.
- DevOps tools consist of configuration management, test and build systems, application deployment, version control and monitoring tools.
- DevOps tools can quickly spin up research labs and test environments in order to try new ideas and build proof of concepts quicker and more cost effectively.

Tools

PERIODIC TABLE OF DEVOPS TOOLS (v1)

Legend:

- Os: Open Source
- Fr: Free
- Fm: Freemium
- Pd: Paid
- En: Enterprise

Tool Categories and Examples:

- Database:** MySQL, PostgreSQL, MongoDB, Redis, etc.
- CI:** Jenkins, Travis CI, CircleCI, etc.
- Deployment:** Docker, Kubernetes, etc.
- Release Mgmt:** Jenkins, Travis CI, CircleCI, etc.
- Logging:** ELK Stack, etc.
- SCM:** Git, GitHub, etc.
- Repo Mgmt:** Jenkins, Travis CI, CircleCI, etc.
- Config / Provisioning:** Ansible, Puppet, Chef, etc.
- Collaboration:** Jenkins, Travis CI, CircleCI, etc.
- Security:** Jenkins, Travis CI, CircleCI, etc.
- Build:** Maven, Gradle, etc.
- Testing:** JUnit, JMeter, etc.
- Containerization:** Docker, Kubernetes, etc.
- Cloud / IaaS / PaaS:** AWS, Azure, etc.
- BI / Monitoring:** Grafana, Prometheus, etc.

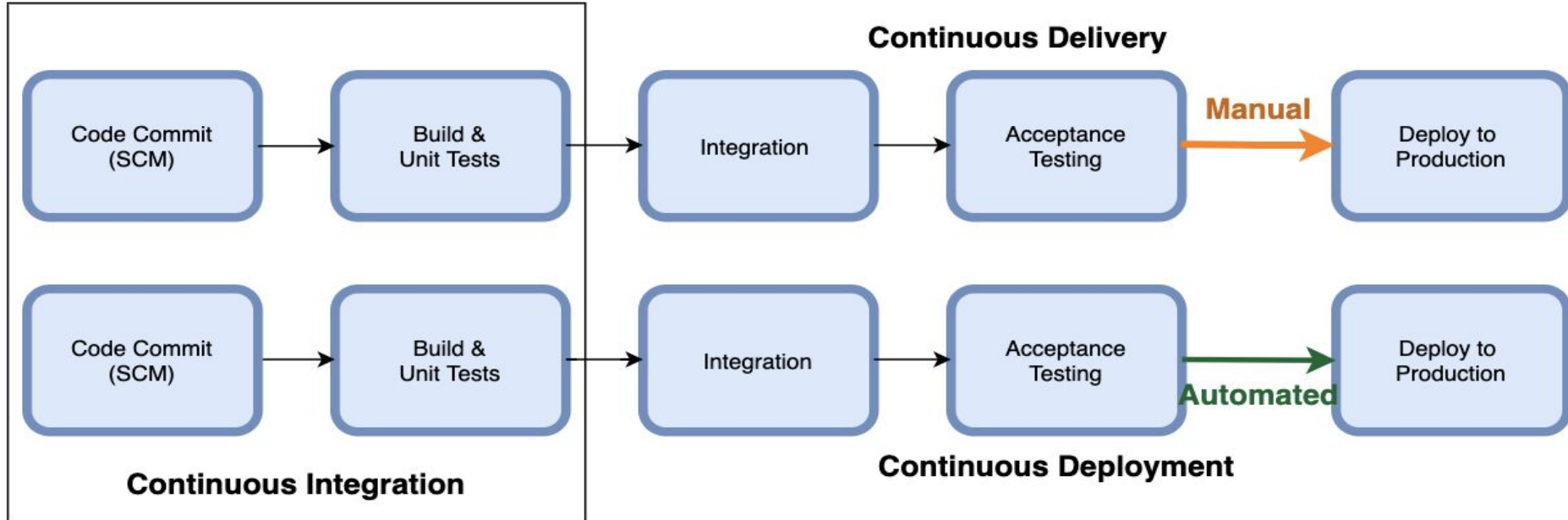
Tool Examples:

- Open Source (Os):** MySQL, PostgreSQL, MongoDB, Redis, Jenkins, Travis CI, CircleCI, Docker, Kubernetes, Ansible, Puppet, Chef, Maven, Gradle, BuildMaster, Codeship, Snap CI, CircleCI, Nexus, Cucumber, Cucumber.js, Qunit, Capistrano, Juju, Rundeck, CFEngine, Packer, Bluemix, Docker, Kubernetes, Ansible, Puppet, Chef, Maven, Gradle, BuildMaster, Codeship, Snap CI, CircleCI, Nexus, Cucumber, Cucumber.js, Qunit, Capistrano, Juju, Rundeck, CFEngine, Packer, Bluemix.
- Free (Fr):** Git, GitHub, Jenkins, Travis CI, CircleCI, Docker, Kubernetes, Ansible, Puppet, Chef, Maven, Gradle, BuildMaster, Codeship, Snap CI, CircleCI, Nexus, Cucumber, Cucumber.js, Qunit, Capistrano, Juju, Rundeck, CFEngine, Packer, Bluemix.
- Freemium (Fm):** MySQL, PostgreSQL, MongoDB, Redis, Jenkins, Travis CI, CircleCI, Docker, Kubernetes, Ansible, Puppet, Chef, Maven, Gradle, BuildMaster, Codeship, Snap CI, CircleCI, Nexus, Cucumber, Cucumber.js, Qunit, Capistrano, Juju, Rundeck, CFEngine, Packer, Bluemix.
- Paid (Pd):** Jenkins, Travis CI, CircleCI, Docker, Kubernetes, Ansible, Puppet, Chef, Maven, Gradle, BuildMaster, Codeship, Snap CI, CircleCI, Nexus, Cucumber, Cucumber.js, Qunit, Capistrano, Juju, Rundeck, CFEngine, Packer, Bluemix.
- Enterprise (En):** Jenkins, Travis CI, CircleCI, Docker, Kubernetes, Ansible, Puppet, Chef, Maven, Gradle, BuildMaster, Codeship, Snap CI, CircleCI, Nexus, Cucumber, Cucumber.js, Qunit, Capistrano, Juju, Rundeck, CFEngine, Packer, Bluemix.

Automation

- Automation is the backbone of Devops “agile” strategy to help reducing the manual hand-offs between development and operations, in order to speed up the processes to release the software continuously.
- It is requires a substantial level of understanding processes, technologies, and complexities of software deployments to see a positive impact in the reliability, testability, security, and other quality attributes of their software.
- Various automation tools for each stage of the SDLC such as version control, build and test, configuration management, deployment and monitoring.
- **Automated testing is recommended or it could become nearly impossible to have continuous delivery.**

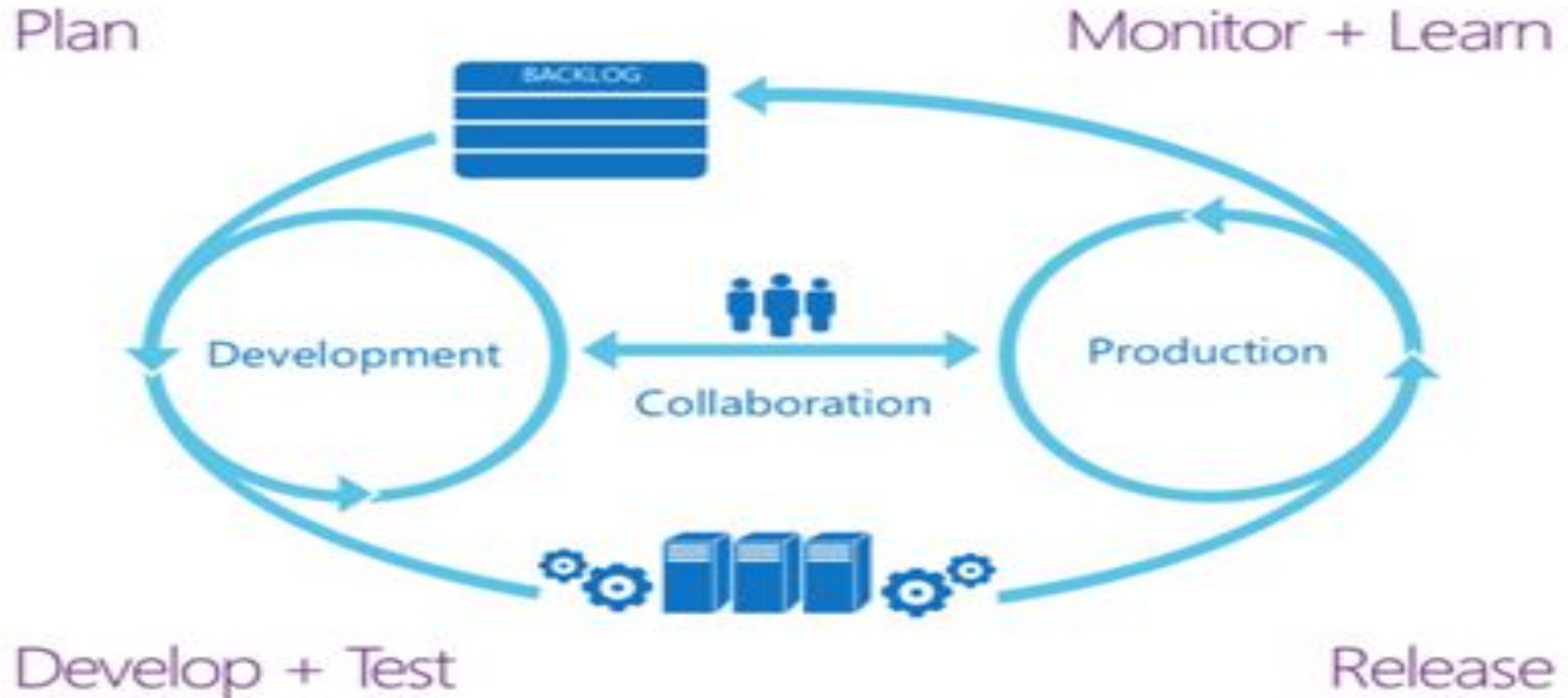
Automation



Communication and Collaboration

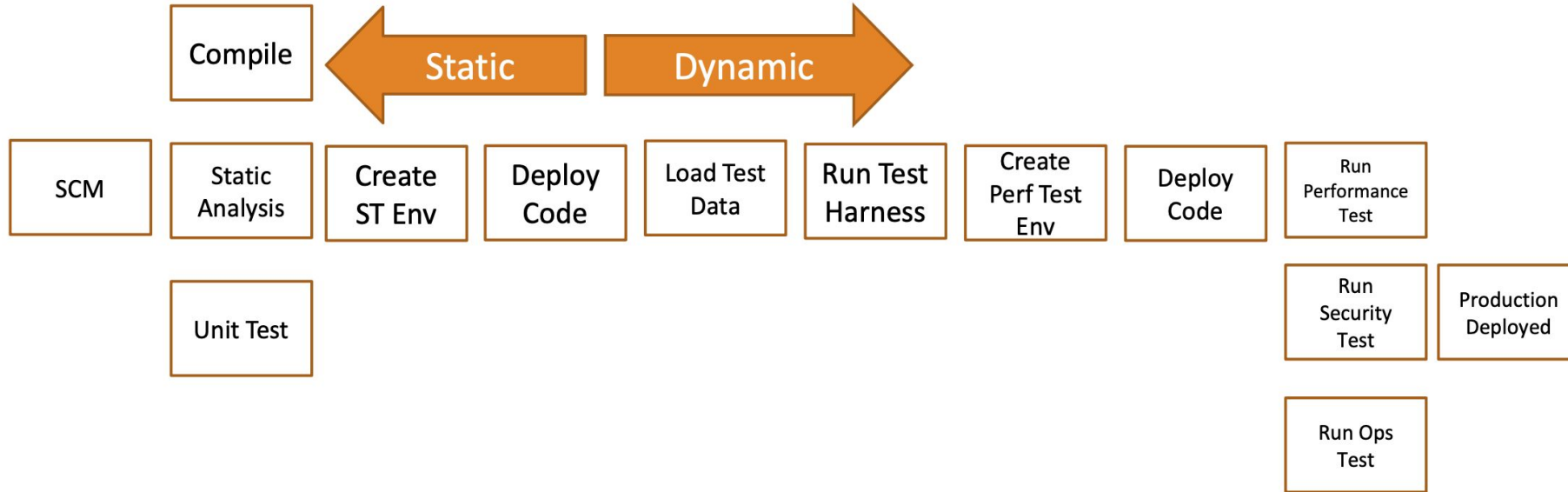
- Strong communication between team members is vital for going forward and driving positive change in how people work together.
- Working together as a team improves interaction and communication to help in working more coordinated way with prompt action for tasks/issues and reduced turnaround time.
- The primary goal of collaborative and close communication is to avoid conflicts and escalations to have proper handover of the tasks at every stage of the SDLC which also simplifies overall delivery process.

Communication and Collaboration

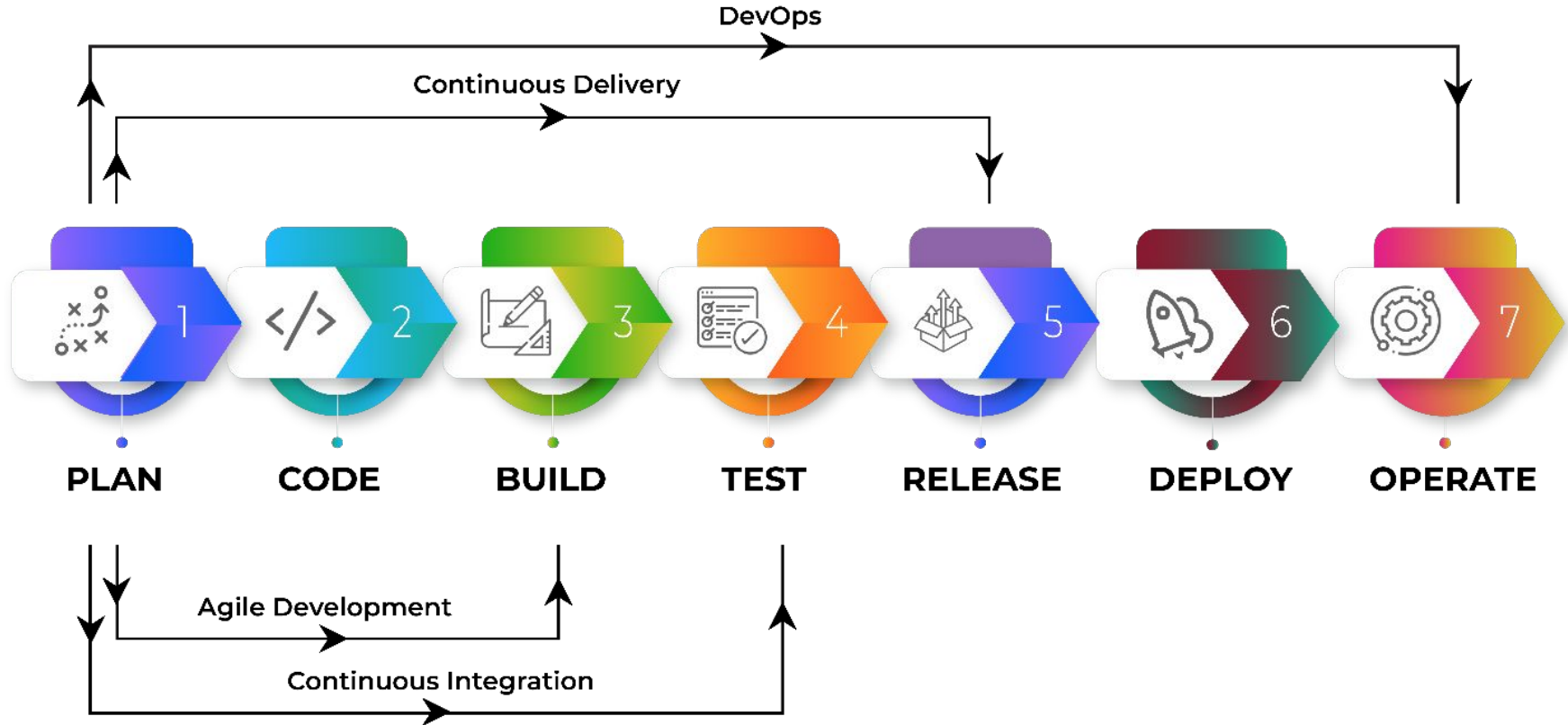


DevOps Automation

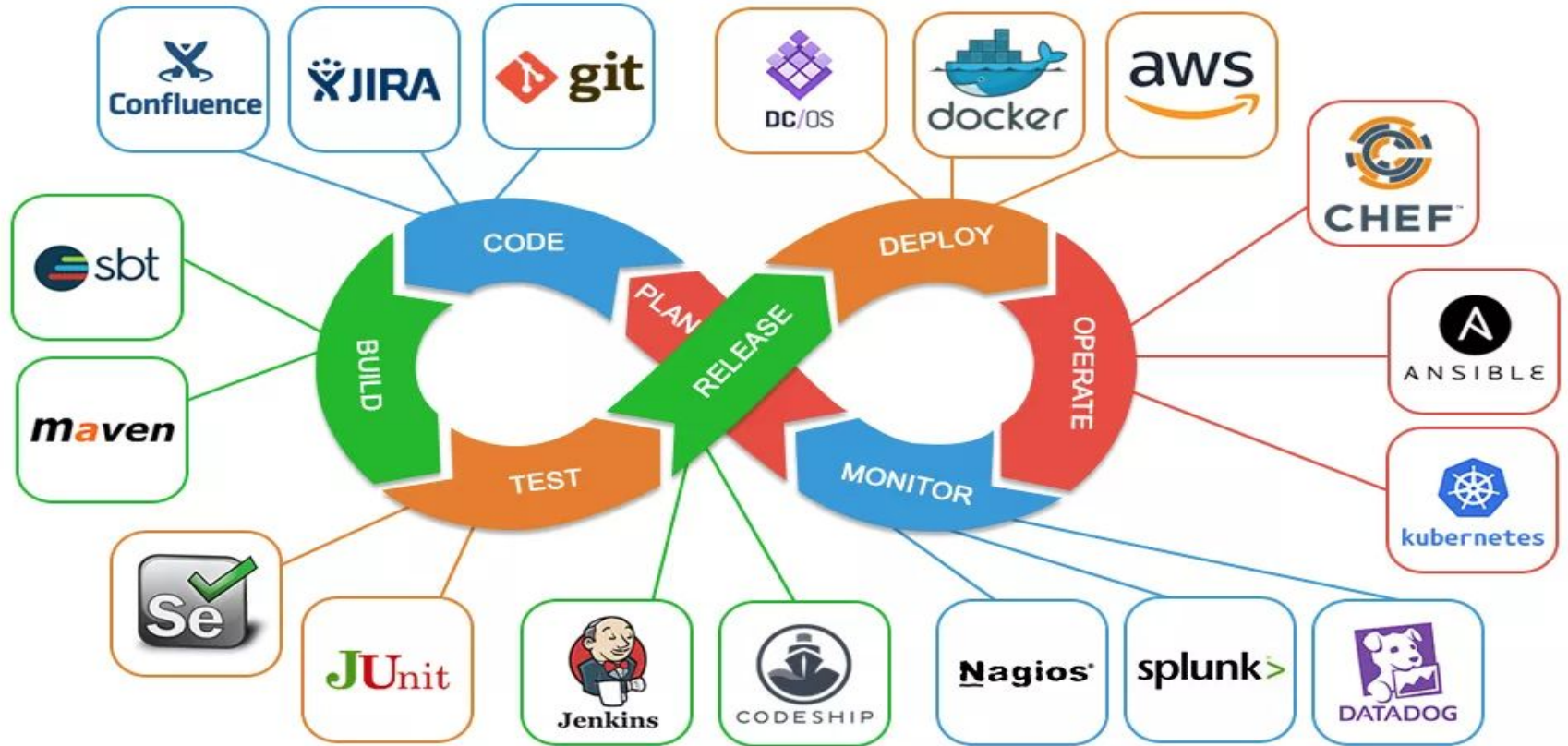
Automation Scope in SDLC



CI/CD as part of DevOps Automation



DevOps Automation Toolchains



SCM Tools

- SCM is a discipline comprising the tools and techniques to manage change to software assets
- It used to ensure integration of code changes
- It used to maintain revision history, what was update, who made updates
- It has ability to merge or undue changes
- Example: **Git, Gerrit, Subversion**

Build and Release Tools

- In DevOps, a software builds and deployments need to run more frequently and in less time
- Continuous integration workflow
- Continuous delivery enabling the ability to deploy to production environments and production like environments at any time
- Example: **Jenkins**, **Bamboo**

Q&A Testing (1)

- Post-Commit Automated Build
- Run automated Unit Tests
- Run automated Static Code Analysis
- Example: **SonarQube, CSS Lint, Food Critic, Cucumber, FitNesse Junit, TestNG**

Q&A Testing (2)

- Pre-Deployment Testing
- Functional Suite
- Test suites against an environment
- UI or API driven – Load Test
- Example: **Selenium, Robot Sauce Labs, HP (functional) Watir, Jmeter Concordian, Casper.js, Watir, Autoit**

Infrastructure Configuration Management

- Manage the configuration of the infrastructure
- Infrastructure is under CM and is automated
- Infrastructure is immutable
- Example : **Chef, Puppet, SaltStack, Ansible, and Kubernetes**