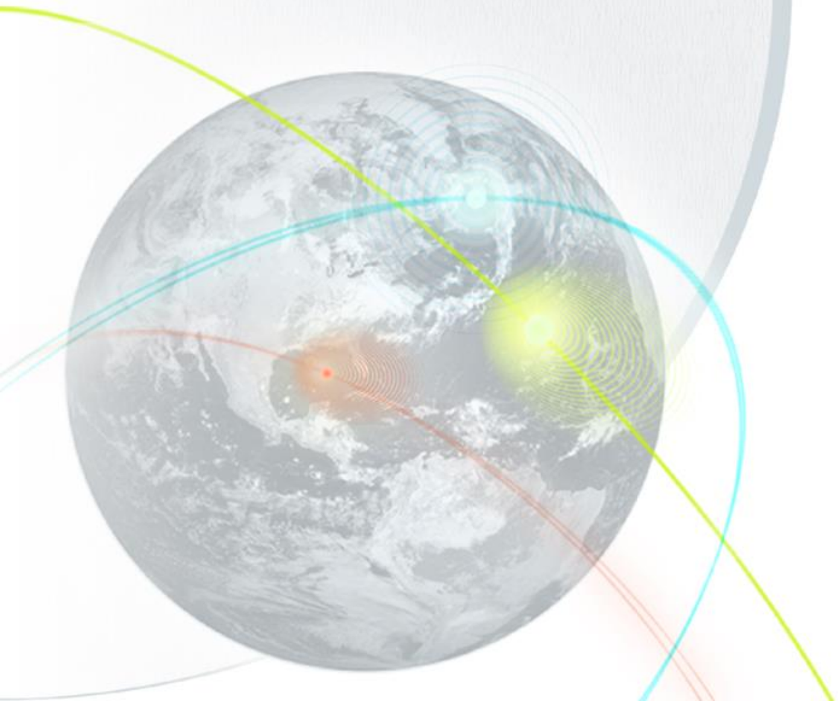
STINGER
GHAFFARIAN
TECHNOLOGIES

*Achieving* **Results** . . . *Exceeding* **Expectations**

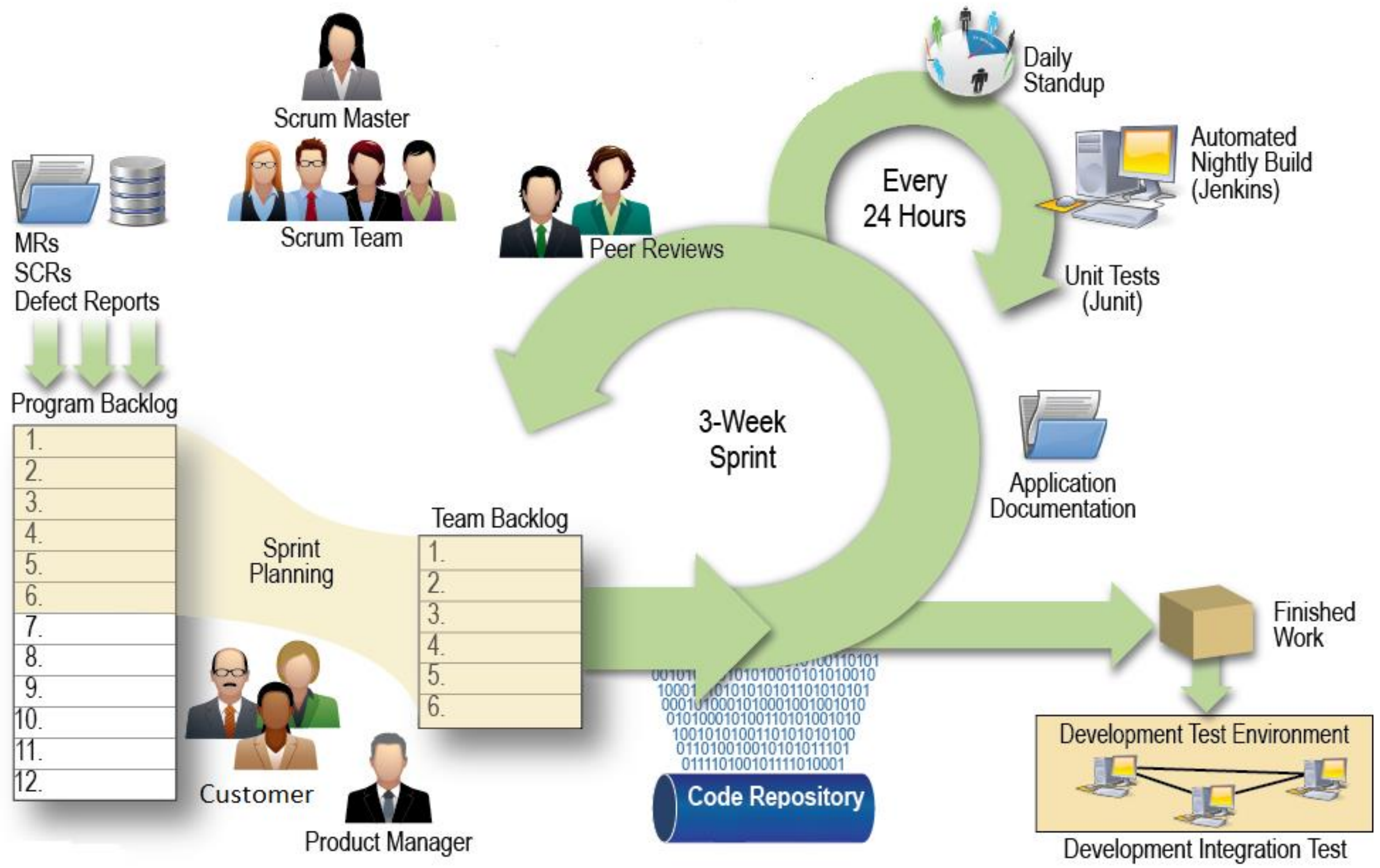


# DevOps

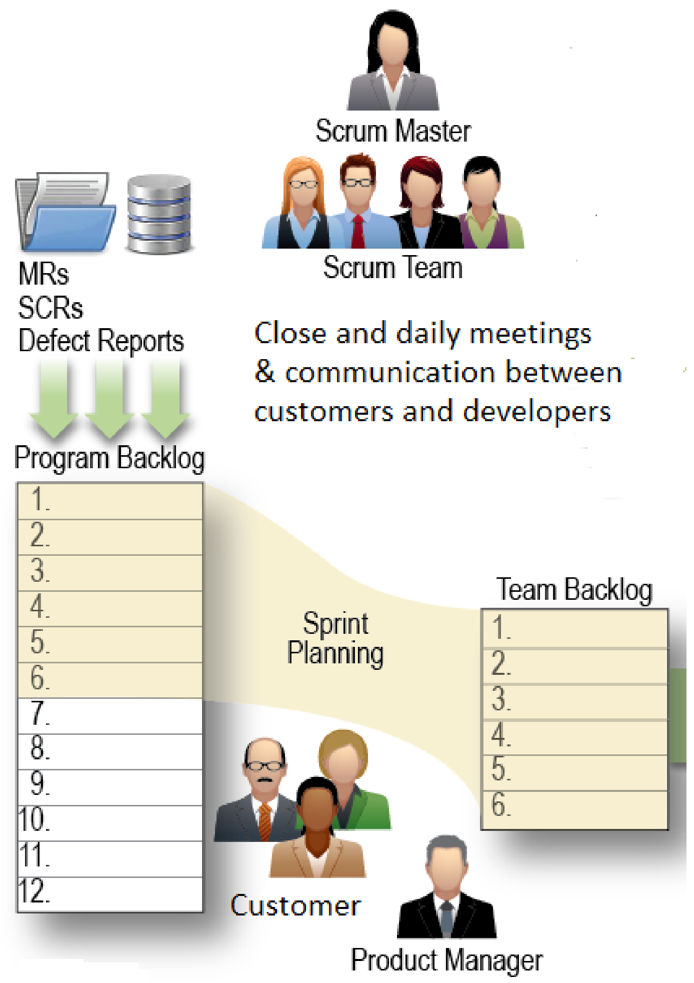
Jesse Pai
Robert Monical

8/14/2015

# Agile Software Development



# Agile Practices



- Adaptive planning
  - Acceptance of changes in requirements and adapting to said changes
- Close and daily meetings/communication with customers and developers
- Small tasks and continuous delivery of software

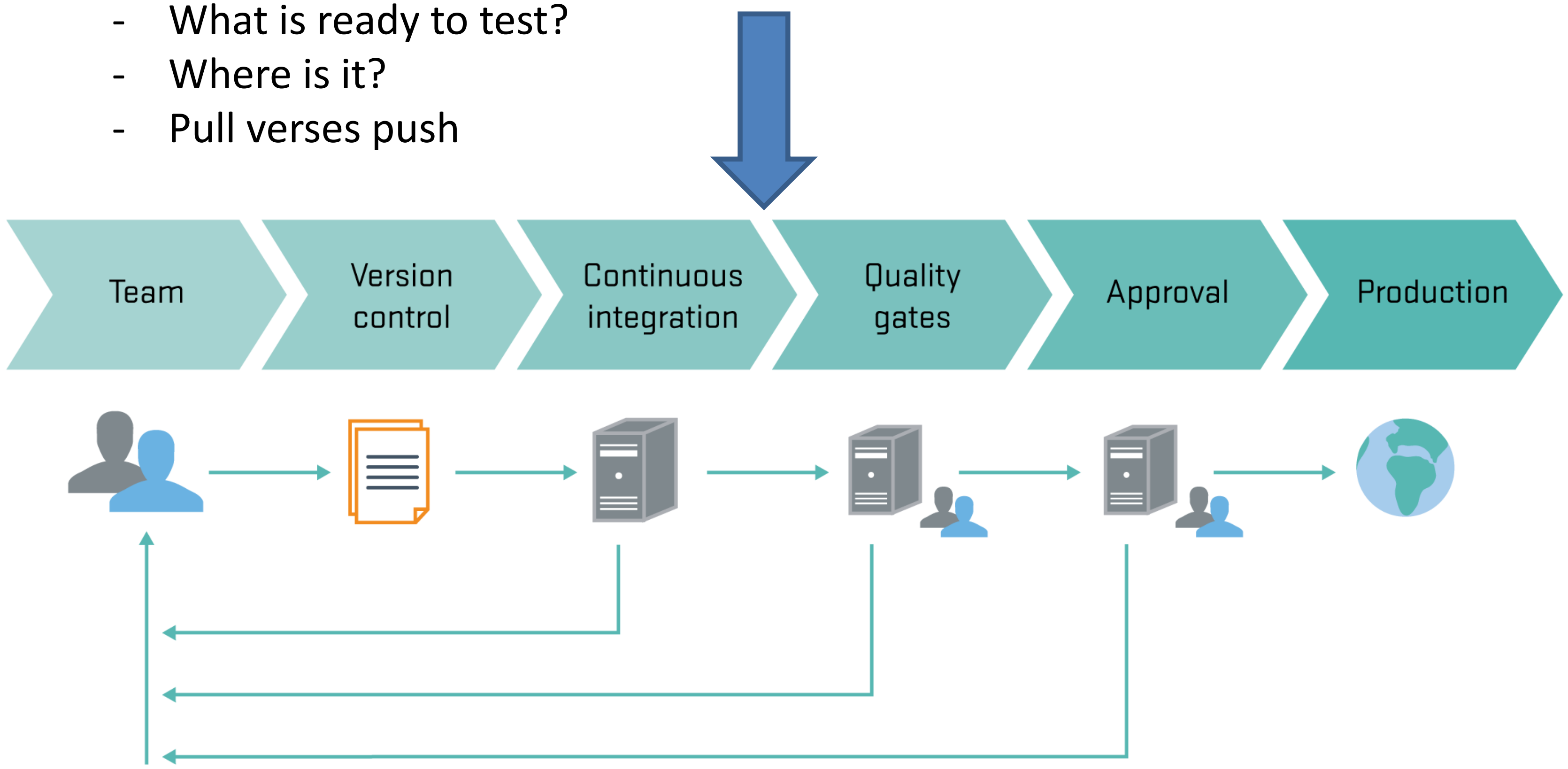
## Agile Goals

- Quicker development
- Faster resolution of problems
- Faster delivery of features
- More time to add additional features/value
- More time for QA and testing
- Reduce the effect of new feature requests, bugs, and requirements changes

# Delivery Pipeline

Chokepoint

- What is ready to test?
- Where is it?
- Pull verses push



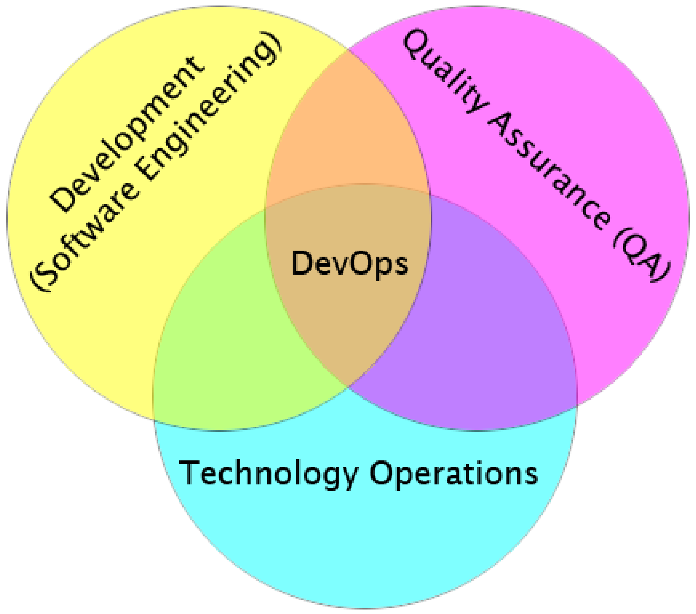
https://www.chef.io/solutions/continuous-delivery/

## Quality Gates

- Create Functional Test Environment
- For each functional test
  - Initialize Test Data
  - Execute test
  - Reset test Environment
- Create non-Functional Test Environment
- For each non-functional test
  - Initialize Test Data
  - Execute test
  - Reset test Environment
- Repeat for each additional test configuration
- Report results
- We may have test automation – rarely have test orchestration

STINGER GHAFFARIAN TECHNOLOGIES
*Achieving* **Results** . . . *Exceeding* **Expectations**

## Solution: DevOps



- Apply agile principles to the entire delivery process
- Orderly evolution of proven agile principles downstream into the software delivery pipeline
- Collaboration is key
- Tools facilitate collaboration – tools do not create DevOps
- Manage by exception

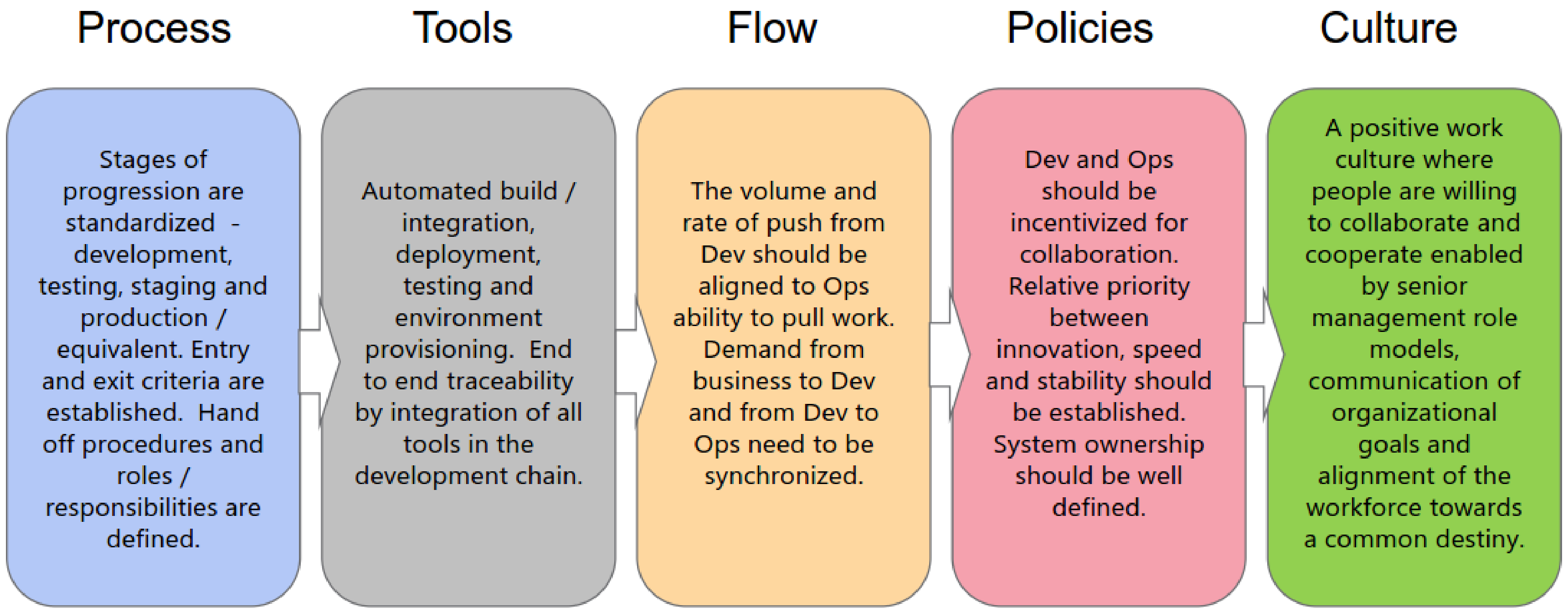
"Devops" by Devops.png: Rajiv.Pantderivative work: Wylve - This file was derived from: Devops.png:. Licensed under CC BY 3.0 via Wikimedia Commons - https://commons.wikimedia.org/wiki/File:Devops.svg#/media/File:Devops.svg

## DevOps Goals

Just like Agile:

- Deliver high-quality, valuable software systems in an efficient, fast, and reliable manner.
  - Automation: To automate as many processes as possible. Manual steps are error prone and there is no way of controlling release and deployment processes
  - Orchestration: Automate transitions between processes
  - Frequency: Increase frequency of releases. More releases means more feedback on changes to the application and its associated configuration

# Achieving DevOps



**Process**

Stages of progression are standardized - development, testing, staging and production / equivalent. Entry and exit criteria are established. Hand off procedures and roles / responsibilities are defined.

**Tools**

Automated build / integration, deployment, testing and environment provisioning. End to end traceability by integration of all tools in the development chain.

**Flow**

The volume and rate of push from Dev should be aligned to Ops ability to pull work. Demand from business to Dev and from Dev to Ops need to be synchronized.

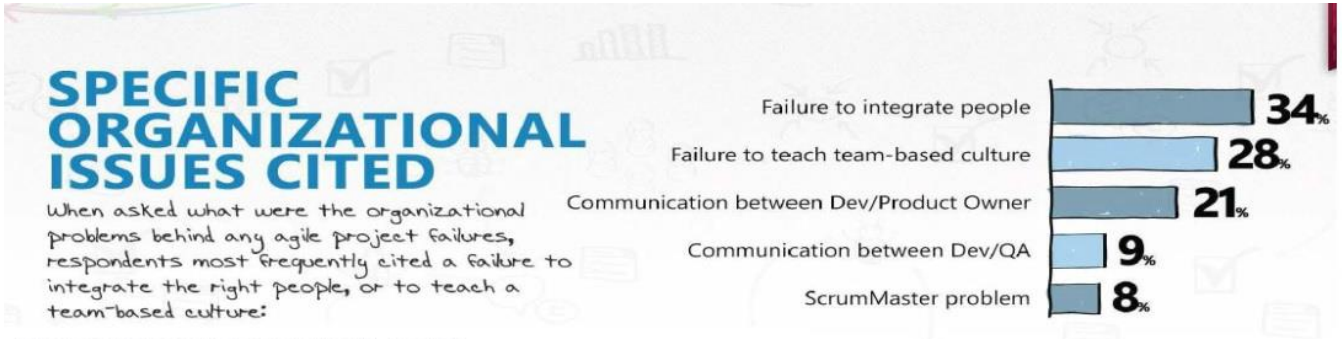
**Policies**

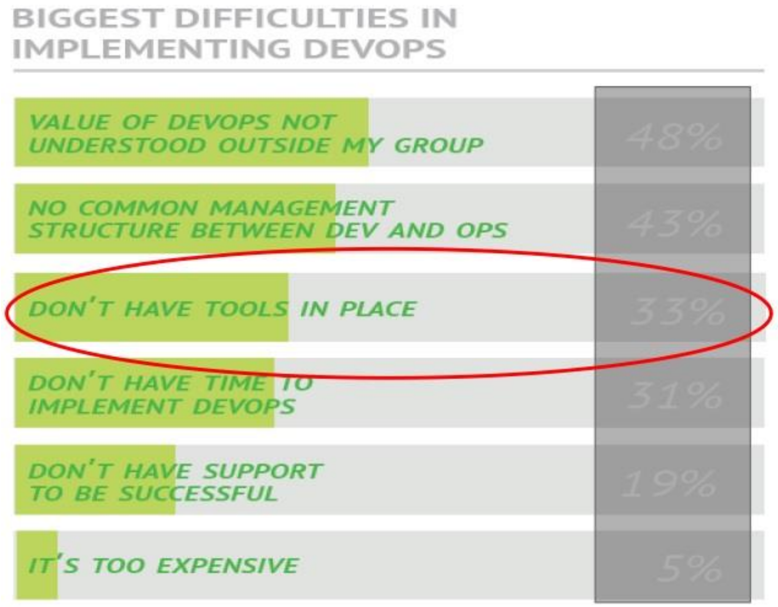
Dev and Ops should be incentivized for collaboration. Relative priority between innovation, speed and stability should be established. System ownership should be well defined.

**Culture**

A positive work culture where people are willing to collaborate and cooperate enabled by senior management role models, communication of organizational goals and alignment of the workforce towards a common destiny.

# Tools Are not That Important





- ❑ Speed
- ❑ Quality

STINGER GHAFFARIAN TECHNOLOGIES
*Achieving* **Results** *. . . Exceeding* **Expectations**

## Benefits of DevOps

- Technical Benefits:
  - Quicker development and deployment
  - Less complex problems to fix
  - Faster resolution of problems
    - Faster bug fixes and system recovery
  - Improved repeatability and reliability of deployments, lower chances of errors in configuration
- Business benefits:
  - Faster delivery of features
  - More stable operation environments
  - More time available to add value (rather than fixing/maintaining the product)

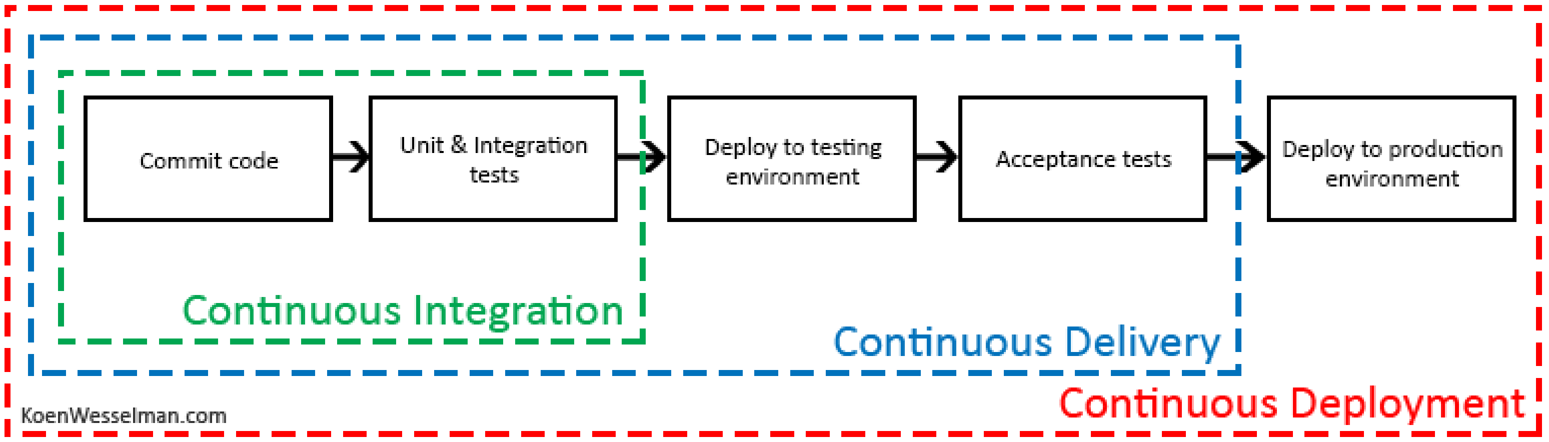
* Source: http://newrelic.com/devops/benefits-of-devops

# DevOps in the Industry

- Users:
  - Flickr, Netflix, Amazon, Twitter
    - Flickr – 10+ Deployments Per Day (http://www.slideshare.net/jallspaw/10-deploys-per-day-dev-and-ops-cooperation-at-flickr)
      - Automated Infrastructure, centralized/shared version control, one step build (allows developers to deploy their own test and QA environments), centralized and shared metrics, Agile culture
- Players:
  - Serena, IBM, AWS, CloudBees, CA, CollabNet
    - Serena – Provides tightly integrated tool chain to monitor, control, and automate the entire delivery pipeline

# Continuous Delivery vs Continuous Deployment

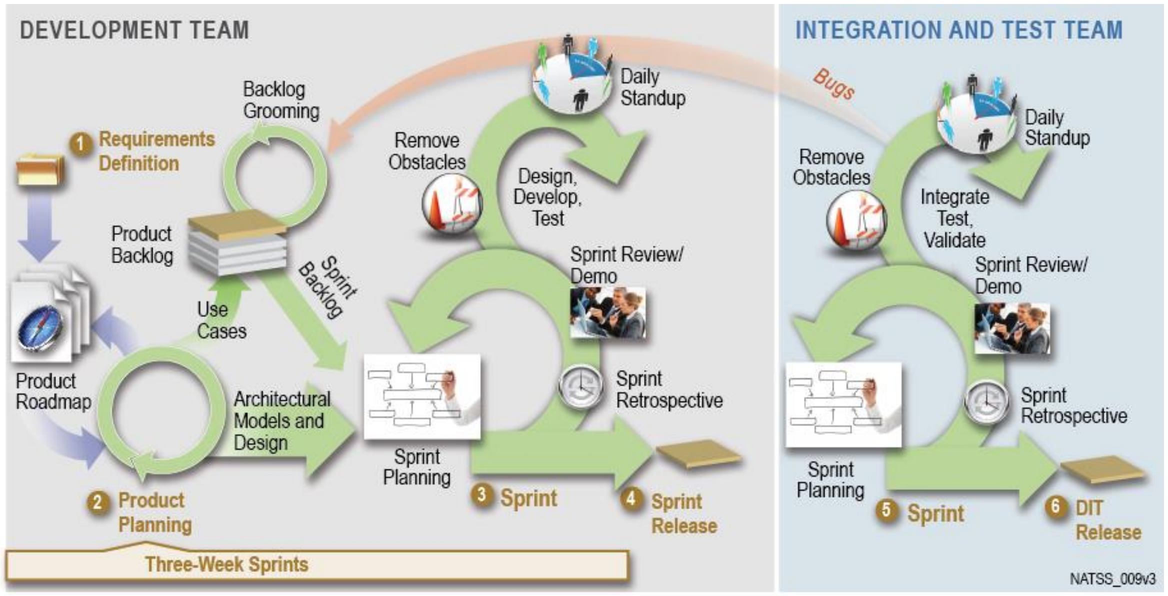
- DevOps does not mean continuous deployment. Continuous  delivery may be a better option for large organizations
- Continuous delivery doesn't mean every change is deployed to production ASAP. It means every change is able to be deployed at any time.



* Source: http://www.koenwesselman.com/difference-between-continuous-integration-continuous-delivery-and-continuous-deployment/

Stinger Ghaffarian Technologies
*Achieving* **Results** . . . *Exceeding* **Expectations**

# Best Practices: Continuous Delivery



Automated Testing

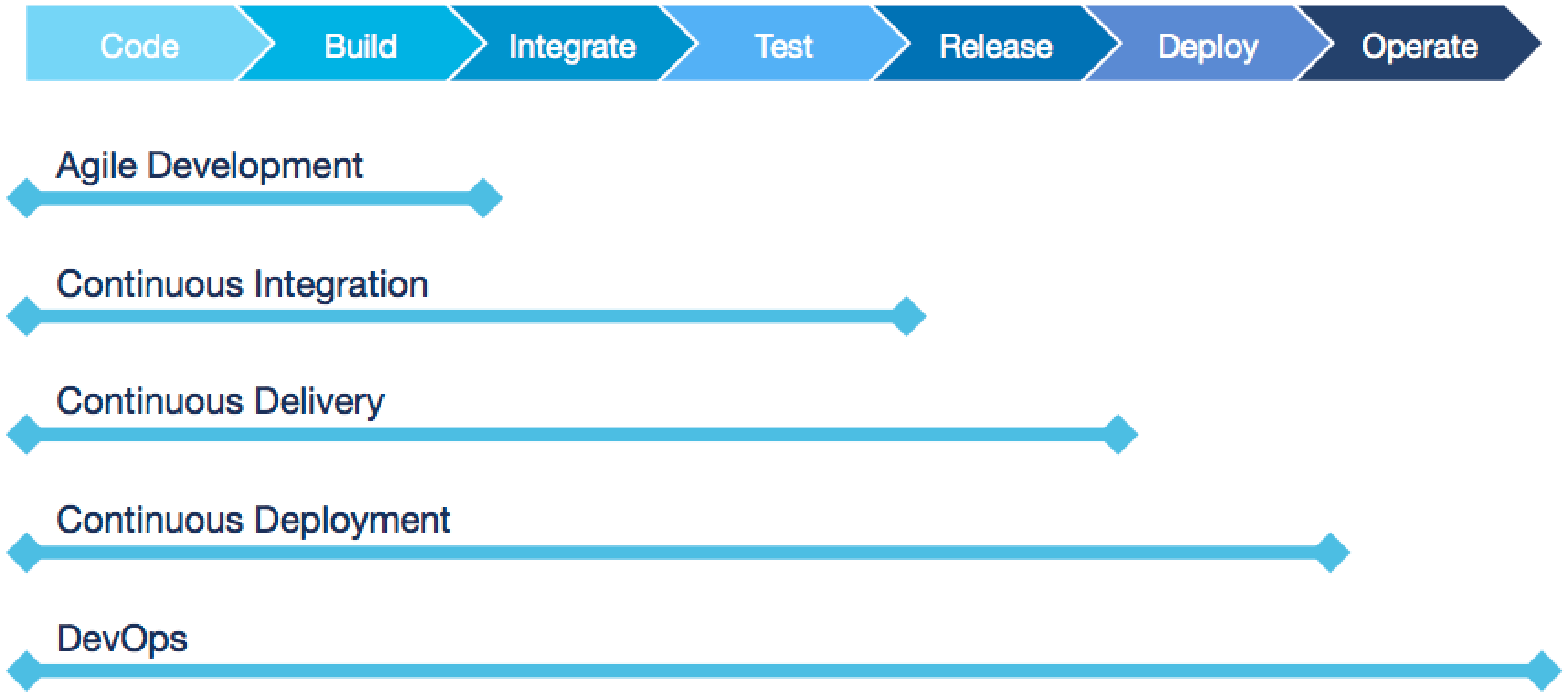
Production Support
- Integrated deployment planning
- Application Monitoring
- Dashboards

Active Stakeholder Participation

Integrated Change and Configuration Mgt.

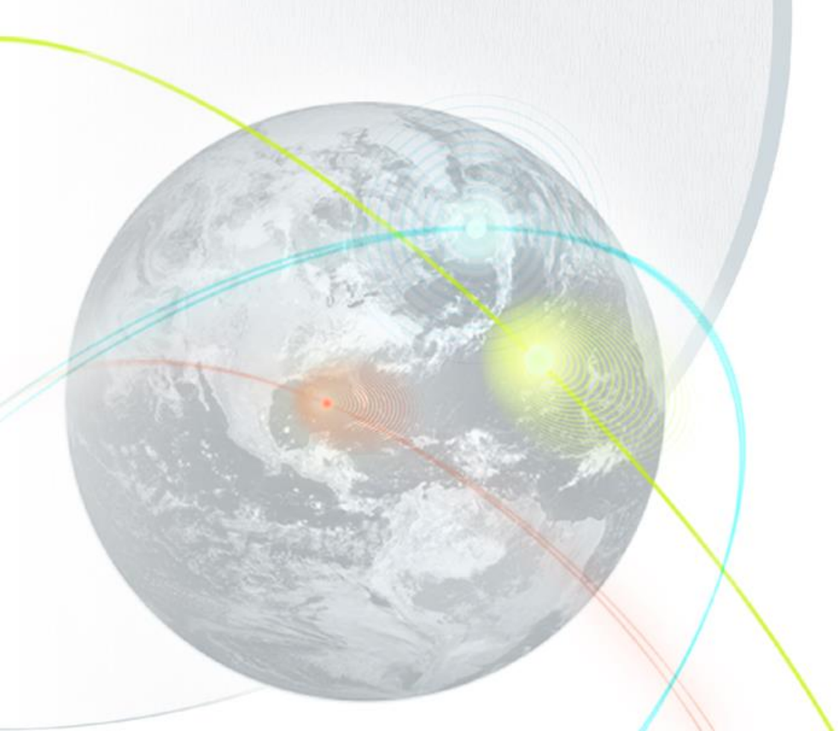
Continuous Integration Deployment and Test

# All Processes Put Together



* Source: http://www.rightscale.com/blog/cloud-management-best-practices/continuous-integration-and-delivery-cloud-how-rightscale-does-it

# SGT and DevOps



Stinger Ghaffarian Technologies
Achieving Results . . . Exceeding Expectations

## SGT and DevOps

- Prototype: Devops Lab
- Loose Tool Chain: Allows for plug-and-play approach to evaluating and selecting tools.
- Automation = Less errors, defects, reworks, and outages between information flow.
- Integration done mostly through APIs (RESTful APIs) or plugins

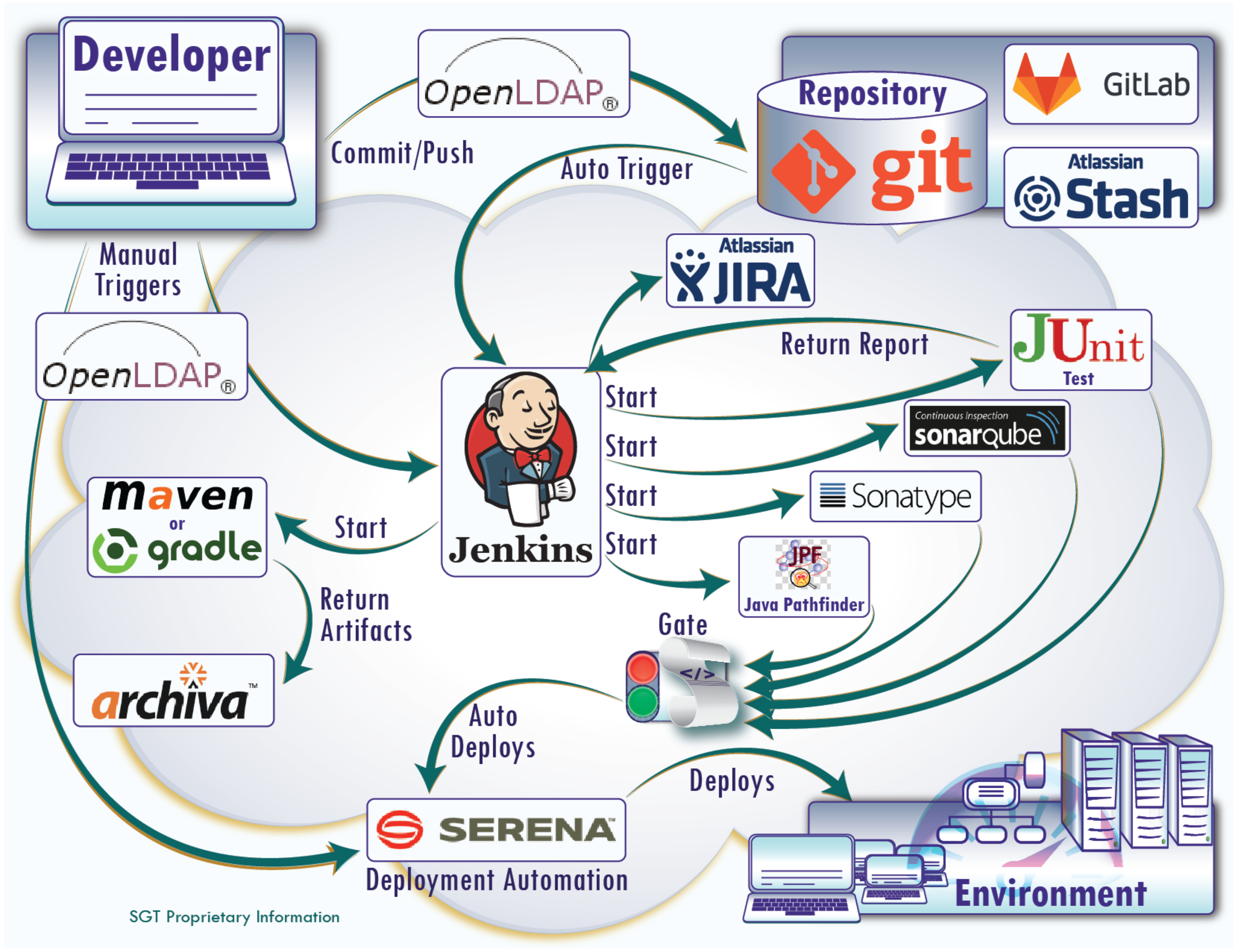
## Commercial, Hybrid, Open Sourced

- Commercial: Full support and tightly integrated tool chain
- Open Sourced: Less support and Loosely integrated tool chain
- Hybrid: Commercial + Open Sourced. Support and features where it counts, cost savings where its not needed.

## DevOps Lab Introduction

- Uses hybrid of commercial and open-source tools to create an integrated tool chain for Agile development and DevOps functionality.
- Provides a test bed to determine whether or not a company is fit to utilize and integrate such tools into its delivery pipeline.
- It provides a full delivery pipeline for development, QA, and operations.

# Tool Chain


Developer
OpenLDAP
Commit/Push
Repository
git
GitLab
Atlassian Stash
Auto Trigger
Atlassian JIRA
Manual Triggers
OpenLDAP
Return Report
JUnit Test
Start
Continuous Inspection sonarqube
Start
Sonatype
Start
Start
Java Pathfinder
maven or gradle
Start
Jenkins
Return Artifacts
Gate
archiva
Auto Deploys
Deploys
SERENA
Deployment Automation
Environment
SGT Proprietary Information

STINGER GHAFFARIAN TECHNOLOGIES
Achieving Results . . . Exceeding Expectations

## Major Features

- Supports Agile Software Development
- Provides continuous integration tools for development automation
- Deploys easily into a cloud environment
- Supports distributed team development
- Automated deployments to multiple environments
- Tracking of multiple environments and deployments
- Automation available for:
  - Code analysis
  - Dependency analysis
  - Tests
  - Issue/tasks

## Tools

- Jira
- GitLab
- Jenkins
  - Junit
- Code Quality Analysis
  - SonarQube
  - Sonatype Nexus Lifecycle
- Serena Deployment Automation

## Jira

- Jira is a commercial issue/task management software that can support Agile software development.
- Supports:
  - Sprints
  - User Stories
  - Tasks, Bugs, and Issues
- Provides an easy to use UI to organize and execute project tasks

## GitLab

- GitLab is an open-sourced centralized git repository
- Community Edition supports up to 10 projects
- Provides webhook and plugin support

## Jenkins

- Jenkins is an open sourced continuous integration tool
- The “heart” of the DevOps lab.
  - Almost every part of the DevOps Lab is connected to Jenkins
- Provides a massive amount of features for building and testing.
  - Plug-ins for additional features
- Enables automatic builds, tests, QA analysis, and deployments

## SonarQube and Sonatype NL

- SonarQube (Open Sourced)
  - Source code analysis tool
  - Analyzes for violations of the 7 axes of code quality.
    - Architecture & Design, Duplications, Unit tests, Complexity, Potential bugs, Coding rules, Comments
- Sonatype Nexus Lifecycle (Commercial)
  - Analyzes dependencies for security concerns and licensing issues
  - Compares dependencies utilized against MavenCentral dependencies/components.
  - Provides an easy to understand report

STINGER GHAFFARIAN TECHNOLOGIES
Achieving Results . . . Exceeding Expectations

## Serena Deployment Automation

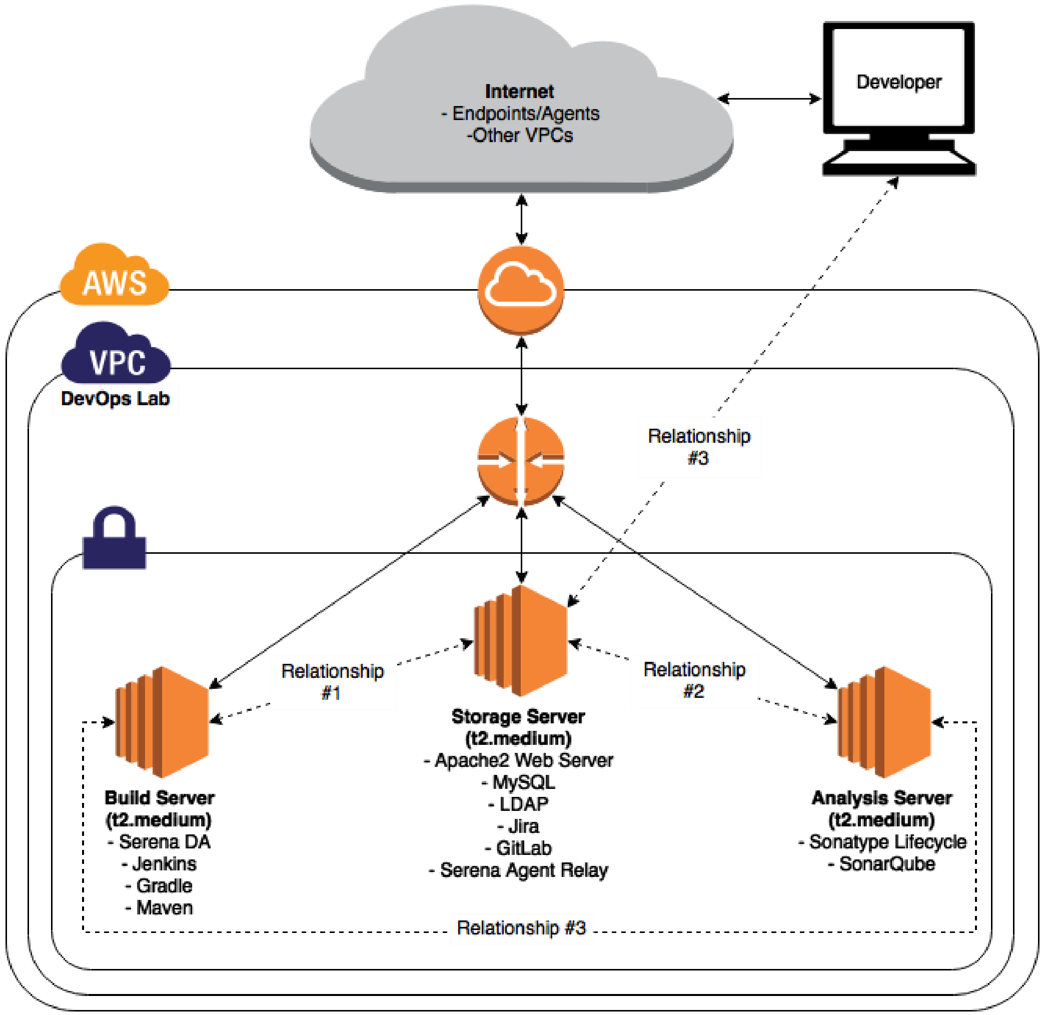
- SDA is a commercial deployment automation tool
- The main tool in organizing and deploying environments and projects
- Provides version tracking on multiple environments (Development, QA, and Production)
- Easy to use GUI based workflow editor
- Plug-in support for additional features/processes

SERENA

Stinger Ghaffarian Technologies

*Achieving* **Results** . . . *Exceeding* **Expectations**

# DevOps Current Architecture

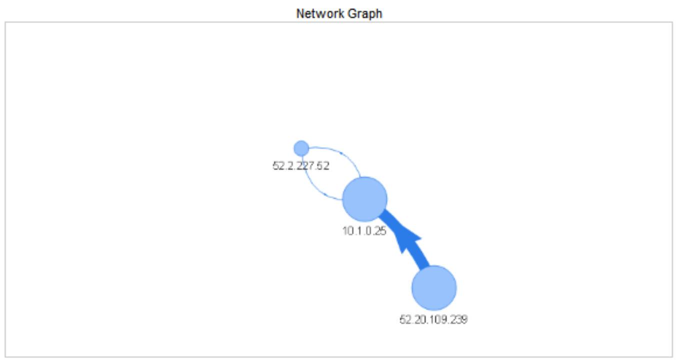


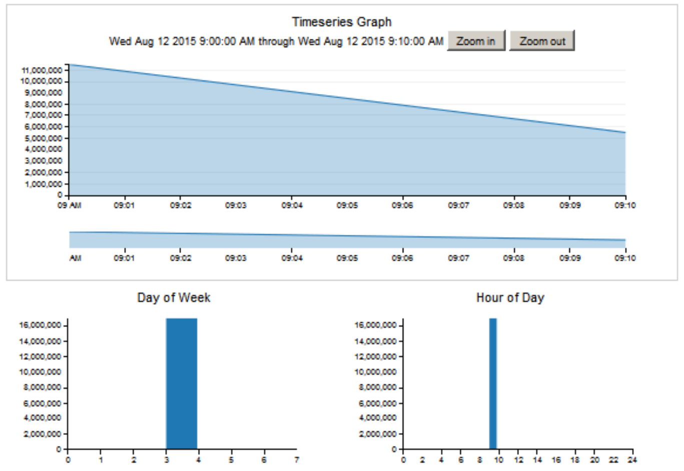
# Deployment: RNAAT

- Real-time Network Activity and Anomaly Tracker
- Network Analysis Tool (Distributed/Cloud Based)
- Multiple servers all manually setup/configured
  - Zookeeper, Apache Storm (Nimbus, UI, and Supervisor, MongoDB, CubeDB, and Kafka
  - Architecture must be setup and ready to go for this to work
- Error prone + time-consuming troubleshooting due to incorrect configuration

Enter DevOps:

- One-click or automated deployment of architecture to specified environment
  - Repeatable and reliable environments
- Automated Junit tests and reporting
- Easier access to code analysis tools due to automated triggering of analysis
- Commit and GO



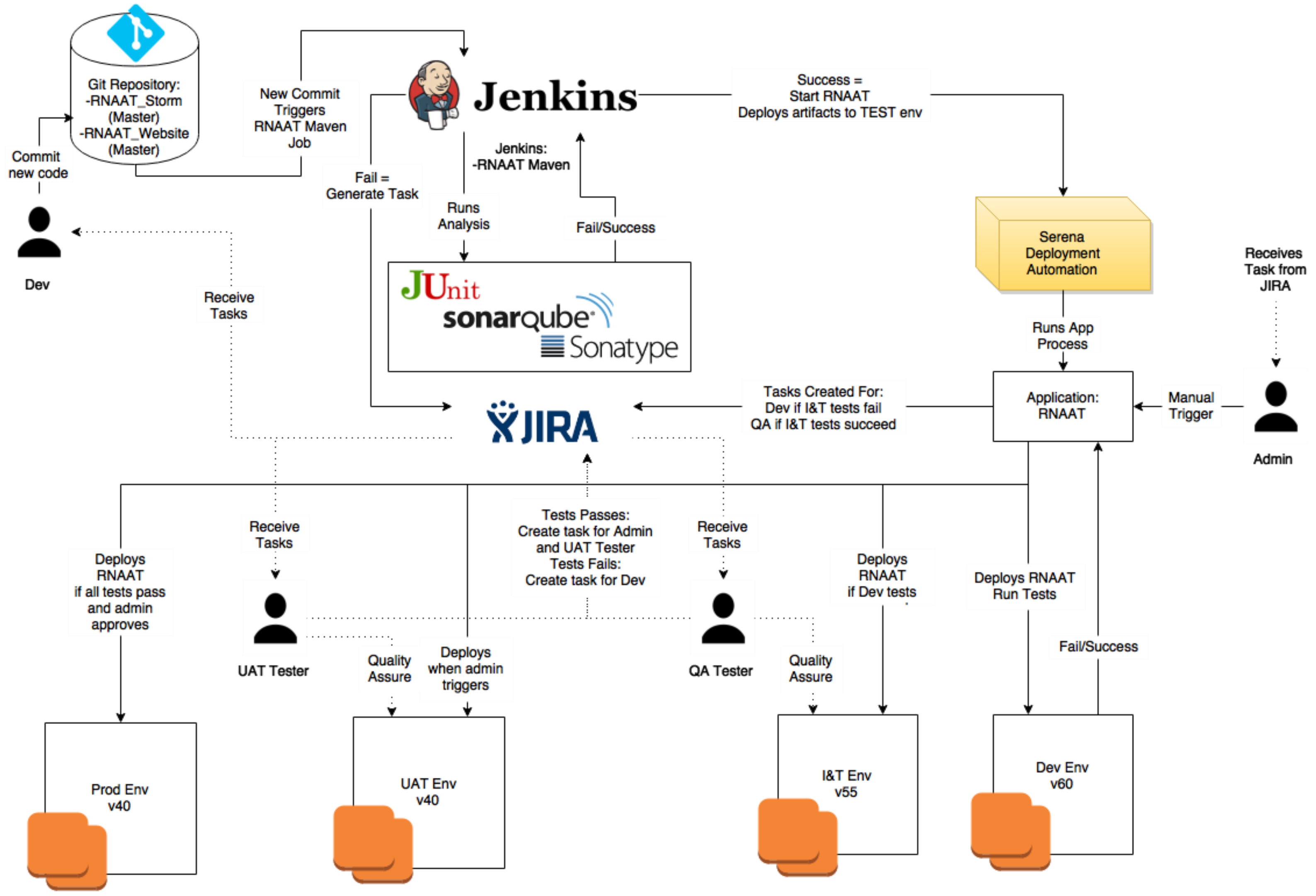


## For more information…

- ITC Website:
  - http://www.innovation.sgt-inc.com/
- DevOps Lab Wiki:
  - http://innovation.sgt-inc.com/dokuwiki/doku.php
  - Select “DevOps Lab”

Stinger Ghaffarian Technologies
*Achieving* **Results** . . . *Exceeding* **Expectations**

# Deployment: RNAAT



Git Repository: -RNAAT_Storm (Master) -RNAAT_Website (Master)
Commit new code
Dev
New Commit Triggers RNAAT Maven Job
Jenkins
Fail = Generate Task
Jenkins: -RNAAT Maven
Runs Analysis
Fail/Success
JUnit
sonarqube
Sonatype
Success = Start RNAAT Deploys artifacts to TEST env
Serena Deployment Automation
Receives Task from JIRA
Runs App Process
Application: RNAAT
Manual Trigger
Admin
Receive Tasks
JIRA
Tasks Created For: Dev if I&T tests fail QA if I&T tests succeed
Receive Tasks
Tests Passes: Create task for Admin and UAT Tester Tests Fails: Create task for Dev
Receive Tasks
Deploys RNAAT if all tests pass and admin approves
UAT Tester
Quality Assure
Deploys when admin triggers
QA Tester
Quality Assure
Deploys RNAAT if Dev tests
Deploys RNAAT Run Tests
Fail/Success
Prod Env v40
UAT Env v40
I&T Env v55
Dev Env v60