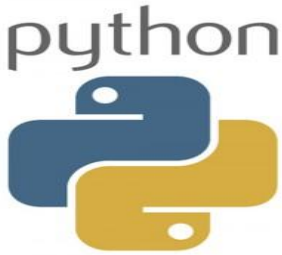
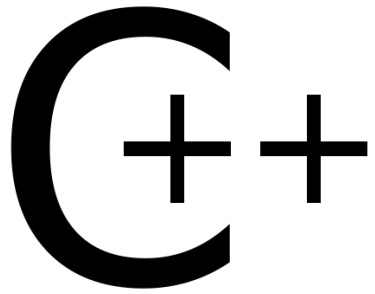
Continuous Integration and Delivery

by

Rajesh Kumar
email – rajesh@scmGalaxy.com

Technology





Source Code Management

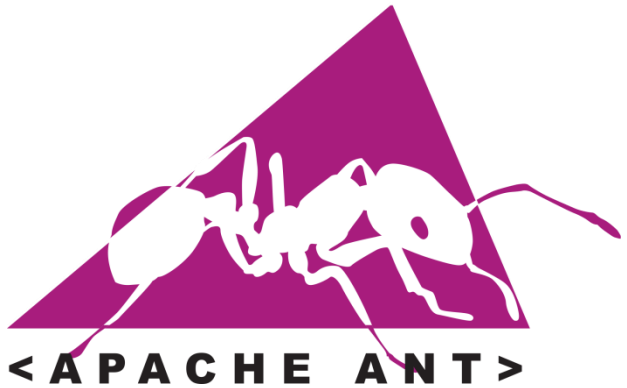
git

SUBVERSION®

Build Tools

Make

Nmake



maven

Usually we don’t have

- Centralized Source Management Tools
- Centralized Build Server for all the JDA Products
- Common Build Script across the organization
- Configuration Management Tools such puppet or salt
- Standard Release and Deployment Process
- Automated Testing Just after build
- Code Review Setup
- Static Code Analysis
- Test Coverage
- Centralized Dashboard for all CI/CD reporting and Notification
- SCM Infrastructure Monitoring Setup

Goals

- Faster Development Process across the JDA Solutions
- Reduces Costs and Burdon on People and Process
- Reduces Engineers options making SCM support easier
- Enforces Uniform Corporate Process
- Standard Version control and Builds
- Fast, Reliable and ready to QA/Deploy Build operation
- Streamline Release and Deployment Process
- Simplified Branching, Tagging, and Directory organization
- Strong Tools integration between CM, Build, QA and Deployment, Feedback etc

- Unify the software delivery process across the JDA product line SDLC
  - Reduce duplication of effort
  - Improve consistency
  - Embrace change
  - Remove known sources of error

Business Benefits

- Developer productivity
- Smoother workflow within organization
- Supports many Development Models
- Open source, Profitability and Growth
- SCM team is focusing on developing CM technology and framework instead of support.

Non-Functional Benefits

- Security
- Backups
- High Availability
- Upgradability
- Faster Development
- Monitoring and Notification
- Scale

Continuous Integration

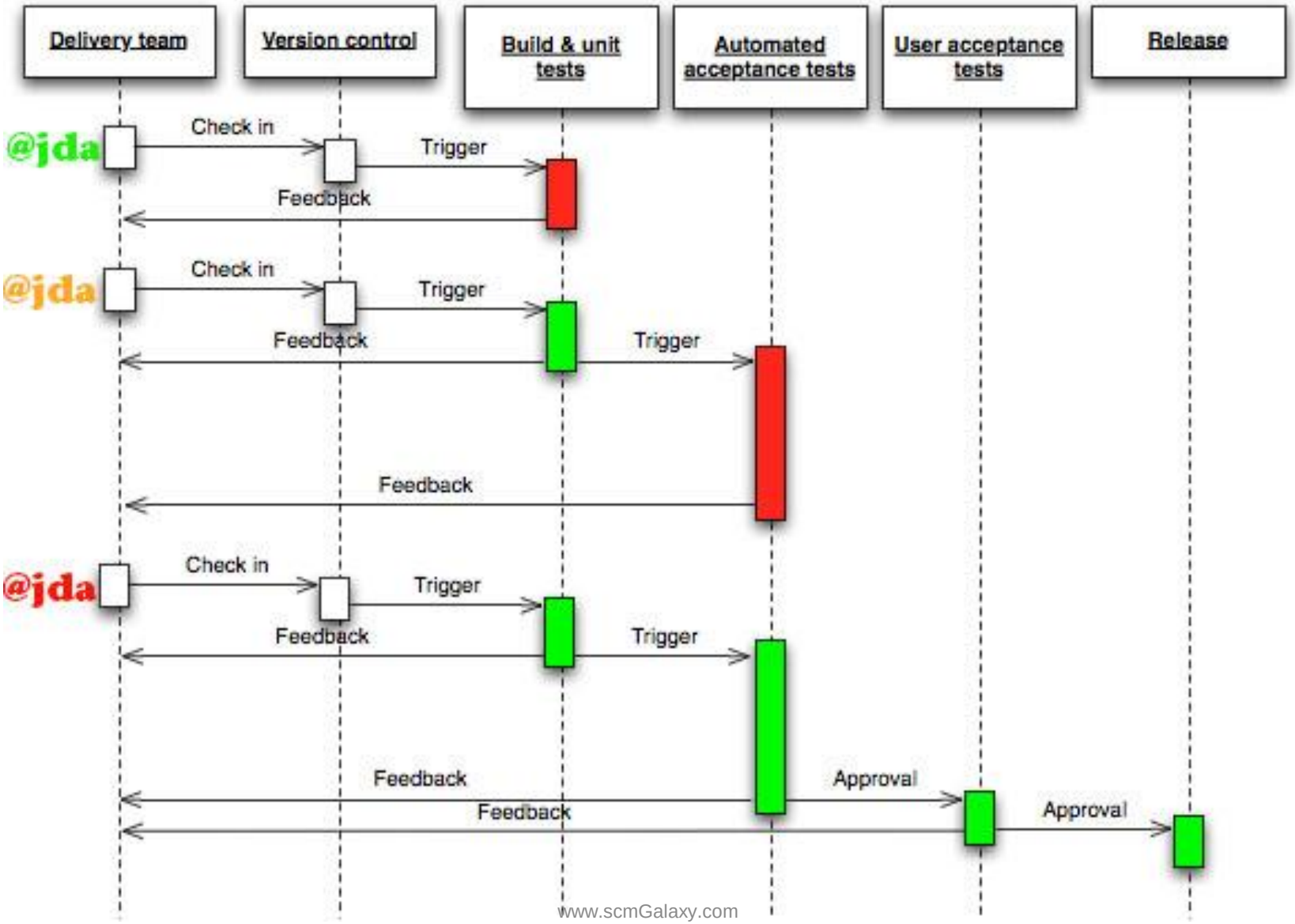
The process of automatically building and testing your software on a regular basis.

Continuous Delivery

Continuous Delivery doesn't mean every change is deployed to production ASAP. It means every change is proven to be deployable at any time

@ccaum

Continuous Integration and Continuous Delivery workflow



Continuous Delivery

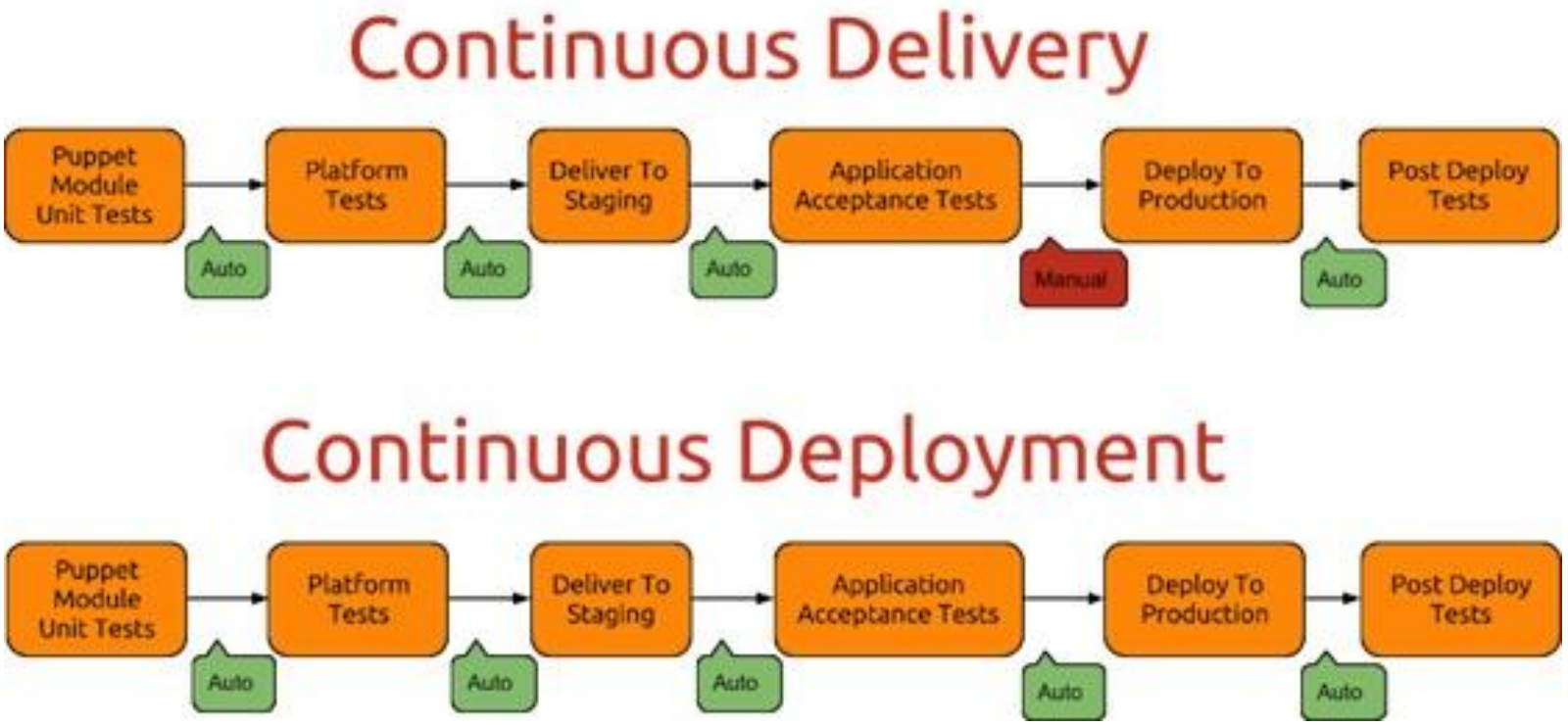
A logical step forward from continuous integration. If your tests are run constantly, and you trust your tests to provide a guarantee of quality, then it becomes possible to release your software at any point in time.

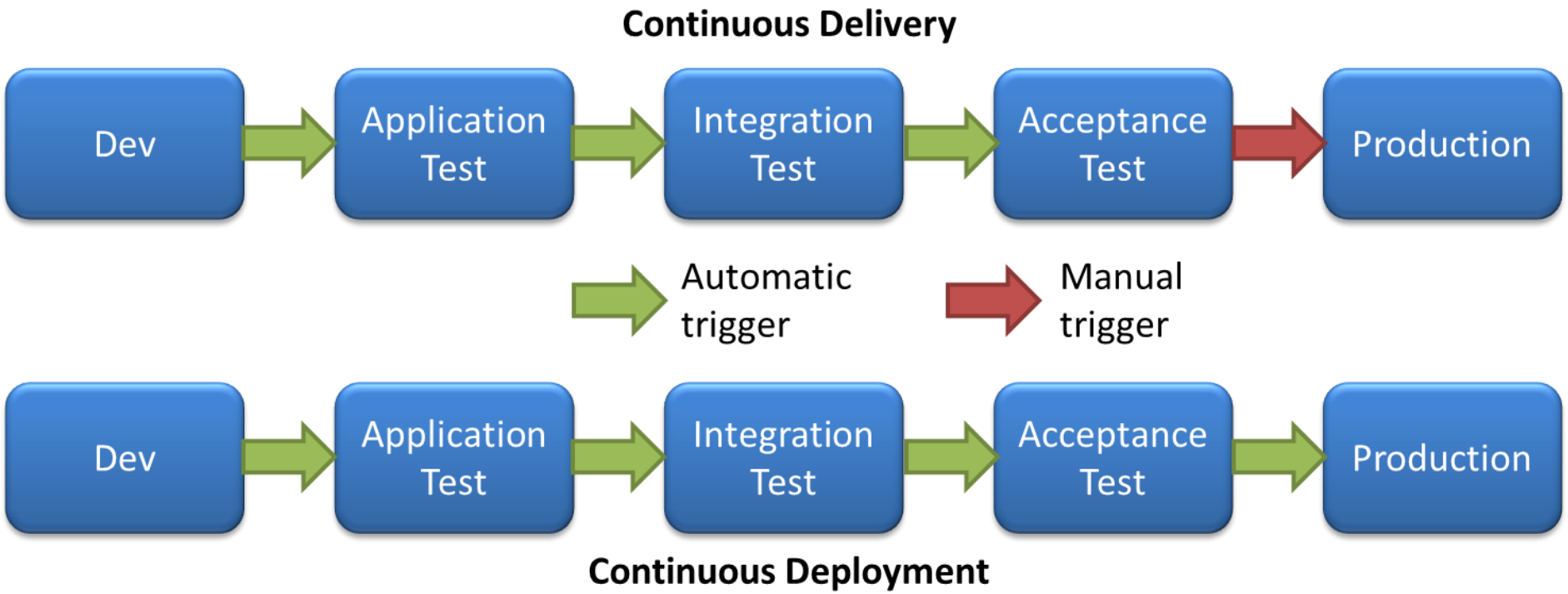
Continuous Deployment

Are we doing it?

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





Continuous Monitoring

So why would you bother?

- Monitoring is for Feedback of
  - Code Quality **poor** | **average** | **good** | **best**
  - Builds **pass** | **fail**
  - Test Quality
  - System Availability
  - Performance of Tools and Products
  - SCM Infrastructure Availability
  - It save time of.
    - Developer
    - Engineer
    - Manager
    - Director
    - CEO

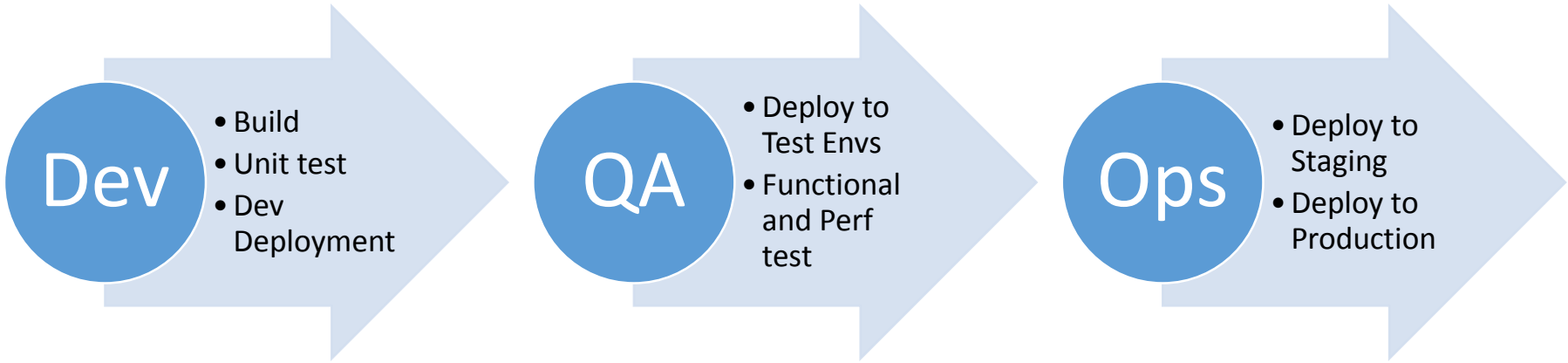
We want…

- Continuous Integration is for immediate feedback
- Automated Testing based on Continuous Integration
- Continuous Delivery based on Automated Testing
- Automated Test Deployments ???????
- Useful Feedback on time
- Measurement!!! & Monitoring!!!

DevOps is a Philosophy

- Agile & Lean applied to the whole software delivery chain, not just developers

- Driven by efficiency and consistencies
  - Building applications
  - Building Environments
  - Configuring Applications and environments
- Optimizing software Delivery to End to End



What is CI/CD

- no definition
- It certainly isn't a person
- No strict rules
- No strict tools
- It's not even new
- If you aren't doing it already ...
- ... you are doing it wrong

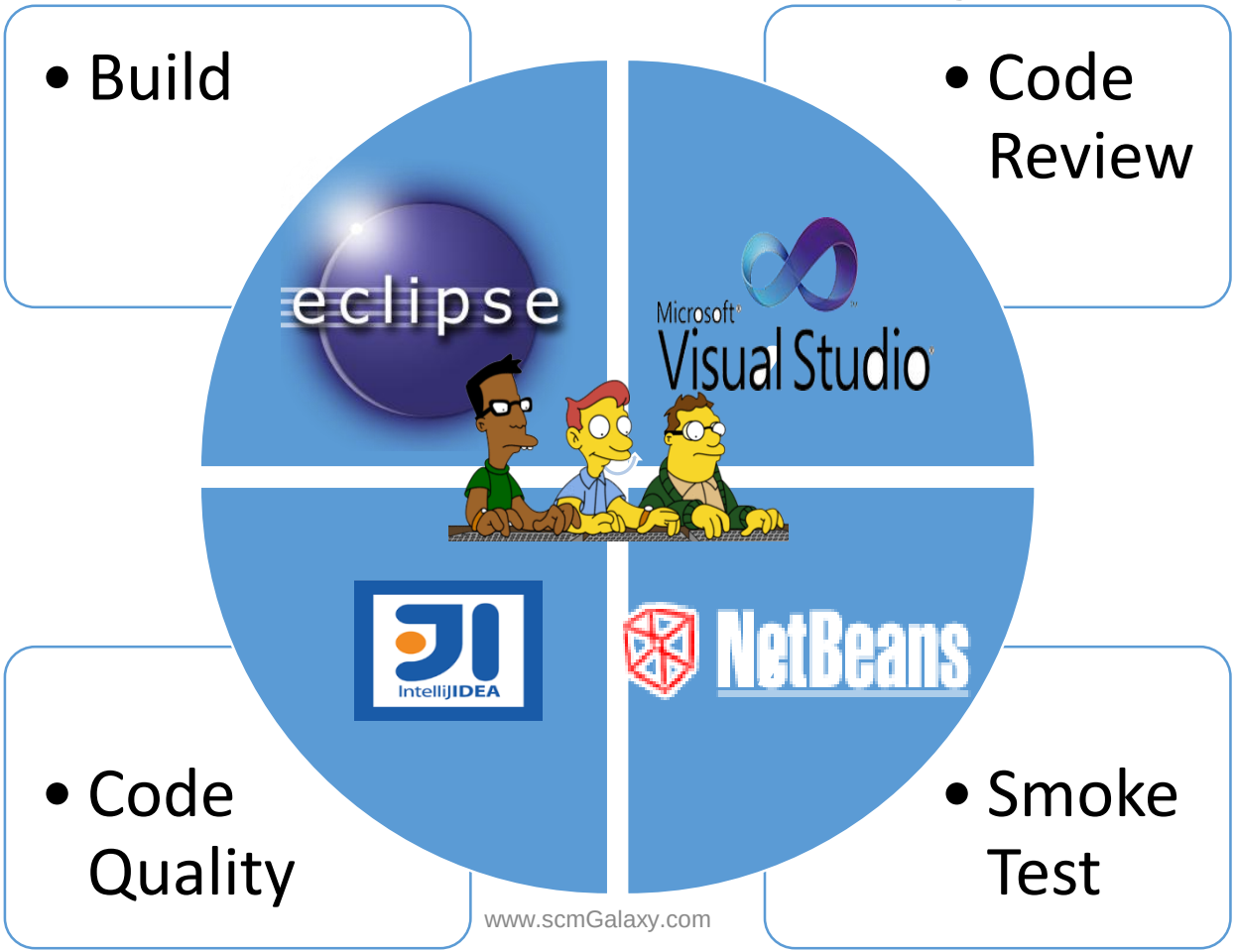
What we have?

IDE - Eclipse, NetBeans, VSStudio
Repos - Clearcase, CVS, SVN, PVCS, Git
CI - Jenkins
ReposMgr - Artifactory
Test Suite
- Junit
- HP Mercury Quality Center
- HP Mercury QuickTest Pro
- NUnit
- Selenium
- Silk Central Test Manager

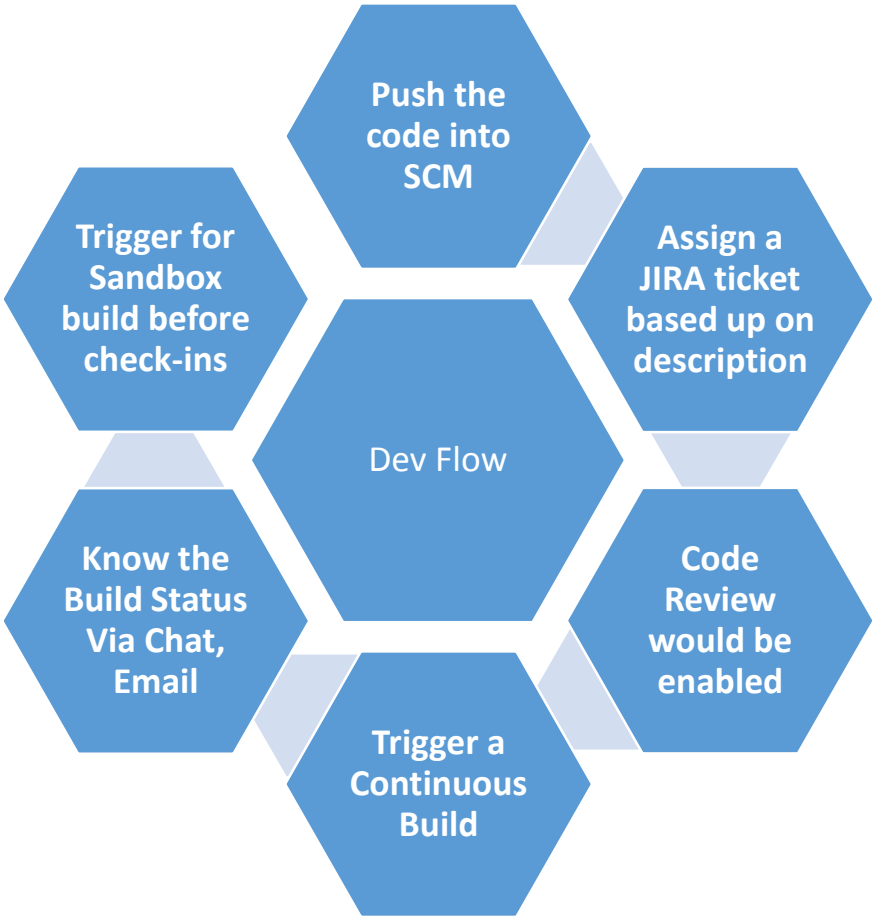
Bug Tracking - Jira
Code Analysis - Sonar
(PMD+Checkstyle+Findbugs)
Wiki - confluence wiki
Code Review - Code Collaborator
Code Search - Fisheye
Line of Code - CLOC, SLOC

Coder Just Code, Everything in IDE

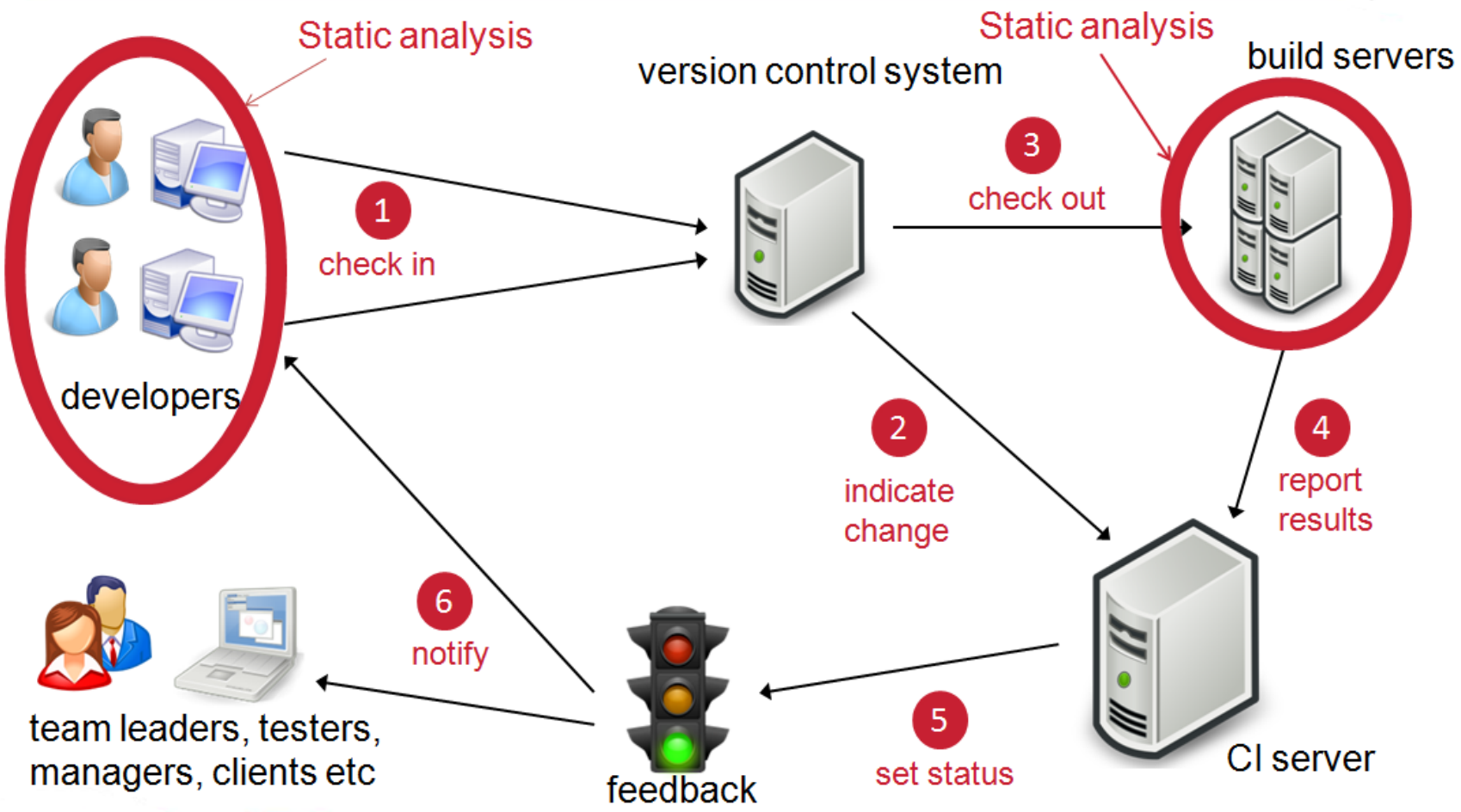
Just Code, Single Platform for all the task, Work faster and avoid context-switching



Coder Just Code



Continuous Build



Continuous Build Cont…

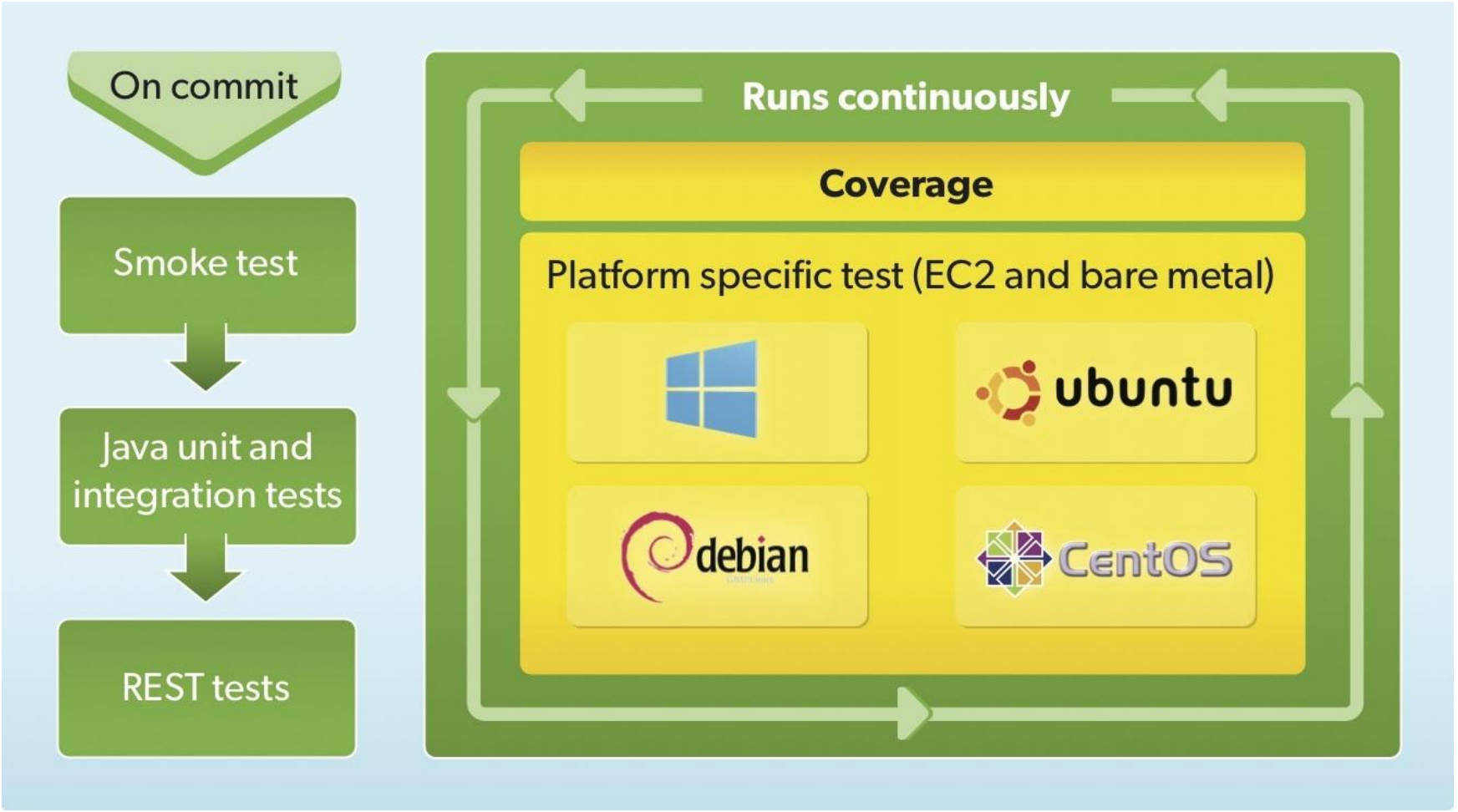
Build Type

- Sandbox Build
  - This Build is for developers code verification before check-ins
- Continuous Build
  - This Build is for code Integration and build quality post check-ins.
- Daily Build
  - This Build is for Unit test | Upload to Artifactory and testing Test server and deployment to various stages.

Continuous Delivery

- Each Daily Build should to be publish on Artifactory
- There could following kinds of Build Candidates would be saved in Artifactory?
  - Daily Build Candidates [Used for Smoke and Unit Test]
  - Release Candidates [Used only for Release and Production Deployment]

Continuous Testing



Continuous Monitoring

Unify…

One Git, One Clearcase, No CVS, NO SVN, NO PVCS)

(I know migration from CVS/Clearcase is challenging but we should start creating buzz in PD for GIT and give them a choice)

5 Solutions 5 CIs Server

(100s of node distributed all around the world)

5 Months == each Phase

One artifactory

We need to have

- Dashboard Solution
  - **Jenkins Dashboard View Plugins**
  - OpenStack Dashboard
  - Jenkins Dashboard View Plugins
  - **Sonar Dashboard**
  - Configuration Manager 2007 Dashboard

Configuration Management Tools

Puppet

SCM Infrastructure Monitoring

Nagios

Code Search and Visualize changes

FishEye

Code Review Tools

Code Collaborator

Need to Evaluate Crucible as well)

3rd Party Components

- Track via Nexus Community version

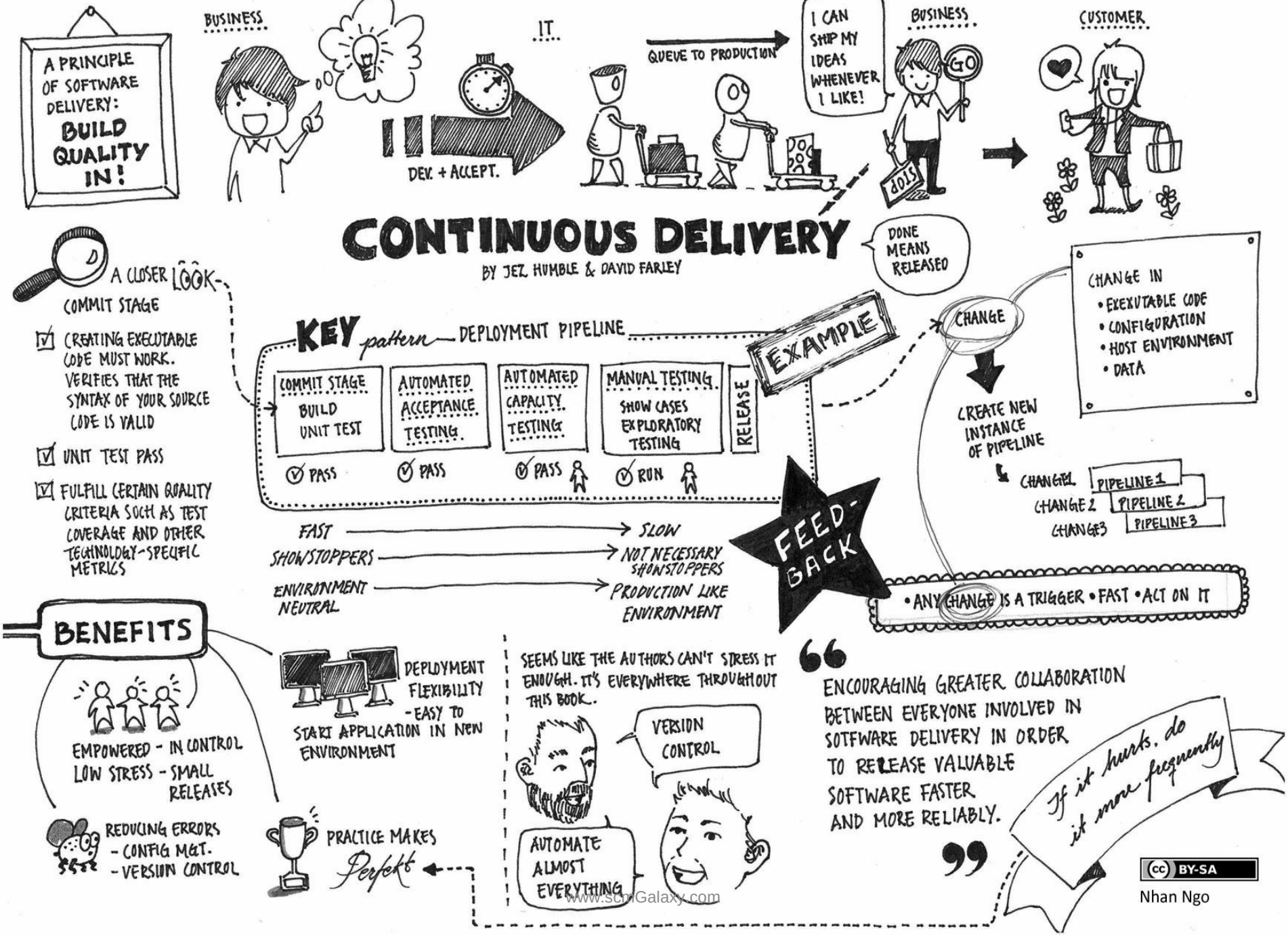
We need to have(Future)

- Log Management
  - Splunk
  - **Graylog2 + Elasticsearch + MongoDB**
  - Fluentd + Elasticsearch + Kibana
  - Logstash + Elasticsearch + Kibana
  - OpenTSDB

We need to have(Future)

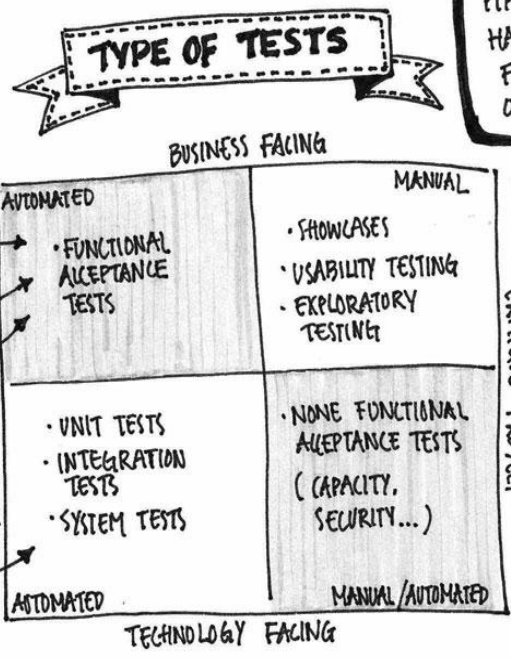
- Application Performance Monitoring Tools
  - New Relic
  - AppDynamics
  - CopperEgg
  - Datadog
  - BigPanda
  - LogicMonitor
  - Stackify
  - Site 24 x 7

Some Board Work….Found on Google



What does CONT. DEL. say about TEST STRATEGY

> TESTING IS A CROSSFUNCTIONAL ACTIVITY THAT INVOLVES THE WHOLE TEAM, AND SHOULD BE DONE CONTINUOUSLY FROM THE BEGINNING OF THE PROJECT.



YOUR DEPLOYMENT PIPELINE SHOULD HAVE ALL THESE FOUR TYPE OF TESTS.

? NOT MUCH INFORMATION REGARDING THIS TYPE OF TESTS IN THE BOOK.

INTEGRATION TEST - TEST THAT ENSURE THAT EACH INDEPENDENT PART OF YOUR APPLICATION WORKS CORRECTLY WITH THE SERVICES IT DEPENDS ON.

WILL FORM PART OF YOUR REGRESSION TEST SUITE

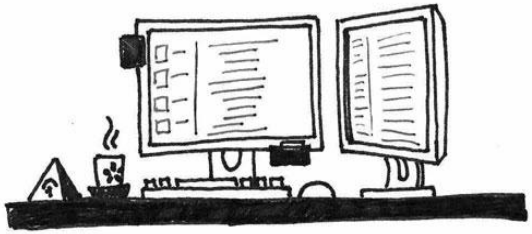
UNIT TEST
COMPONENT TEST
DEPLOYMENT TEST

REGRESSION TEST? NOT MENTIONED IN THE DIAGRAM. THEY ARE CROSSCUTTING CATEGORY.

ANY PLAN THAT DEFERS TESTING TO THE END OF THE PROJECT IS **BROKEN**.

Nhan Ngo

What does CONT. DEL say about AUTOMATED ACCEPTANCE TESTING



OBJECTIVE: PROVE THAT OUR APPLICATION DOES WHAT THE CUSTOMER MEANT IT TO, NOT THAT IT WORKS THE WAY IT'S PROGRAMMERS THINK IT SHOULD.

FAIL FAST ⇅ FAST FEEDBACK

UNIT TESTS SHOW THAT A SINGLE PART OF THE APPLICATION DOES WHAT THE PROGRAMMER INTENDS IT TO.

COST A LOT

RESPONSE

CAN BE COST EFFECTIVE IF WE DESIGN IT SMARTLY.

PERFORMANCE

- REFACTOR TESTS → ATOMIC TESTS
- CREATE A CLEAN RUNNING INSTANCE OF THE SYSTEM UNDER TEST AT THE BEGINNING OF THE ACCEPTANCE TEST RUN, RUN ALL OF THE ACCEPTANCE TESTS AGAINST THAT INSTANCE AND SHUT IT DOWN AT THE END.
- USING COMPUTE GRIDS
- PARALLEL TESTING

CREATING ACCEPTANCE TEST IS A COLLABORATING PROCESS.

Q: WHY?
A: TRANSPARENCY + TAKE AWAY ASSUMPTIONS + SHARE KNOWLEDGE

DEFINE ALL CRITERIA IN COLLABORATION WITH TESTER

ANALYST

ROLES: ONE PERSON CAN PLAY MORE THAN ONE ROLE

ANALYST DESCRIBES REQUIREMENT AND BUSINESS CONTEXT + GO THROUGH ALL CRITERIA WITH DEVELOPER AND TESTER

DEVELOPER

TESTER

DESIGN TESTS

MAINTAINABLE ACCEPTANCE TEST SUITE

- LAYERS
- ATOMIC
  NO DEPENDENCIES BETWEEN TESTS. THE ORDER IN WHICH THEY EXECUTE DOES NOT MATTER.
- USE TEST STUBS
- OWNED BY DEVELOPERS & TESTERS

ACCEPTANCE CRITERIA
GIVEN...
WHEN...
THEN...

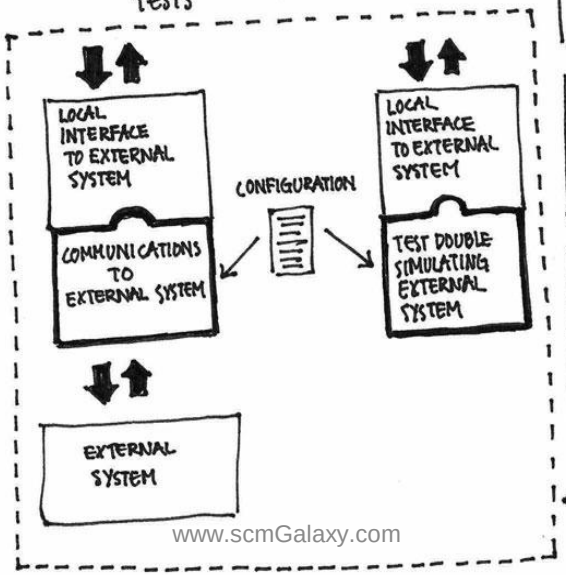
TEST IMPLEMENTATION
CODE USES DOMAIN LANGUAGE.
NO REF. TO UI ELEMENTS

APPLICATION DRIVER LAYER
UNDERSTANDS HOW TO INTERACT WITH THE APPLICATION TO PERFORM ACTIONS AND RETURN RESULTS.

EXTERNAL INTEGRATION POINTS - (INTEGRATION TEST STRATEGY)

1. CREATE SMALL NUMBER OF TESTS TO COVER OBVIOUS SCENARIOS.
2. WE WILL MISS PROBLEMS → WE WILL ADDRESS BREAKAGES AS WE FIND THEM BY WRITING TEST TO CATCH EACH CASE.

⚠ NOT A PERFECT STRATEGY, BUT TO ATTEMTING TO GET PERFECT COVERAGE IN SUCH SCENARIOS IS USUALLY VERY DIFFICULT AND THE RETURNS OF EFFORT VERSUS REWARD DIMINISH VERY QUICKLY.

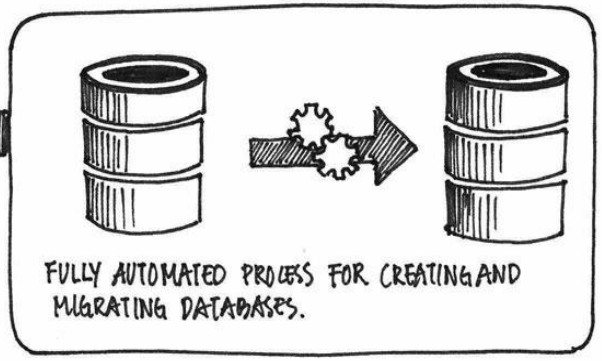


Nhan Ngo

What does

CONT. DEL. say about MANAGING DATA

Nhan Ngo



FULLY AUTOMATED PROCESS FOR CREATING AND MIGRATING DATABASES.

NO REAL DATA BASE

BENEFIT : (LAYERS)

FOCUS ON BUSINESS BEHAVIOUR

+

DATA ACCESS CODE KEPT TOGETHER

IN MEMORY DATABASE

- CONFIGURABLE (ALLOW YOU TO SWITCH TO ANYTHING SUITABLE)
- BENEFIT: DECOUPLED CODE

MANAGING TEST DATA

2 CONCERNS

- TEST PERFORMANCE
- TEST ISOLATION

MANAGING THE COUPLING BETWEEN TEST AND DATA

TEST ISOLATION

EACH TEST'S DATA IS ONLY VISIBLE FOR THAT TEST.

ADAPTIVE TEST

EACH TEST IS DESIGNED TO EVALUATE IT'S DATA ENVIRONMENT AND ADAPT ITS BEHAVIOUR TO SUIT THE DATA IT SEES.

CONSEQUENCE

MORE COMPLEX AND LARGER TESTS.

TEST SEQUENCING

TEST ARE DESIGNED TO RUN KNOWN SEQUENCES, EACH DEPENDING FOR INPUTS ON THE OUTPUTS OF IT'S PREDECESSORS.

CONSEQUENCE

FAIL CAUSING SUBSEQUENT TEST NOT TO BE RUN

IF YOU WANT TO TEST DIFFERENT VARIATIONS OF THIS TEST YOU ARE FORCED TO RUN THE PREDECESSORS

SETUP & TEAR DOWN

COMMIT STAGE

MUST RUN QUICKLY

- MINIMUM TEST DATA TO ASSERT THAT THE UNIT UNDER TEST EXHIBIT THE EXPECTED RESULT
- TEST NOT CLOSELY TIED TO IMPLEMENTATION. WILL OTHERWISE INHIBIT CHANGE.

+

AUTOMATED ACCEPTANCE TEST

3 TYPES OF DATA

- TEST SPECIFIC – TEST ISOLATION STRATEGY
- TEST REF. DATA – SUPPORTING CAST
- APPLICATION REF DATA – IRRELEVANT TO BEHAVIOUR UNDER TEST. NEEDS TO BE THERE FOR APPLICATION TO START UP.

+

CAPACITY TESTING

AMPLIFY TO GET THE LARGE SCALE

MANUAL TEST

SUBSET OF PRODUCTION DATA

Thanks