

SUCCESSFUL TESTING THE CONTINUOUS DELIVERY PROCESS

INTRODUCTION



Pascal Dufour
Agile Tester

@Pascal_Dufour



Harald Rietman
Developer
Scrum Master

@hrietman



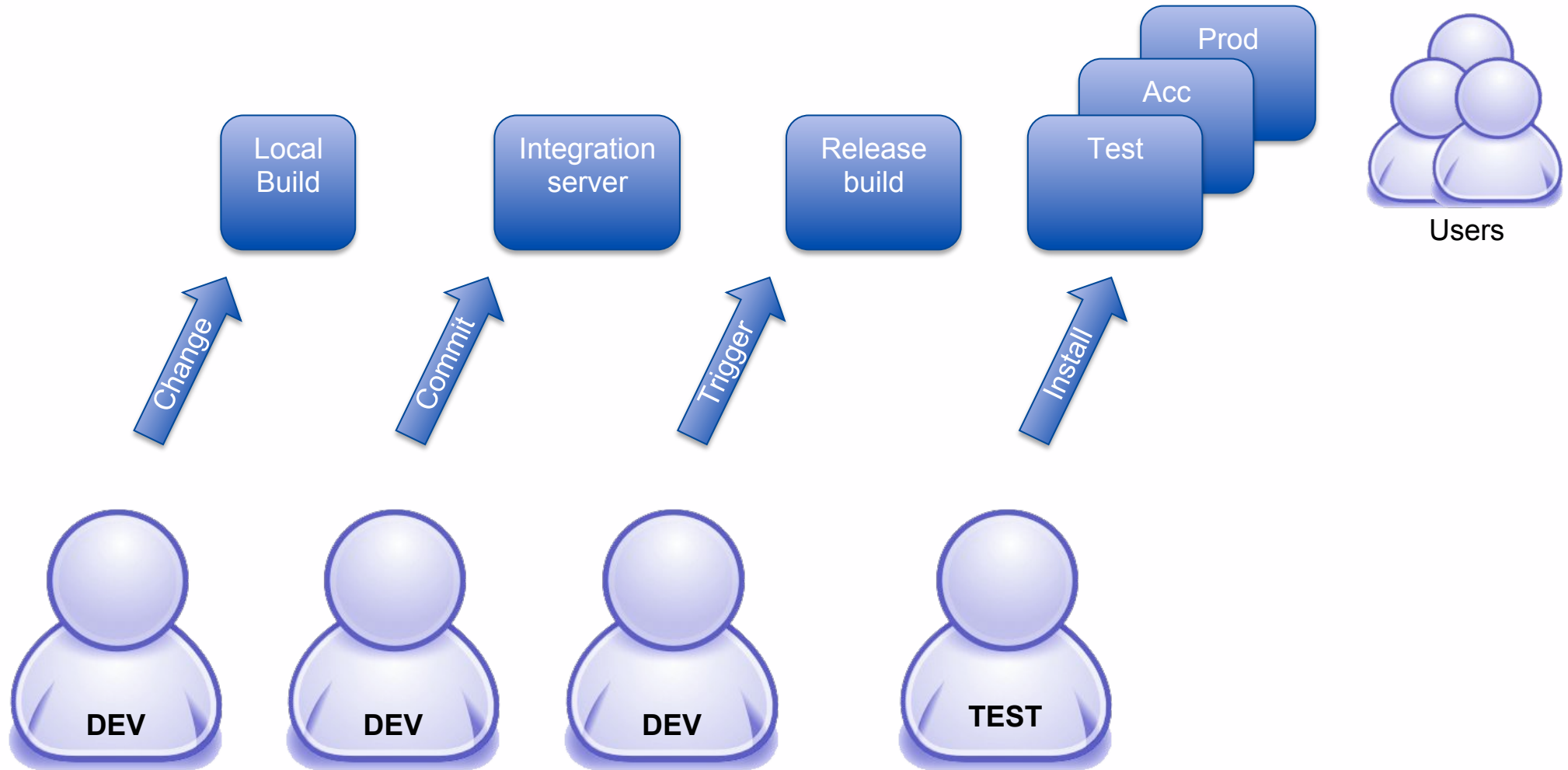
TYPICAL

- Experience with Continuous Delivery?
- As a tester, do you need to wait for an acceptable release from developers?
- Loose time doing all repetitive manual stuff?
- Maintain different environments and software versions manually, over-and-over-and-over...
- Is this really release 1.6.5?

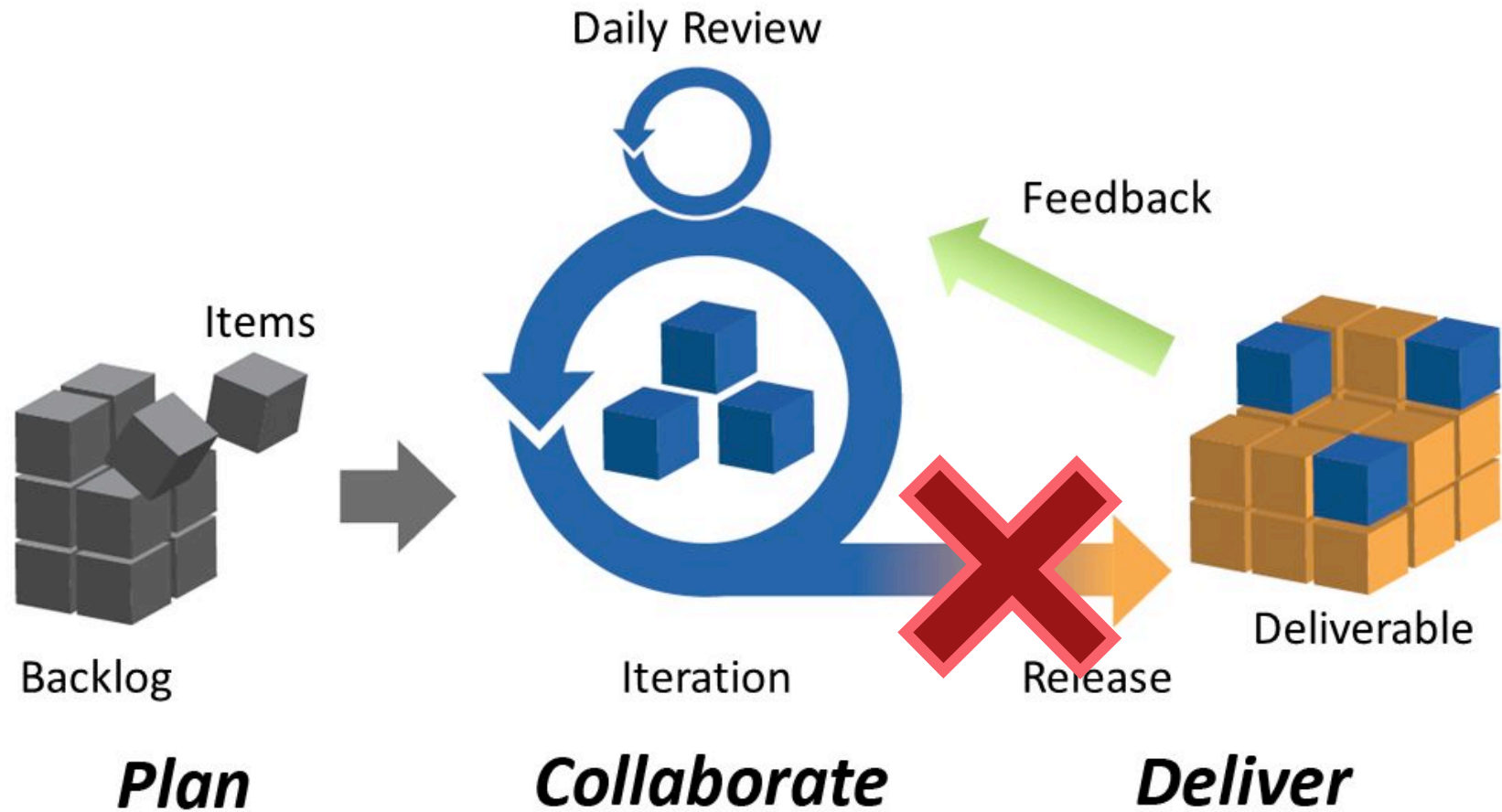
- All sounds familiar?



BUILDING A RELEASE



WHY CONTINUOUS DELIVERY



Agile Project Management: Iteration

GOALS

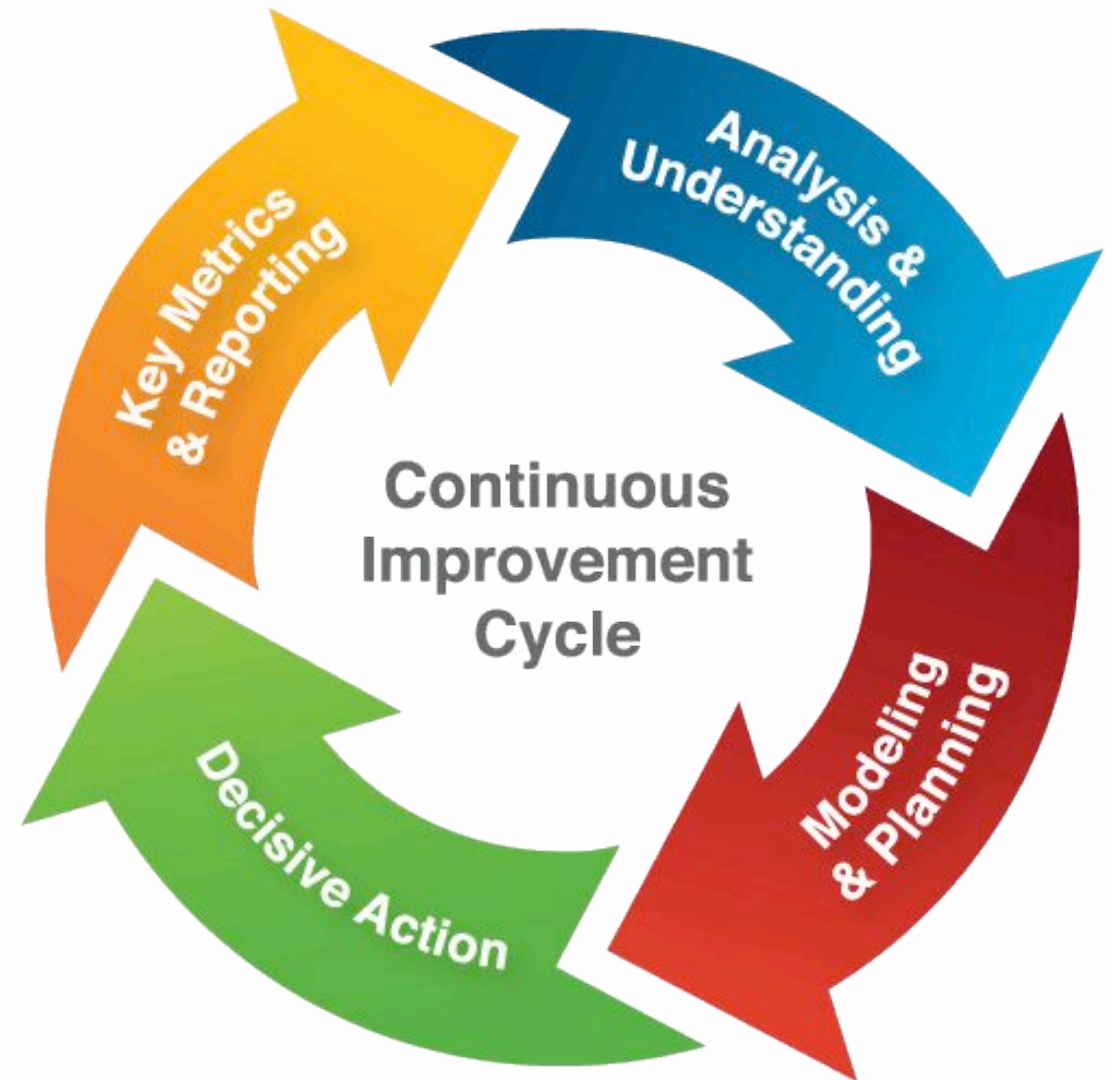
- Predictability
- Quality
- Speed



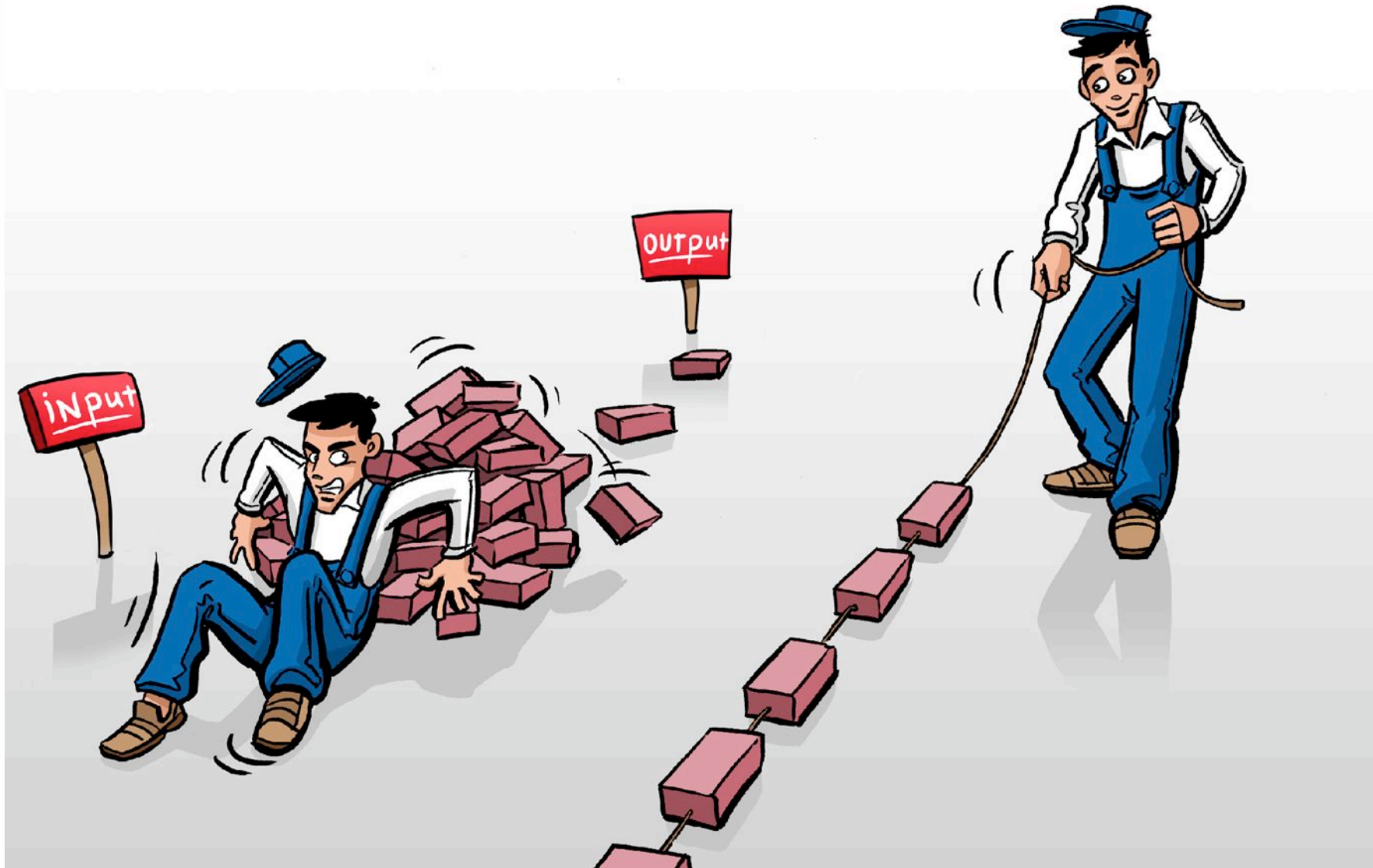
CONTINUOUS IMPROVEMENT CYCLE

Requires:

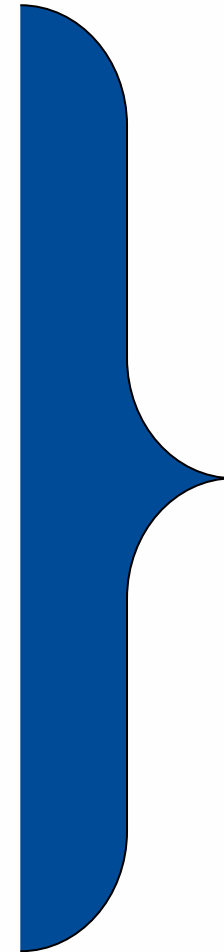
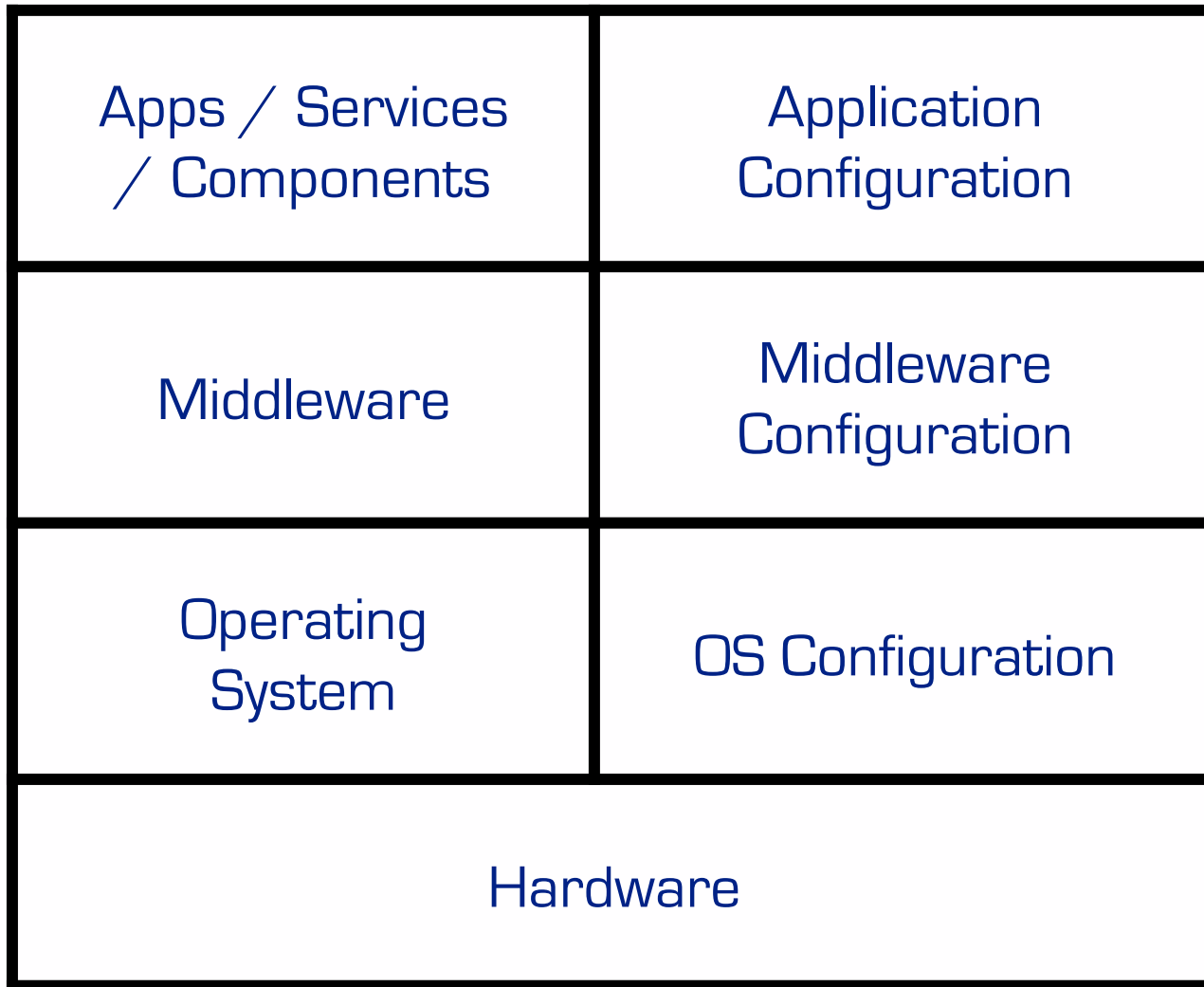
Discipline &
Responsibility



PUSH VS. PULL



APPLICATION STACK



TESTING

STOP THE LINE!



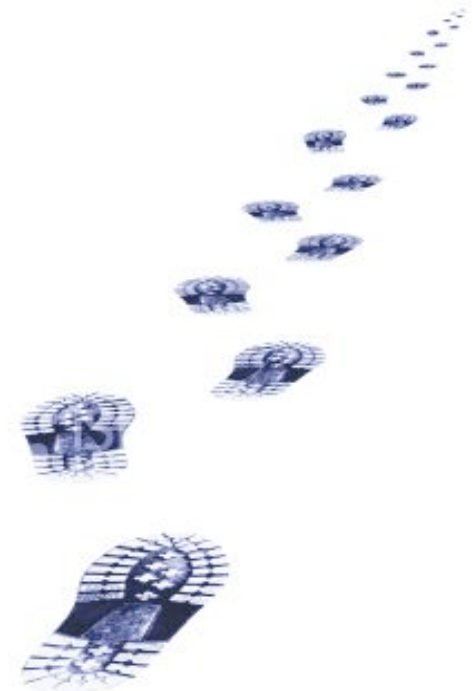
WHAT IS CONTINUOUS DELIVERY



- Continuous Delivery pipeline
- Automation
- Manual control
- Feedback

WHAT ARE THE MAIN STEPS OF CONTINUOUS DELIVERY?

- Build – compile, unit test, version, package
- Quality – metrics, documentation
- Test – acceptance-, regression- and performance tests
- Provision environments – deployment to test- and staging environment
- Production – green/blue deployment to production

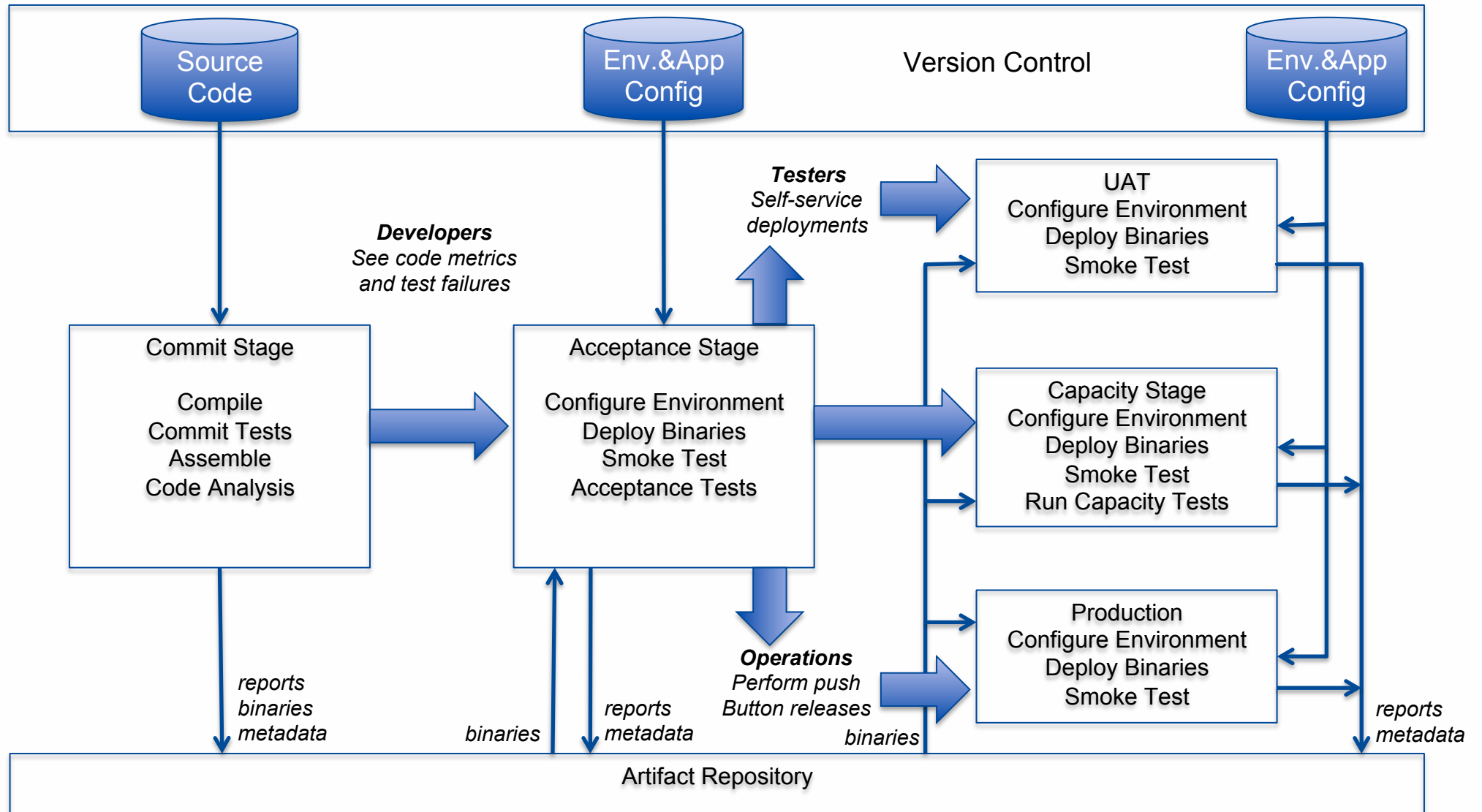


WHAT ARE THE KEY BENEFITS OF CONTINUOUS DELIVERY?



- Low-risk releases
- Faster return on investment in software projects
- Improvement of competitiveness and responsiveness
- Quality improvement of new software versions

ANATOMY IN DETAIL



MAIN STEPS TO CREATE A CONTINUOUS DELIVERY PIPELINE?

- Getting started with virtual environments, e.g. Amazon EC2
- Configure your Continuous Integration Server
- Provisioning your Test, Staging and Production environments
- Configure your Continuous Delivery Pipeline
- Create a Dashboard of your Systems

EXAMPLE



Continuous Delivery :: Production Step :: select version and environment for deployment

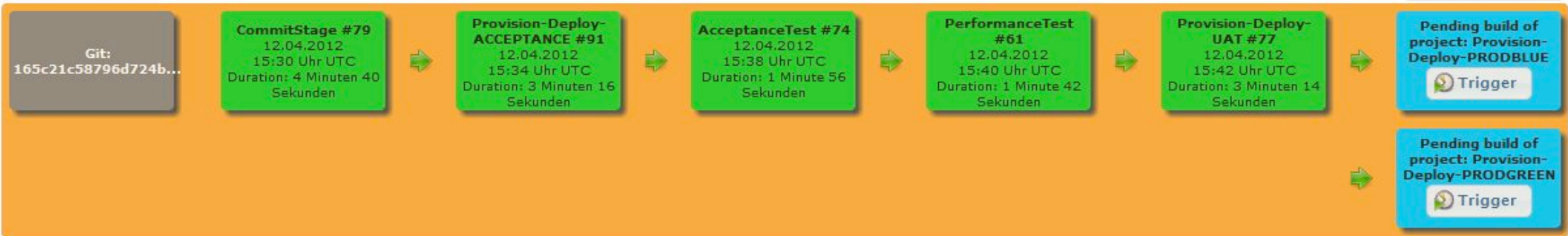
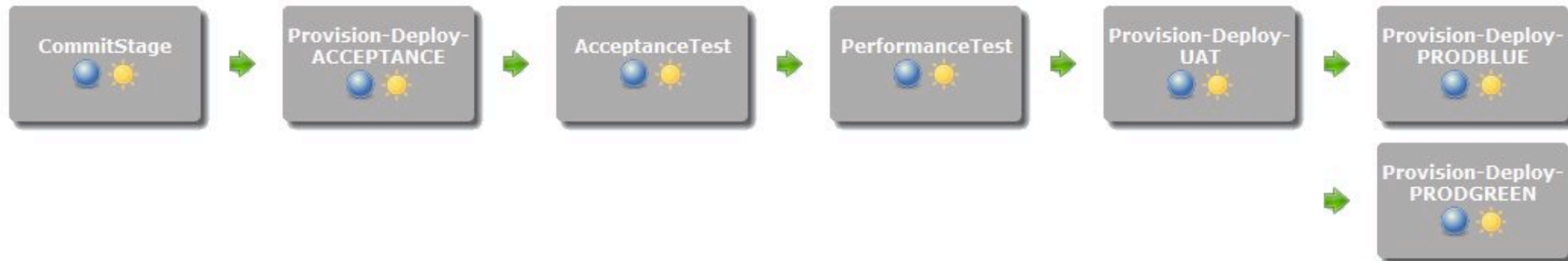
Available Releases

Deploy VERSION to ENVIRONMENT

Continuous Delivery :: System Status :: displays server status and versions deployed

Acceptance/Performance Test	UAT Environment	Production Green	Production Blue
Version: 1.4.237	Version: 1.4.236	Version: 1.4.174	Version: 1.4.225
Go to ACCEPTANCE	UAT-ENVIRONMENT	not running	PRODUCTION-BLUE
Status: running AWS Instance ID: i-44b9bc20 Instance-Type: m1.small Time started: 2012-04-12T10:25:14+0000 AWS-TAG: ACCEPTANCE	Status: running AWS Instance ID: i-abd722cc Instance-Type: t1.micro Time started: 2012-06-22T07:39:53+0000 AWS-TAG: UAT-ENVIRONMENT	Status: not running AWS Instance ID: i-375eab50 Instance-Type: t1.micro Time started: AWS-TAG: PRODUCTION-GREEN	Status: running AWS Instance ID: i-40181224 Instance-Type: t1.micro Time started: 2012-06-22T08:09:47+0000 AWS-TAG: PRODUCTION-BLUE

WHAT ARE THE MAIN STEPS OF CONTINUOUS DELIVERY?



TOOLING THAT CAN HELP YOU

maven



Cucumber

Jenkins



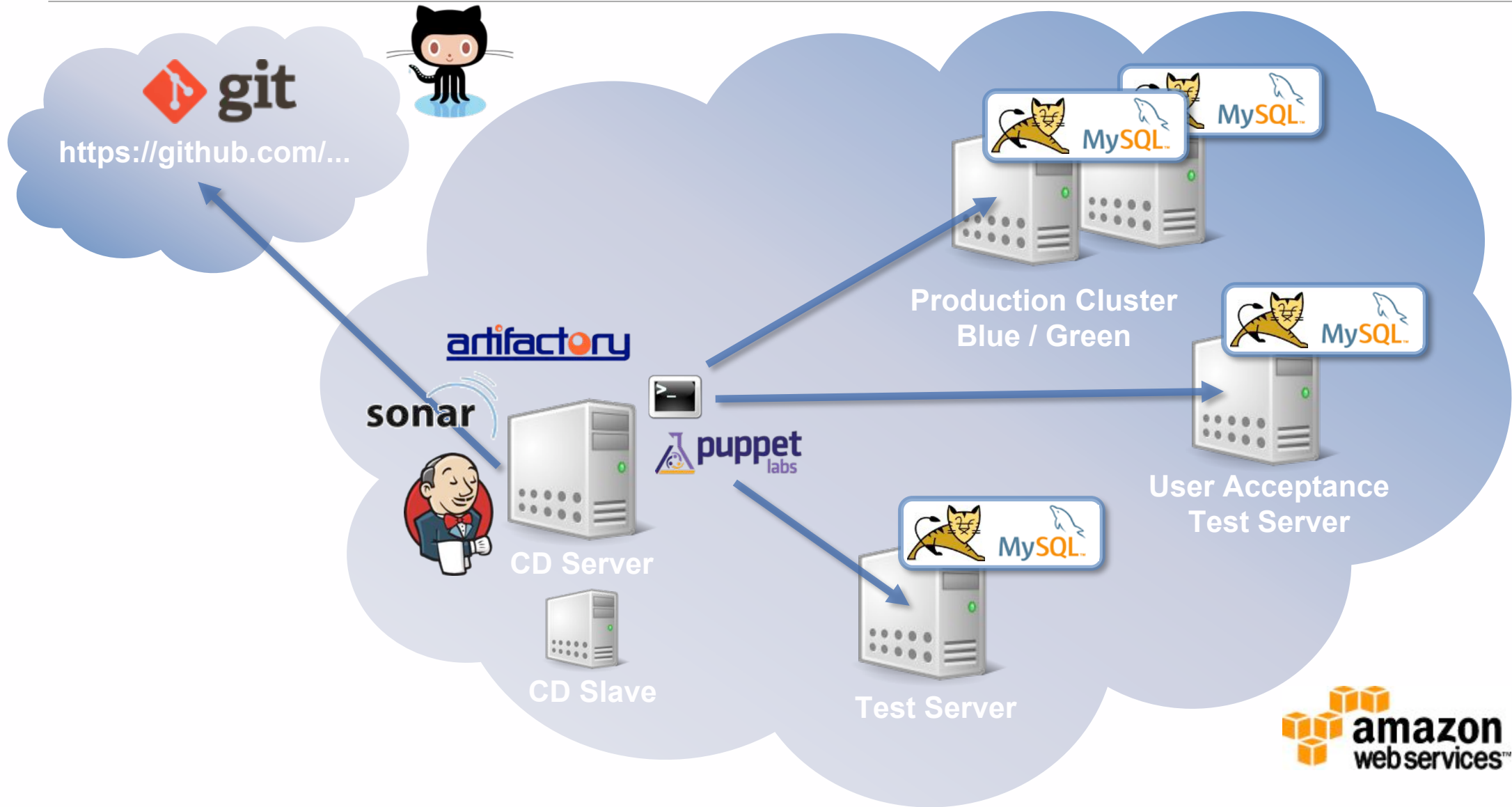
vmware®



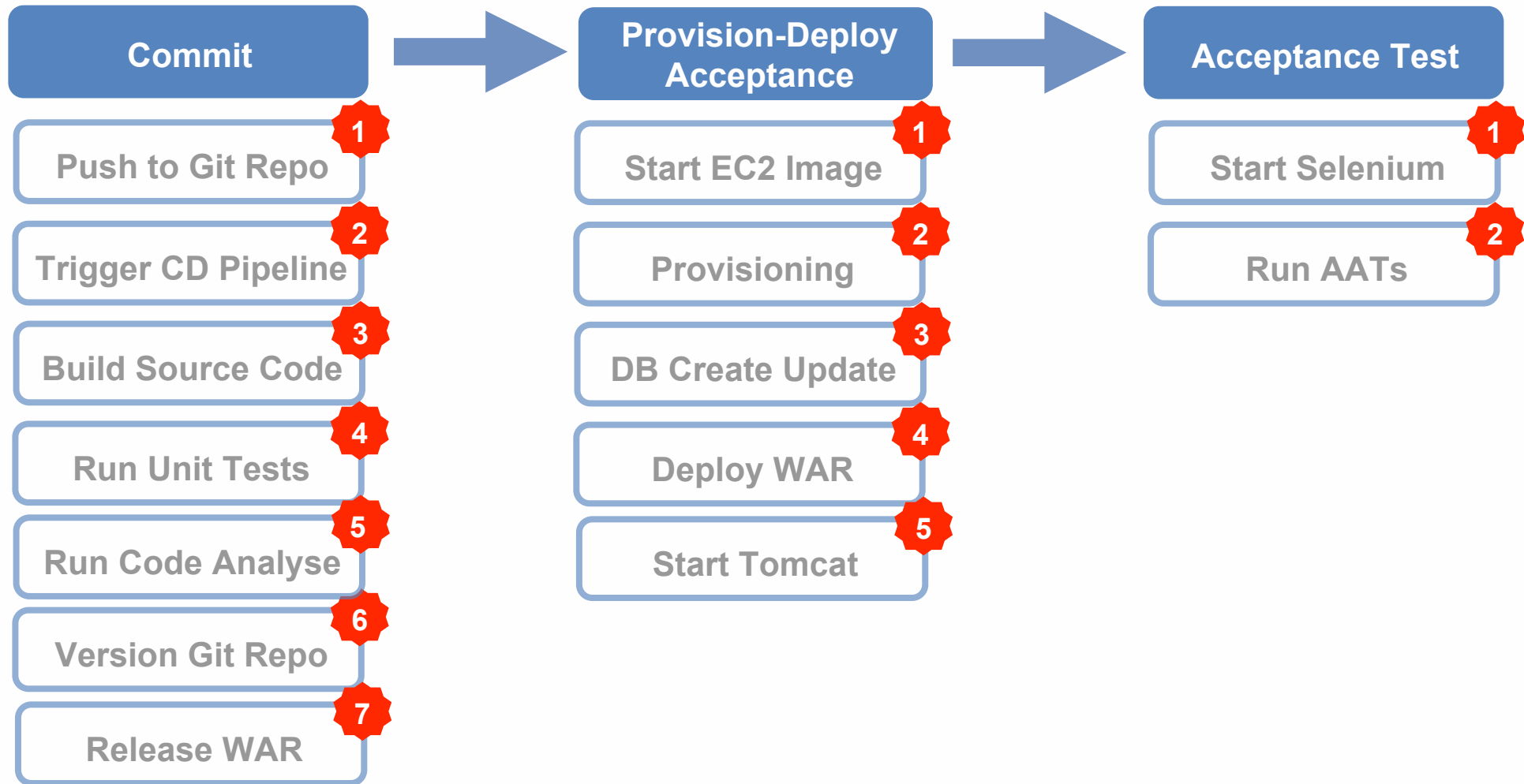
EXAMPLE – TECHNOLOGY STACK

- Amazon EC2 (Virtualization, Cloud)
- Jenkins (CI Server)
- Git/Github (Version Control)
- Sonar (Code Quality)
- jUnit (Unit Tests)
- jBehave (Acceptancetests)
- Selenium (UI Tests)
- Puppet (Provisioning)
- Tomcat (Application Server)
- Artifactory (Artifact Repository)
- Maven (Build Tool)
- jMeter (Performance Tests)
- AppDynamics (Performance Tests)
- Liquibase (DB Versioning)

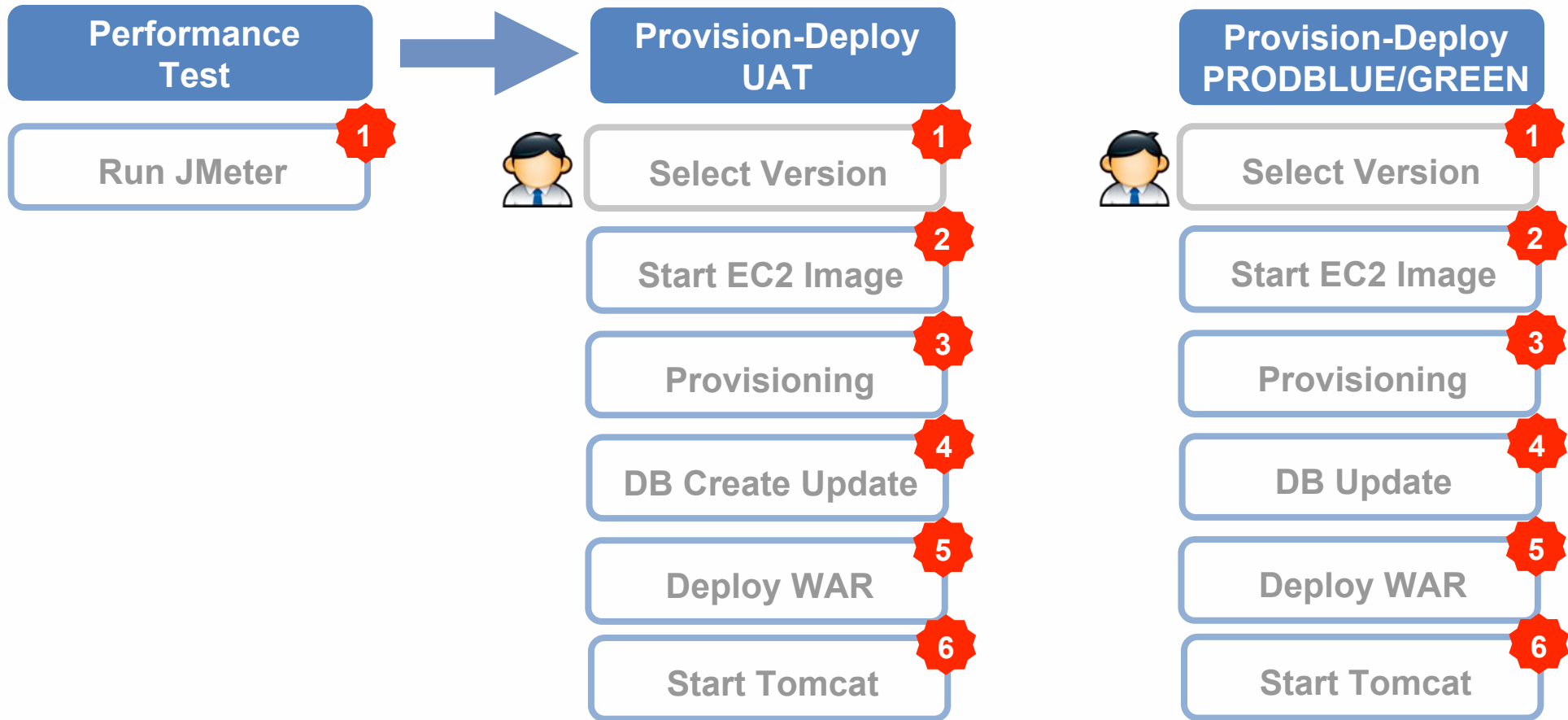
OUR SETUP



EXAMPLE – DELIVERY PIPELINE 1/2

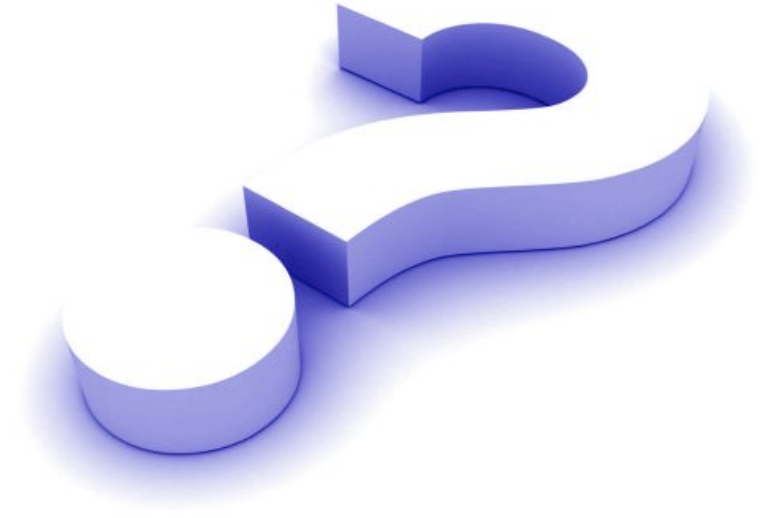


EXAMPLE – DELIVERY PIPELINE 2/2



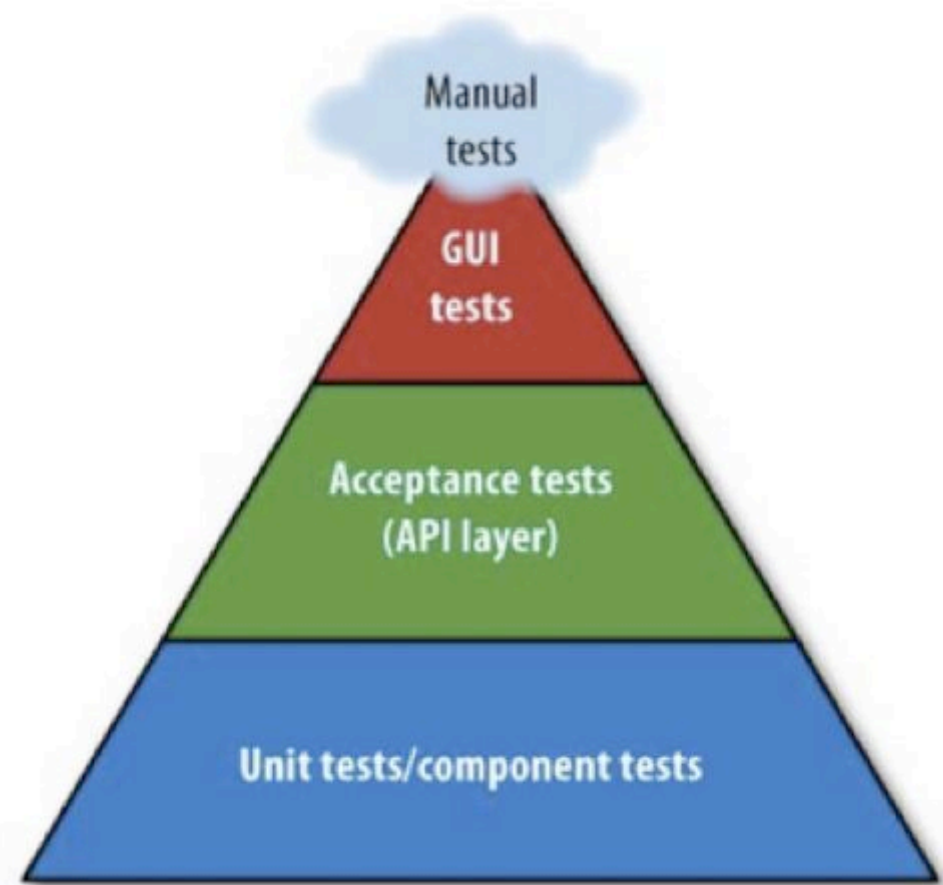
ASK YOURSELF

- Which parts are tested how?
- What remains to be tested?



WHAT DO WE TEST WHERE AND WHEN?

- Automation is key
- Do what we do best
- Team effort



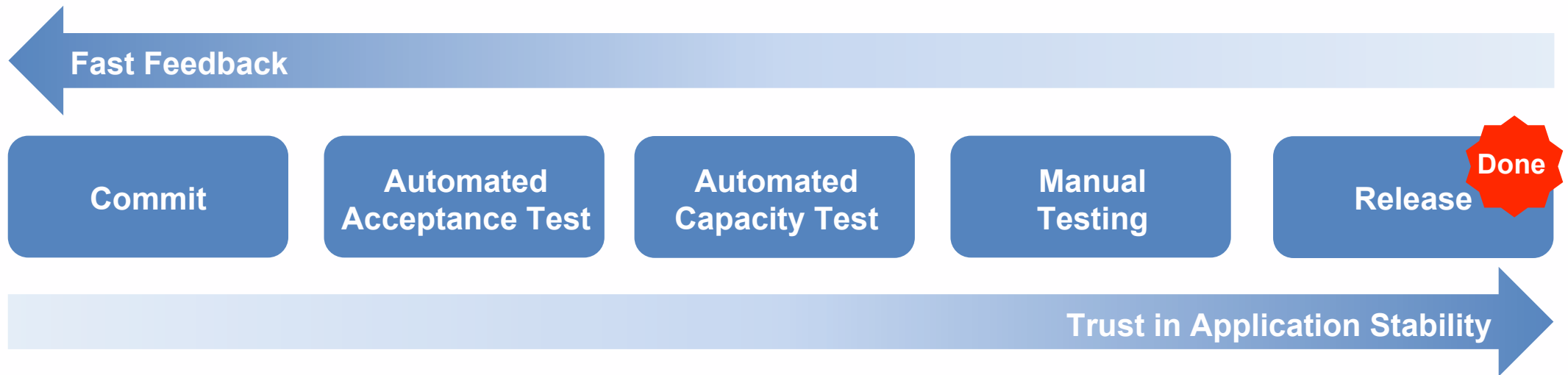
MANUAL TESTING

Still required:

- Smoke testing
- Exploratory testing
- User acceptance testing



ANATOMY OF A DEPLOYMENT PIPELINE



- Every change results in a trigger of the deployment process
- Software is build once and only once
- The same deployment process for every environment
- Deployment in production-like environments

Available Releases

Deploy VERSION to ENVIRONMENT

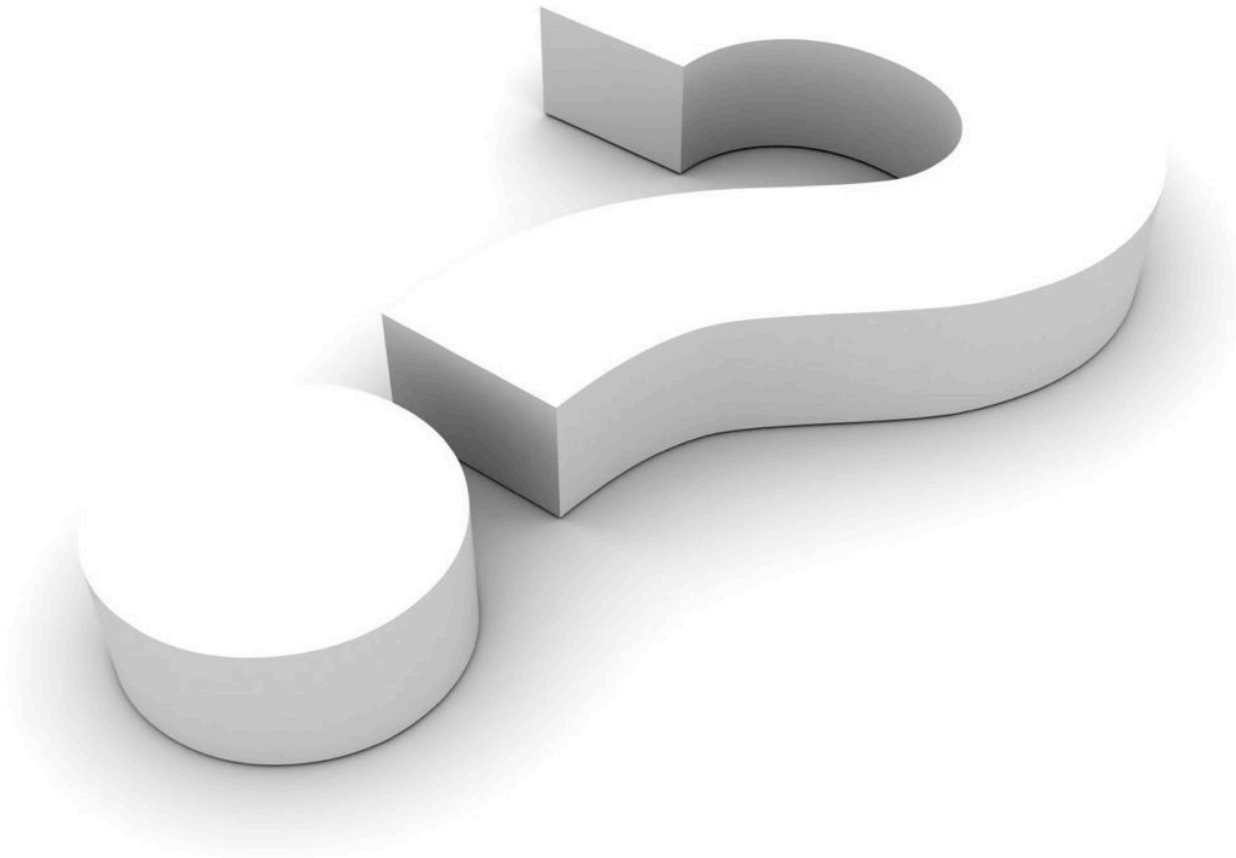
- Version 1.4.237
- Version 1.4.236
- Version 1.4.235
- Version 1.4.234
- Version 1.4.233
- Version 1.4.232
- Version 1.4.231
- Version 1.4.230
- Version 1.4.228
- Version 1.4.227
- Version 1.4.225
- Version 1.4.224
- Version 1.4.223
- Version 1.4.222
- Version 1.4.219
- Version 1.4.218
- Version 1.4.217
- Version 1.4.216
- Version 1.4.215
- Version 1.4.214

Acceptance/Performance Test	UAT-ENVIRONMENT	Production Green	Production Blue
Version: 1.4.237	Version: 1.4.237	Version: 1.4.174	Version: 1.4.225
Go to ACCEPTANCE	UAT-ENVIRONMENT	not running	PRODUCTION-BLUE
Status: running AWS Instance ID: i-44b9bc20 Instance-Type: m1.small Time started: 2012-04-12T10:25:14+0000 AWS-TAG: ACCEPTANCE	Status: running AWS Instance ID: i-44b9bc20 Instance-Type: m1.small Time started: 2012-06-22T07:39:53+0000 AWS-TAG: UAT-ENVIRONMENT	Status: not running AWS Instance ID: i-375eab50 Instance-Type: t1.micro Time started: AWS-TAG: PRODUCTION-GREEN	Status: running AWS Instance ID: i-40181224 Instance-Type: t1.micro Time started: 2012-06-22T08:09:47+0000 AWS-TAG: PRODUCTION-BLUE
			

SUMMARY

- Continuous delivery is a way to create a stable release process
- Manual testing is (still) necessary
- Pull vs Push
- Dashboard

THANK YOU FOR YOUR ATTENTION – ANY QUESTIONS?



CONTACT INFORMATION

Pascal.Dufour@codecentric.nl

+31 (0) 6 819 288 98

@pascal_dufour

Harald.Rietman@codecentric.nl

+31 (0) 6 549 633 47

@hrietman



See our website for more information:

<http://www.codecentric.nl/portfolio/continuous-delivery/>