DevOps – Build and Release Engineer Training

By

Rajesh Kumar

DevOps Architect & Blogger

https://twitter.com/**RajeshKumarIN**

Agenda

- What is SCM?
- Major activities of SCM?
- Day to Day activities?
- Roles in SCM?
- Tools & Technology
- Best Practices
  - Continuous Integration
  - Continuous Delivery
  - DevOps

Why SCM?

- Increased efficiencies, stability and control by improving visibility and tracking.
- Cost reduction by having detailed knowledge of all the elements of the configuration which allows for unnecessary duplication to be avoided.
- Enhanced system reliability through more rapid detection and correction of improper configurations that could negatively impact performance.
- The ability to define and enforce formal policies and procedures that govern asset identification, status monitoring, and auditing.

Why SCM? Cont..

- Greater agility and faster problem resolution, thus giving better quality of service.
- Decreased risk and greater levels of security.
- More efficient change management by knowing what the prior structure is in order to design changes that do not produce new incompatibilities and/or problems.
- Faster restoration of service. If you know the desired state of the configuration and how they are interrelated, recovering the live configuration will be much easier and quicker.

SCM involves in each phase of SDLC Cycles

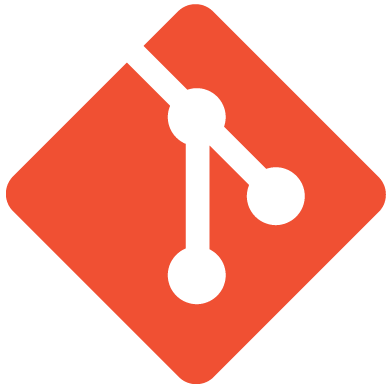
- **Requirement Analysis**
- **Feasibility Study**
- **Design**
- **Development**
- **QA**
- **Release | Delivery**

Clients of SCM

- Developers
- QA
- Mgmt
- Clients

Activities of SCM

Source Code Management



git

SUBVERSION

Why Source Code Management?

Build Management

Make

Nmake

maven

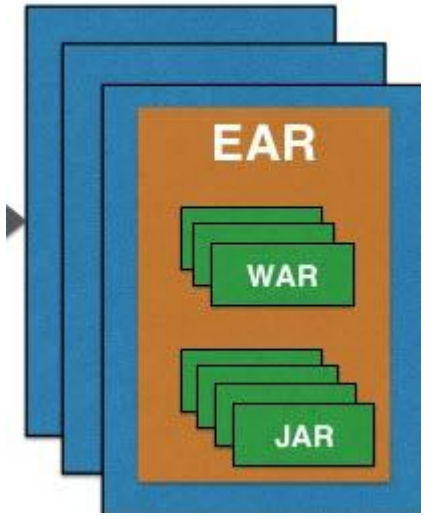
MSBuild

Why build tools?

Packaging

InstallShield

FLEXERA SOFTWARE®
InstallAnywhere®



DEBIAN / UBUNTU
PACKAGE MANAGEMENT

nuget

Why Packaging

Artifact Management (Repository)

Nexus

artifactory

archiva™

nuget

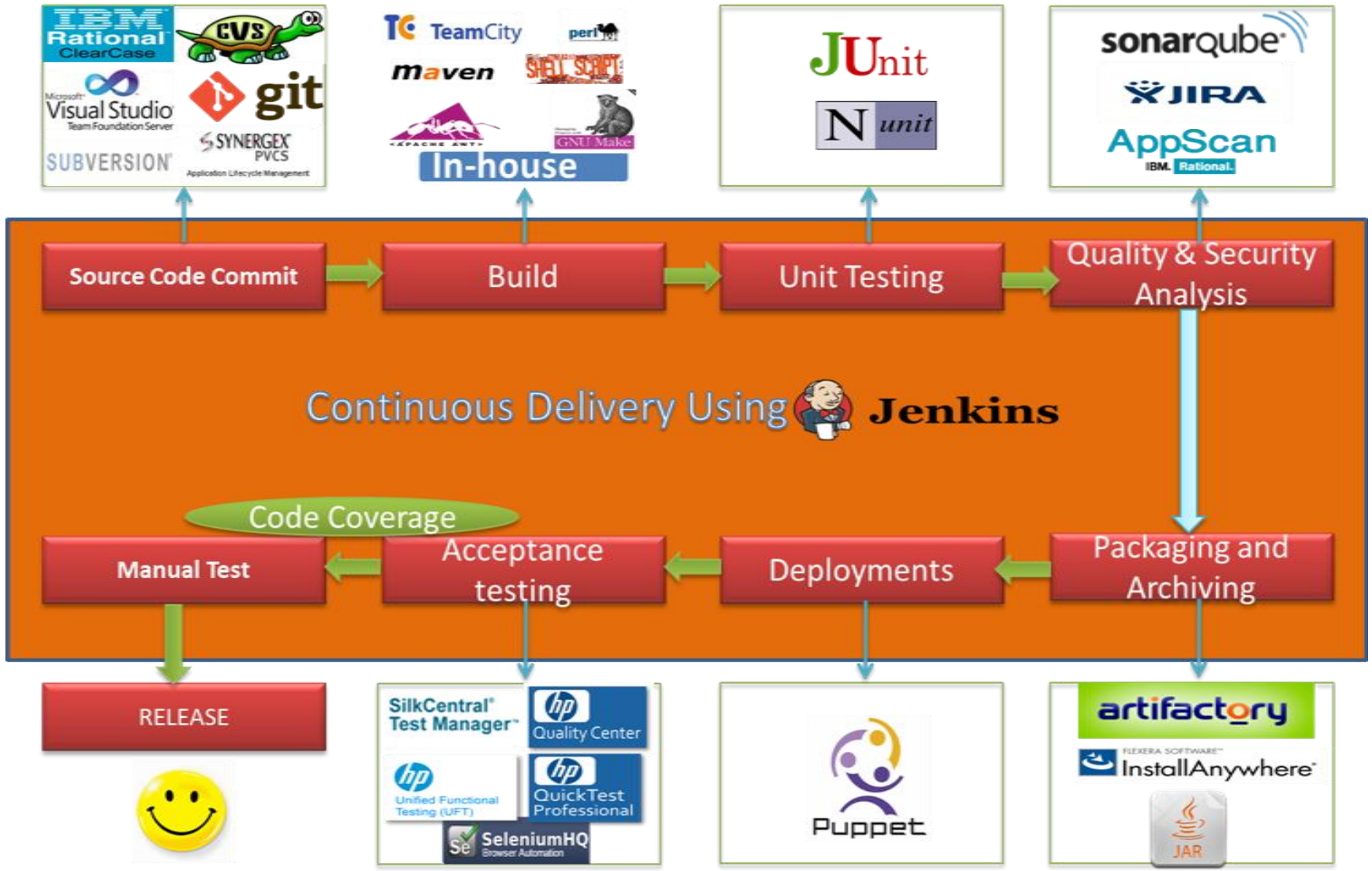
Why Artifact Management?

Deployment & Configuration Management

Why Deployment & Configuration Management?

Release Management

What do we want?


IBM Rational ClearCase
CVS
Visual Studio Team Foundation Server
git
SYNERGEX PVCS
SUBVERSION
TeamCity
perl
Maven
SHELL SCRIPT
GNU Make
In-house
JUnit
N unit
sonarqube
JIRA
AppScan
IBM Rational.
Source Code Commit
Build
Unit Testing
Quality & Security Analysis
Continuous Delivery Using Jenkins
Code Coverage
Manual Test
Acceptance testing
Deployments
Packaging and Archiving
RELEASE
SilkCentral Test Manager
Quality Center
Unified Functional Testing (UFT)
QuickTest Professional
SeleniumHQ
Puppet
artifactory
InstallAnywhere
JAR

Do we have one?

Don’t we have one tool which can integrate to all these tools?

Framework? We call it CI tools

Jenkins

TeamCity

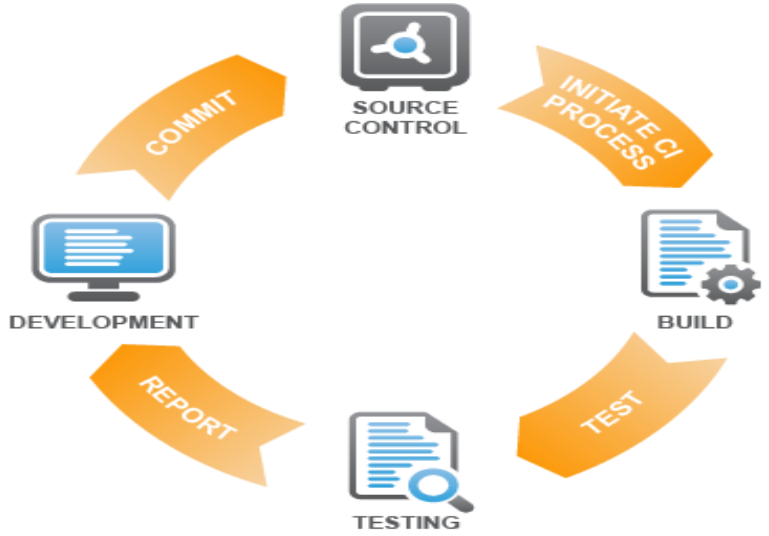
So what is CI?

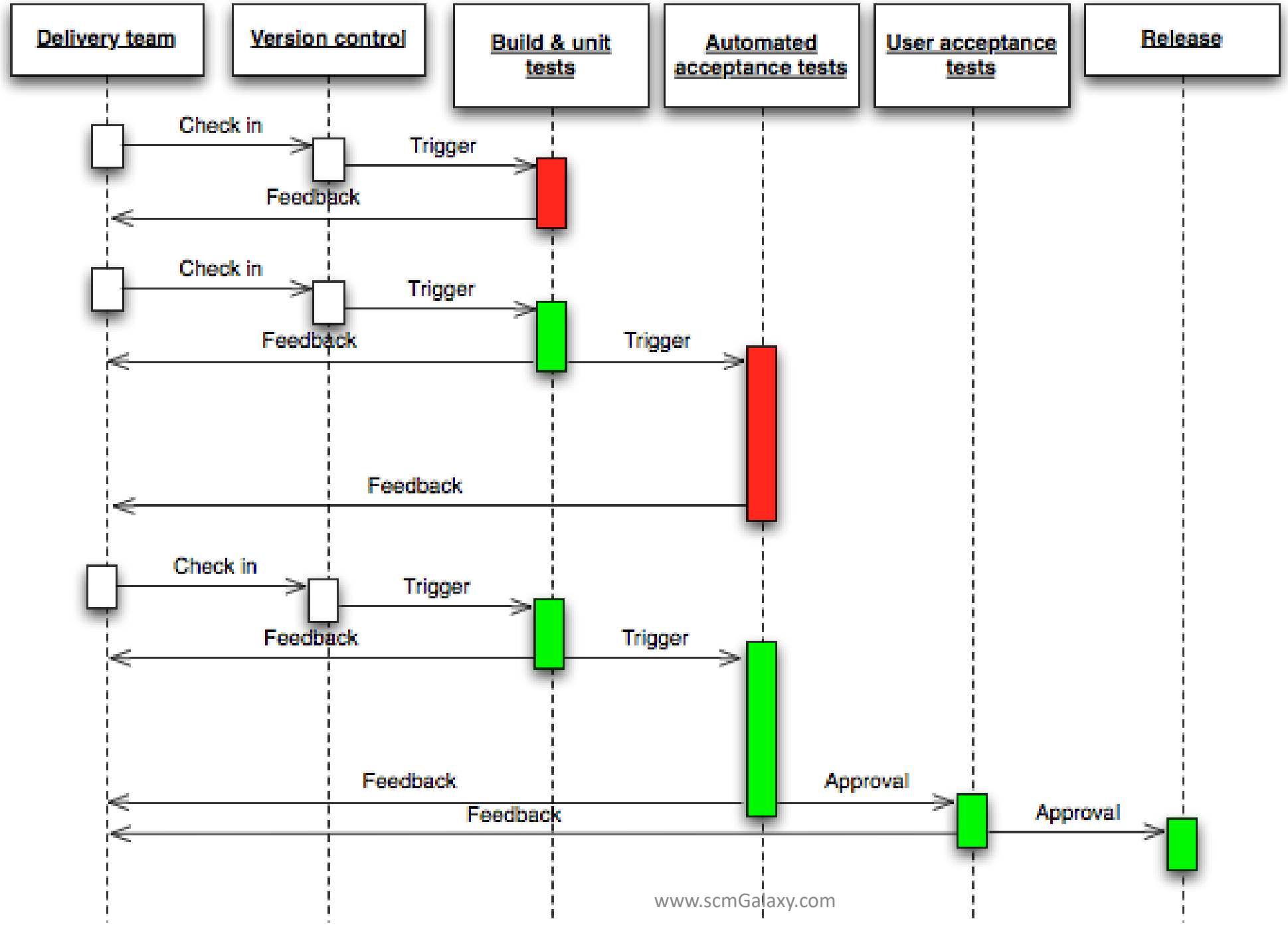
What is Continuous Integration (CI)?

Immediate Feedback

Continuous Integration is a software development practice where members of a team integrate their work frequently, usually each person integrates at least daily - leading to multiple integrations per day. Each integration is verified by an automated build (including test) to detect integration errors as quickly as possible.

- Martin Fowler





www.scmGalaxy.com

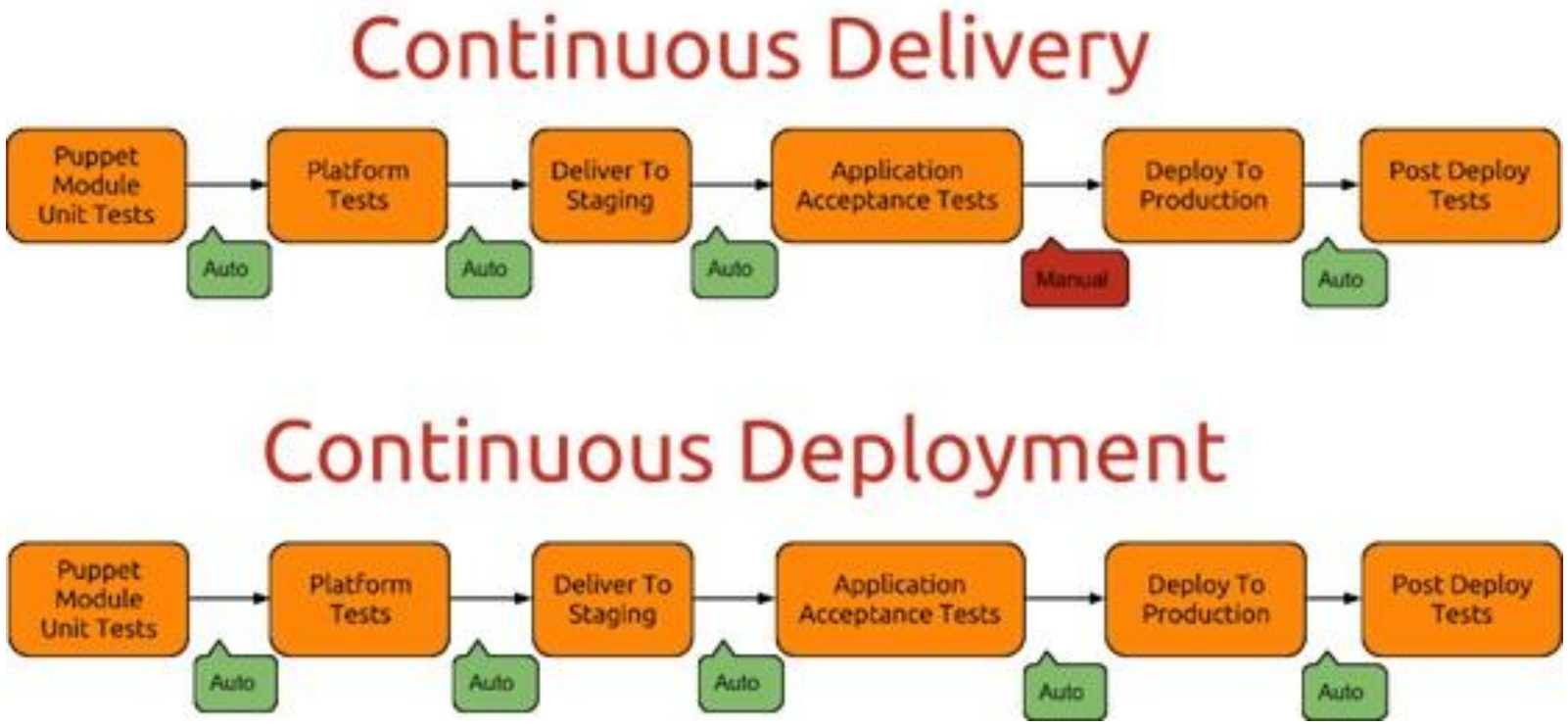
What is Continuous Delivery (CD)?

Continuous Delivery is the continual delivery of code to an environment once the developer feels the code is ready to ship. The idea is to deliver code to a user base - QA (UAT or Staging) or customers (Production environment), for continual review and inspection.

Continuous Deployment

The ultimate culmination of this process; it’s the actual delivery of features and fixes to the customer as soon as they are ready..

Delivery vs Deployment



Our Responsibilities

System Issues

1. Disk Space
2. Memory
3. Mother Board
4. Disk Corruption
5. I/O issues
6. OS issues

Applications Issues

1. App bugs
2. Security
3. Overloading / performance
4. DB issues

Third Party Dependency issues

1. Network issues
2. Dependent services

Our Responsibilities

- Source Code
    - Training Dev
    - Backup & Restore
    - Authentication & Authorization
    - Branching & Merging
    - Tagging & Labeling
    - Account
- Build
    - Building code
    - Jar
    - Deployment
    - Package

Our Roles?

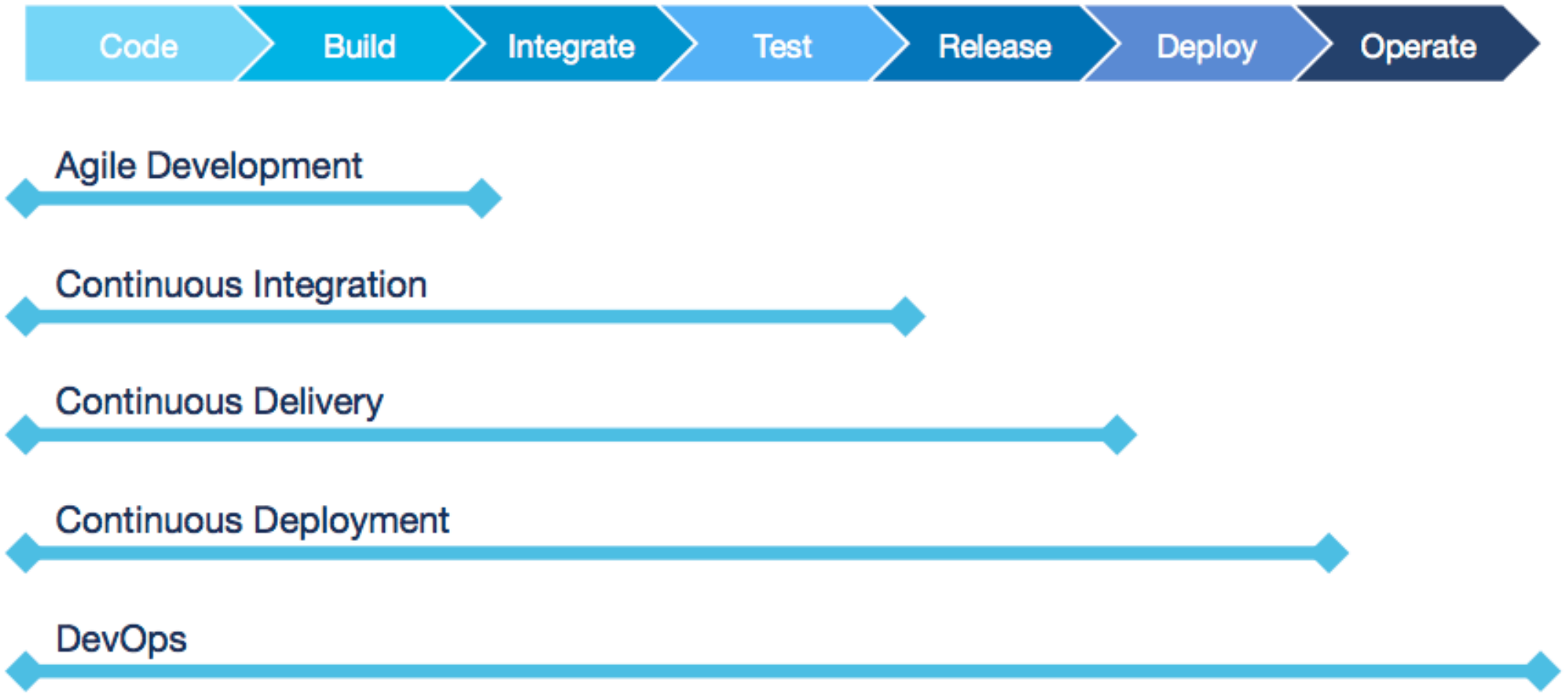
- SCM Engineer
- Build Engineer
- Install Engineer
- Application Packaging Engineer
- Release Engineer
- **AppOps Engineer**
- **DevOps Engineer**

Some time roles may differ….

- Source CM- Git Engineer, Git Admin, CM Engineer
- Build Management – Build Engineer, B & R eng
- Packaging – Application Pack Eng, Install Eng.
- Release and Deployment(Chef) – Release Engineer, Release Mgr, DevOps Eng
- System Admin – Chef – SCM Admin, SCM System Admin

So where is DevOps?

What is DevOps?



Questions?