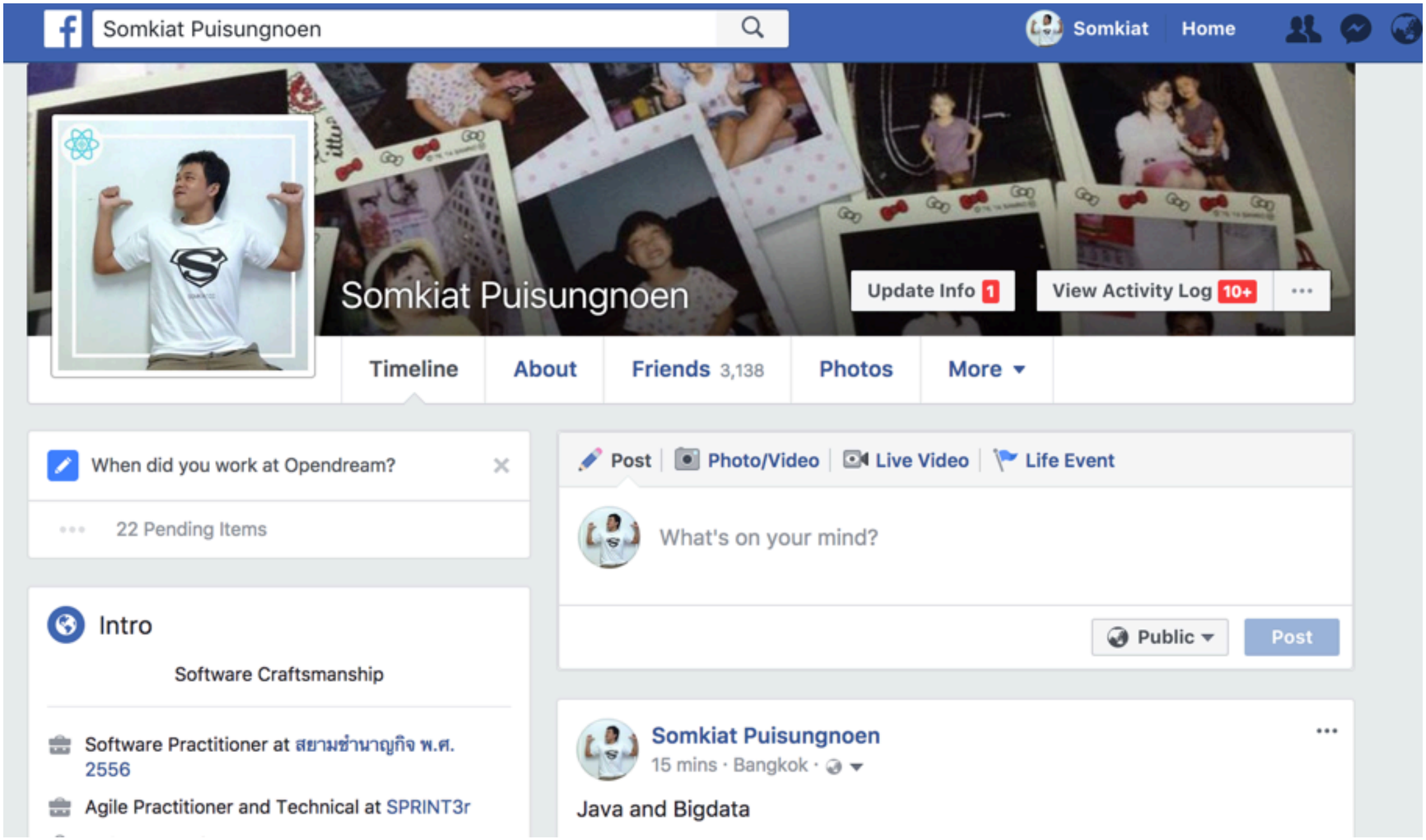
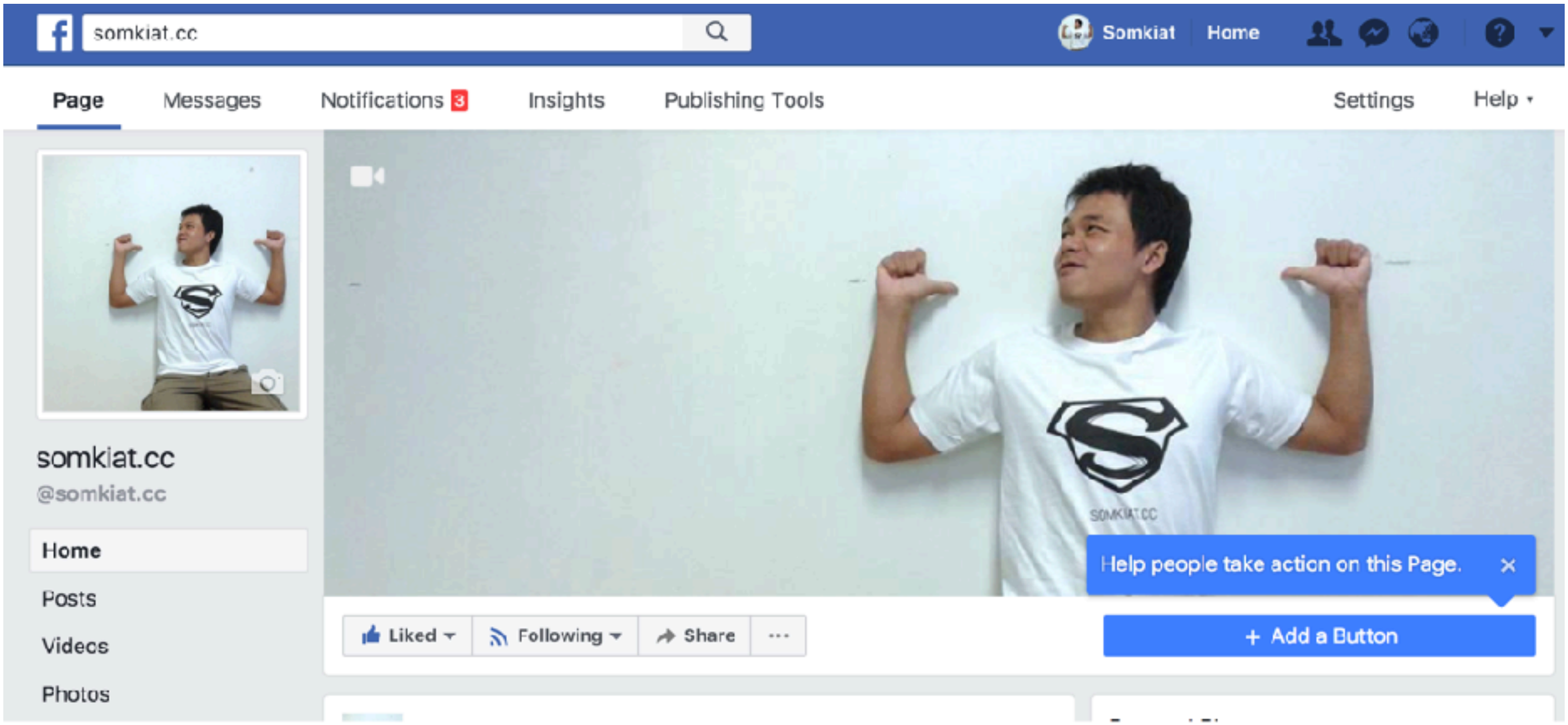
CI/CD series

CI/CD with Jenkins





Agenda Day 1

1. Concept of Continuous Integration
2. Concept of Continuous Delivery/Deployment
3. Practices of Continuous Integration
4. Build pipeline
5. Build your CI/CD system with Jenkins
6. Installation and Configuration Jenkins
7. Build your pipeline
8. Workshop

Agenda Day 2

1. Working with automated testing
2. Working with code quality
3. Automated deployment with Jenkins
4. Pipeline as a Code with Jenkins
5. Backup and Restore
6. Scalable Jenkins with Master/slave
7. Workshop

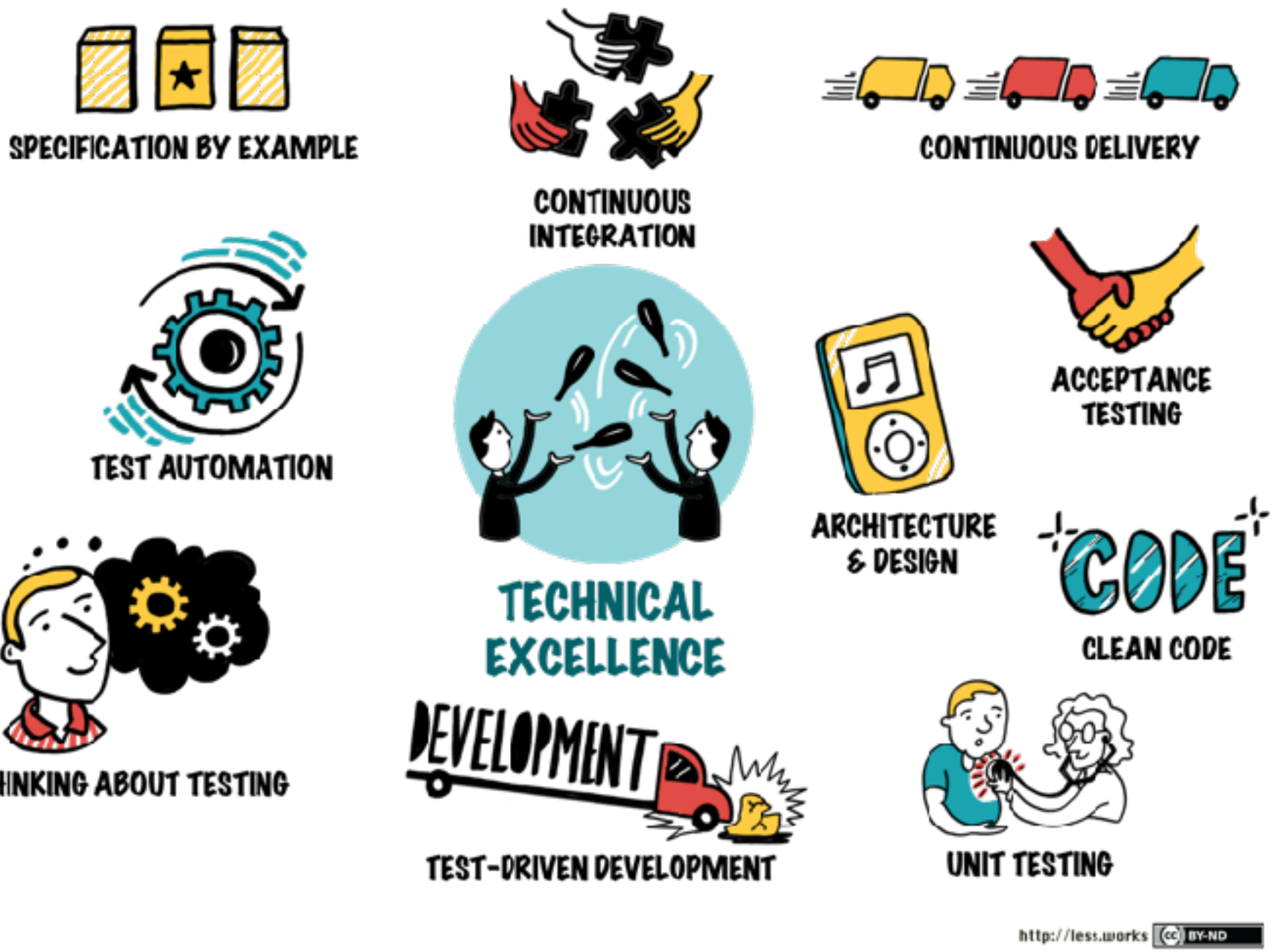
https://github.com/up1/course-ci-cd-with-jenkins

Software development

Quality vs. Quantity

Continuous Integration ?

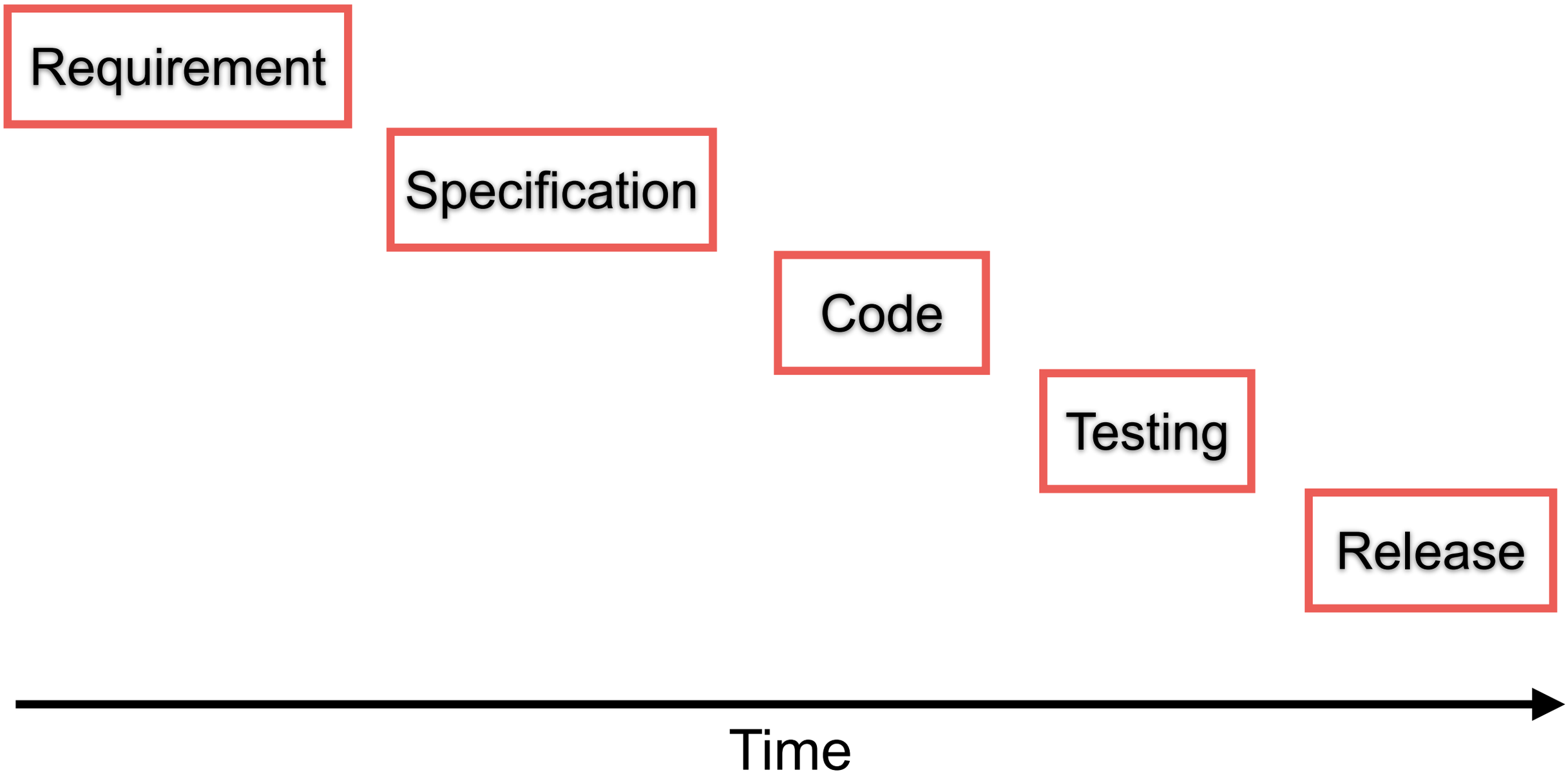
Technical Excellence



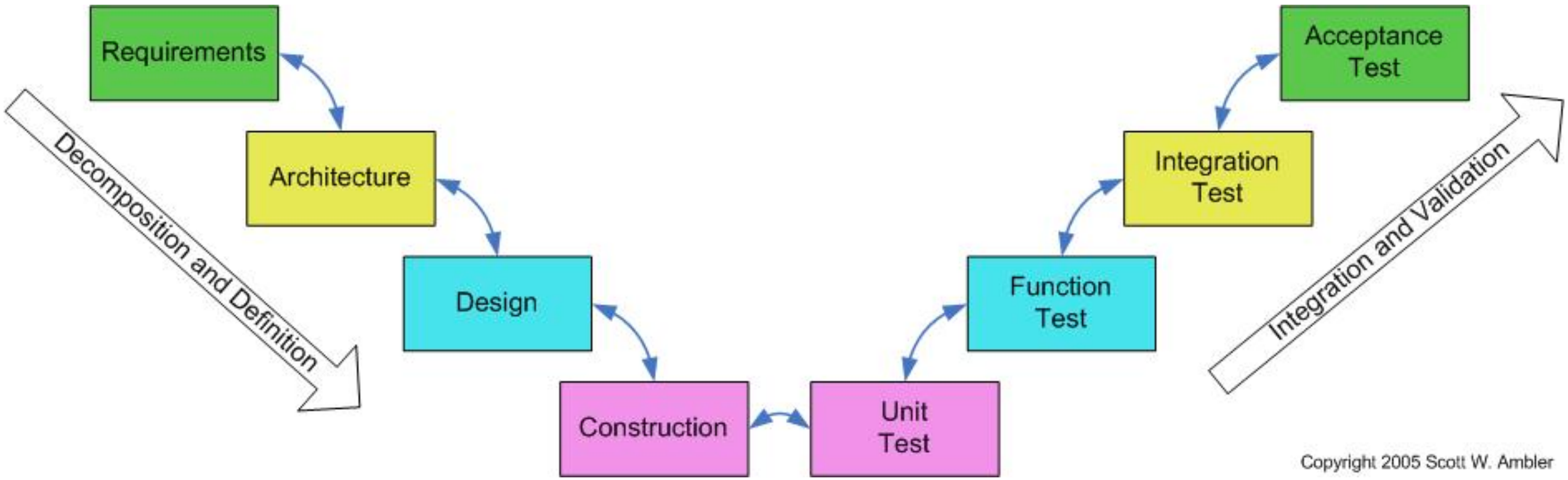
https://less.works/less/technical-excellence

Why CI/CD ?

Software Delivery Process



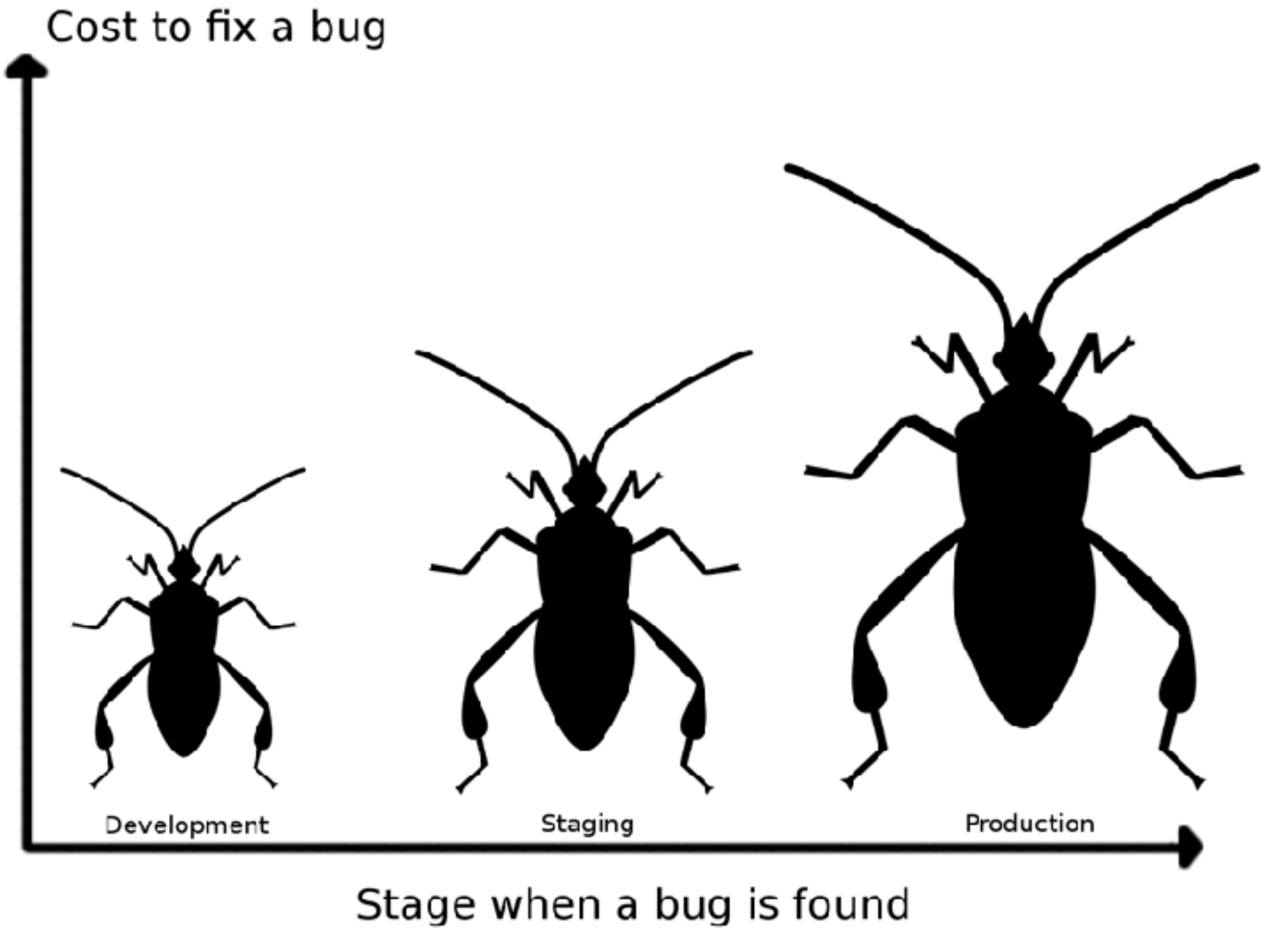
V Model



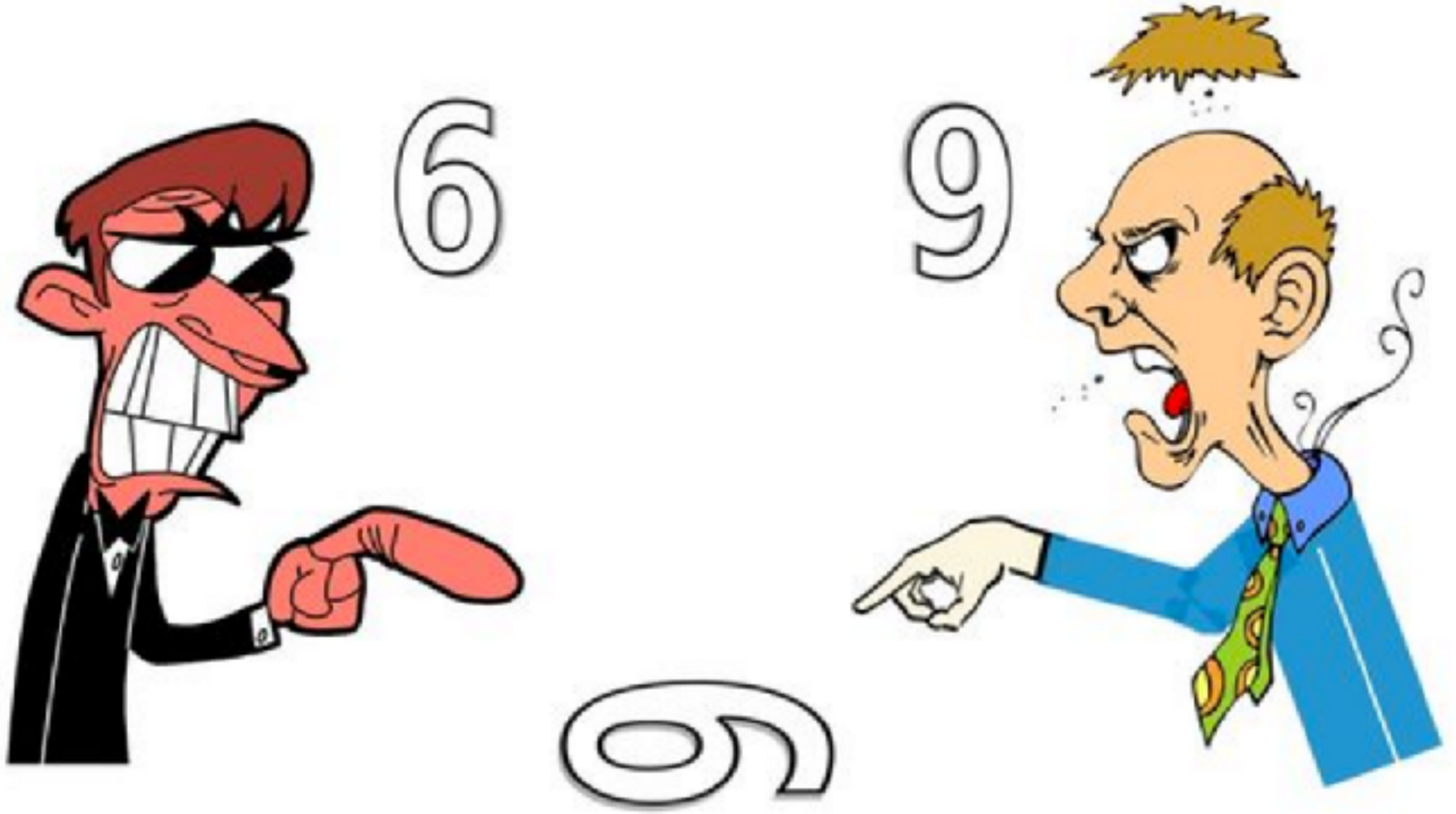
The cost of integration

1. Merging the code
2. Duplicate changes
3. Test again again !!
4. Fixing bugs
5. Impact on stability

The cost of integration



Developer vs Tester



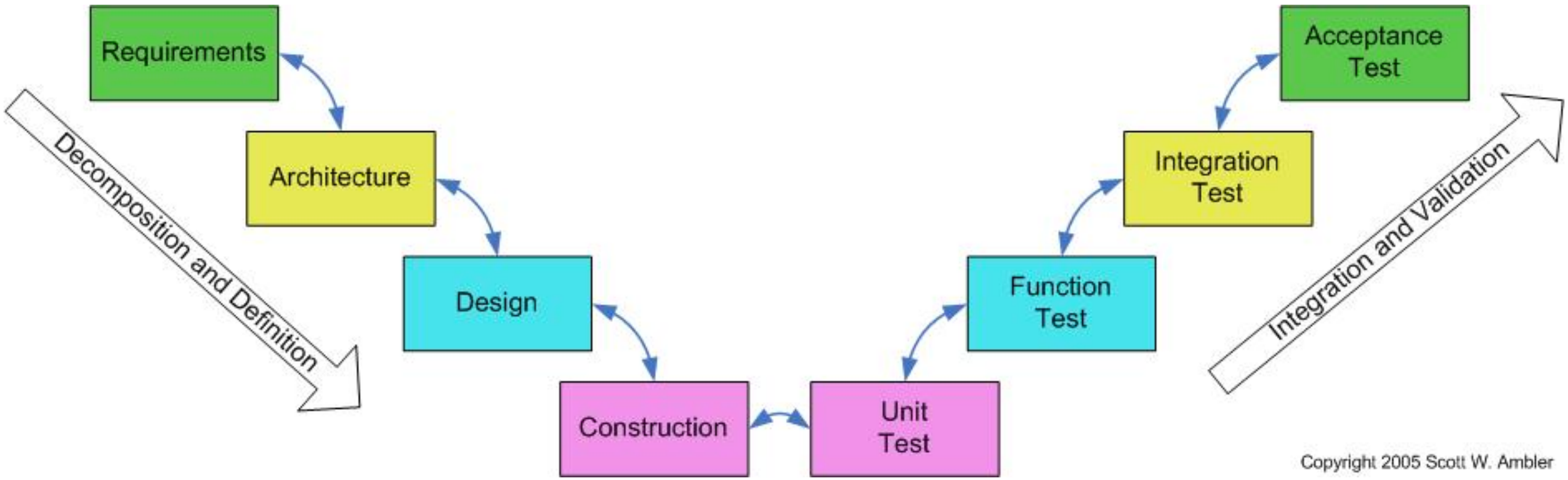
Testing process ?

Test first

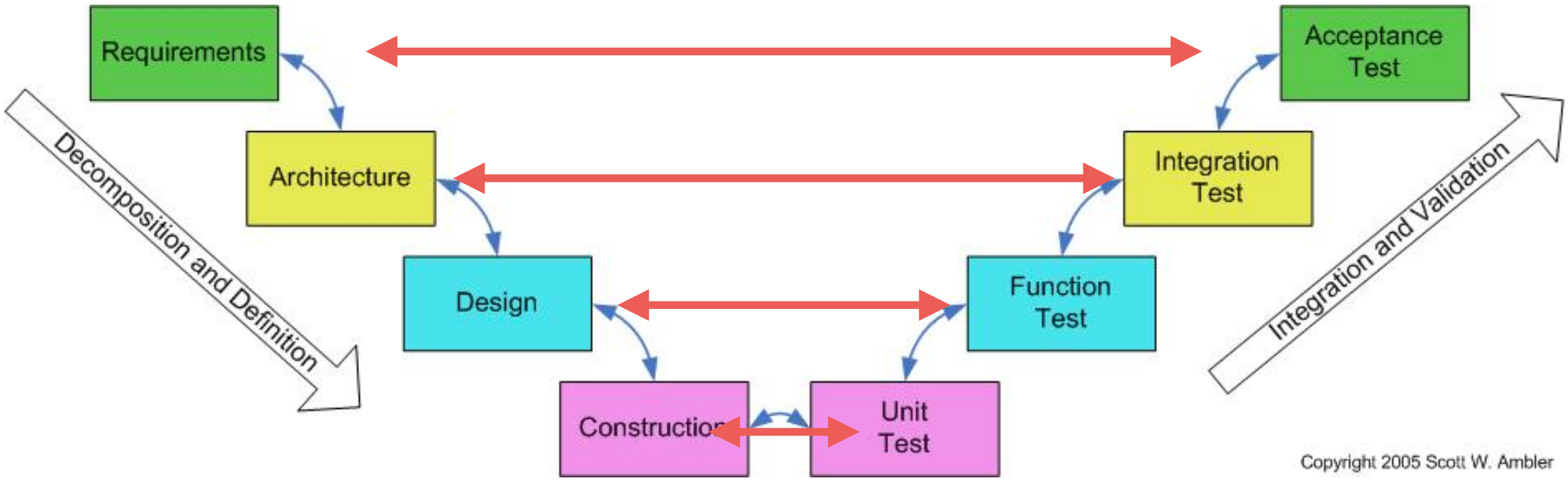
Test last

Test later

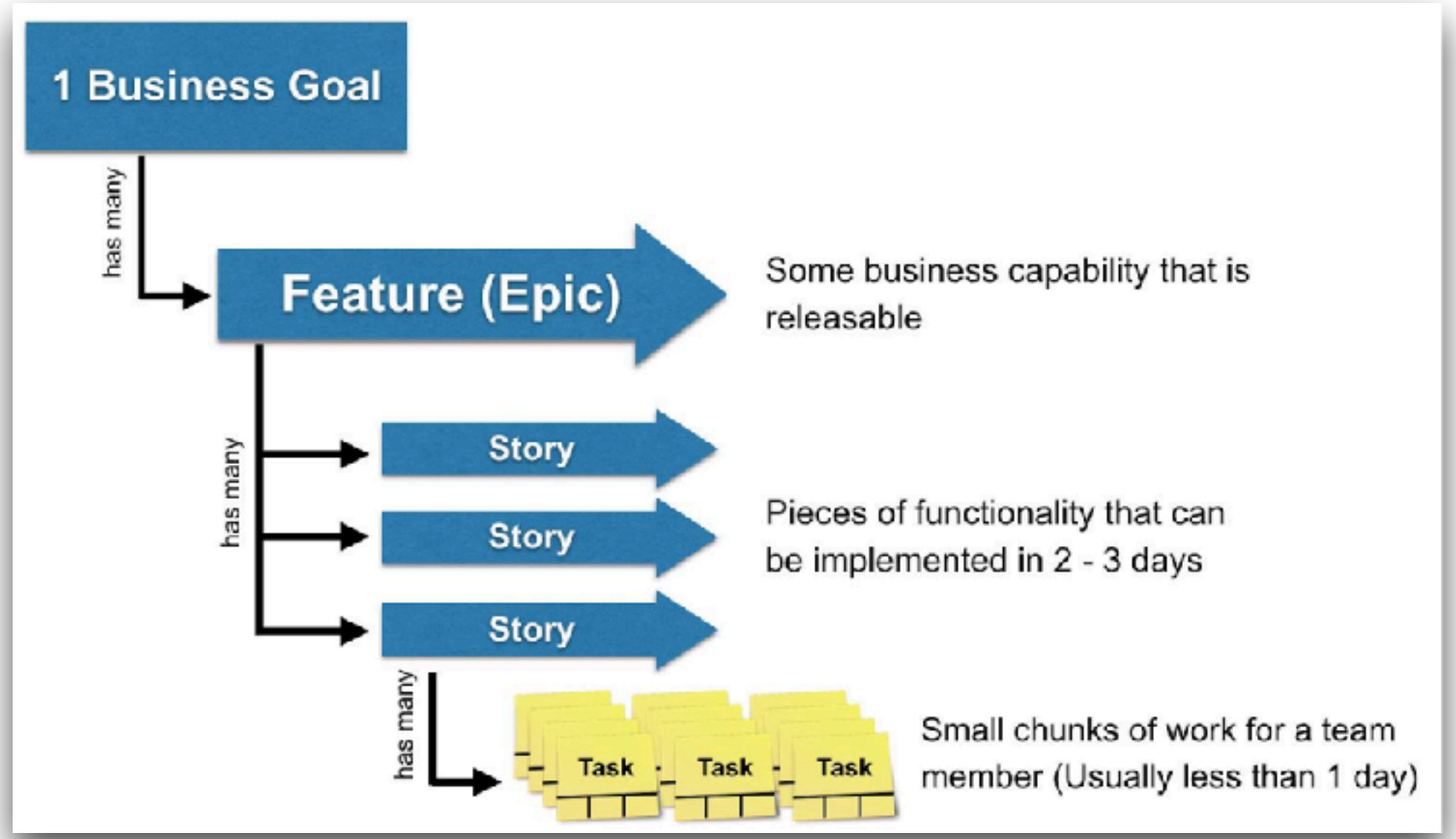
V Model



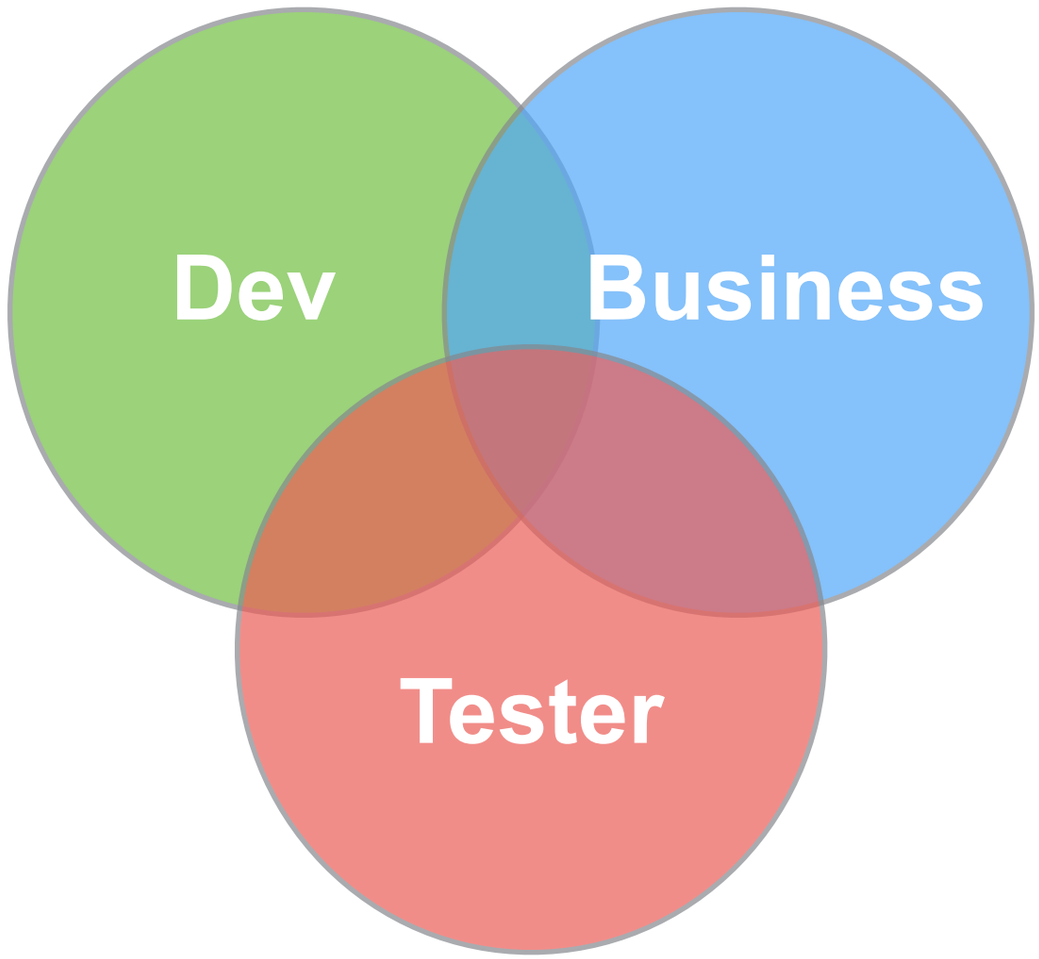
V Model



Start with business goal to tasks



Power of Three



Test case

=

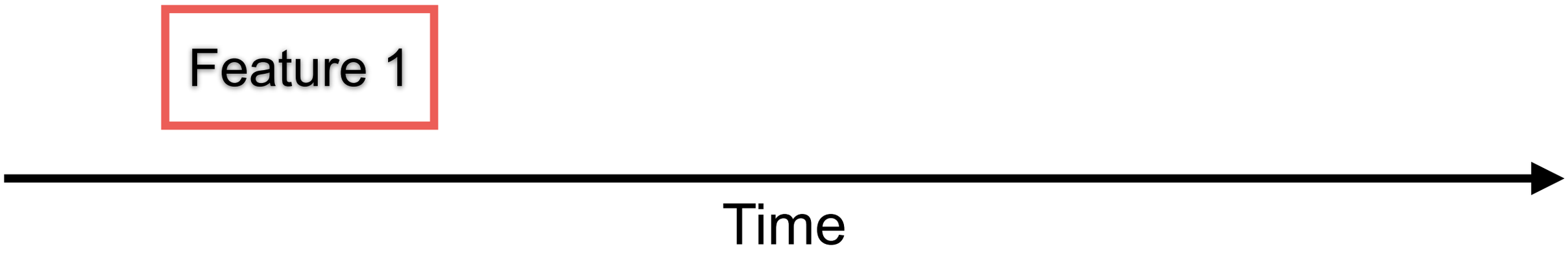
Business Criteria

+

Examples (real data)

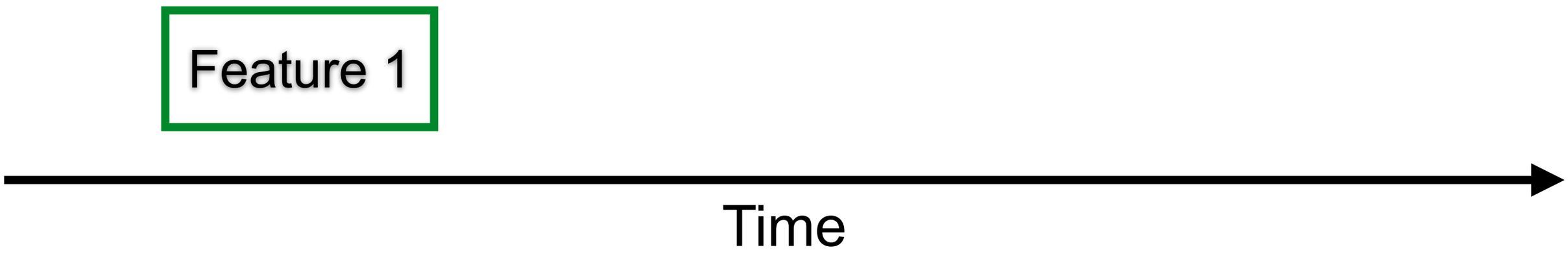
Quality process with Test ?

Iterative and incremental process



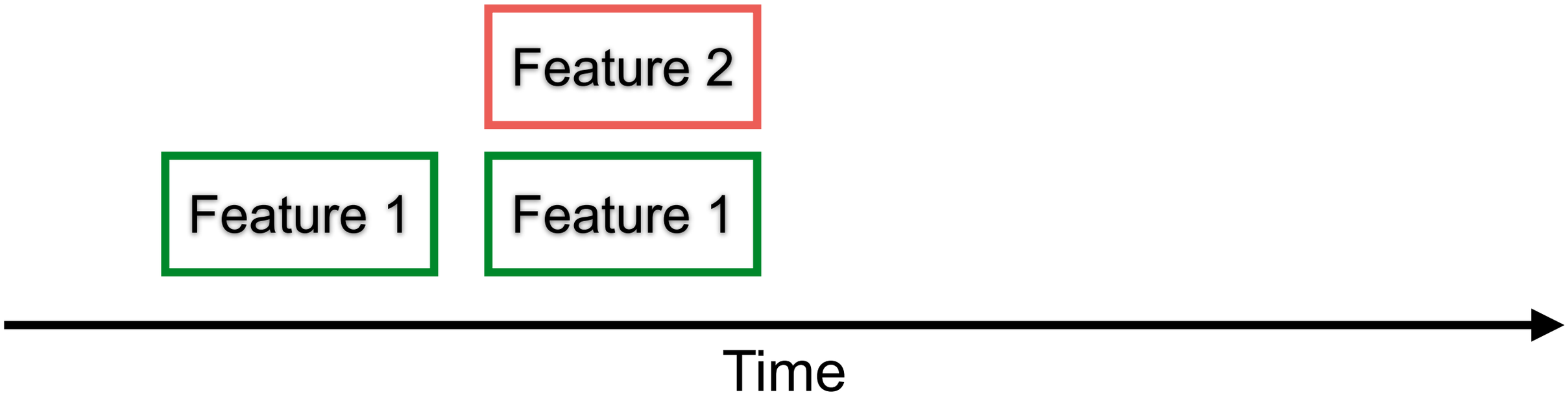
Iterative and incremental process

Done = coded and tested



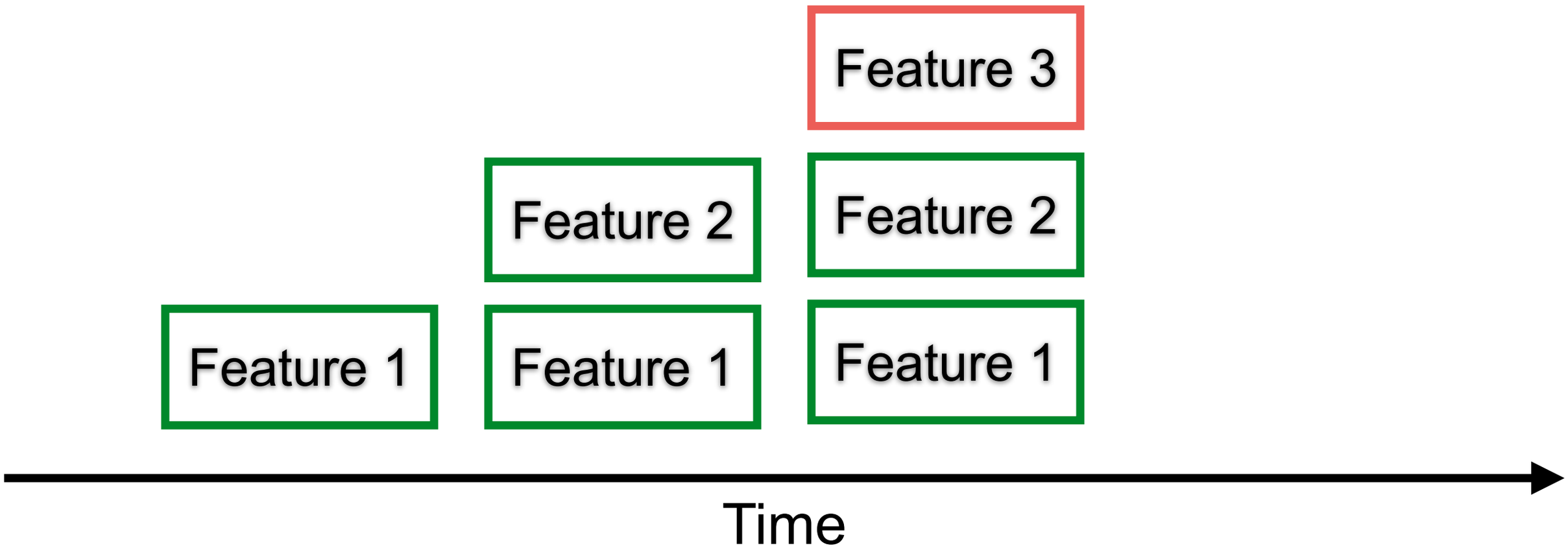
Iterative and incremental process

Done = coded and tested



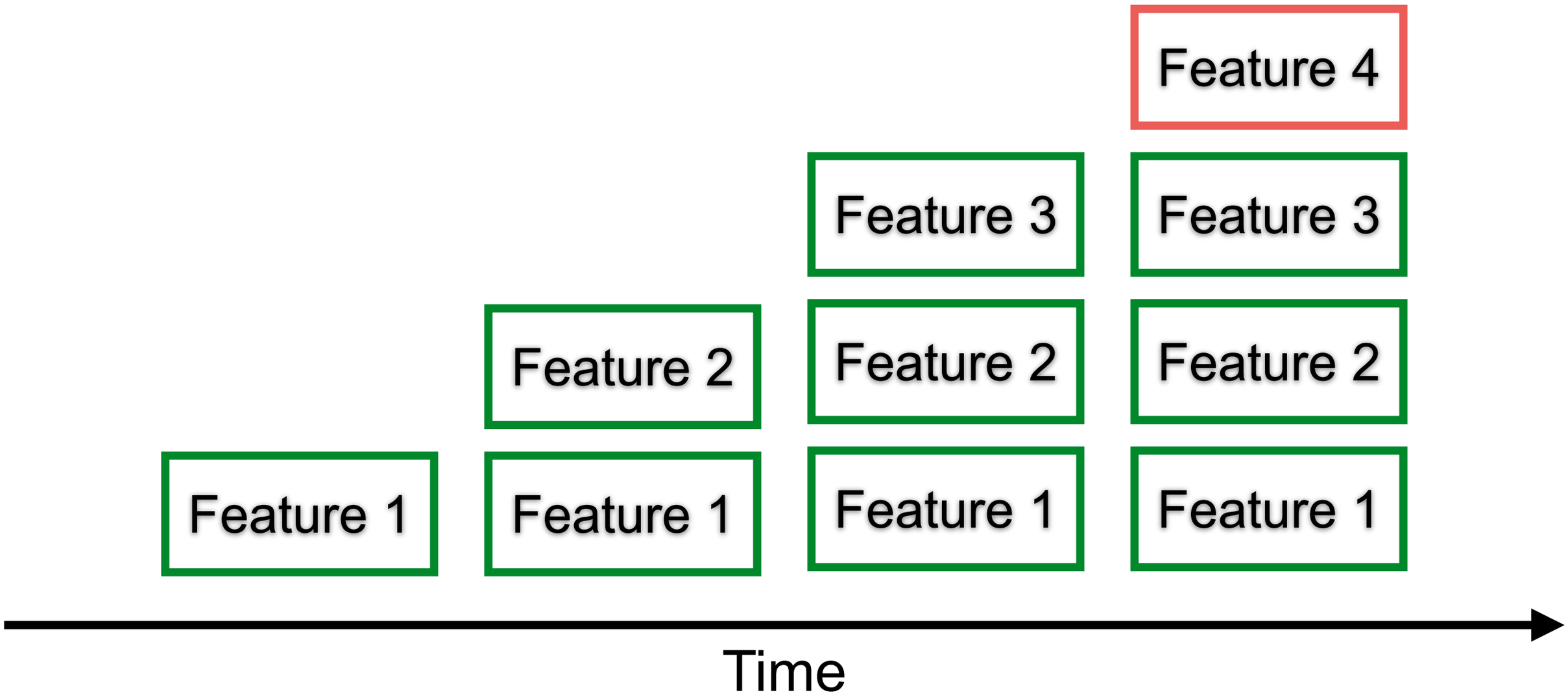
Iterative and incremental process

Done = coded and tested



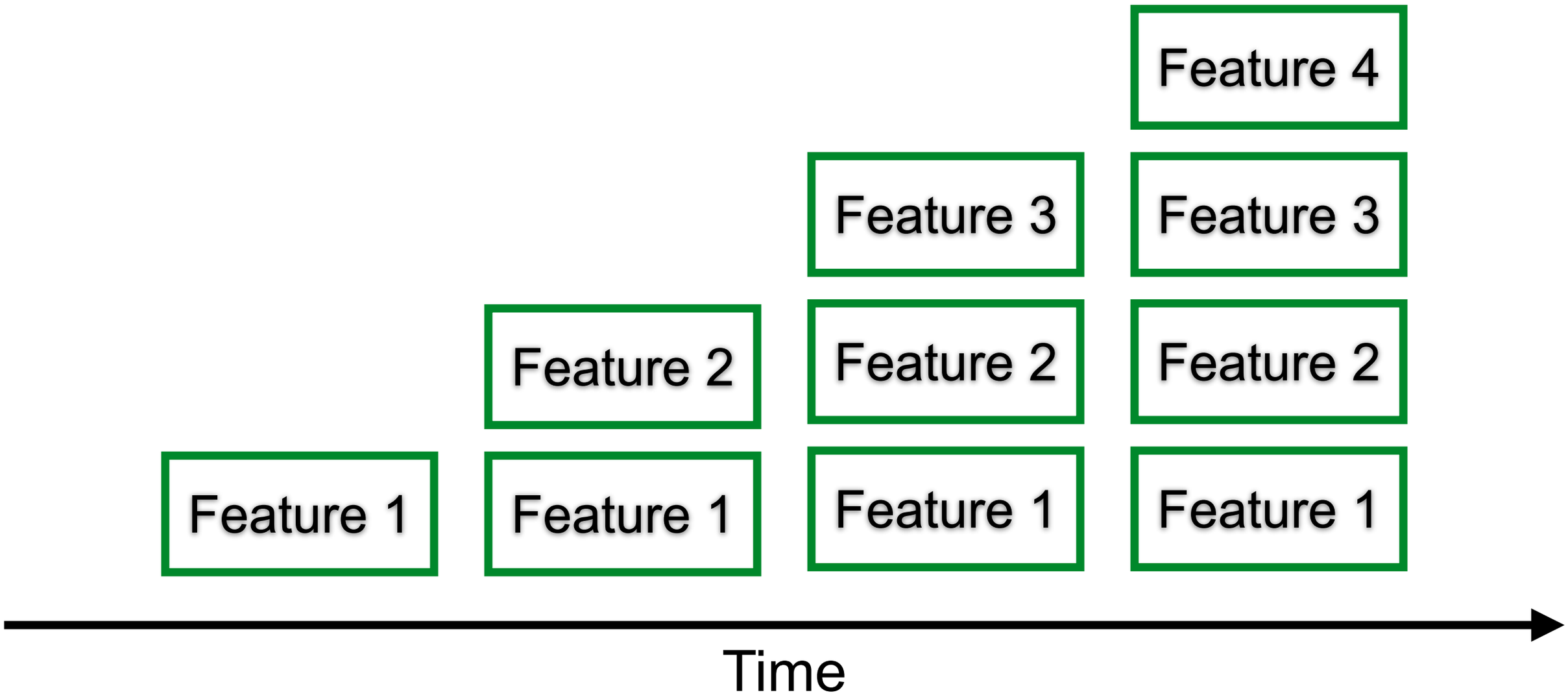
Iterative and incremental process

Done = coded and tested



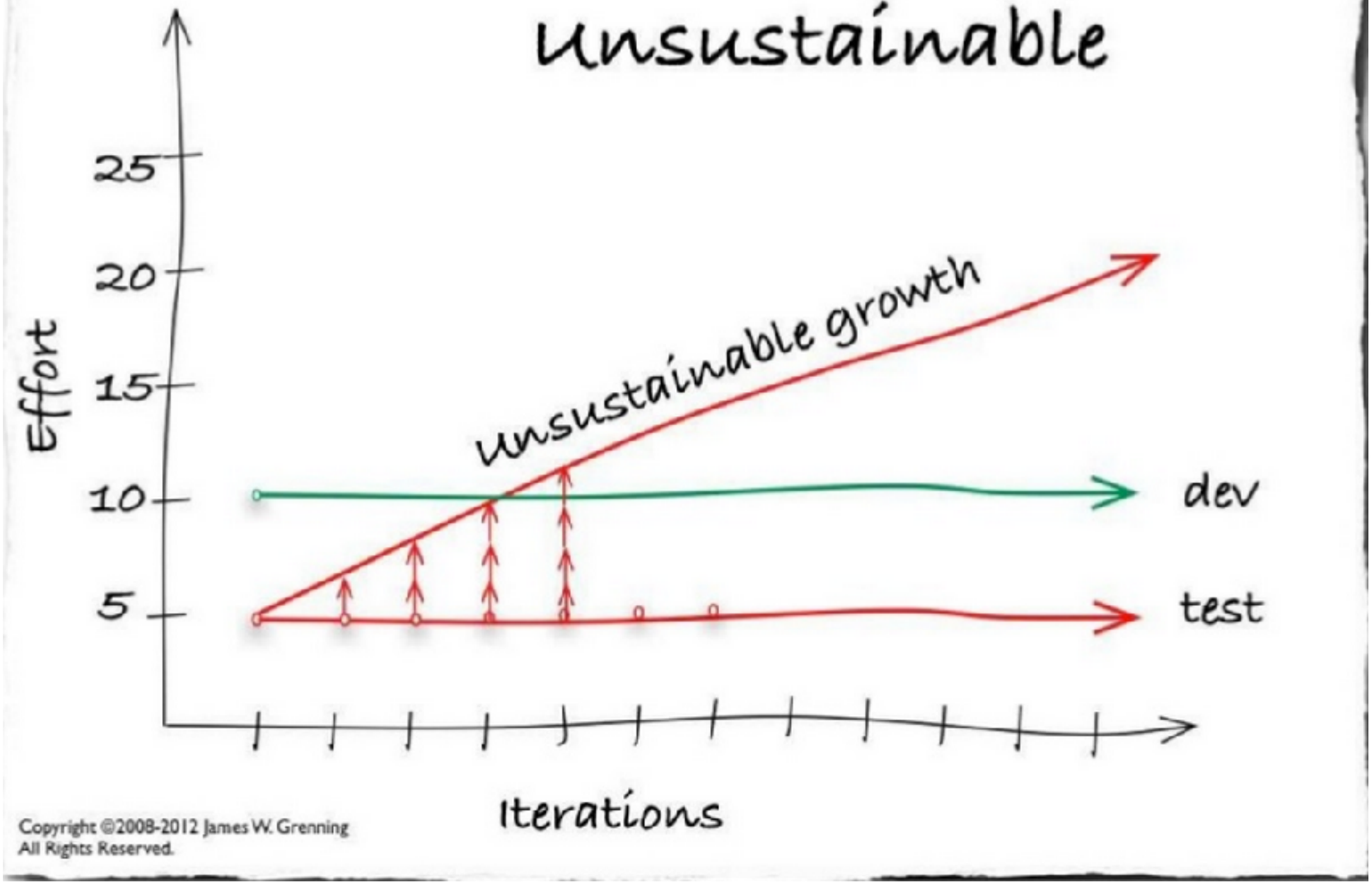
Iterative and incremental process

Done = coded and tested

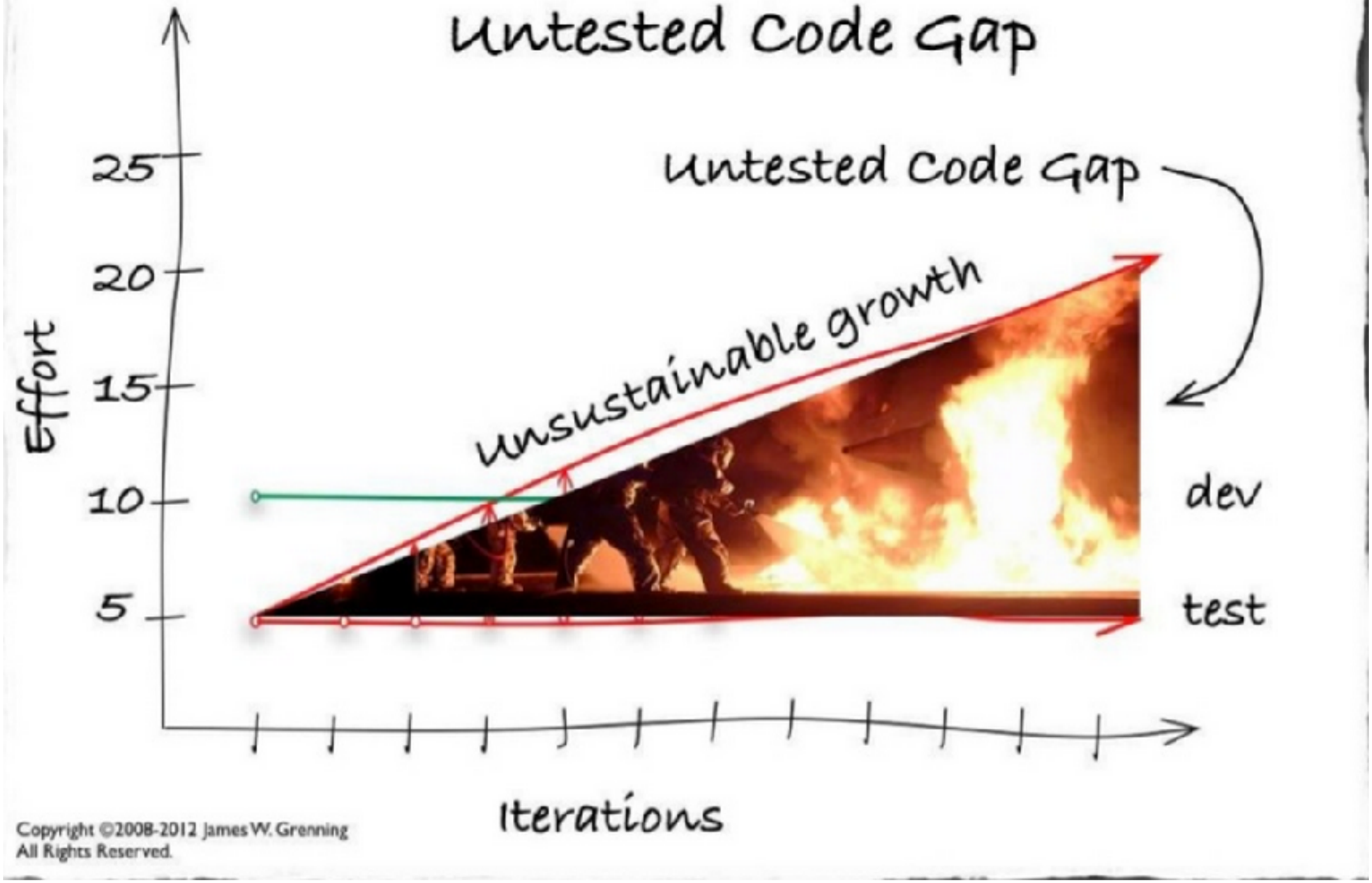


Manual testing ?

Manual Test is Unsustainable



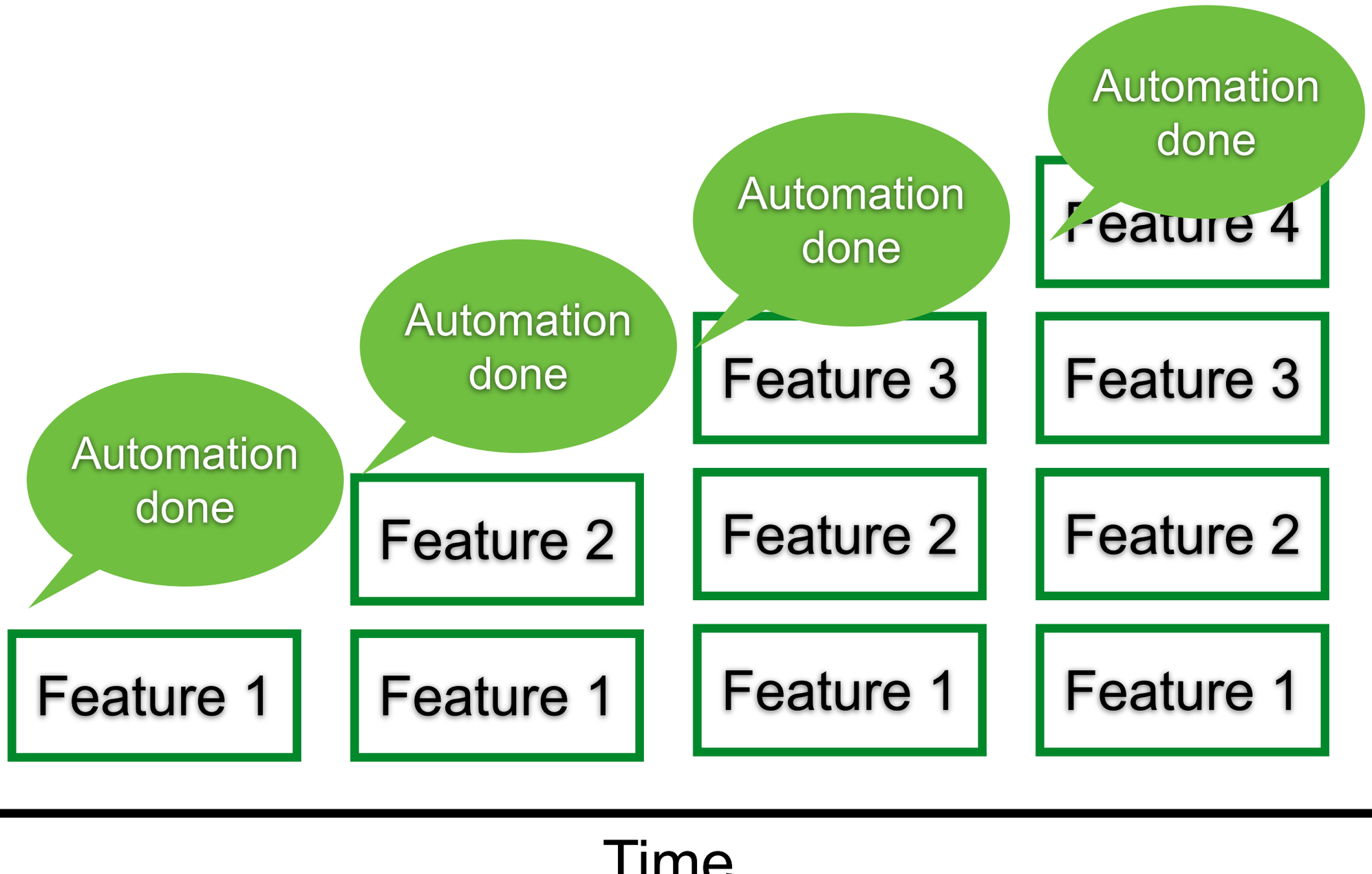
Risk Accumulates in the Untested Code Gap



We need automation testing

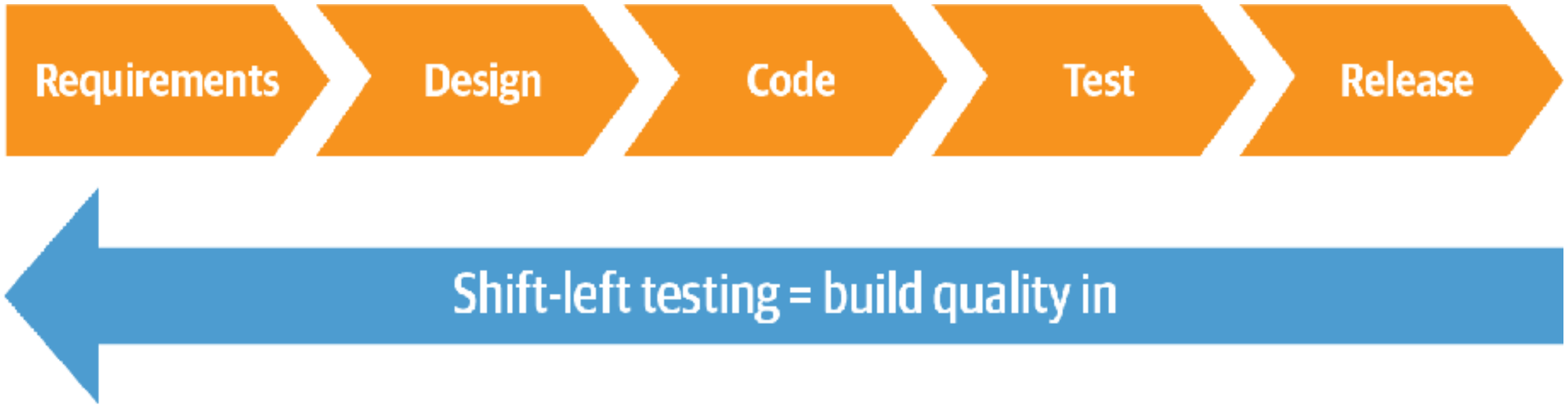
Iterative and incremental process

Done = coded and tested



Test is activity

Shift Left Testing



Functional testing	Non-functional testing

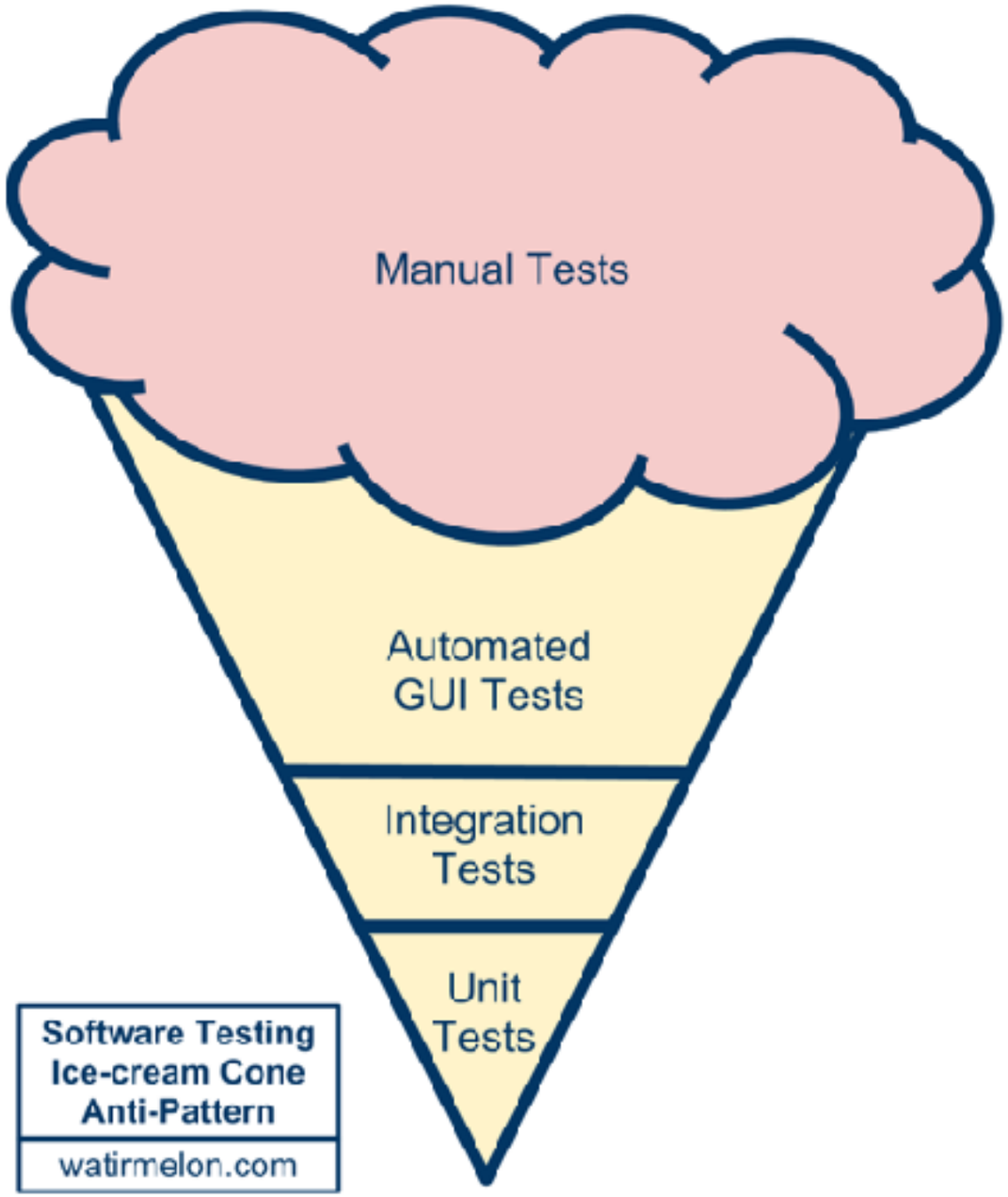
Good Test

F.I.R.S.T + U

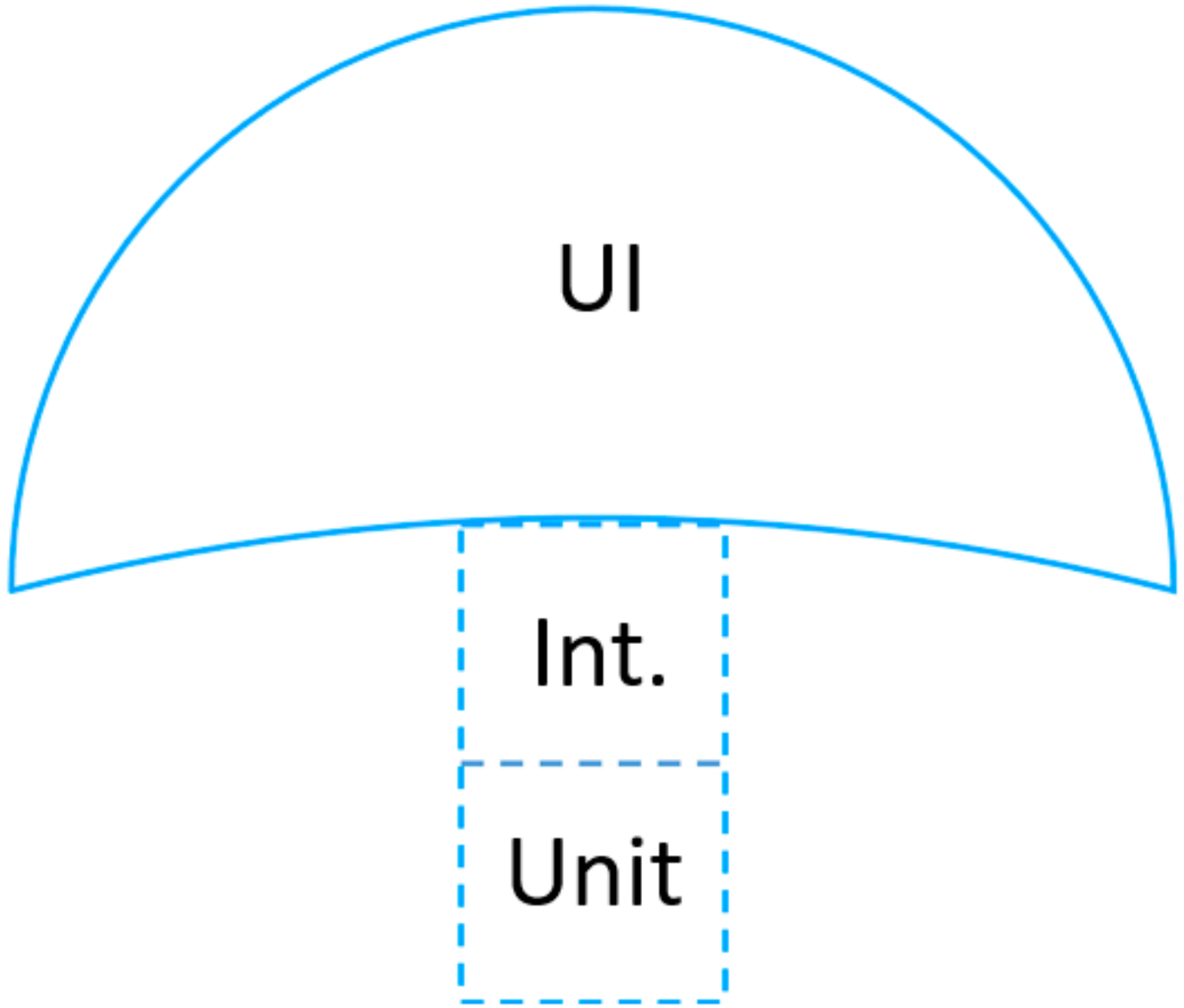
Fast	Isolate	Repeat
Self-verify	Timely	Understand

Testing ?

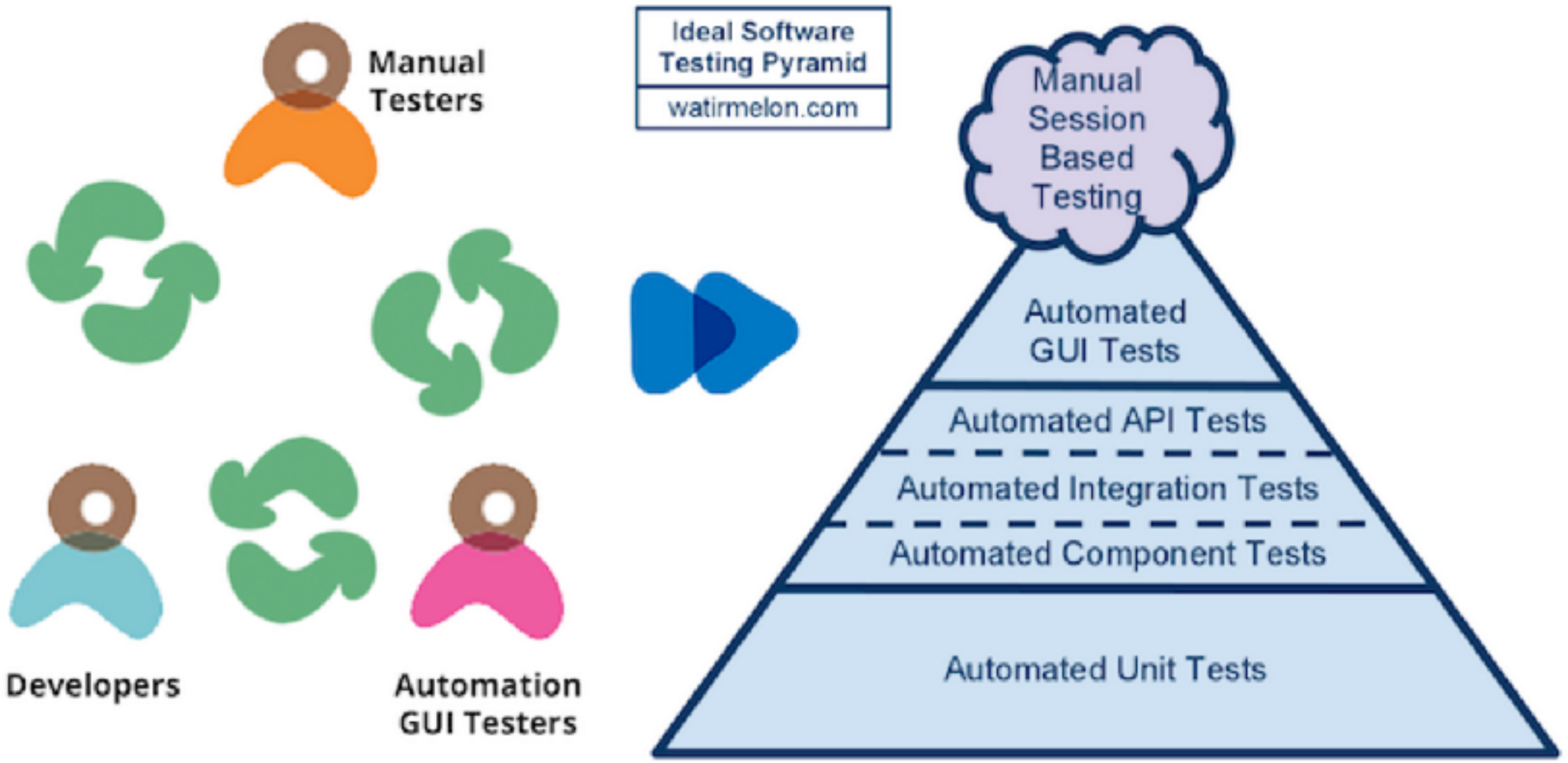
Ice cream testing



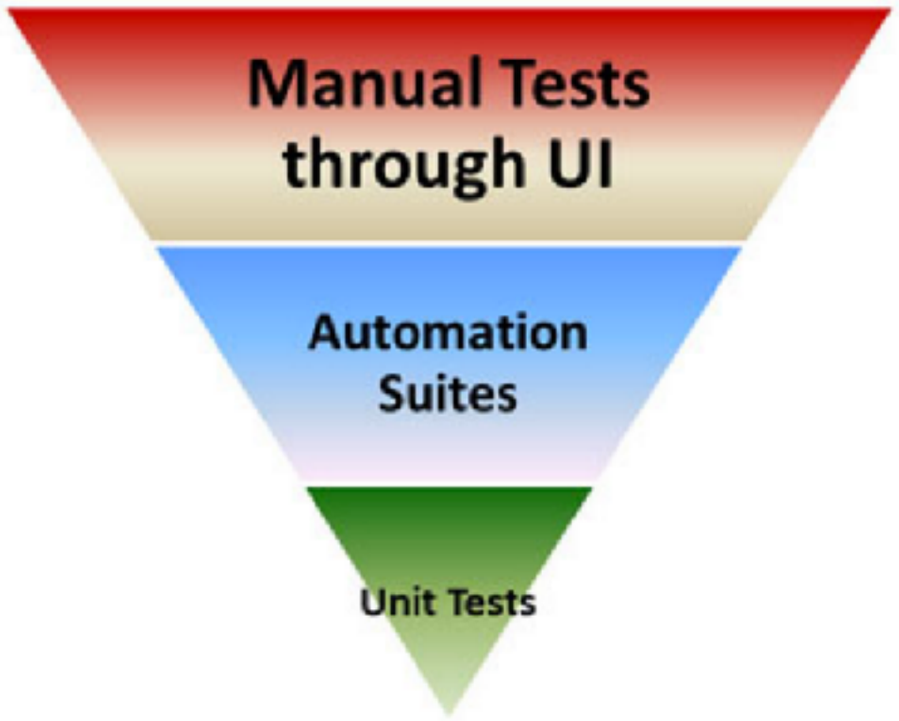
Mashroom testing



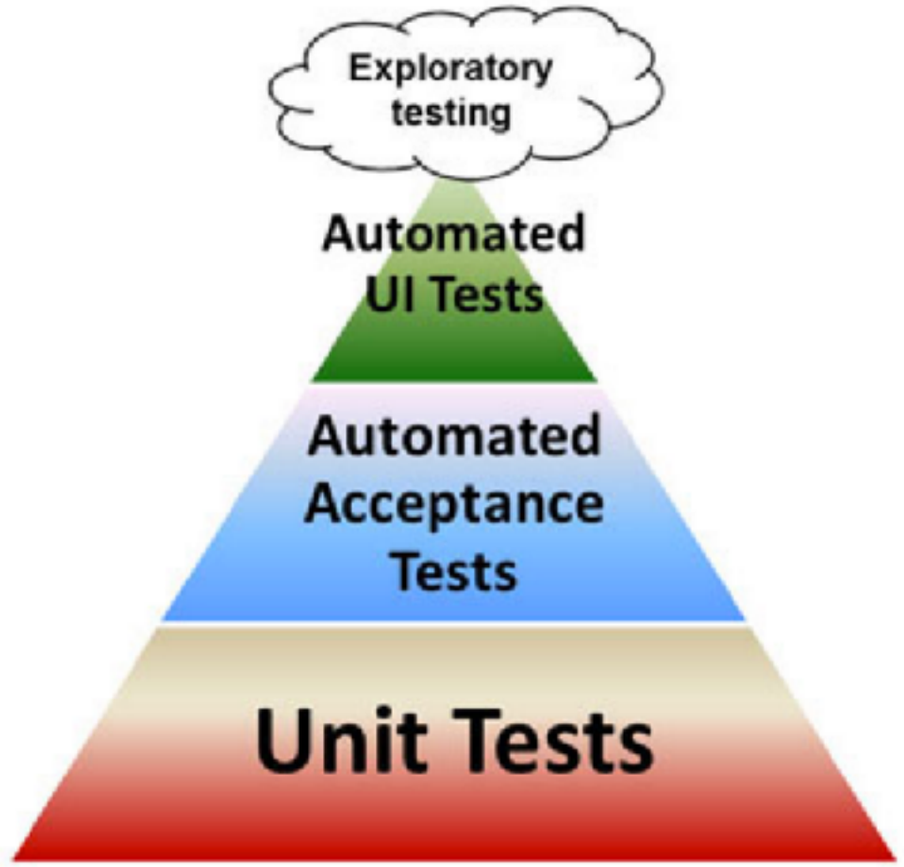
Pyramid testing



Prevention over find bugs

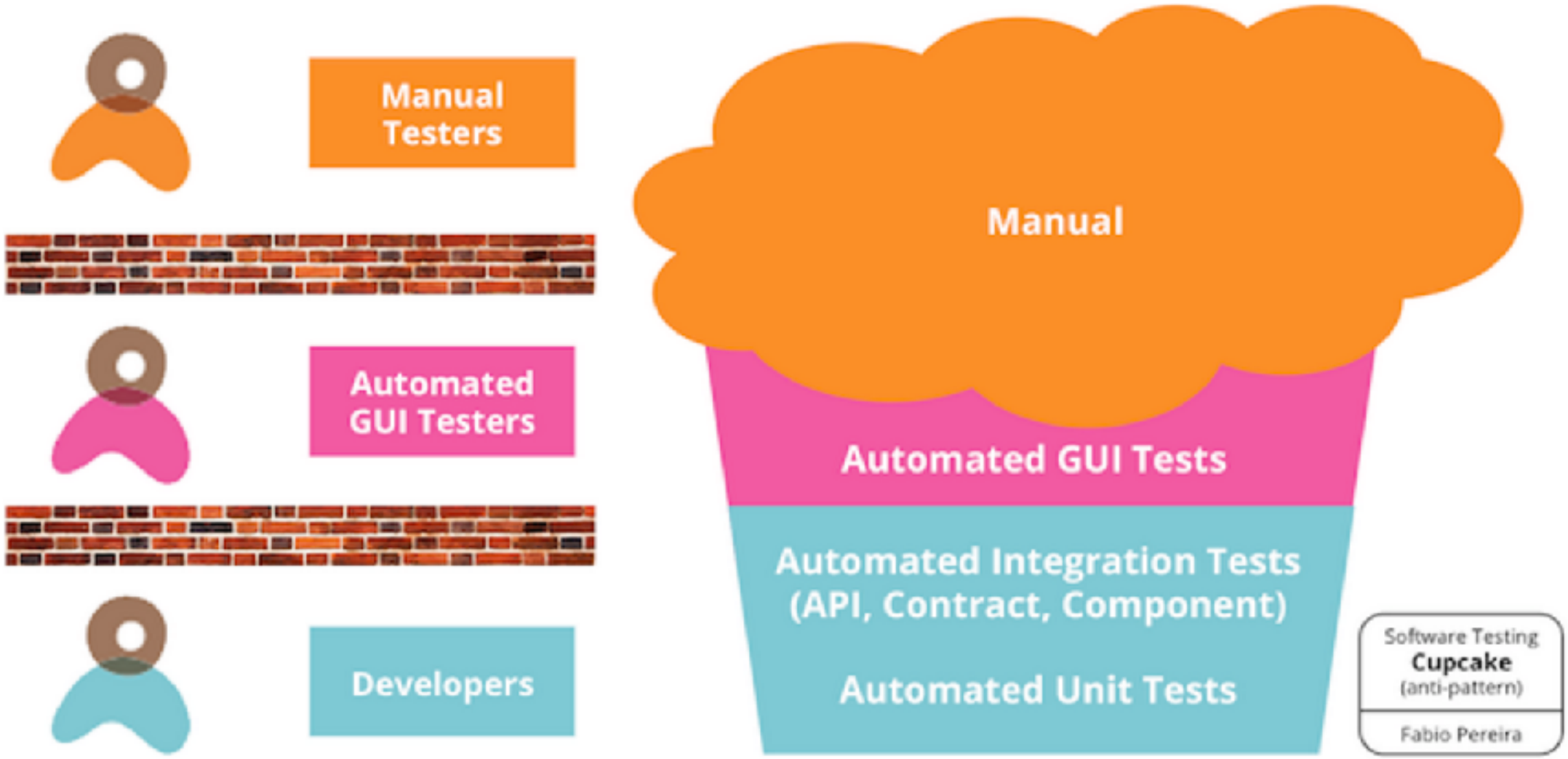


Traditional
(find bugs)

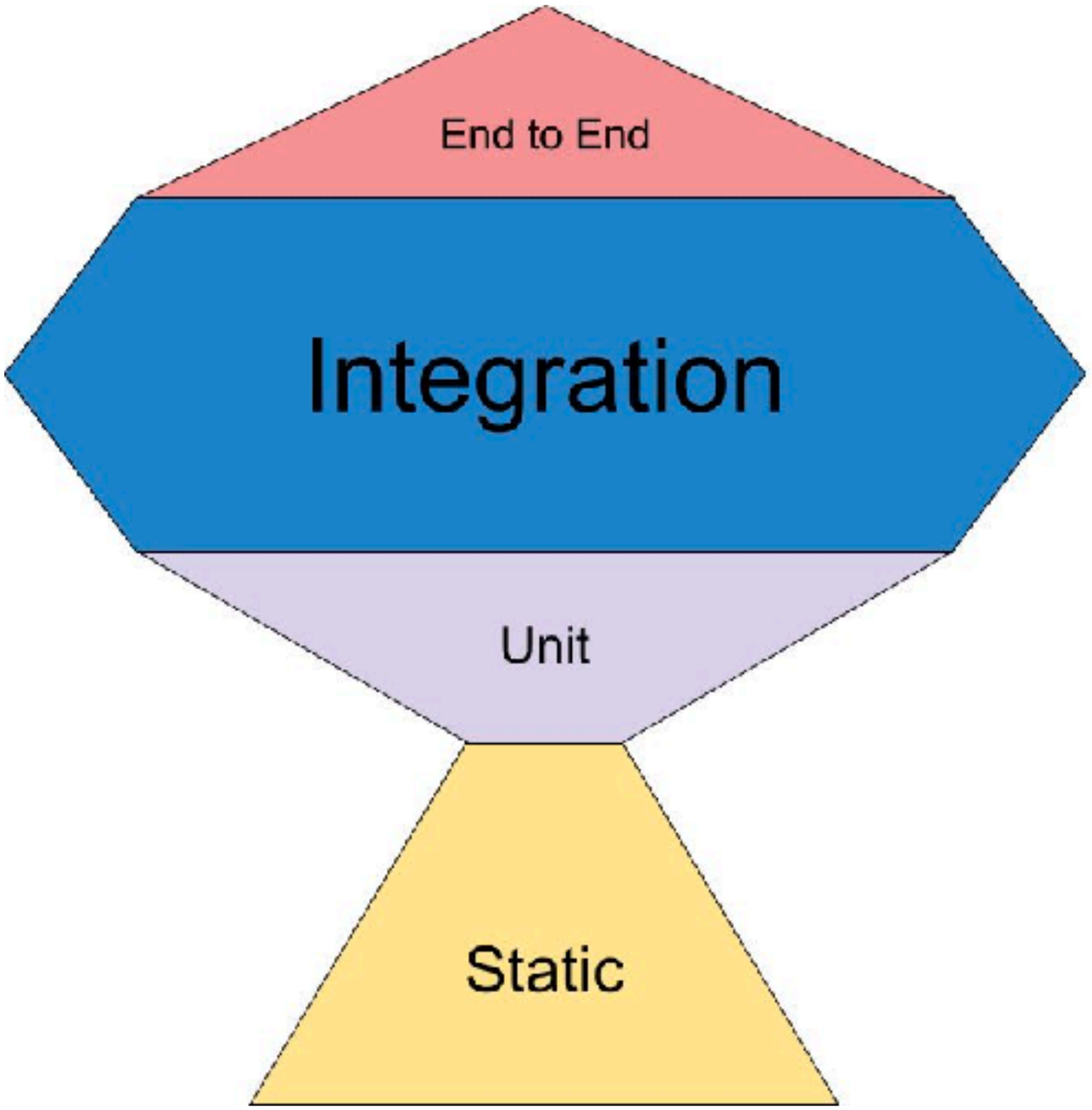


Agile
(prevent bugs)

Cup cake testing



Trophy testing



Mind-set switch

Instead of

We are here to find bug

We are here to ensure requirement are met

We are here to break the software

Think

What can I do to help deliver the software successfully !!

Remember !!

Test pyramid is a tool
To talk about automation tasks
How to prioritize and help to do automation ?
Way to make **visible to the whole team**

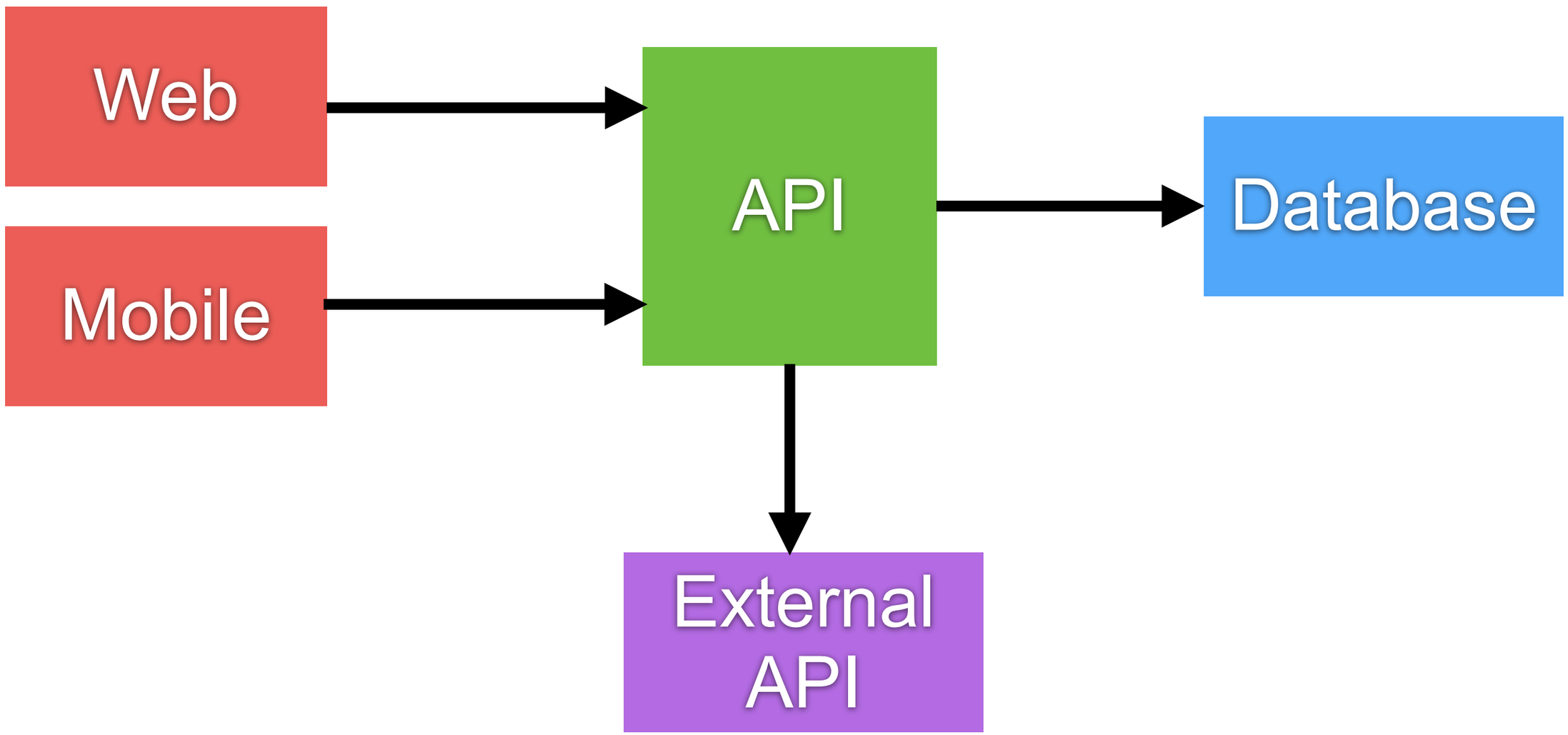
Testing strategies

Test Strategy ?

Unit
Integration
Service Component
Contract
End-to-end

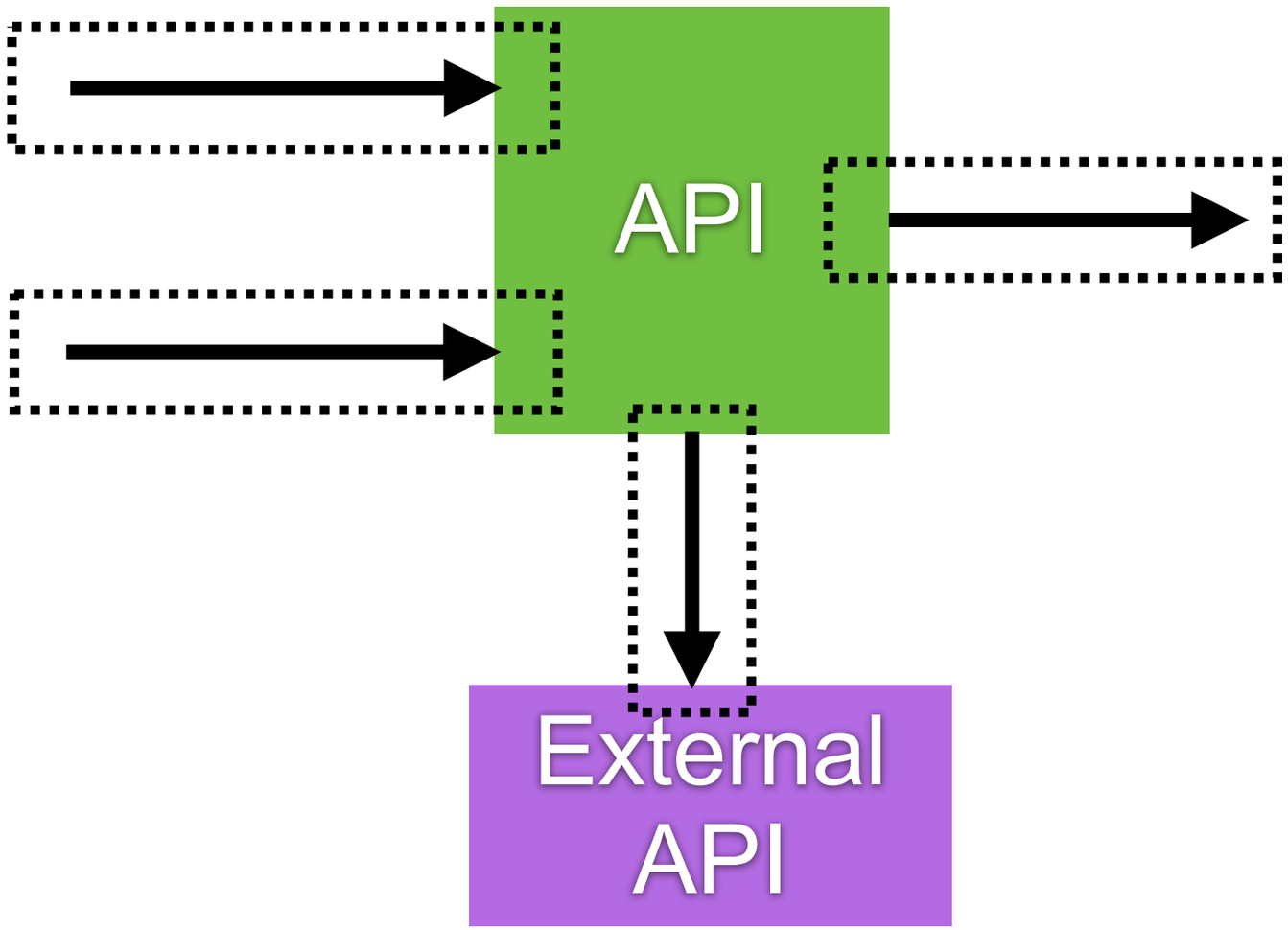
Start with your architecture

Simple architecture

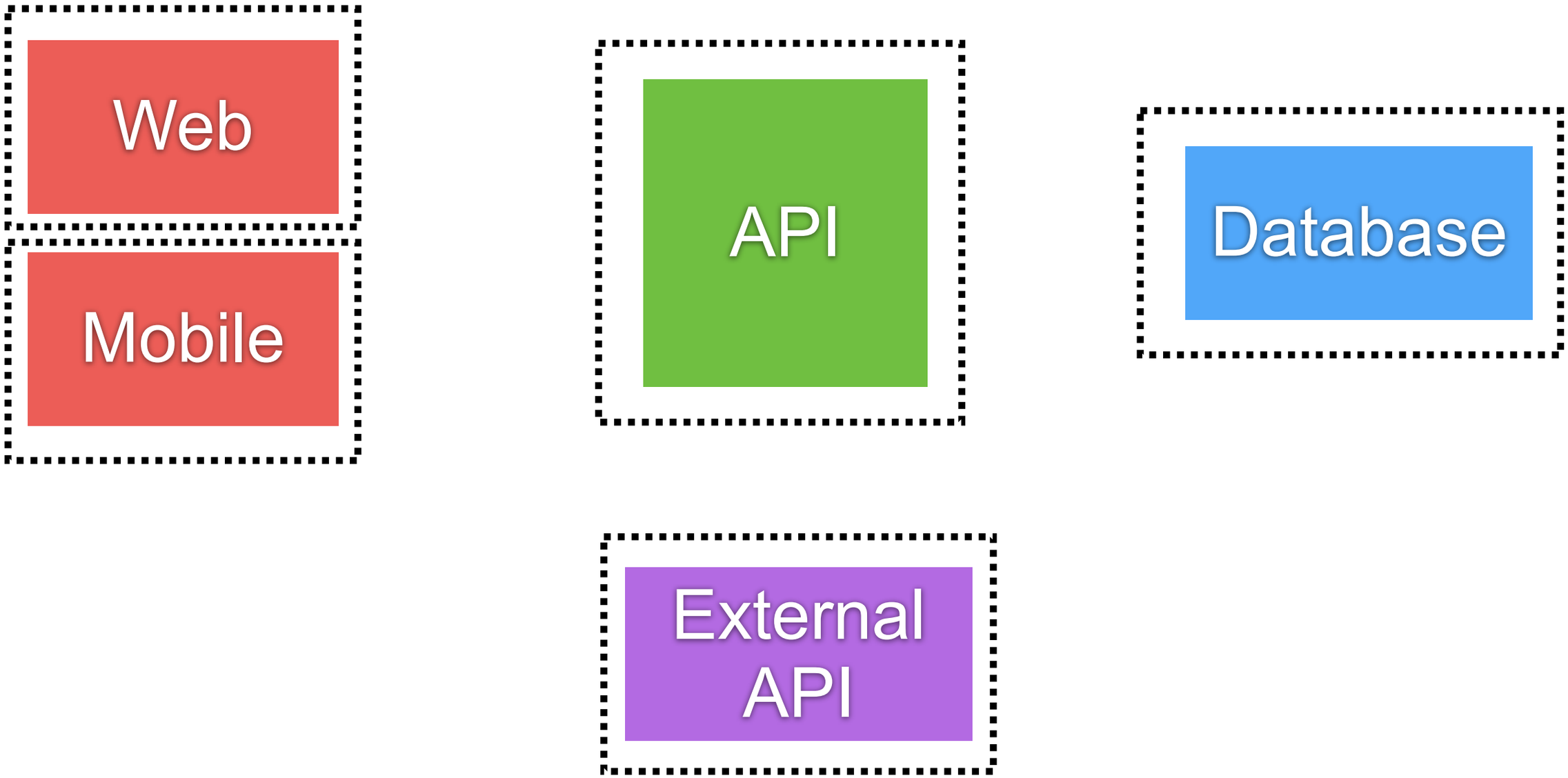


Testing strategies ?

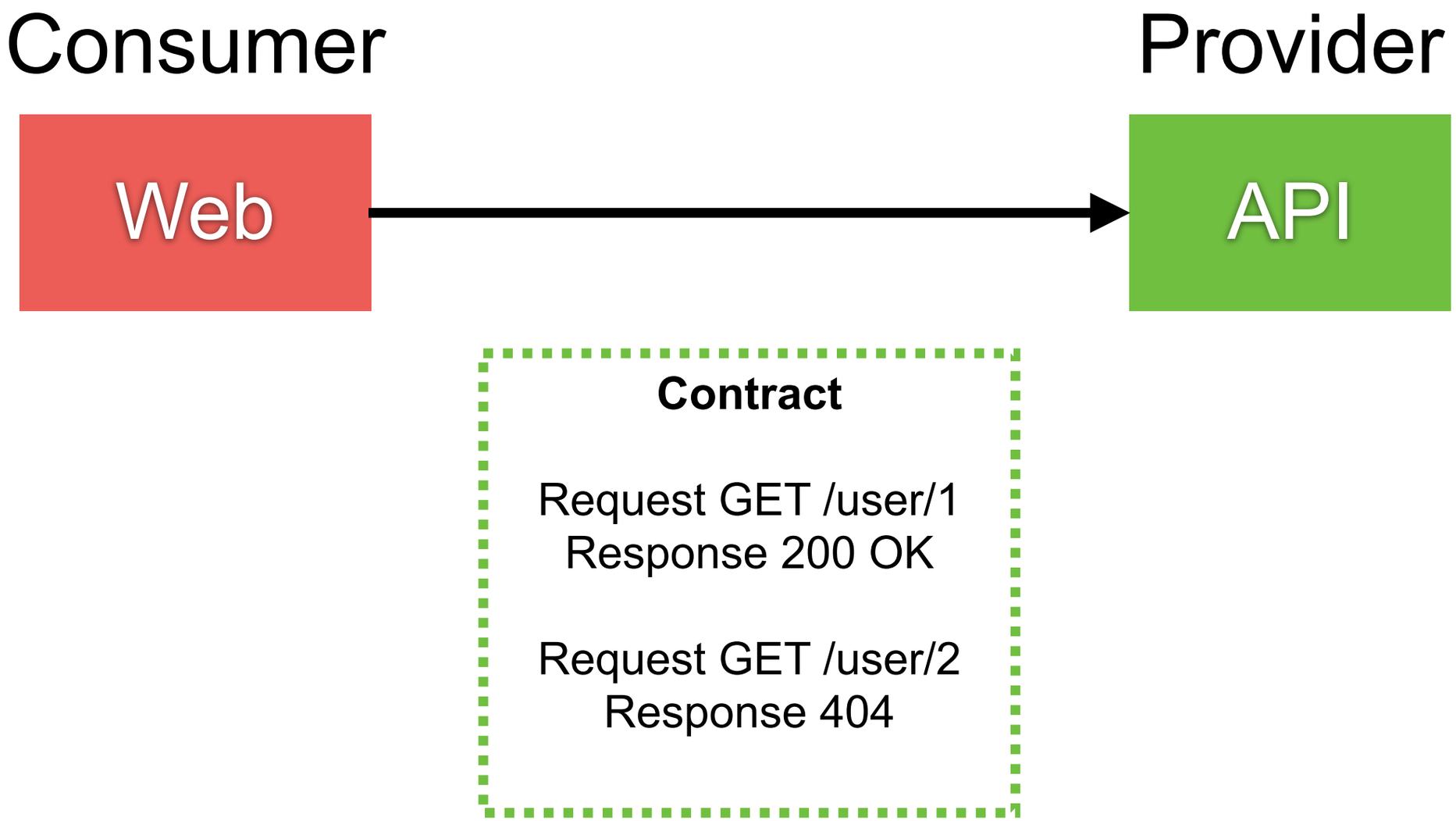
Integration testing



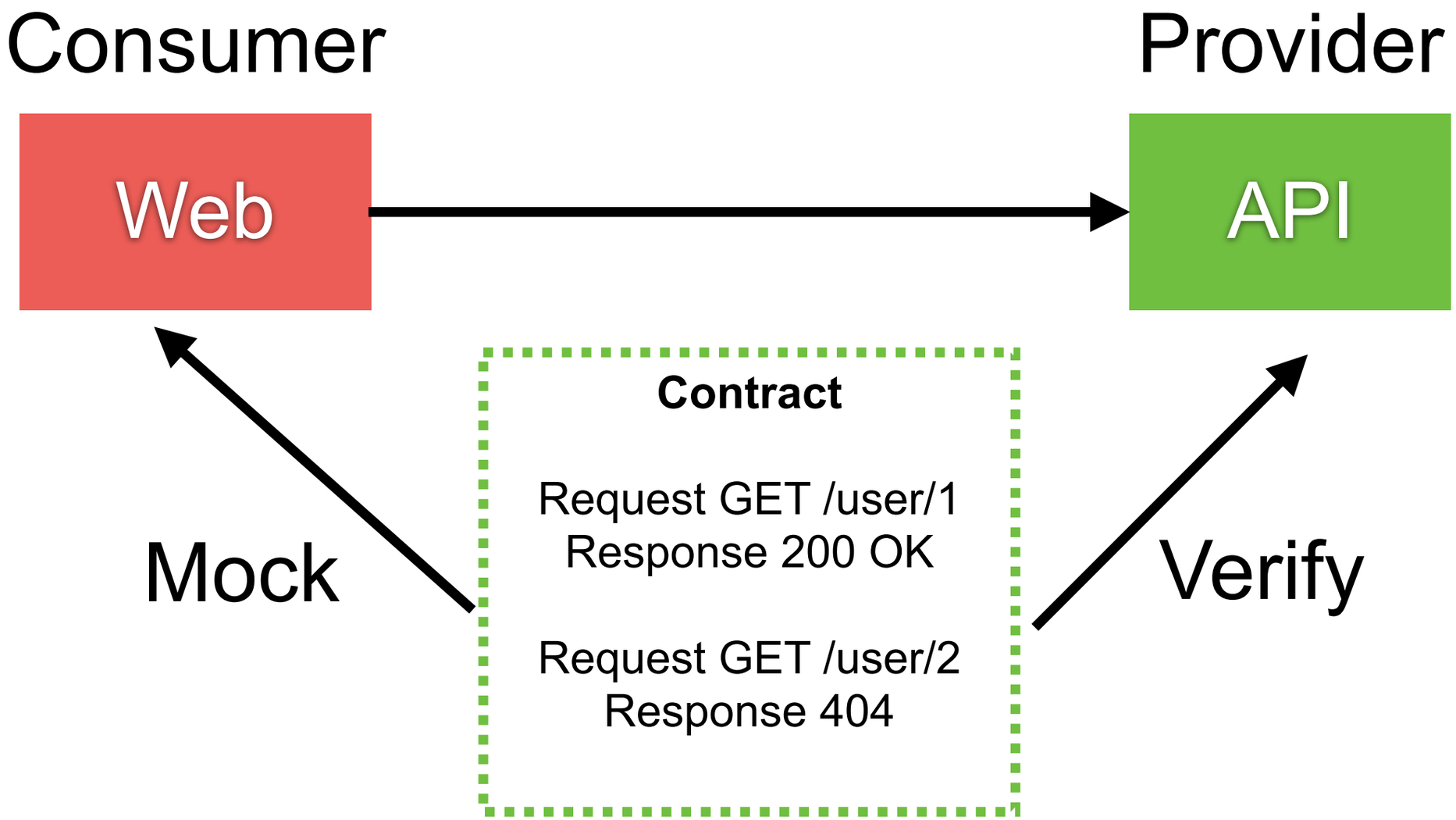
Service component testing



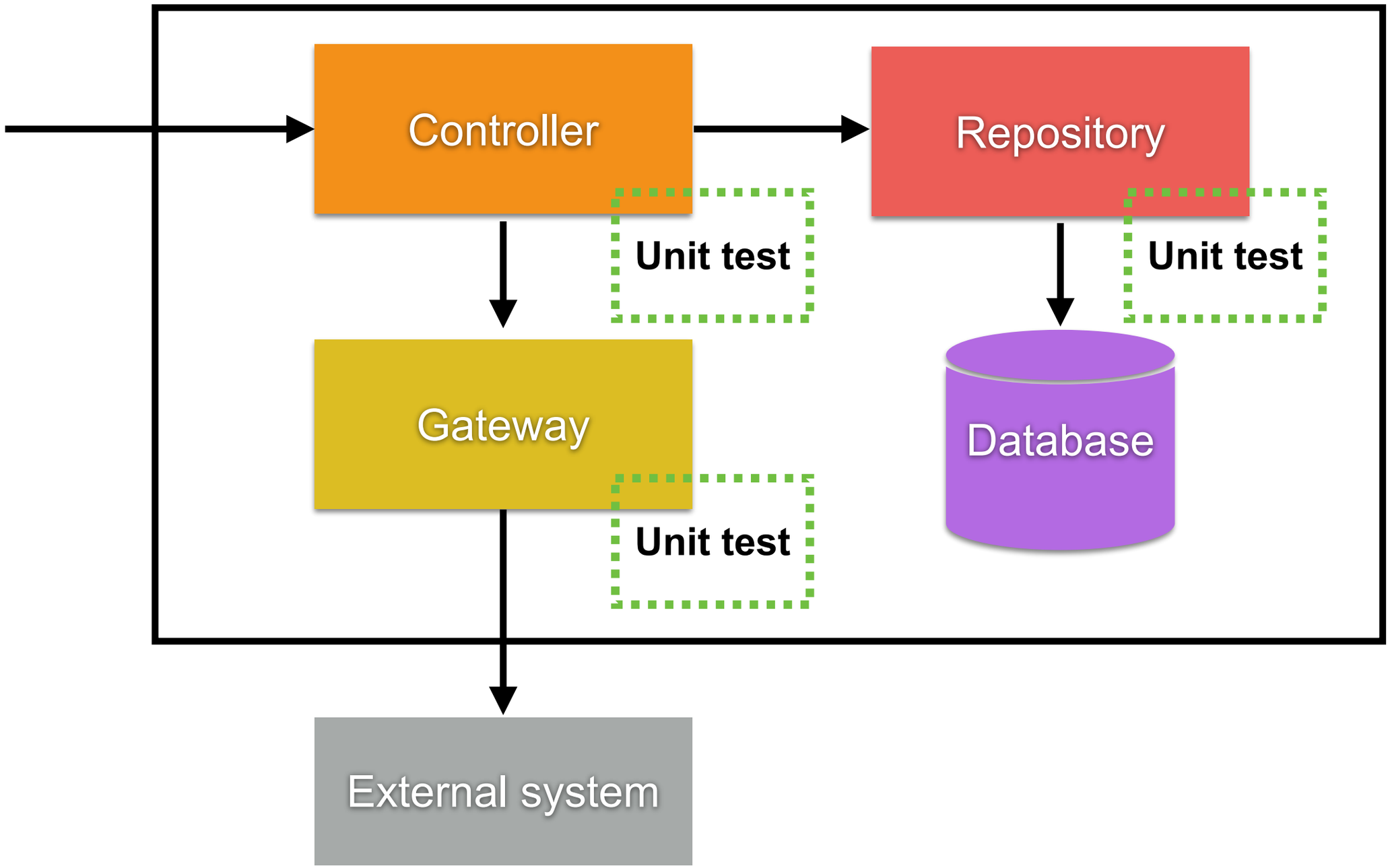
Contract testing (1)



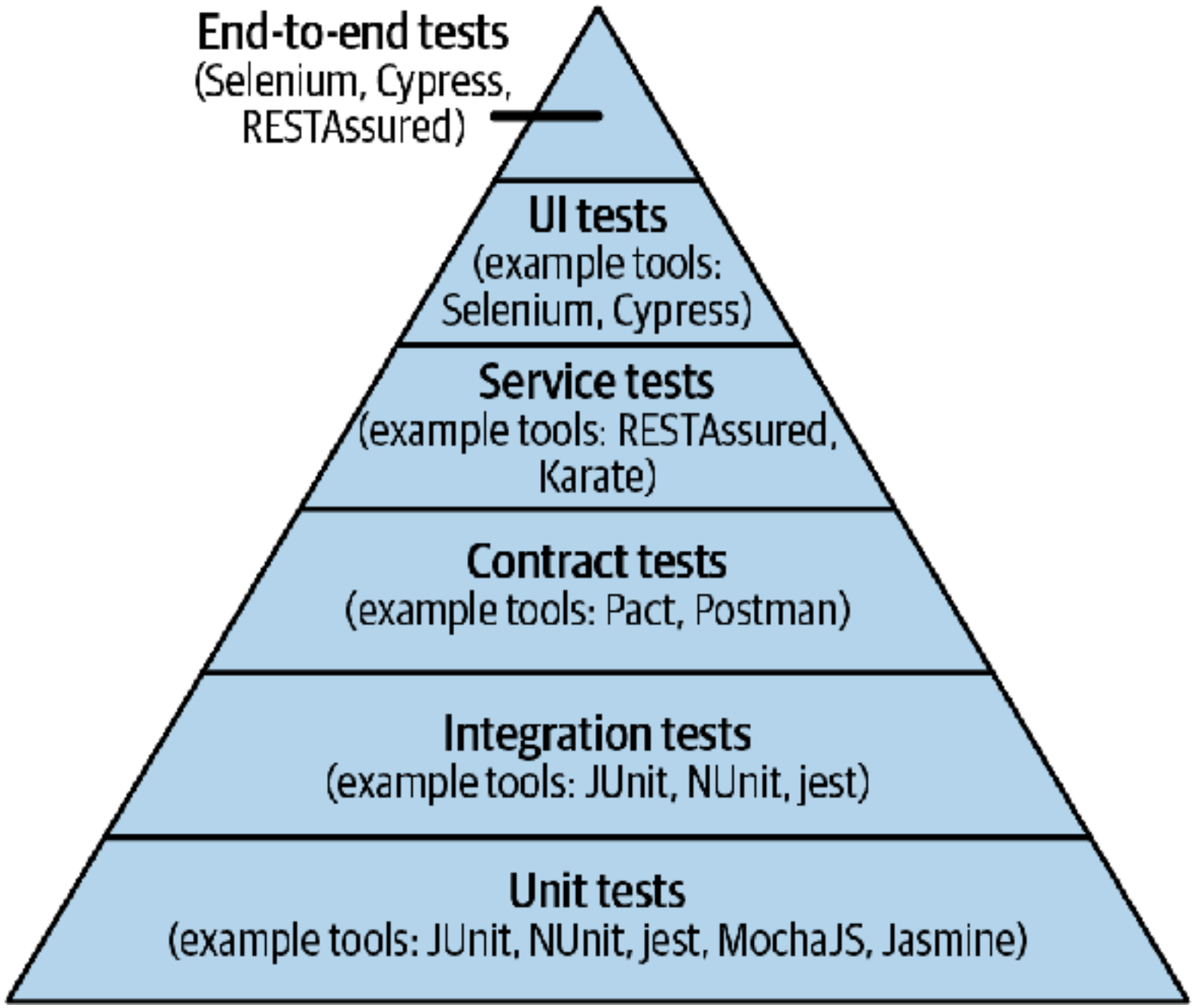
Contract testing (2)



Unit testing

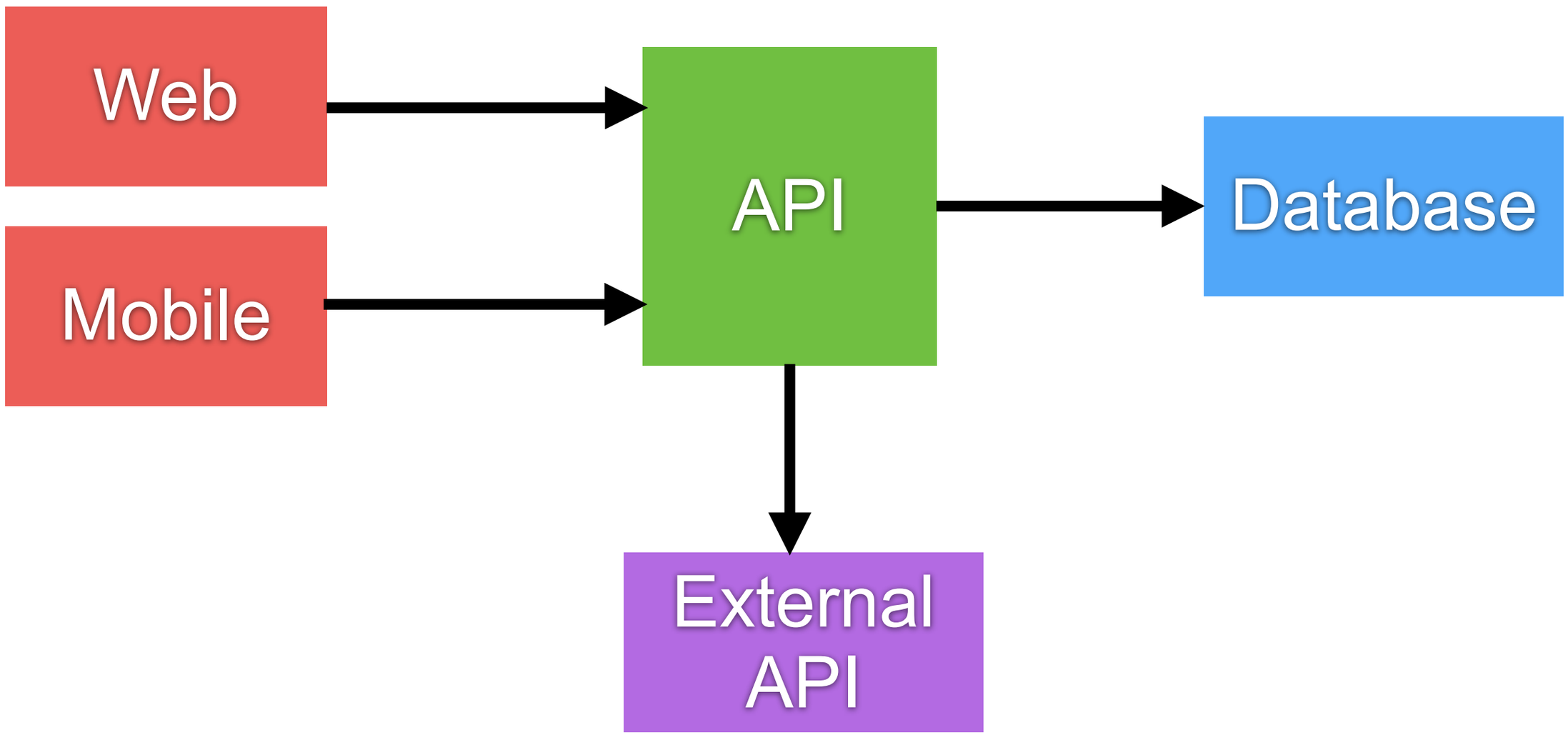


Number of your test cases ?



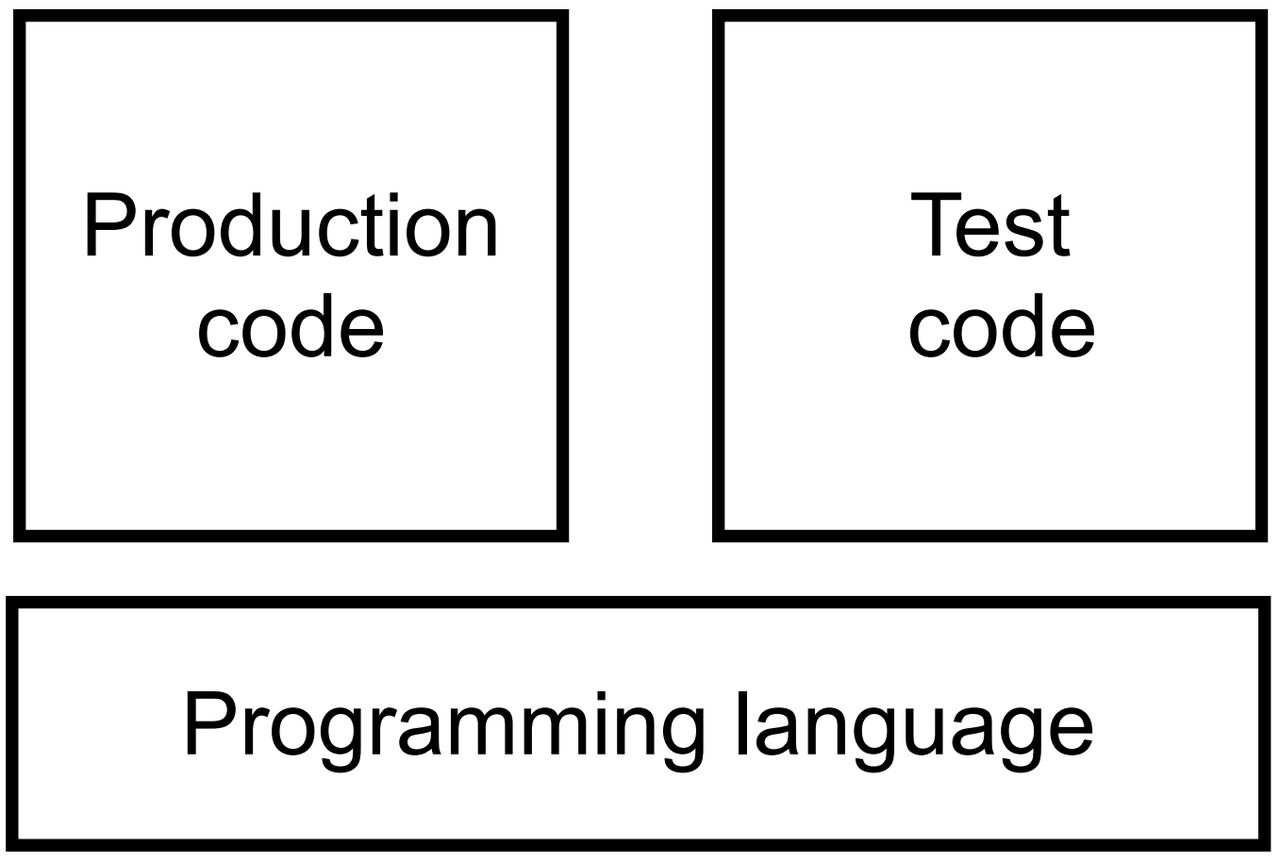
Testing tools and framework ?

Testing tools ?



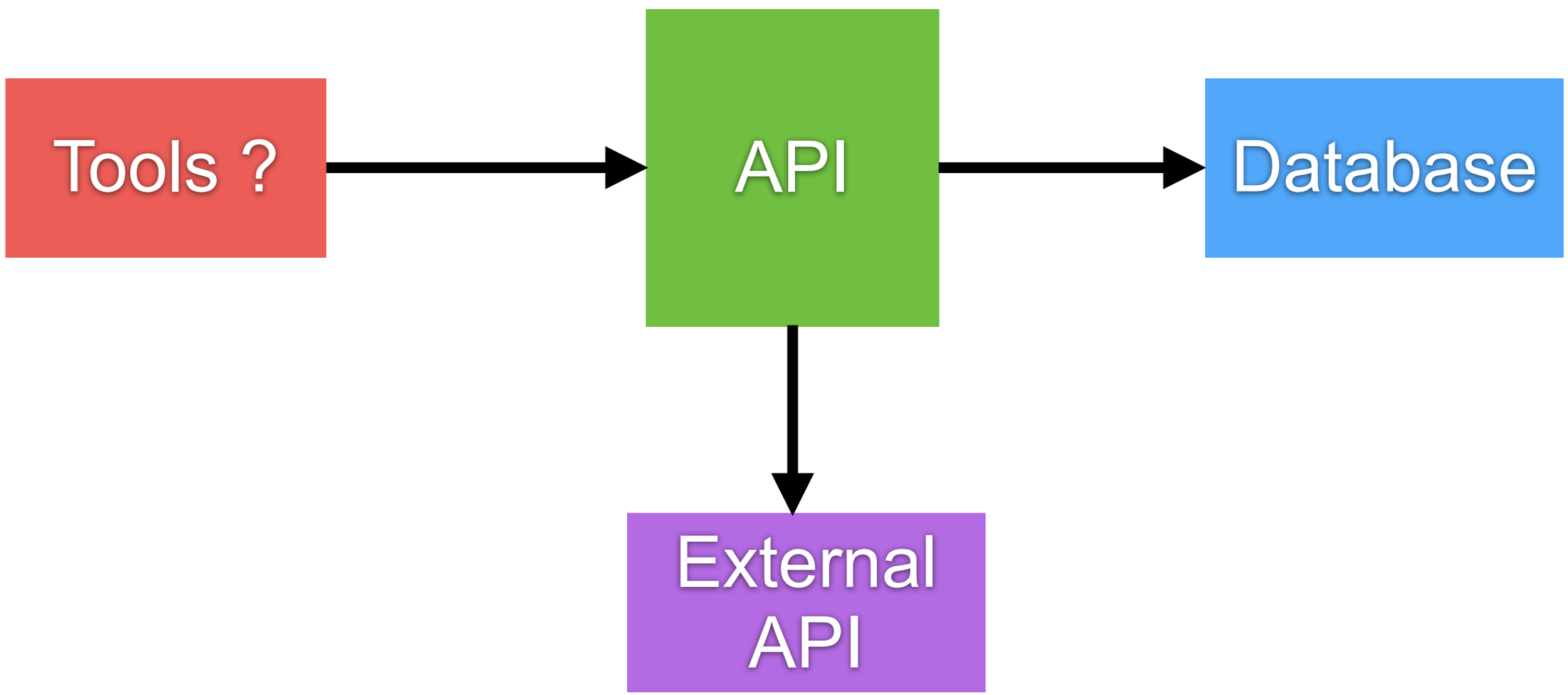
Testing ?

Write test cases with programming language



Testing for API

API Testing ?



Testing for API

REST API

GraphQL

gRPC

WebSocket

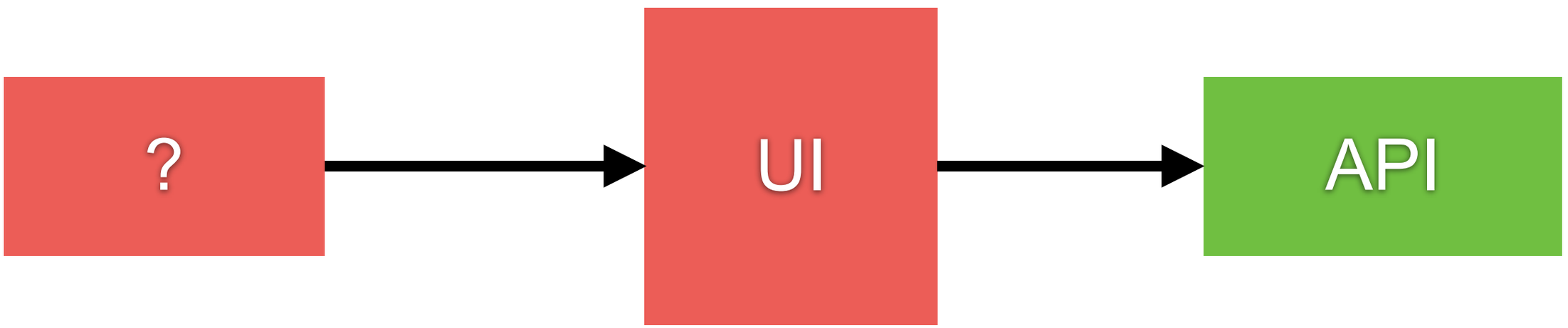
Socket.io

Messaging

Testing for API

Testing for Web UI

UI Testing ?



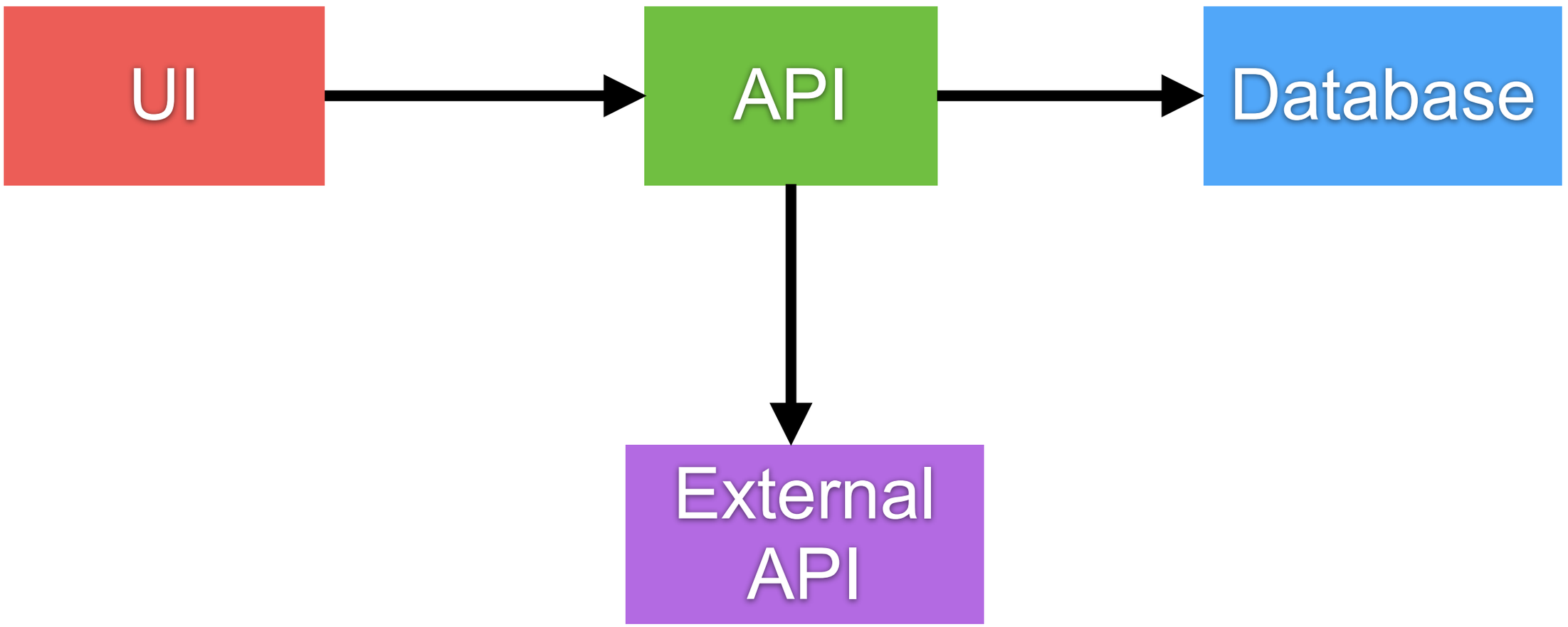
External testing for UI

Write test script and coding

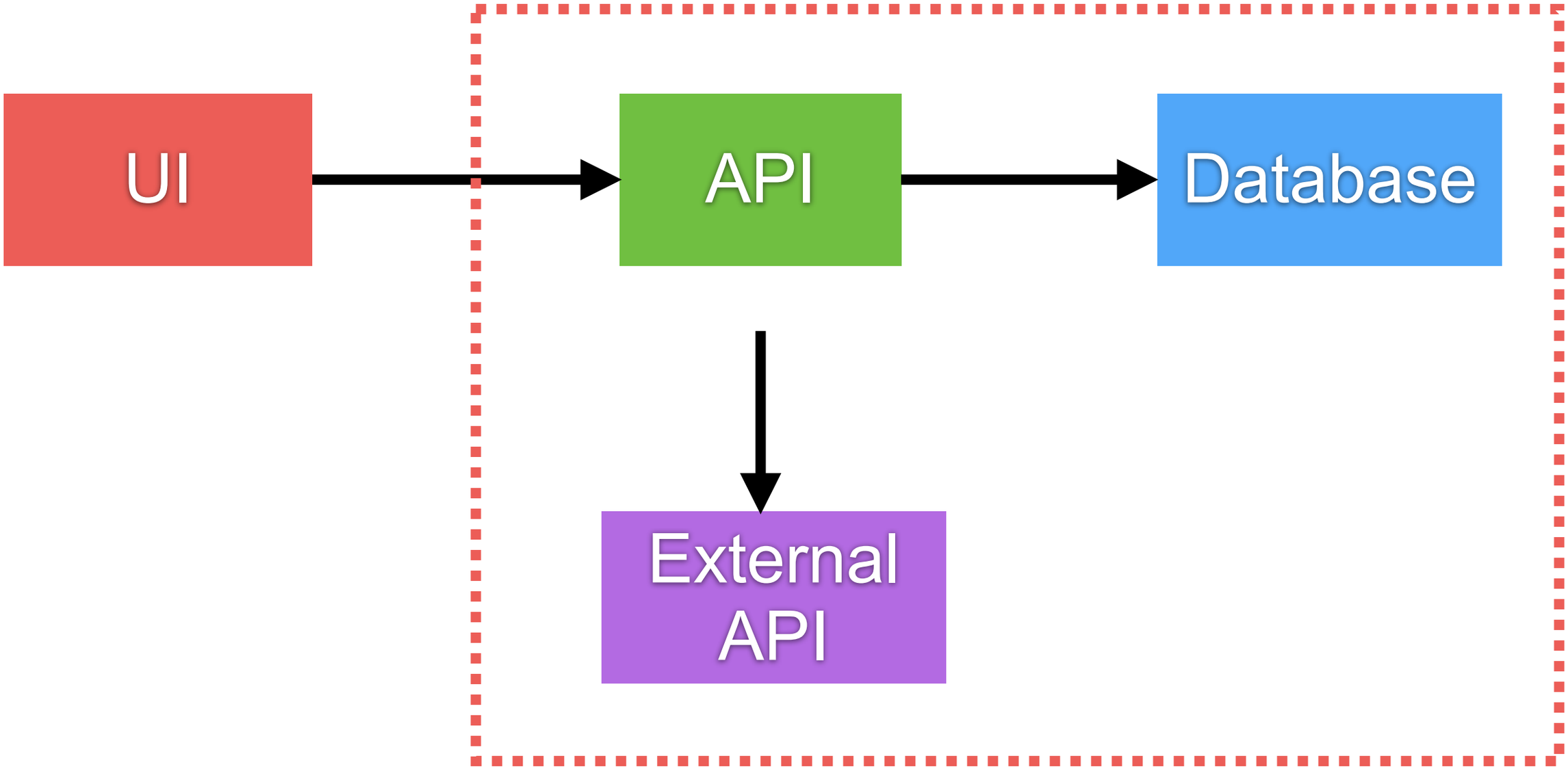
Playwright

Manage dependencies

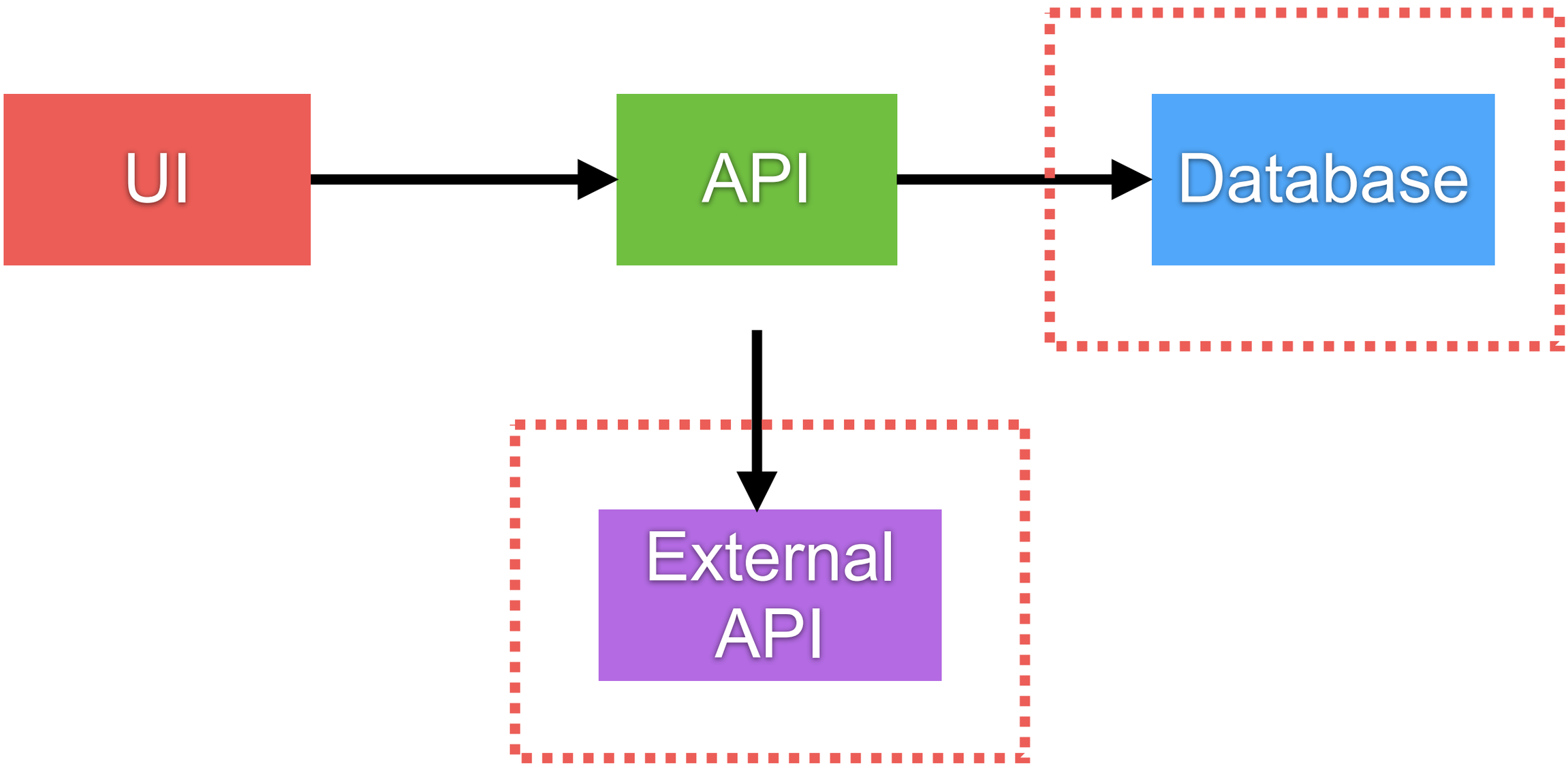
Manage dependencies ?



UI Perspective

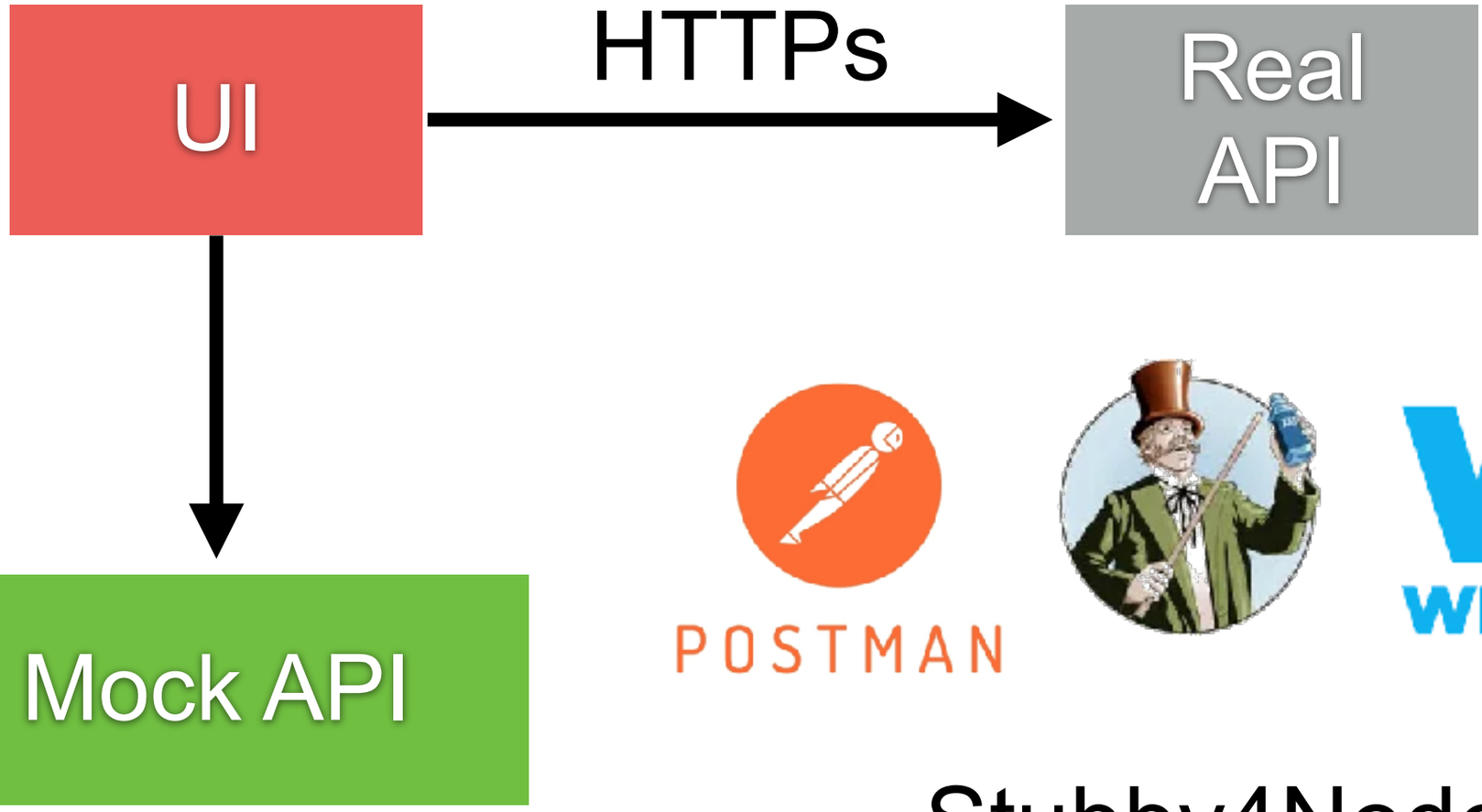


API Perspective



Manage dependencies of UI

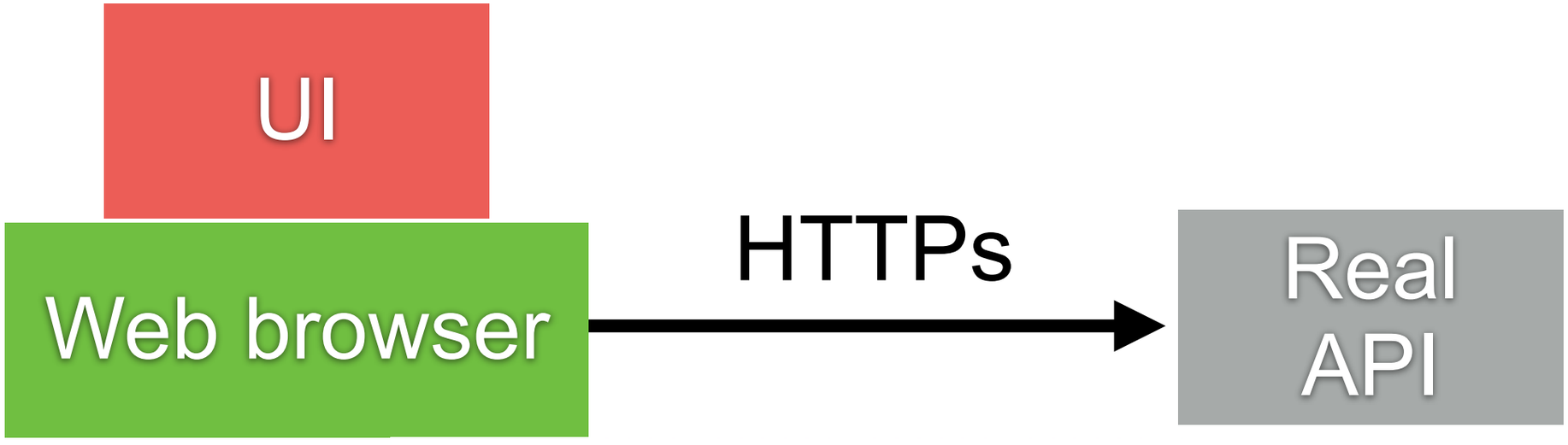
External Mock API Server



Stubby4Node

JsonServer

Intercept in Web browser

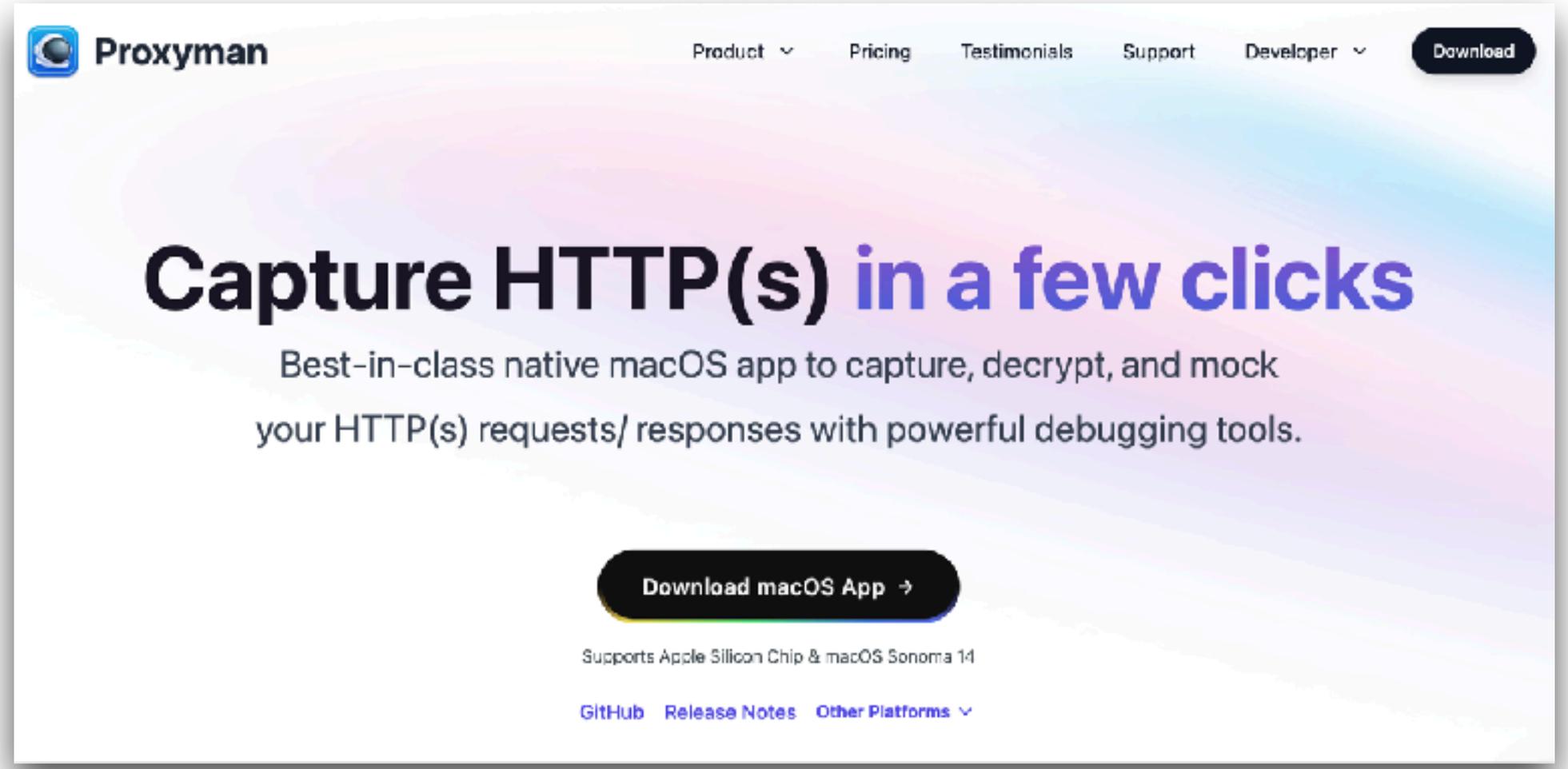


Playwright

cypress

Proxy Man

Capture HTTP request !!

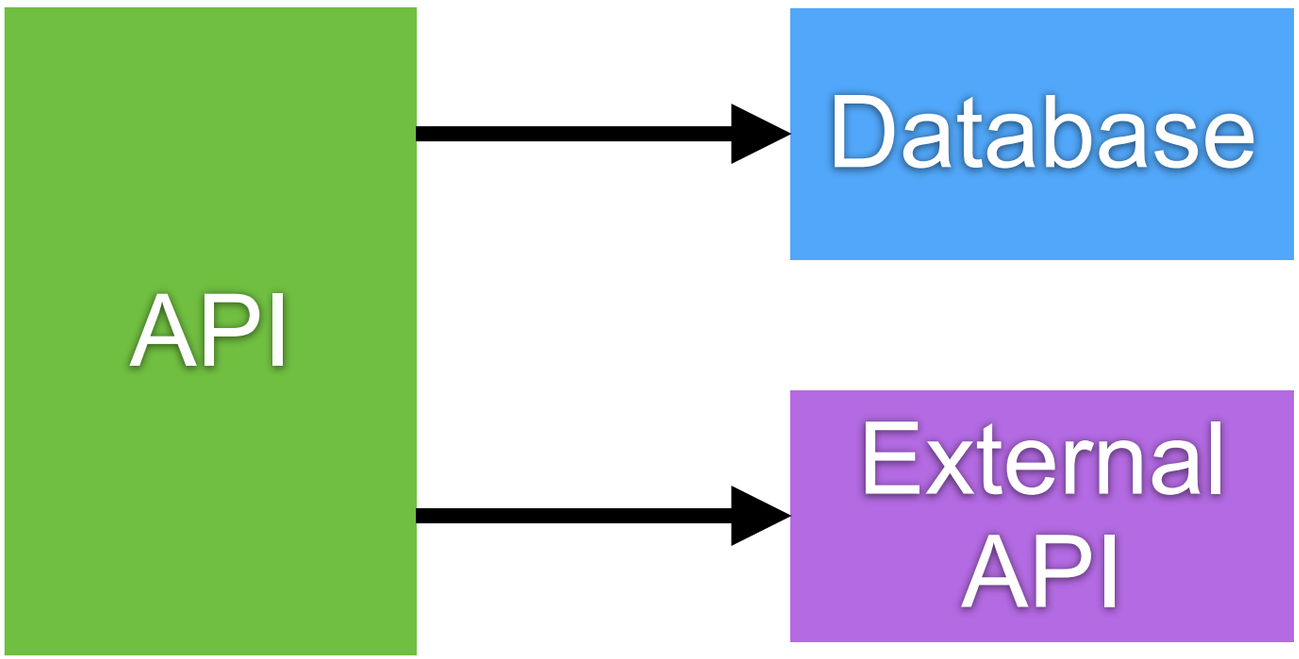


https://proxyman.io/

Manage dependencies of API

Manage dependencies of API

Database
External API



Database

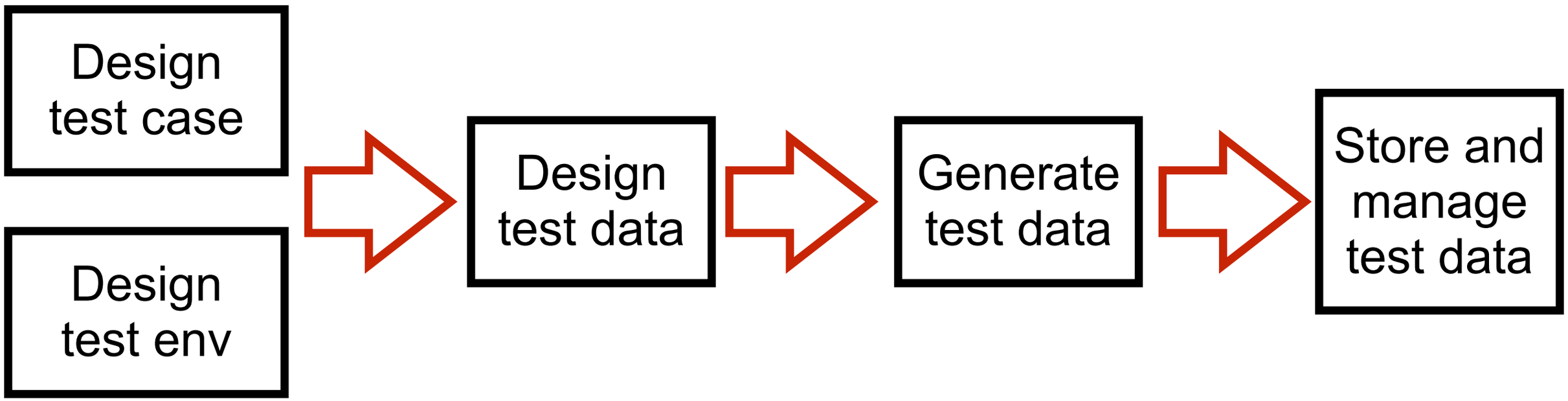
In-memory database
Container-based
Real database

Tables

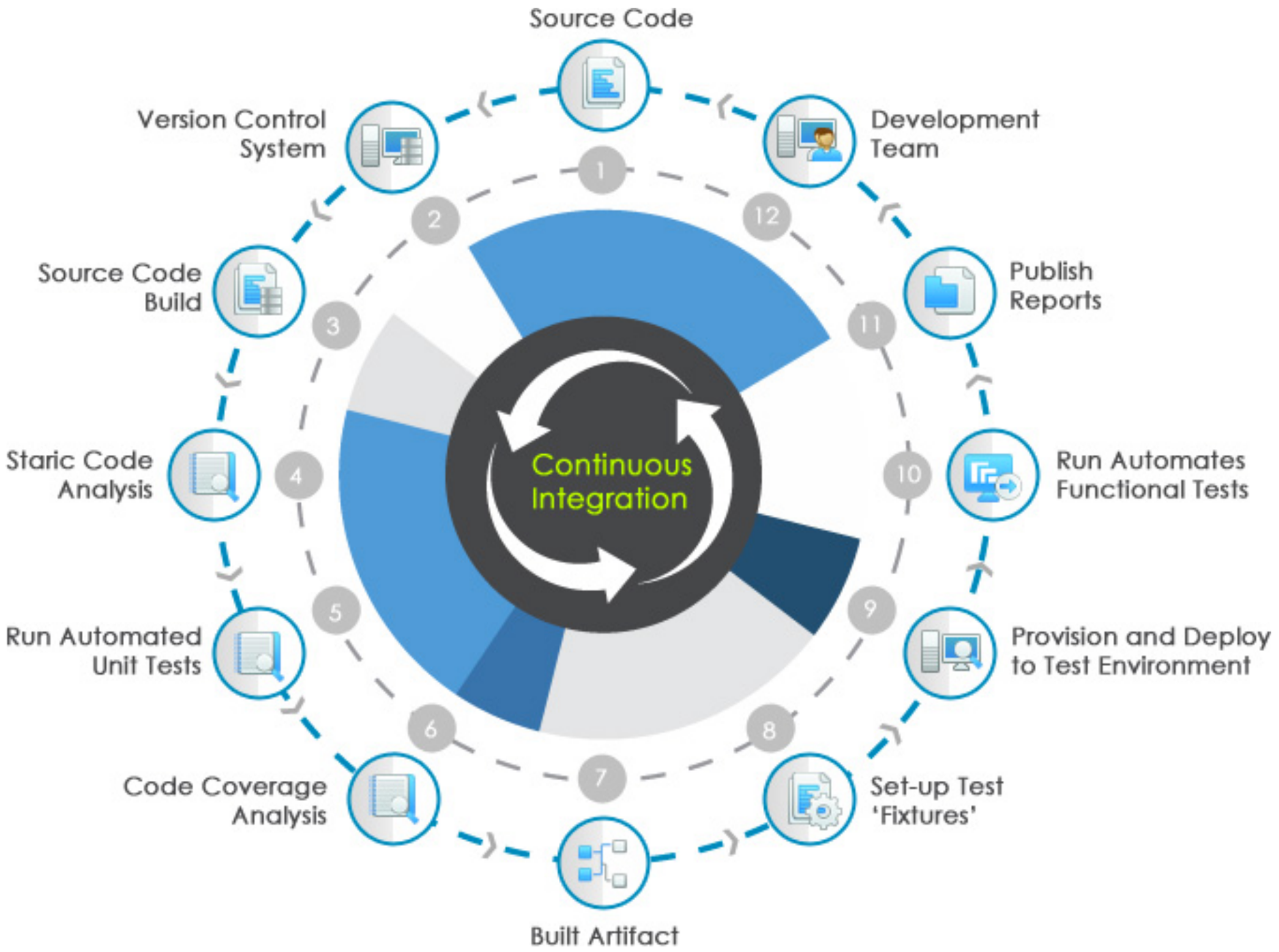
Data for test

Database migration tools

Design test data ?



Delivery pipeline/process ?



Jenkins

Bamboo

TeamCity

go

Visual Studio Team Foundation Server

Hudson

Travis

wercker

circleci

CI is about what people do not about what tools they use

Continuous Integration

Discipline to **integrate frequently**

Continuous Integration

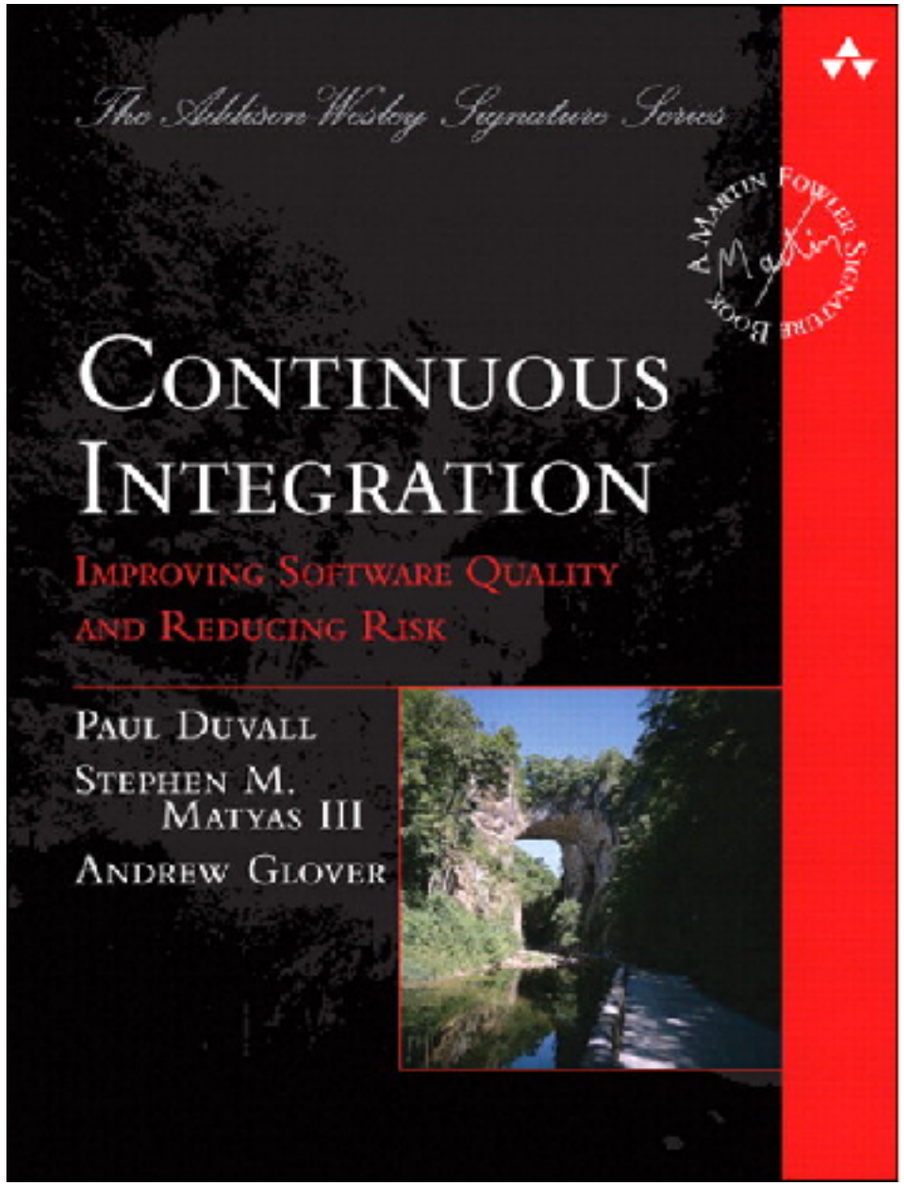
Strive to make **small change**

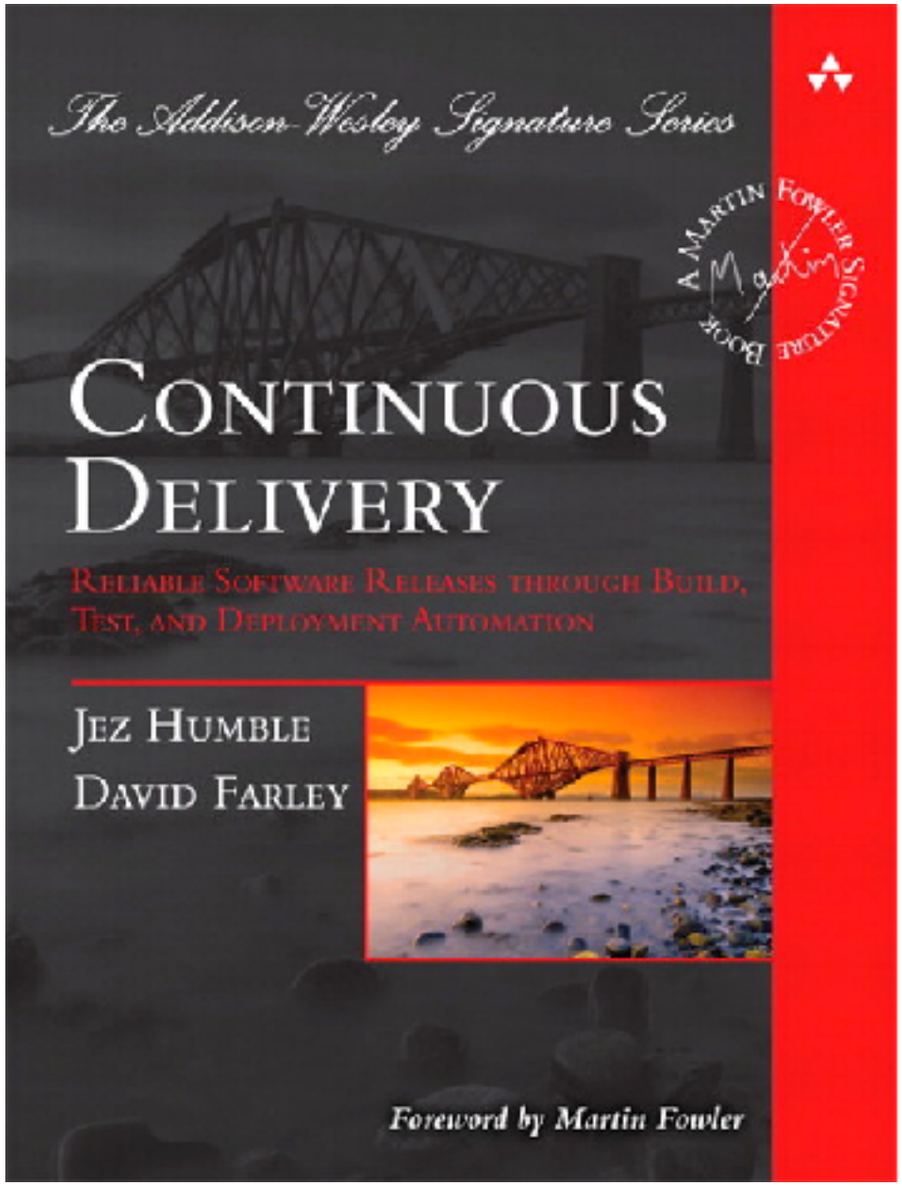
Continuous Integration

Strive for **fast feedback**

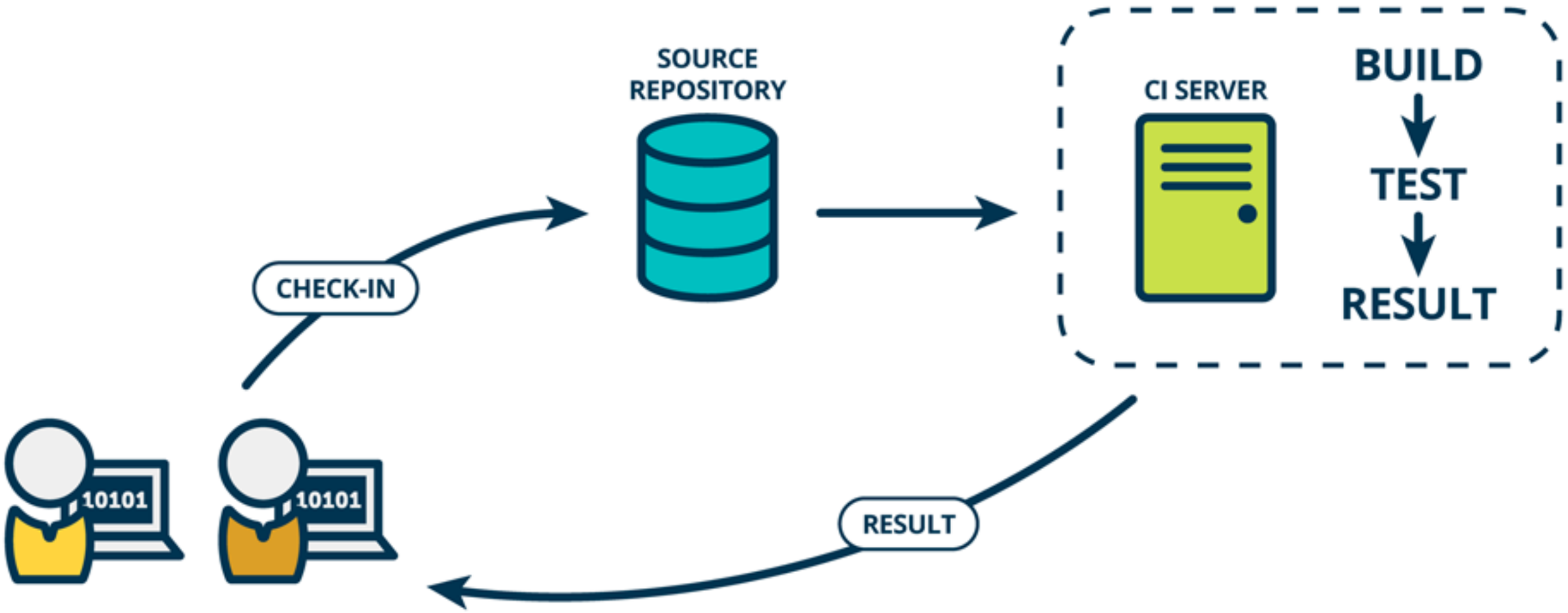
Practices of Continuous Integration

Improve quality and reduce risk

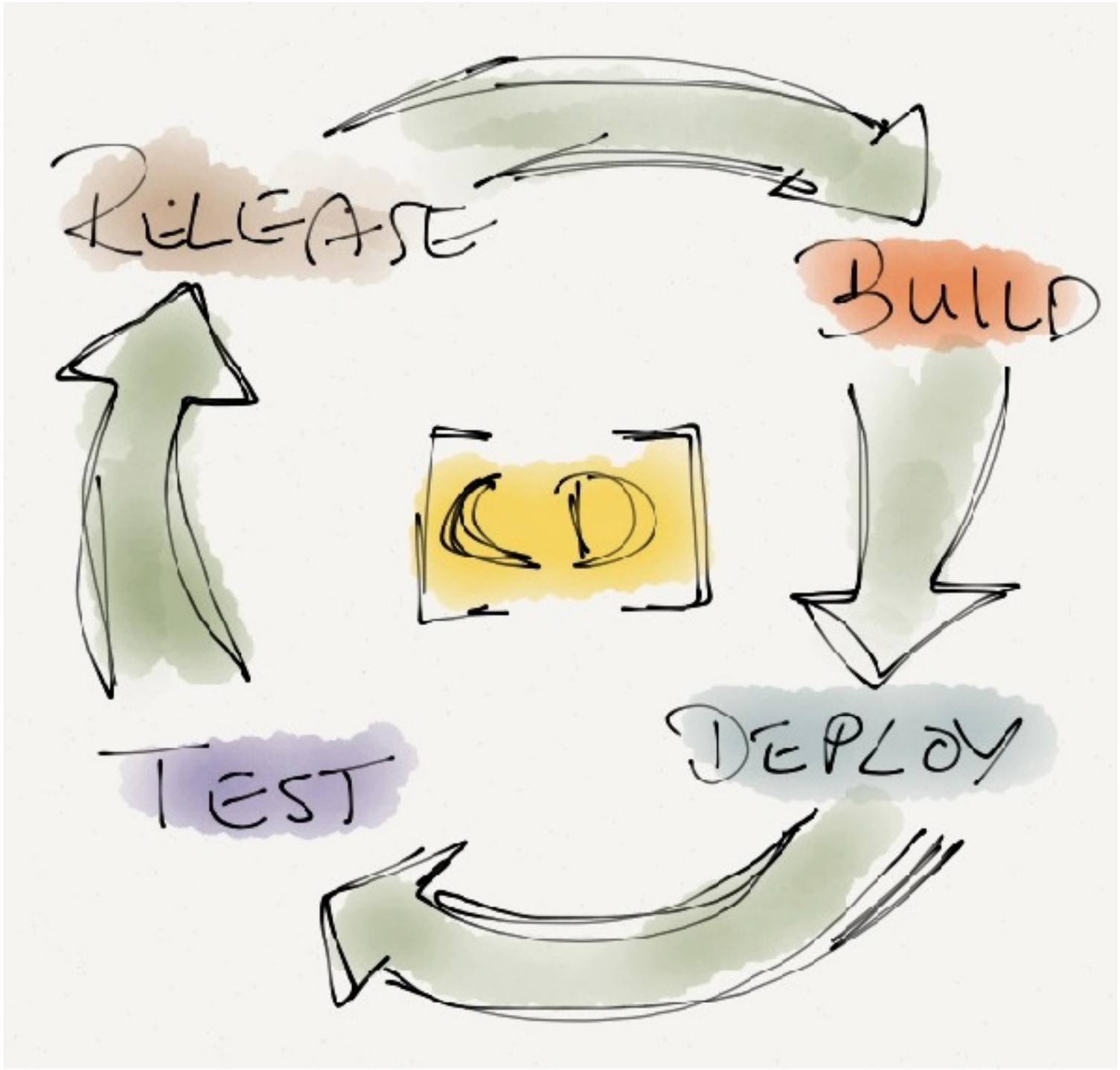




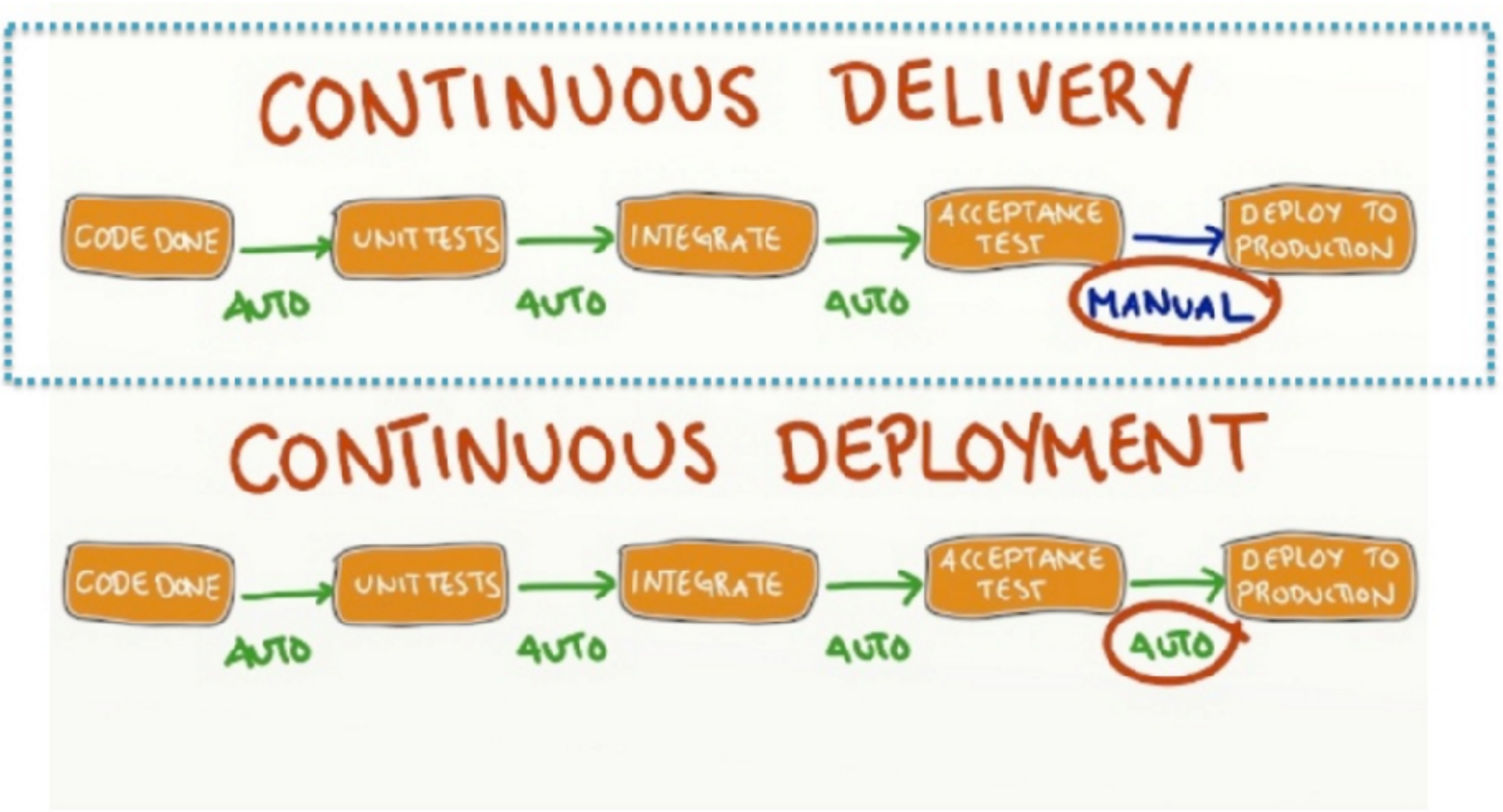
Continuous Integration



CD ?

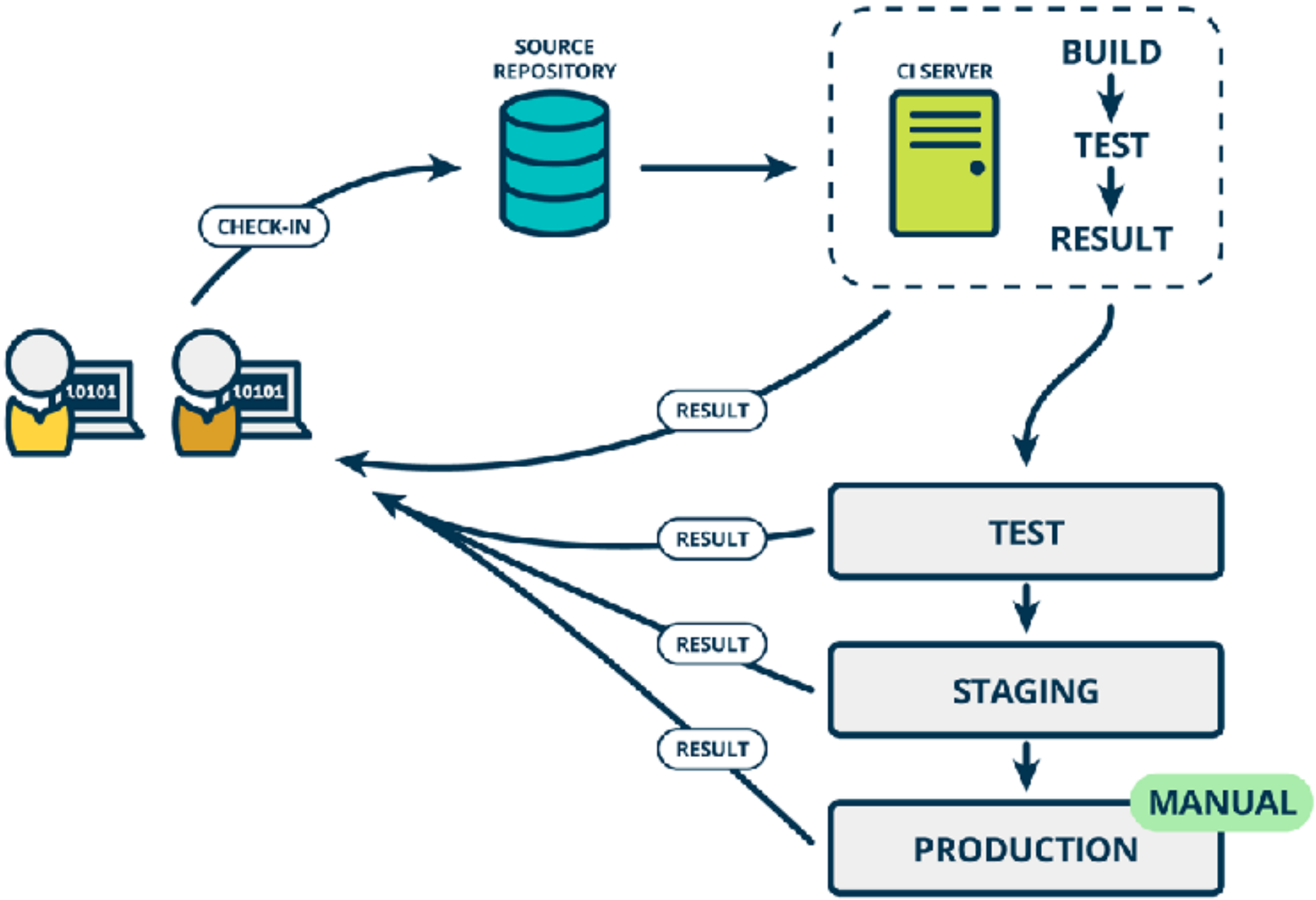


CD ?

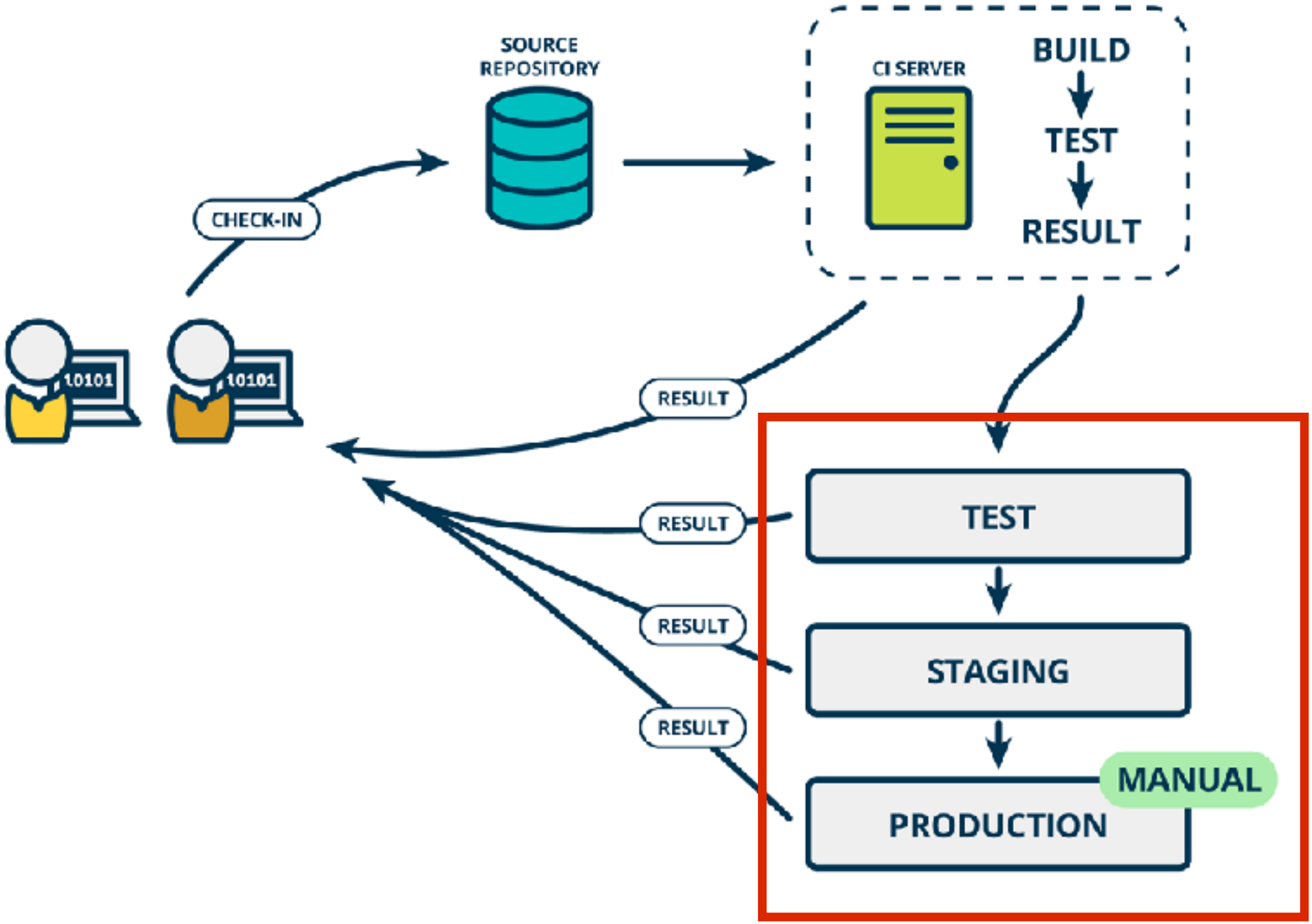


http://blog.crisp.se/2013/02/05/yassalsundman/continuous-delivery-vs-continuous-deployment

Continuous Delivery



Rise of DevOps



Continuous Integration

is a Software development **practices**

Practice 1

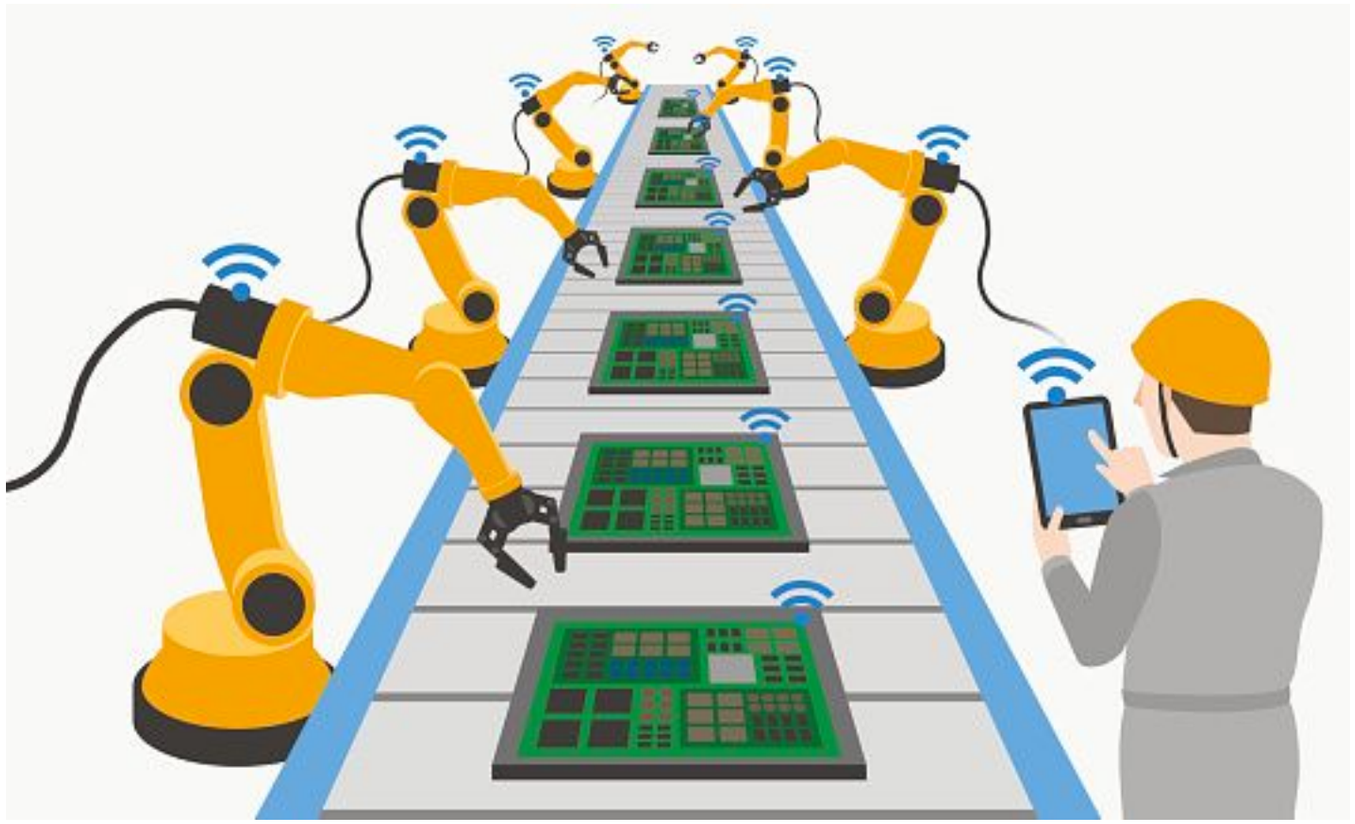
Maintain a single source repository

In general, you should store in source control everything you need to build anything

Practice 2

Automated the build

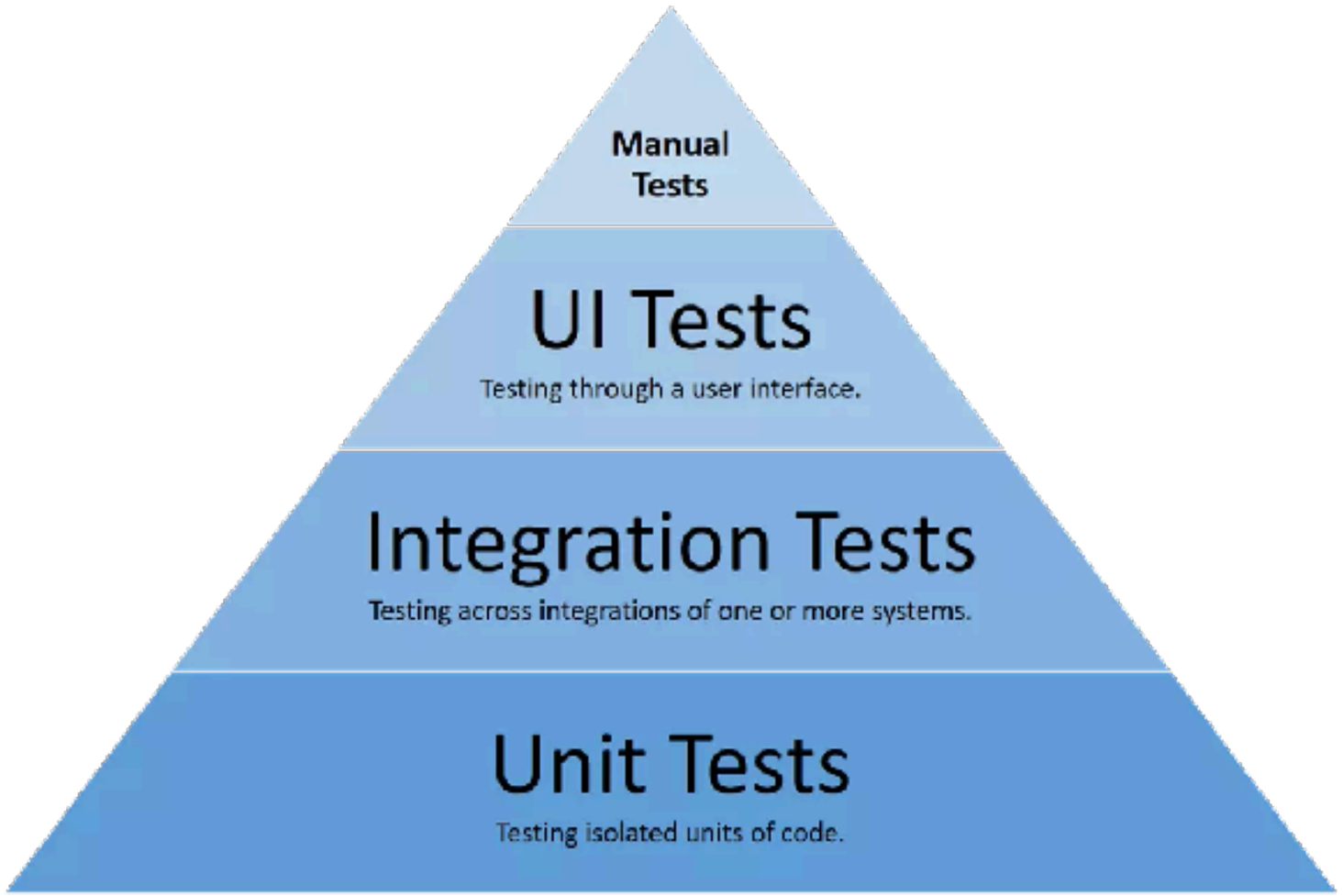
Automated environment for builds



Practice 3

Make your build **self-testing**

Build process => compile, linking and **testing**

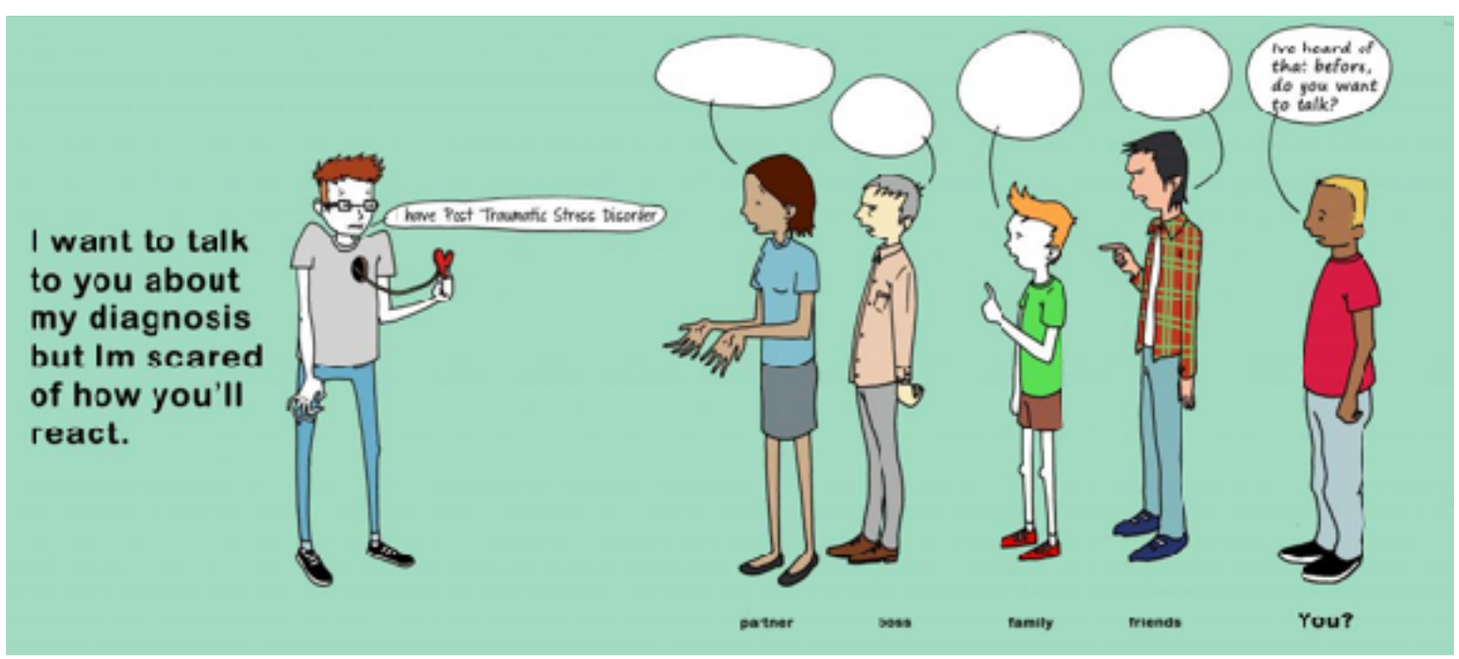


Practice 4

Everyone **commits** to the mainline everyday

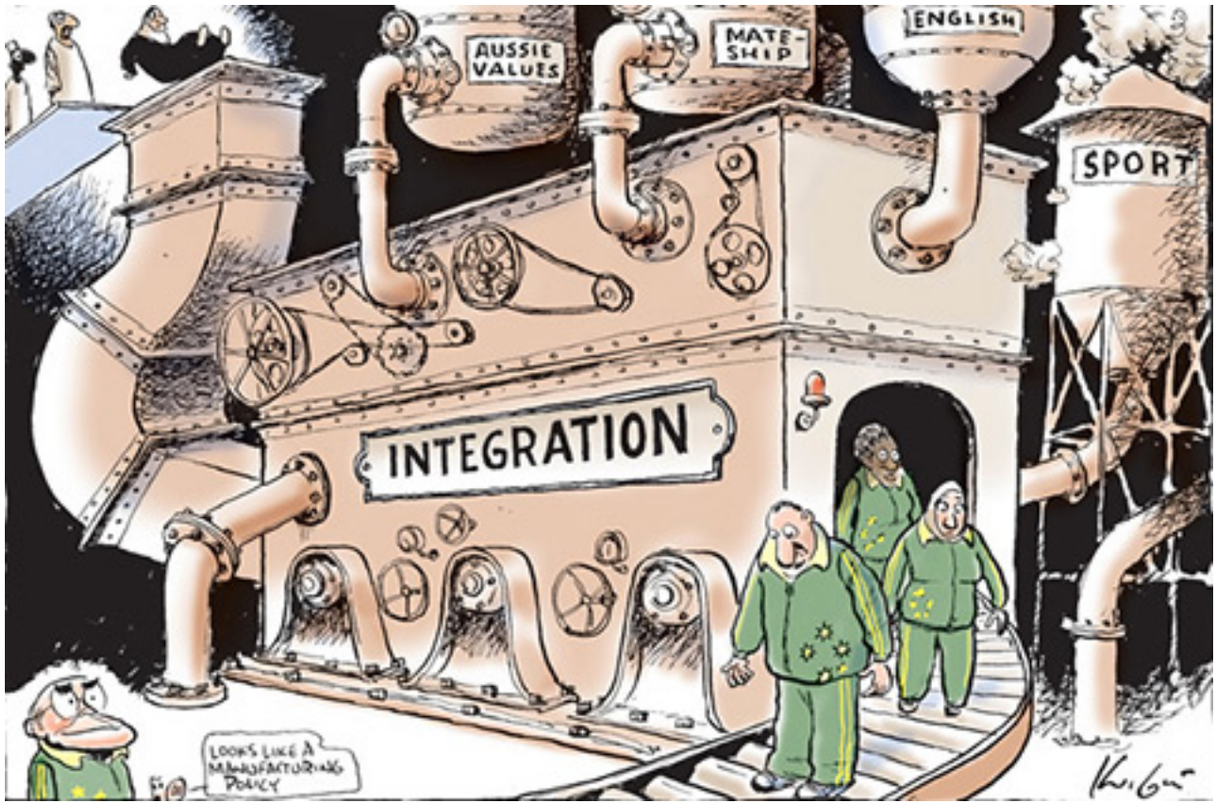
Integration is about **communication**

Integration allows developers to **tell** other developers



Practice 5

Every commits should build the mainline on an **Integration machine**



Nightly build is not enough for Continuous Integration

Practice 6

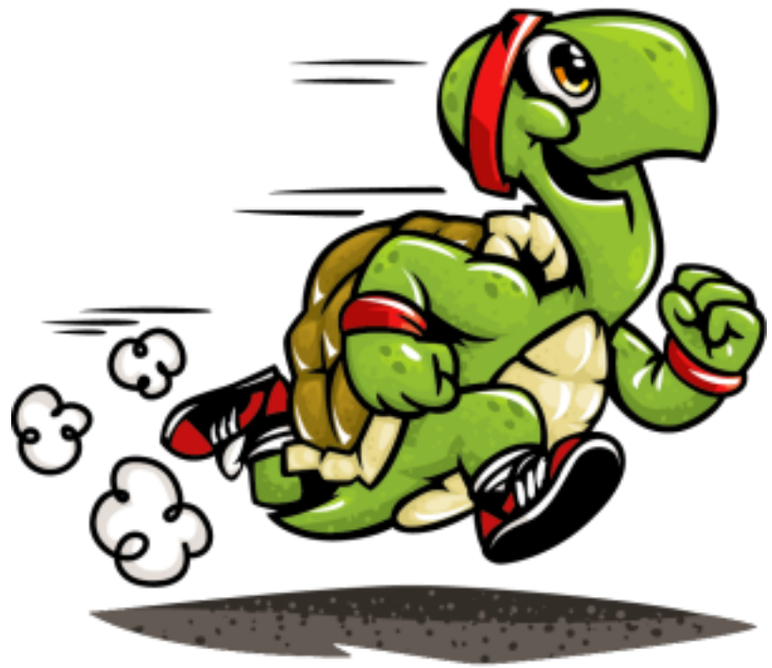
Fix broken builds **immediately**

"Nobody has a higher priority task than fixing the build"

Practice 7

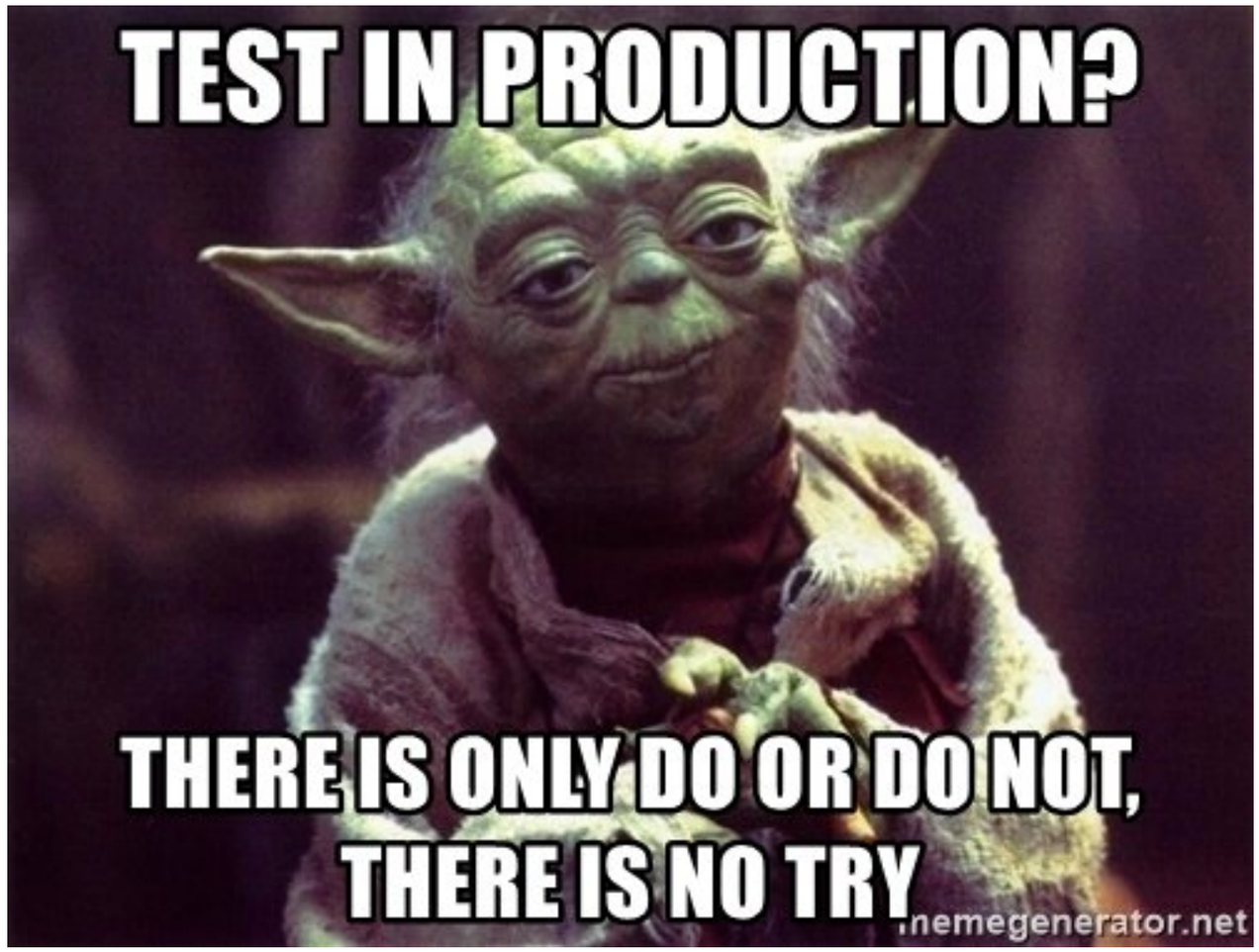
Keep the build **fast**

Continuous Integration is to provide rapid feedback



Practice 8

Test in clone of the **Production** environment



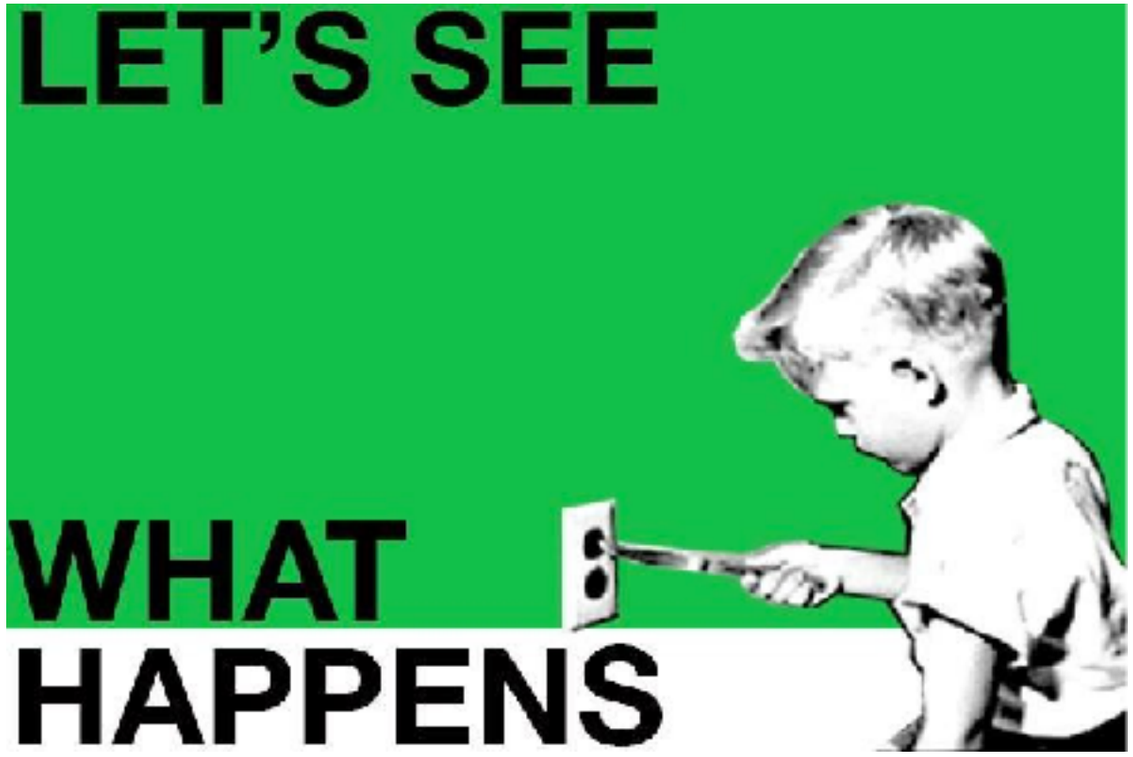
Practice 9

Make it easy for anyone to get
the latest executable

Make sure well known place where people can find

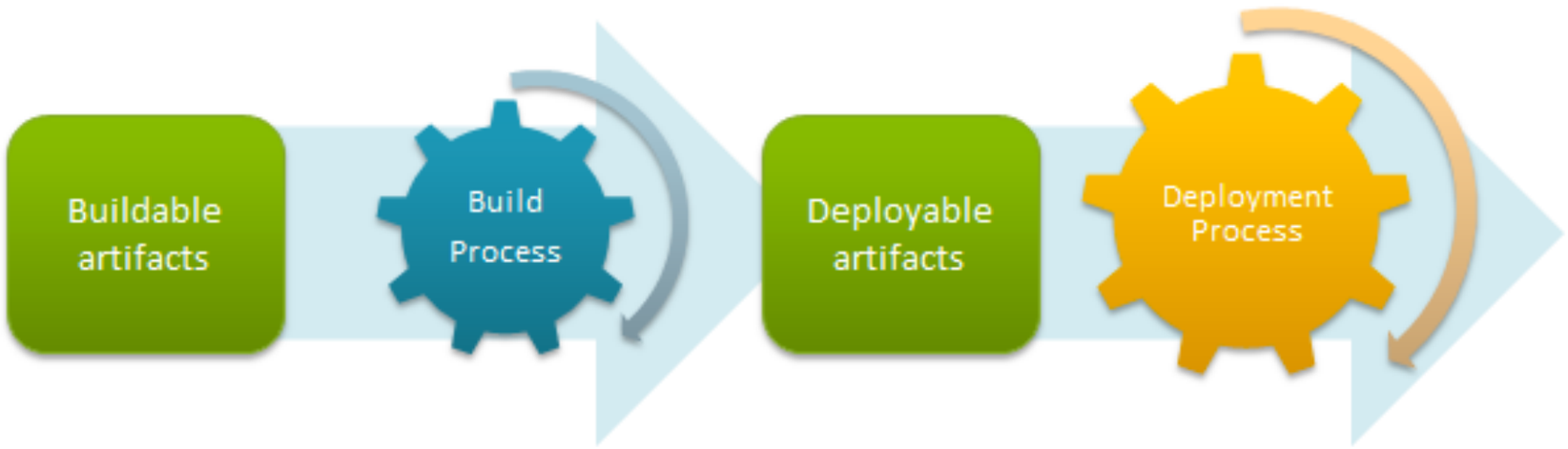
Practice 10

Everyone can see what's happening
Easier to see the state of the system and changes
Show the good information



Practice 11

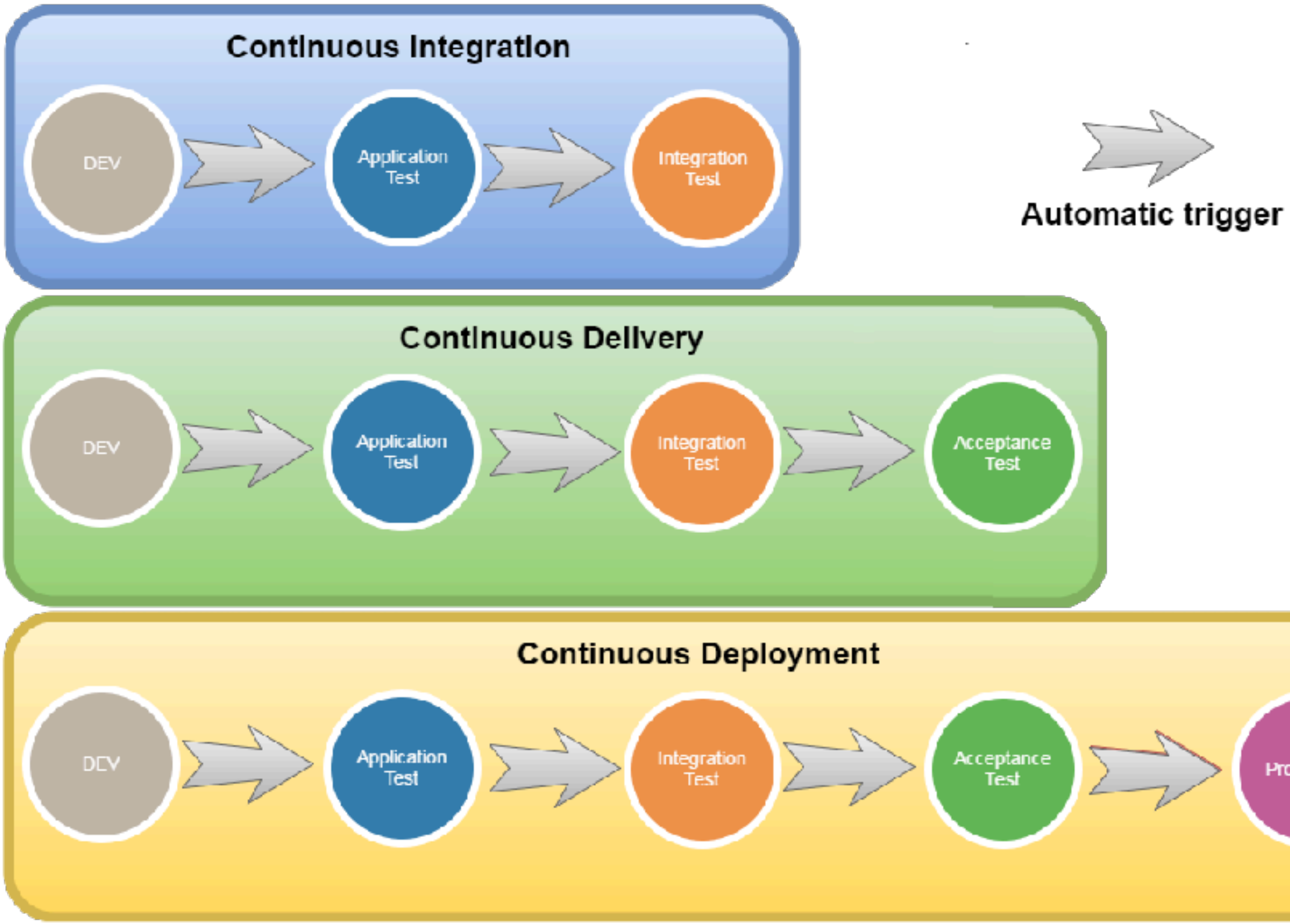
Automated deployment



Continuous Delivery

Continuous Delivery

Use version control for all production artifacts
Automate your deployment process
Implement continuous integration (CI)
Use trunk-based development methods
Implement test automation
Support test data management
Integrate security into software development process



How to achieve the CI ?

1. Use good version control

Local

Centralize

Distributed

2. Choose Branch strategy

Main only

Development isolation

Feature isolation

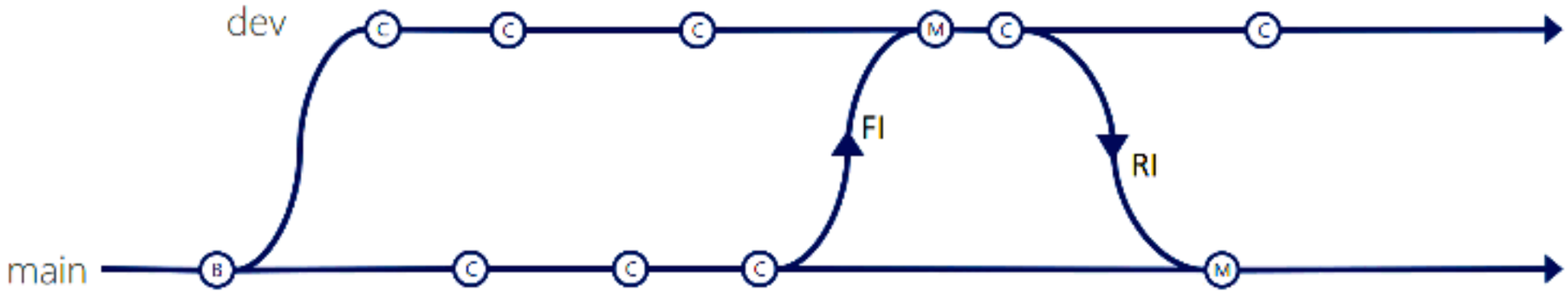
Release isolation

Service and Release isolation

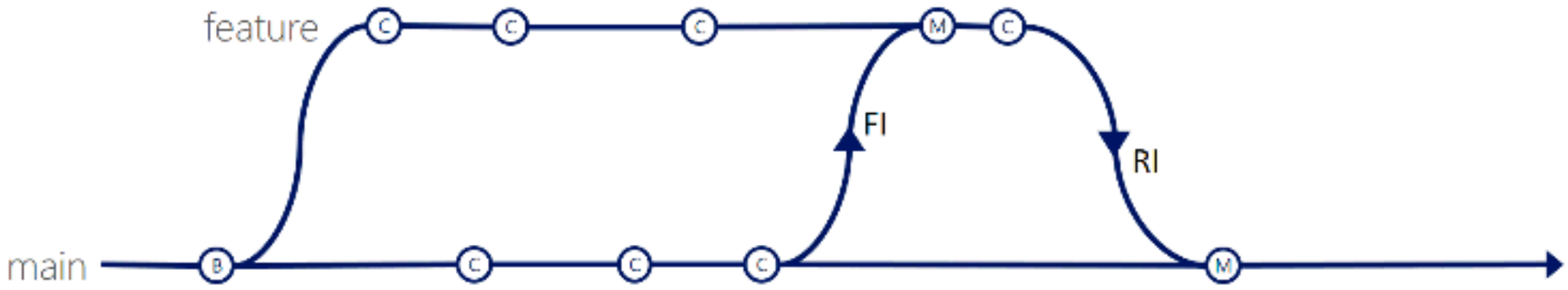
Service, Hotfix and Release isolation

Main only

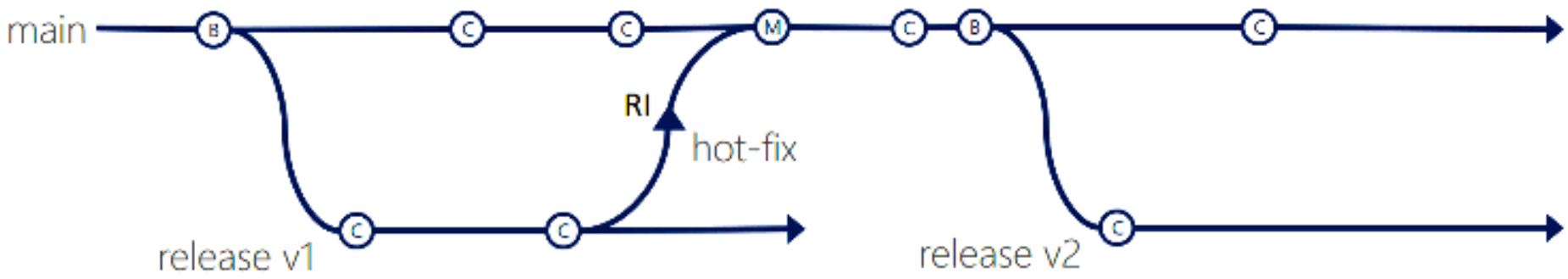
Development isolation



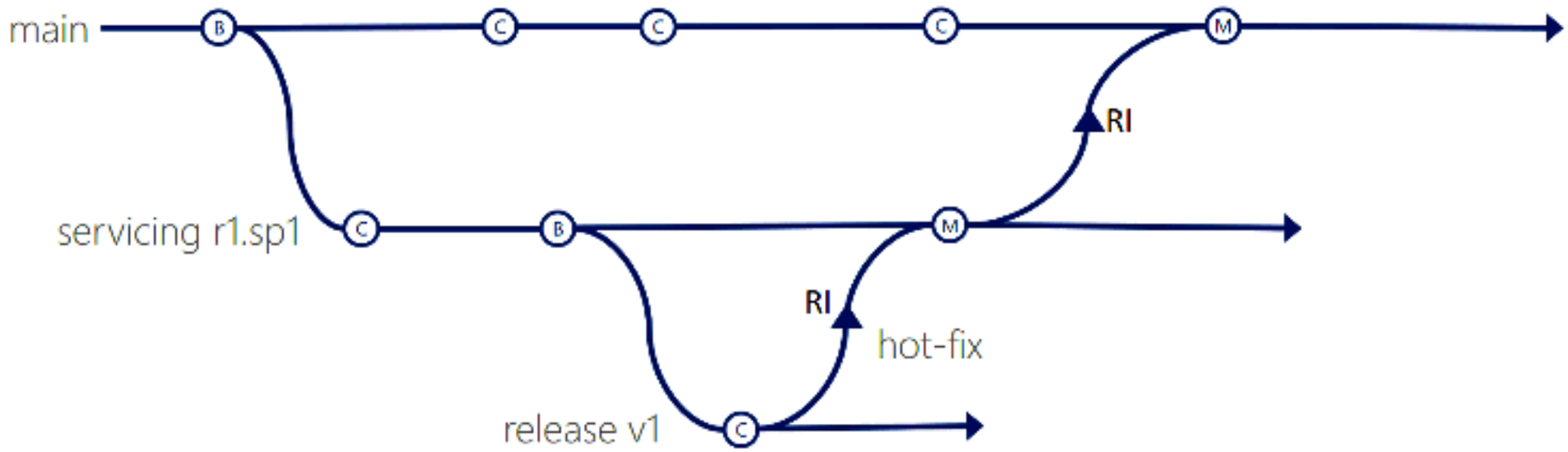
Feature isolation



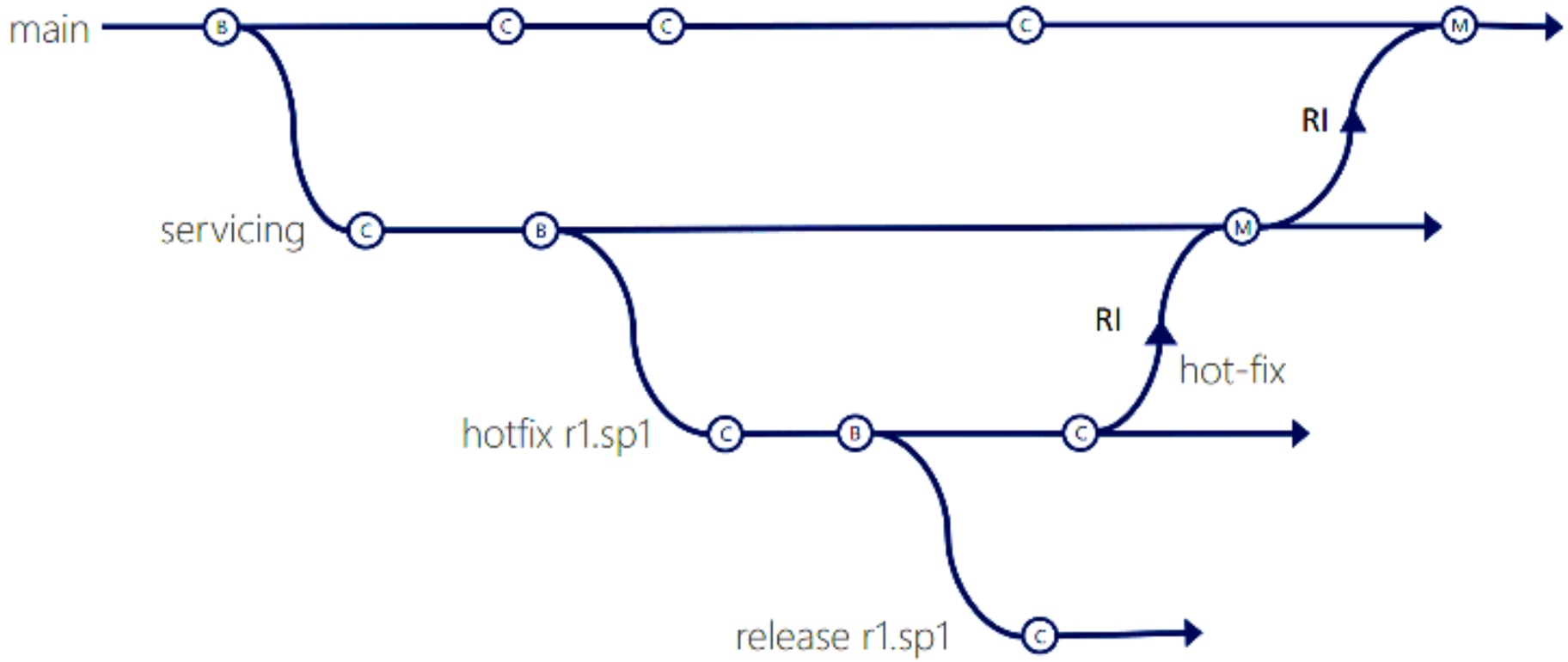
Release isolation



Service and Release isolation



Service, Hotfix, Release isolation

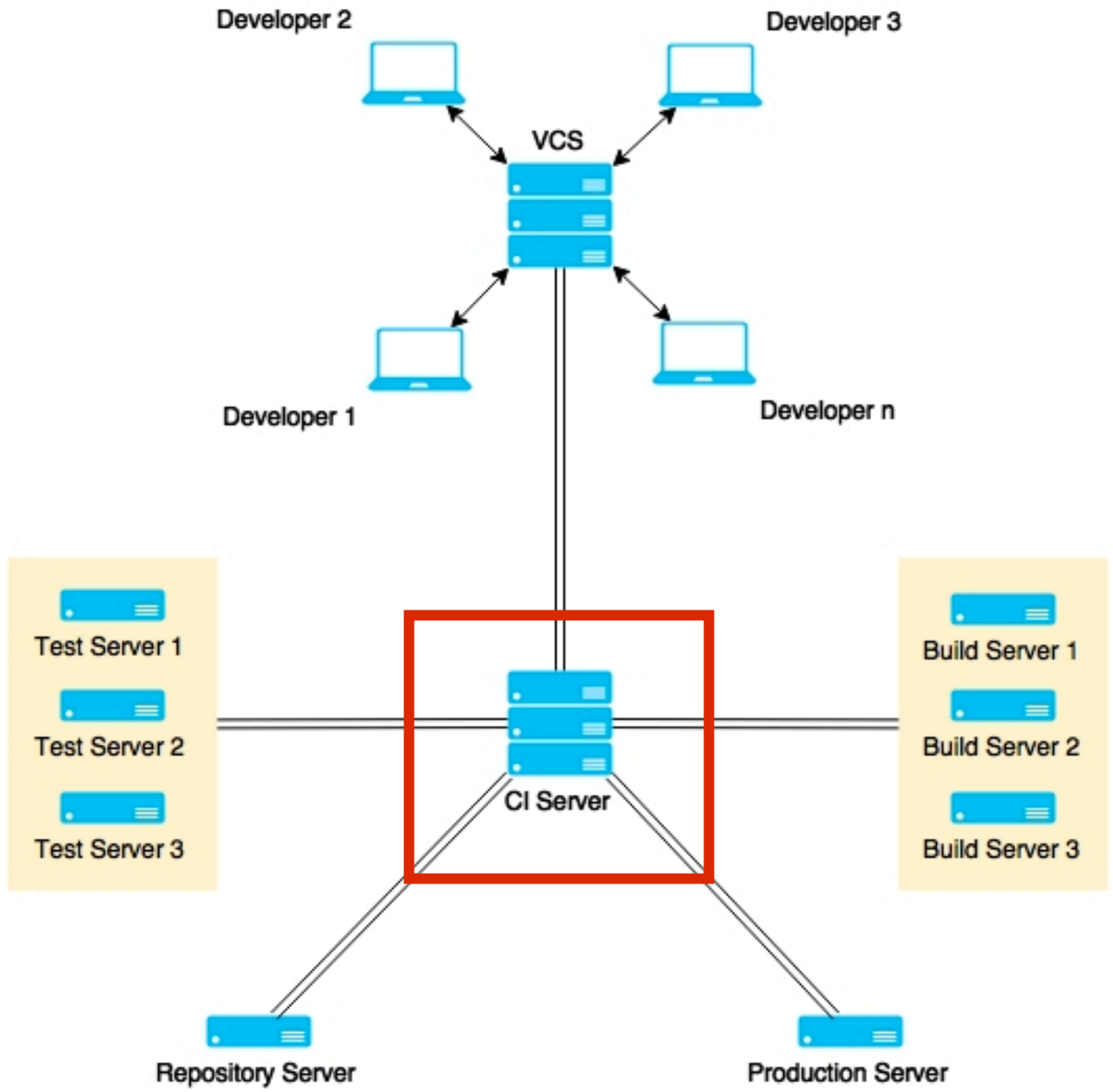


Validate, Validate and Validate

Suggestion

Keeping branches short-lived, merge conflicts are keep to as few as possible

3. Use good CI tool



Jenkins

Bamboo

TeamCity

go

Visual Studio Team Foundation Server

Hudson

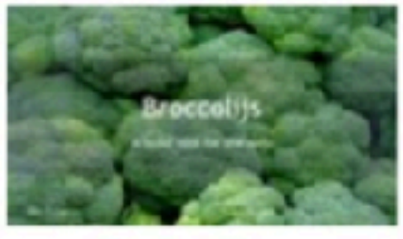
Travis

wercker

circleci

4. Use good build tool

- Javascript
  - Gulp, Grunt, Brocolli



- C#/.NET
  - Nant, MSBuild

- Java/JVM
  - Ant, Maven, Gradle, SBT, Leiningen

More …

Use static code analysis

Automated testing

Automated deployment

People discipline/habit

“Behind every successful agile project, there is a

Continuous Integration Server”

Let’s start with Jenkins

Application and framework to manage and monitor the executable of **repeated tasks**

Jenkins

https://jenkins.io/

Why Jenkins ?

Easy !!

Extensible

Scalable

Opensource

Large community

Lot of plugins

Who use Jenkins ?

We thank the following organizations for their major commitments to support the Jenkins project.

Microsoft · cloudbees · OSL OPEN SOURCE LAB

rackspace redhat

We thank the following organizations for their support of the Jenkins project through free and/or open source licensing programs.

Atlassian Datadog JFrog Mac Cloud PagerDuty XMission

https://jenkins.io/

Hardware requirements

For Jenkins server

RAM +2GB

More CPU

More Disk

Installation

Jenkins in containers

Apache Tomcat

Jetty

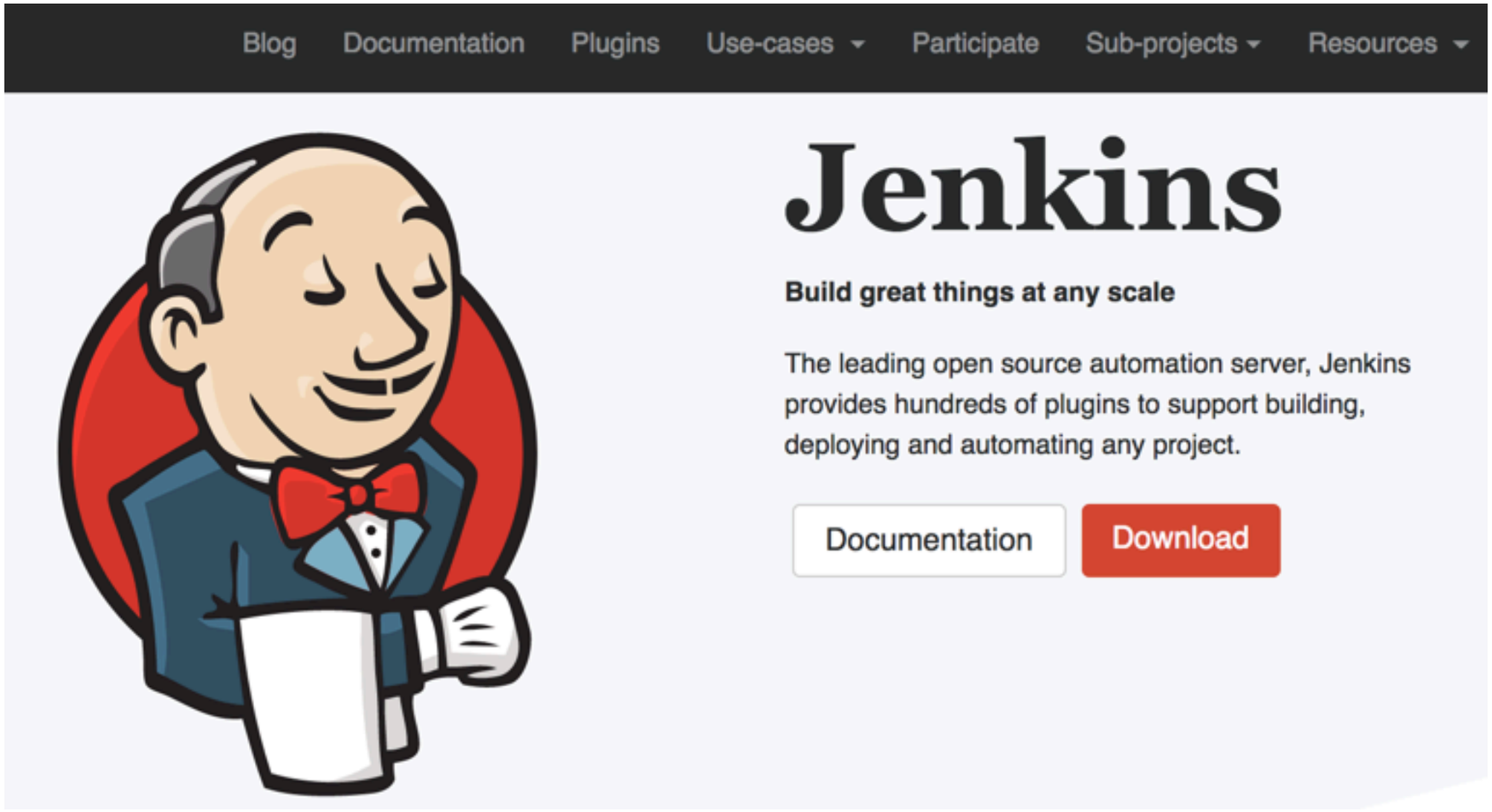
JBoss

Websphere

WebLogic

Glassfish

Download Jenkins



https://jenkins.io/

Use Long Term Support (LTS)

Getting started with Jenkins

The Jenkins project produces two release lines, LTS and weekly. Depending on your organization's needs, one may be preferred over the other.

Both release lines are distributed as `.war` files, native packages, installers, and Docker containers.

Long-term Support (LTS)

LTS (Long-Term Support) releases are chosen every 12 weeks from the stream of regular releases as the stable release for that time period. Learn more...

Changelog | Upgrade Guide | Past Releases

Deploy Jenkins 2.46.3

Deploy to Azure

Download Jenkins 2.46.3 for:

- Docker
- FreeBSD

Weekly

A new release is produced weekly to deliver bug fixes and features to users and plugin developers.

Changelog | Past Releases

Download Jenkins 2.65 for:

- Arch Linux
- Docker
- FreeBSD
- Gentoo

Start Jenkins

$java -jar jenkins.war

Default port of server is **8080**

```
 org.eclipse.jetty.server.AbstractConnector doStart
ector@3e2fc448{HTTP/1.1,[http/1.1]}{0.0.0.0:8080}
 org.eclipse.jetty.server.Server doStart
```

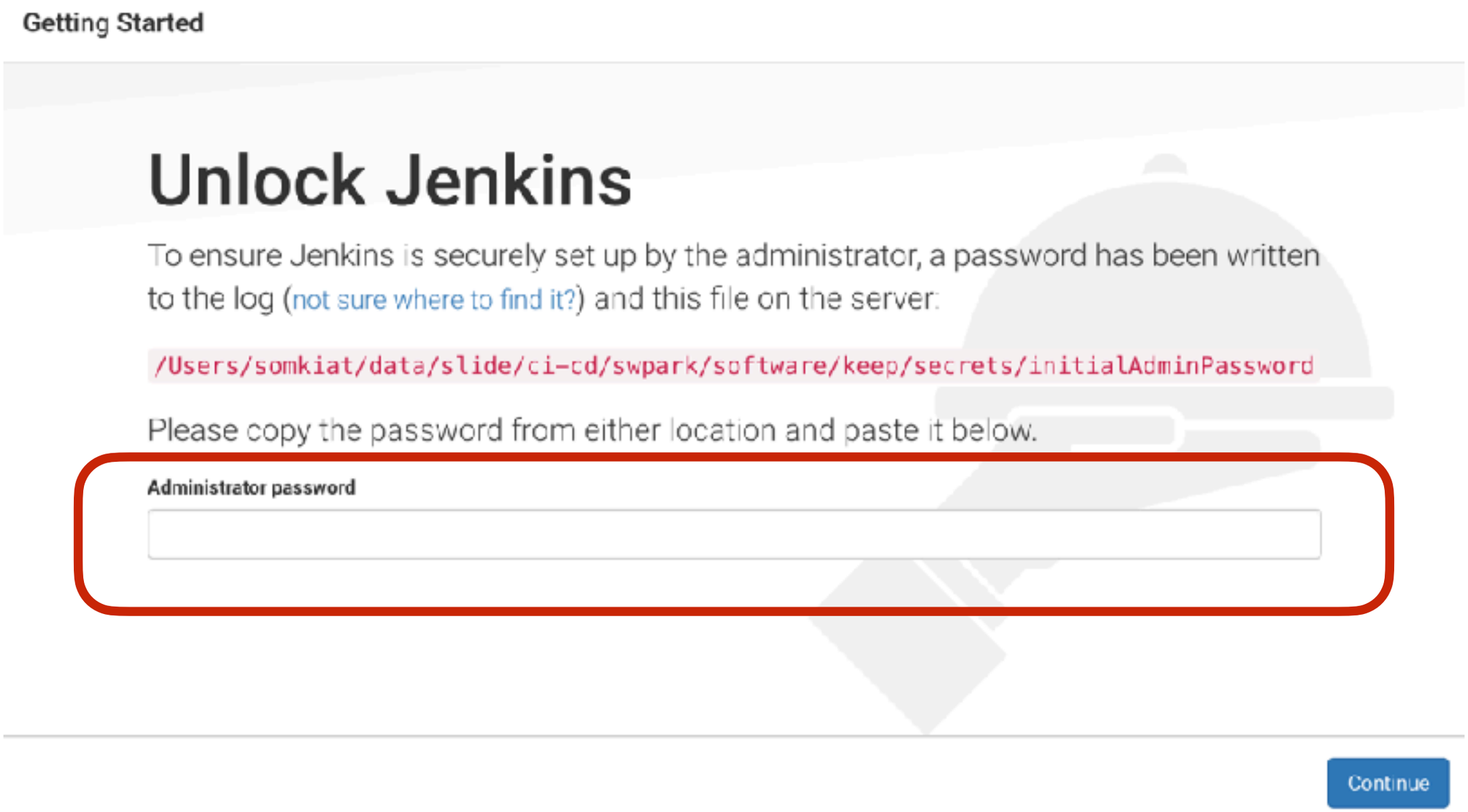
```
 winstone.Logger logInternal
gine v4.0 running: controlPort=disabled
 jenkins.InitReactorRunner$1 onAttained
ion
 jenkins.InitReactorRunner$1 onAttained

 jenkins.InitReactorRunner$1 onAttained
```

Change port of Jenkins

$java -jar jenkins.war **--httpPort=<port>**

Open in browser

http://localhost:8080

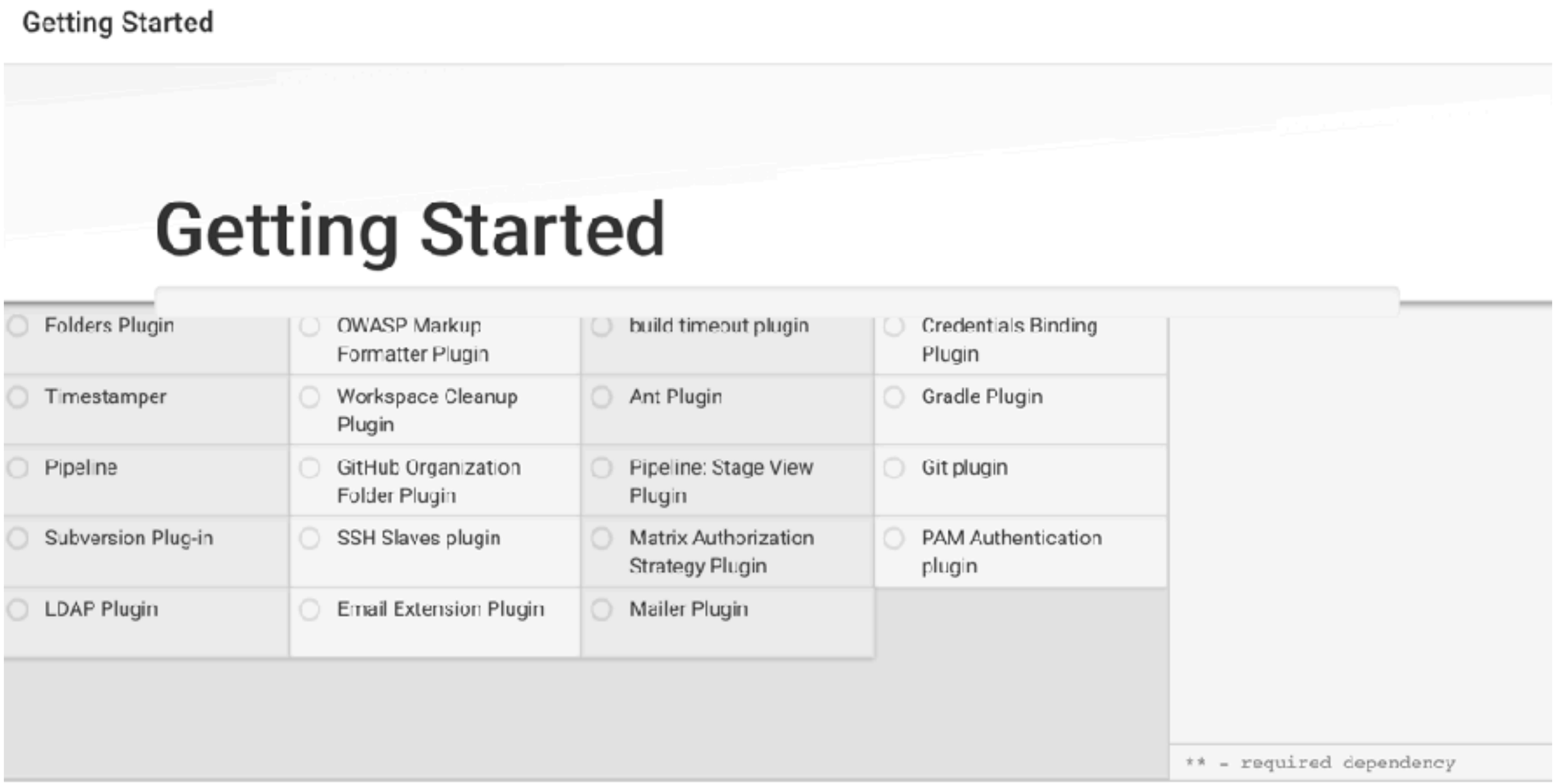


Copy password from console

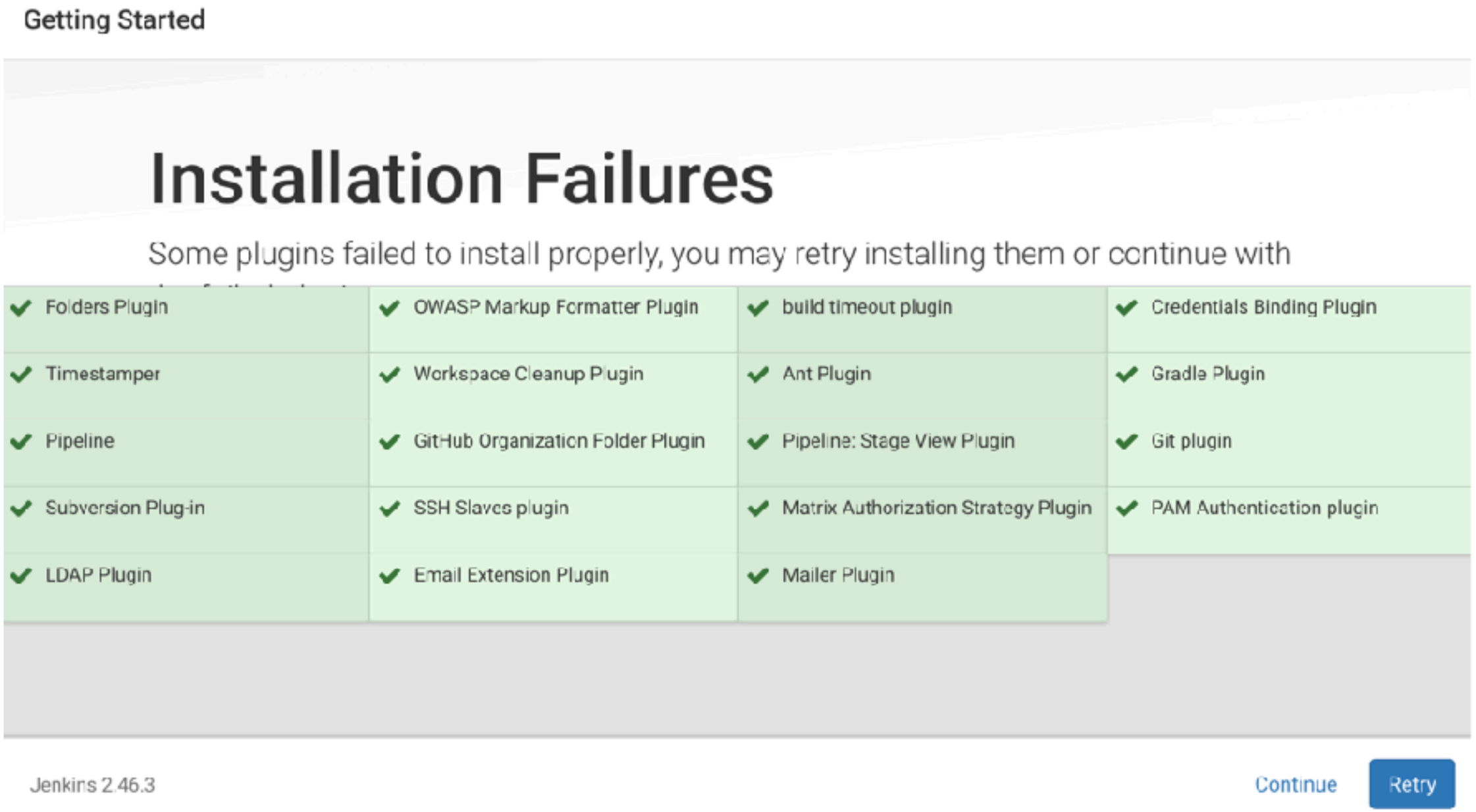
```
************************************************************
************************************************************
************************************************************

Jenkins initial setup is required. An admin user has been creat
.
Please use the following password to proceed to installation:

a4b3a5231b8048419192d0c5afd3fce8

This may also be found at: /Users/somkiat/data/slide/ci-cd/swpa
itialAdminPassword

************************************************************
************************************************************
************************************************************
```

Customize plugins

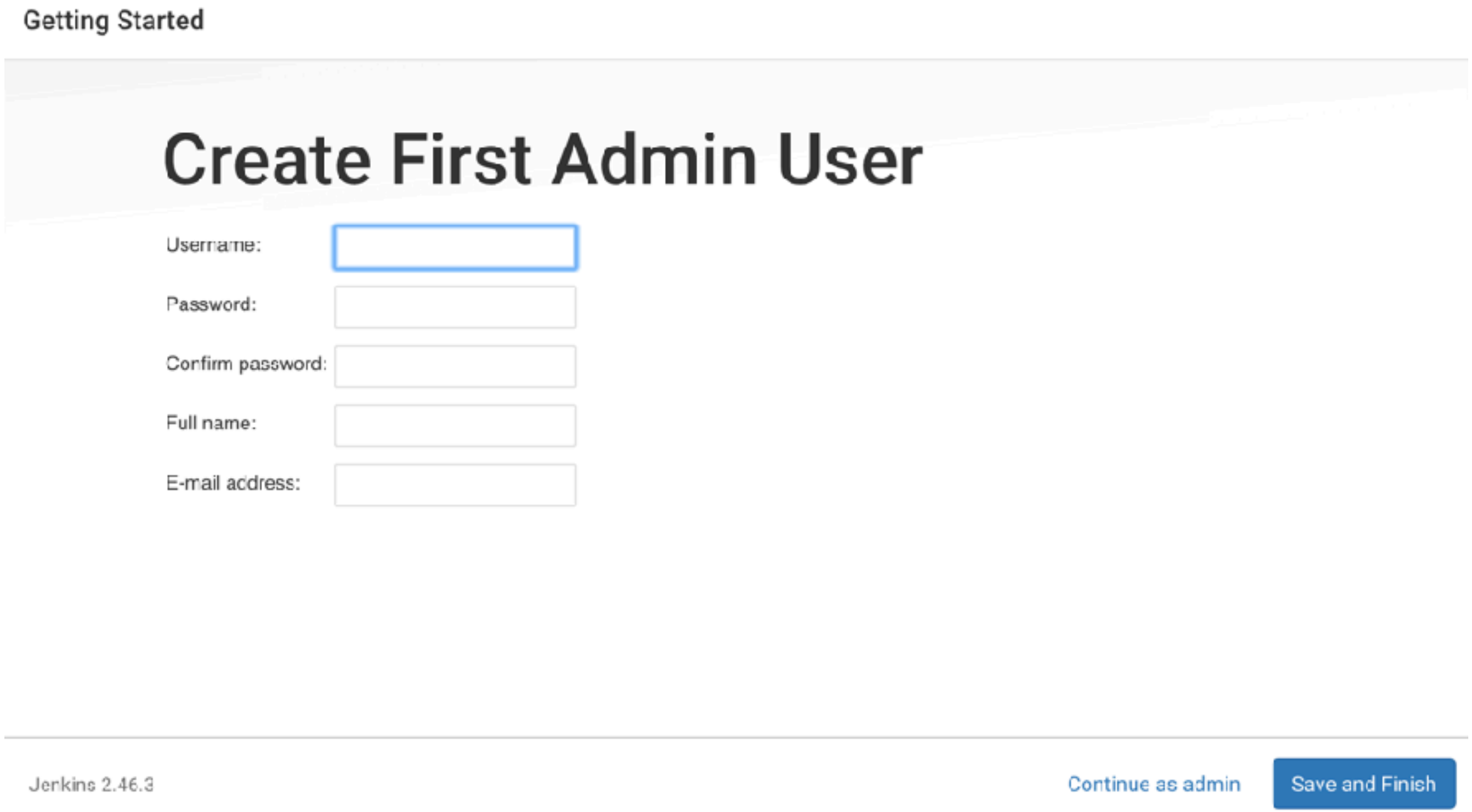
Waiting



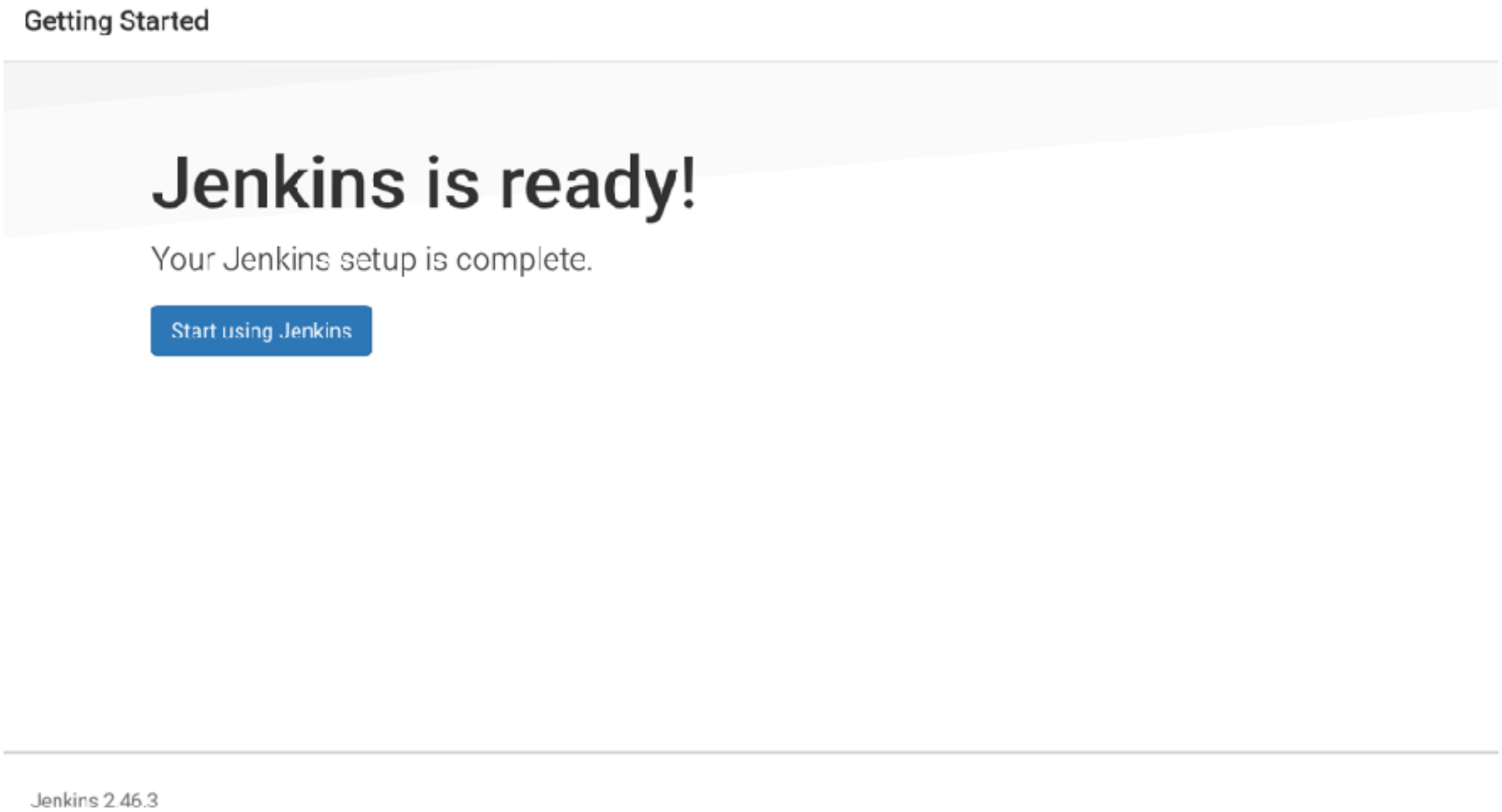
Success



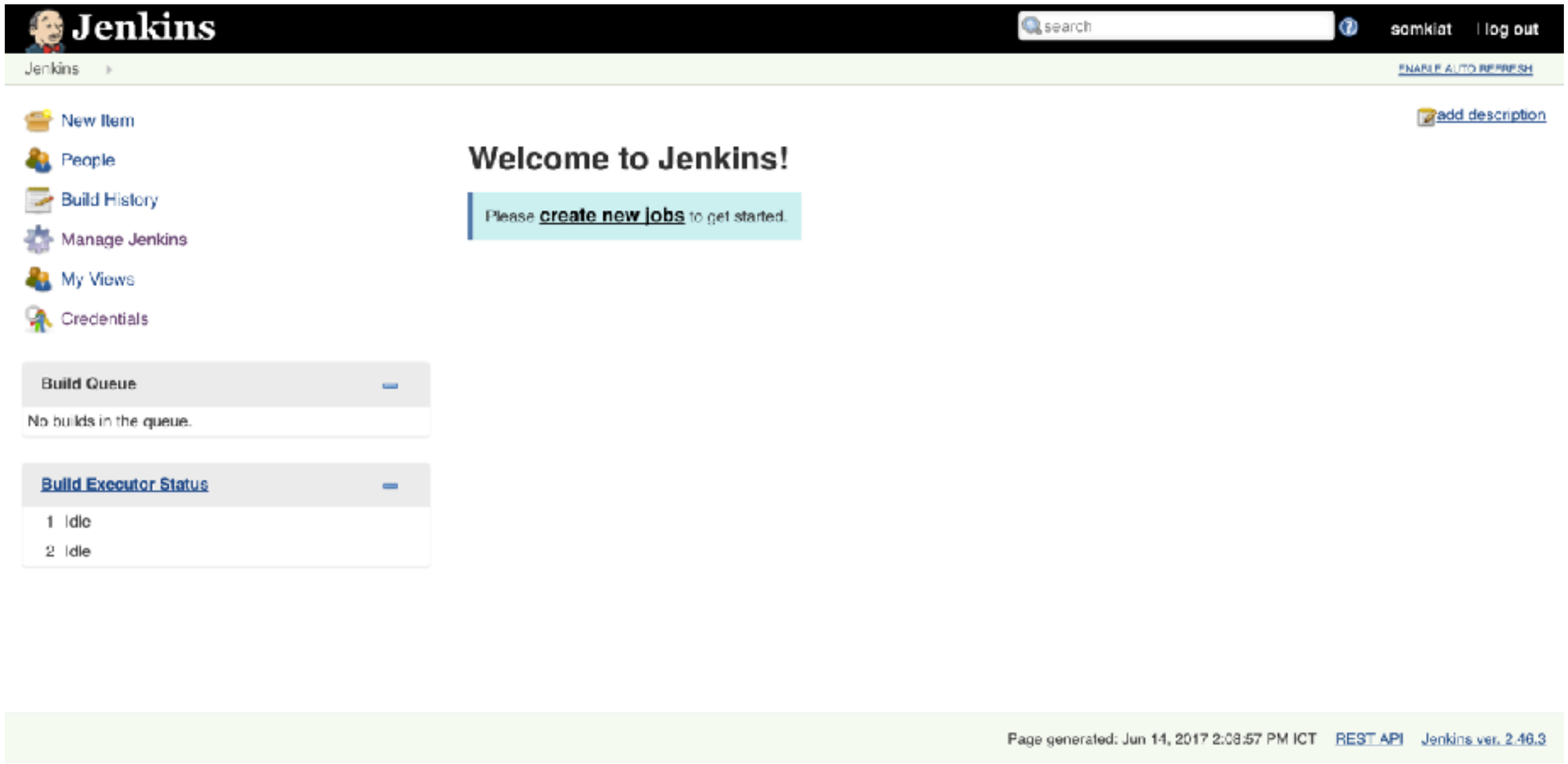
Create a new user



Ready to use



Welcome to Jenkins



About Jenkins's HOME

Default of **JENKINS_HOME** is

<path of user>/.jenkins

About Jenkins's HOME

Data in JENKINS_HOME

```
/Users/somkiat/.jenkins
├── config.xml
├── failed-boot-attempts.txt
├── hudson.model.UpdateCenter.xml
├── jenkins.CLI.xml
├── jenkins.install.UpgradeWizard.state
├── jobs
├── logs
├── nodeMonitors.xml
├── nodes
├── plugins
├── queue.xml
├── queue.xml.bak
├── secret.key
├── secret.key.not-so-secret
├── secrets
├── updates
├── userContent
├── users
└── war
```

About Jenkins's HOME

File and Folder name	Description
config.xml	All about configuration
jobs	Keep all jobs/project
plugins	Keep all plugins
nodes	Keep all nodes
logs	Keep all logs

Change Jenkins's HOME

For Windows

set JENKINS_HOME=<your path>

For Linux/Mac

export JENKINS_HOME=<your path>

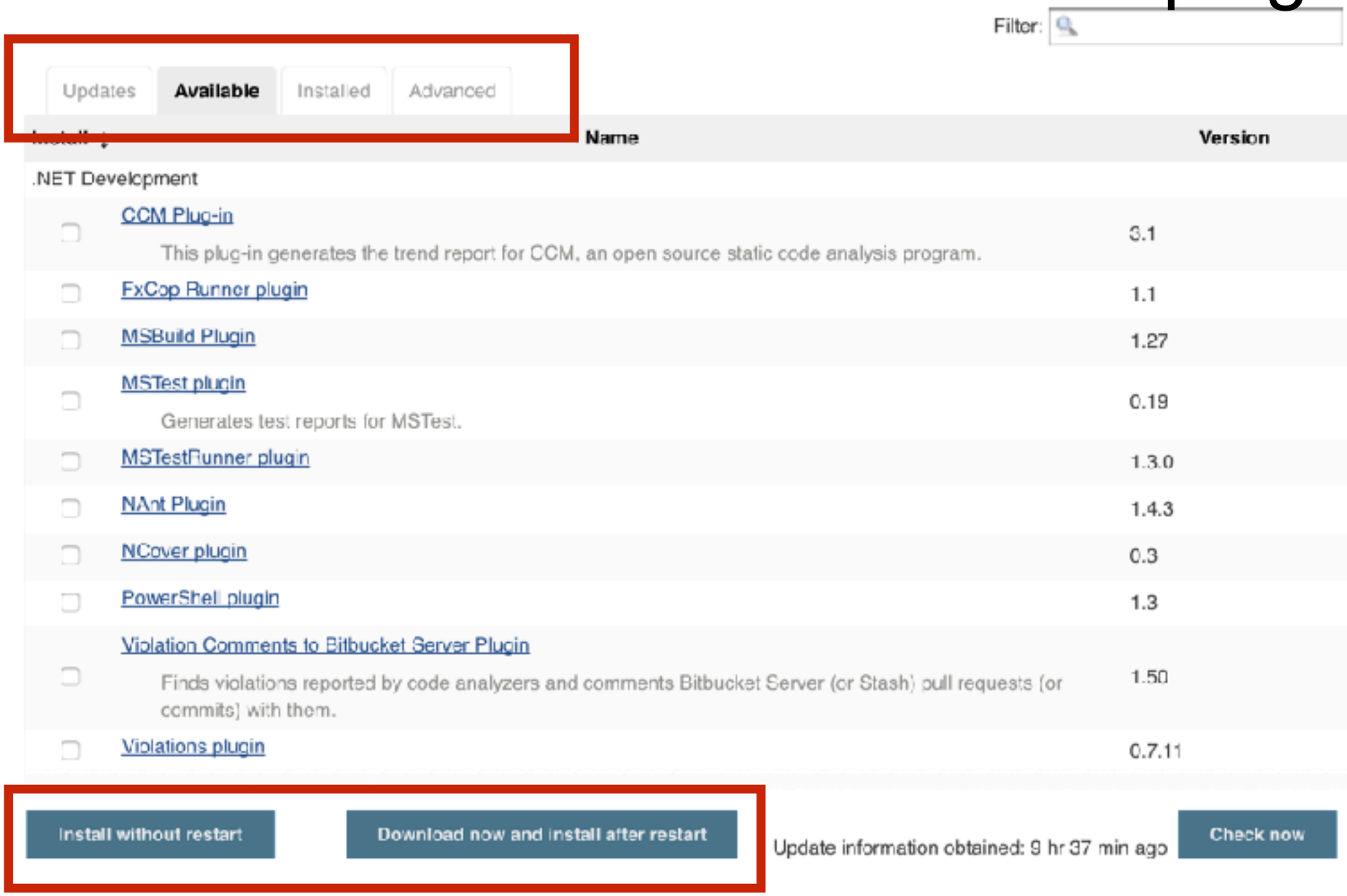
try to restart Jenkins …

Finds Jenkins's plugin

https://plugins.jenkins.io/

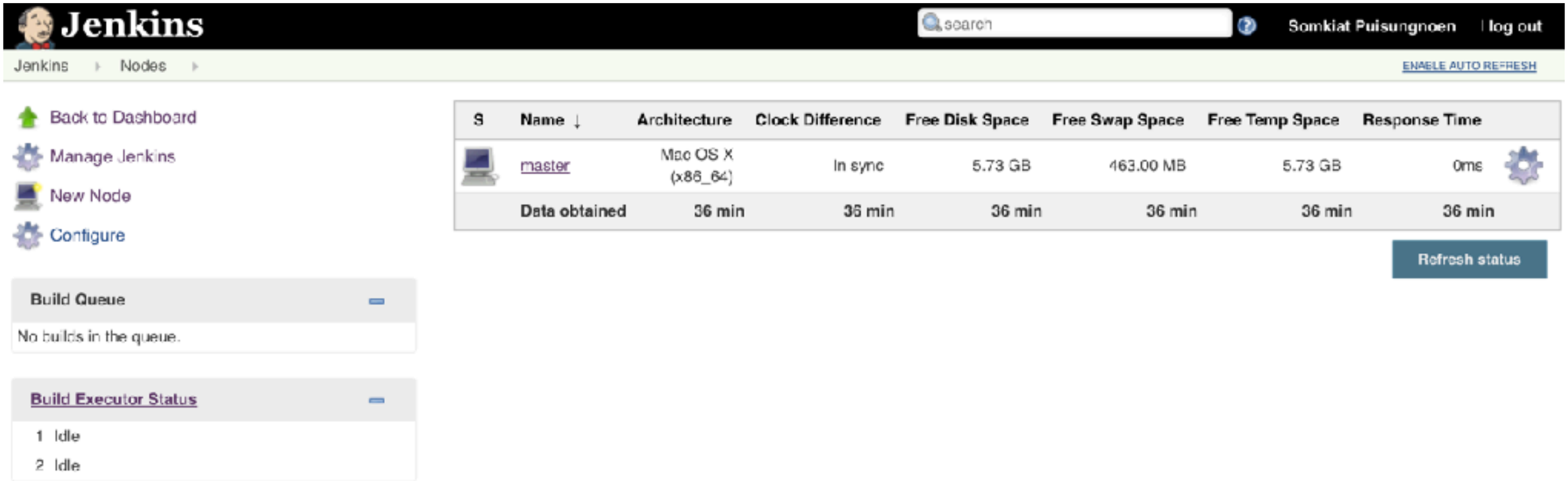
Try to install a first plugin

Choose Available tab and select a plugin



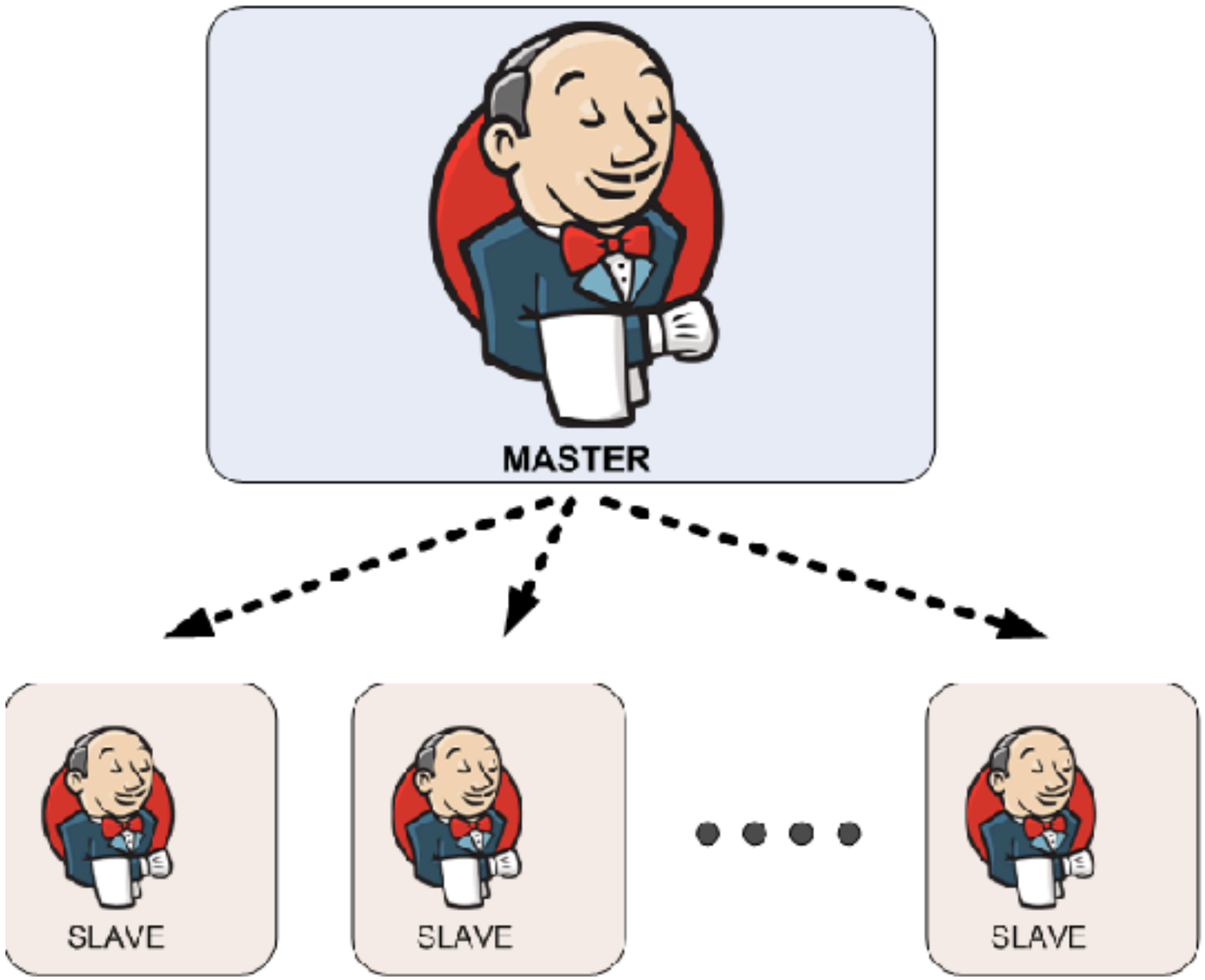
Manage Nodes

Add, remove, status of nodes



Manage Nodes

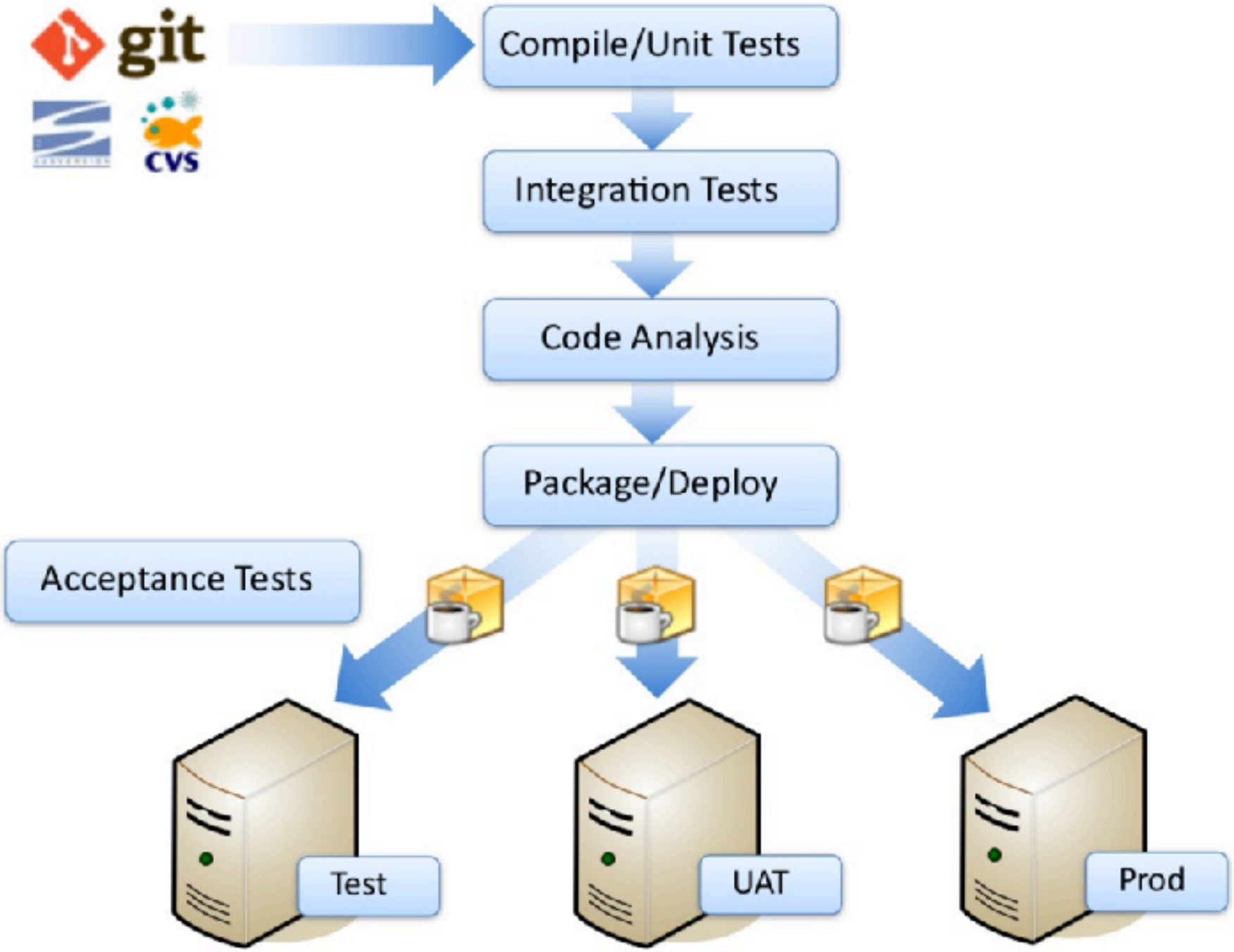
Master-slave concept to scale Jenkins



Create a first Job

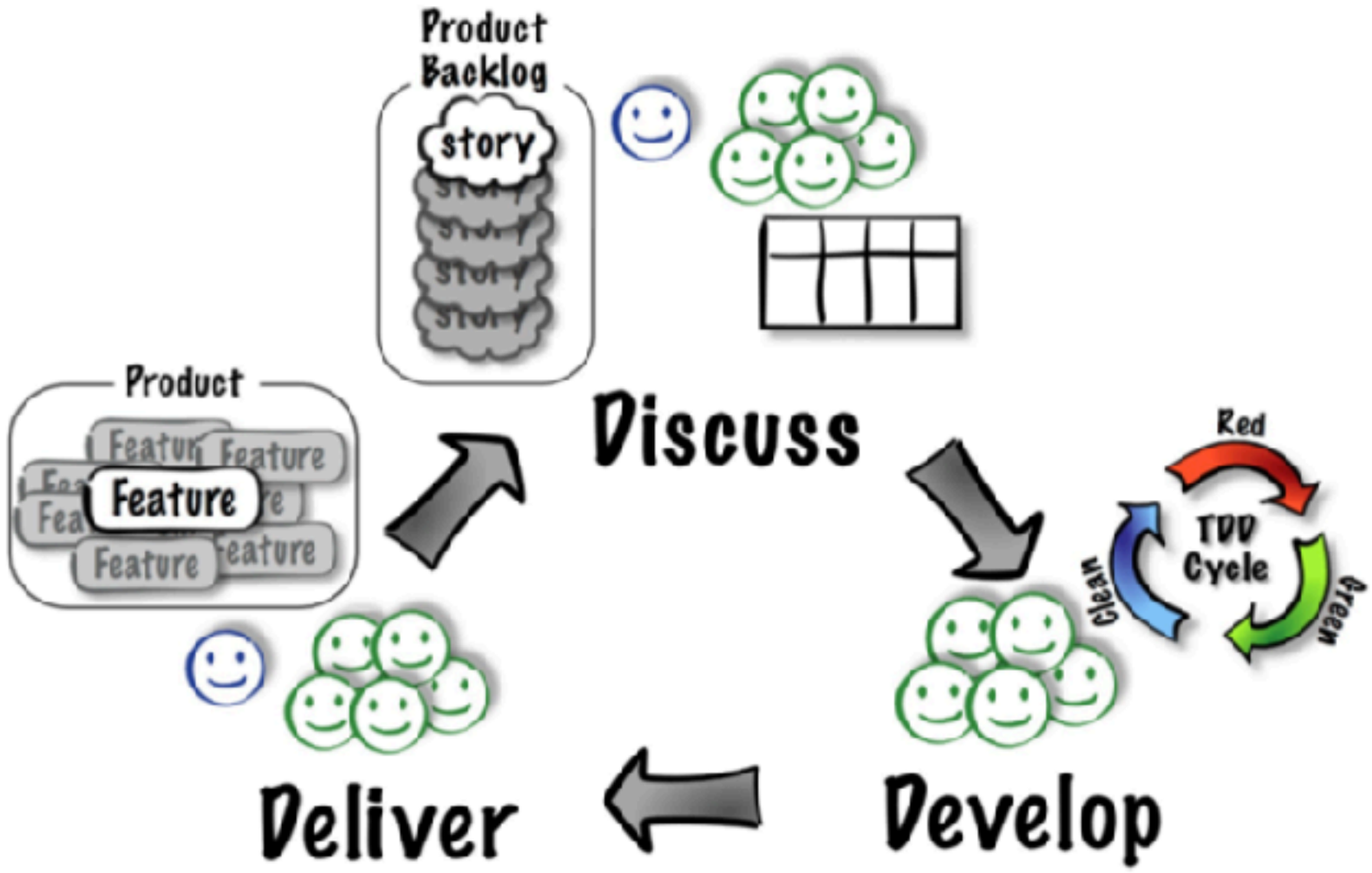
Workshop with Java Web Application

Pipeline of this project



Suggestion

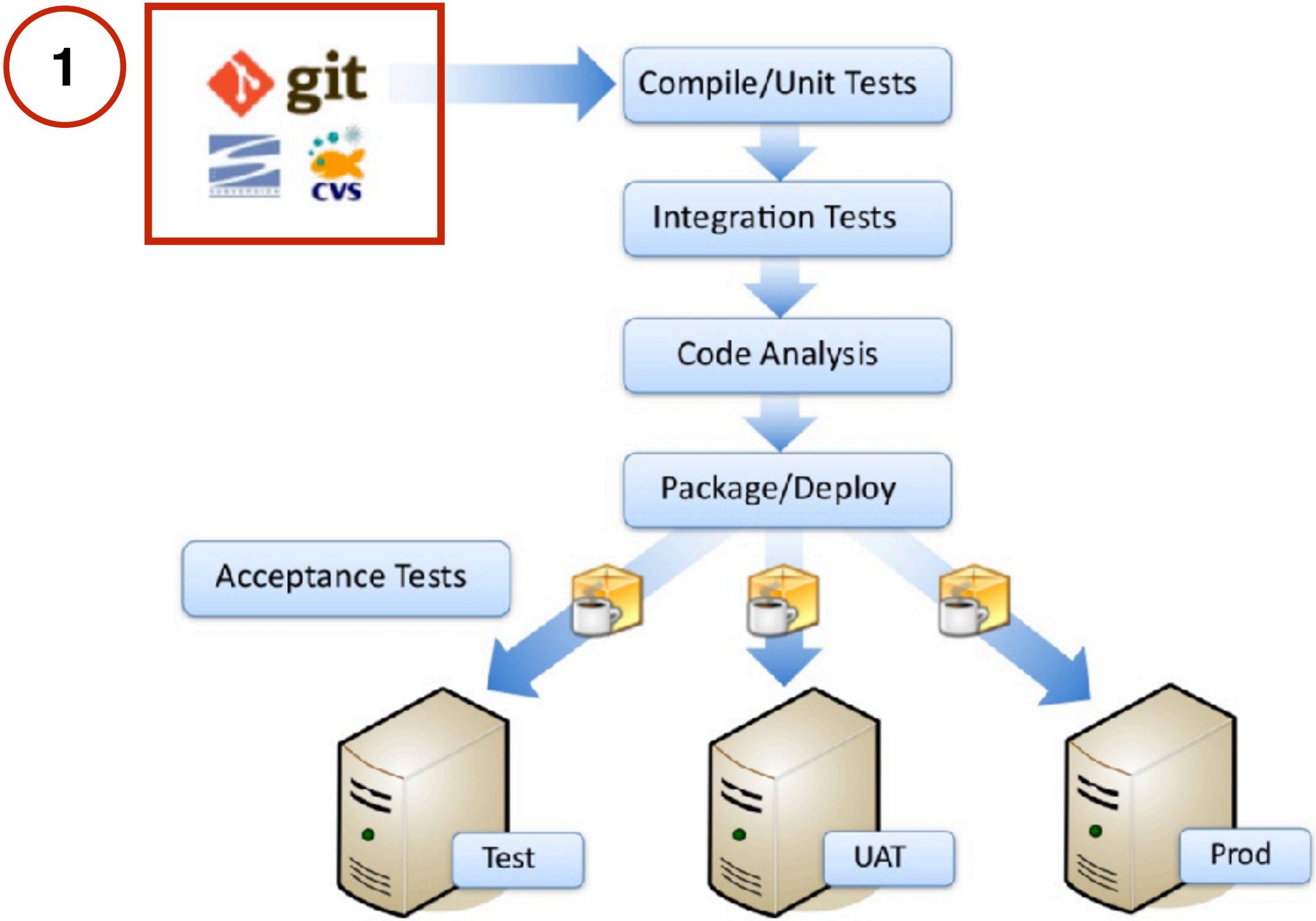
"Must setup/configuration CI server before the first feature"

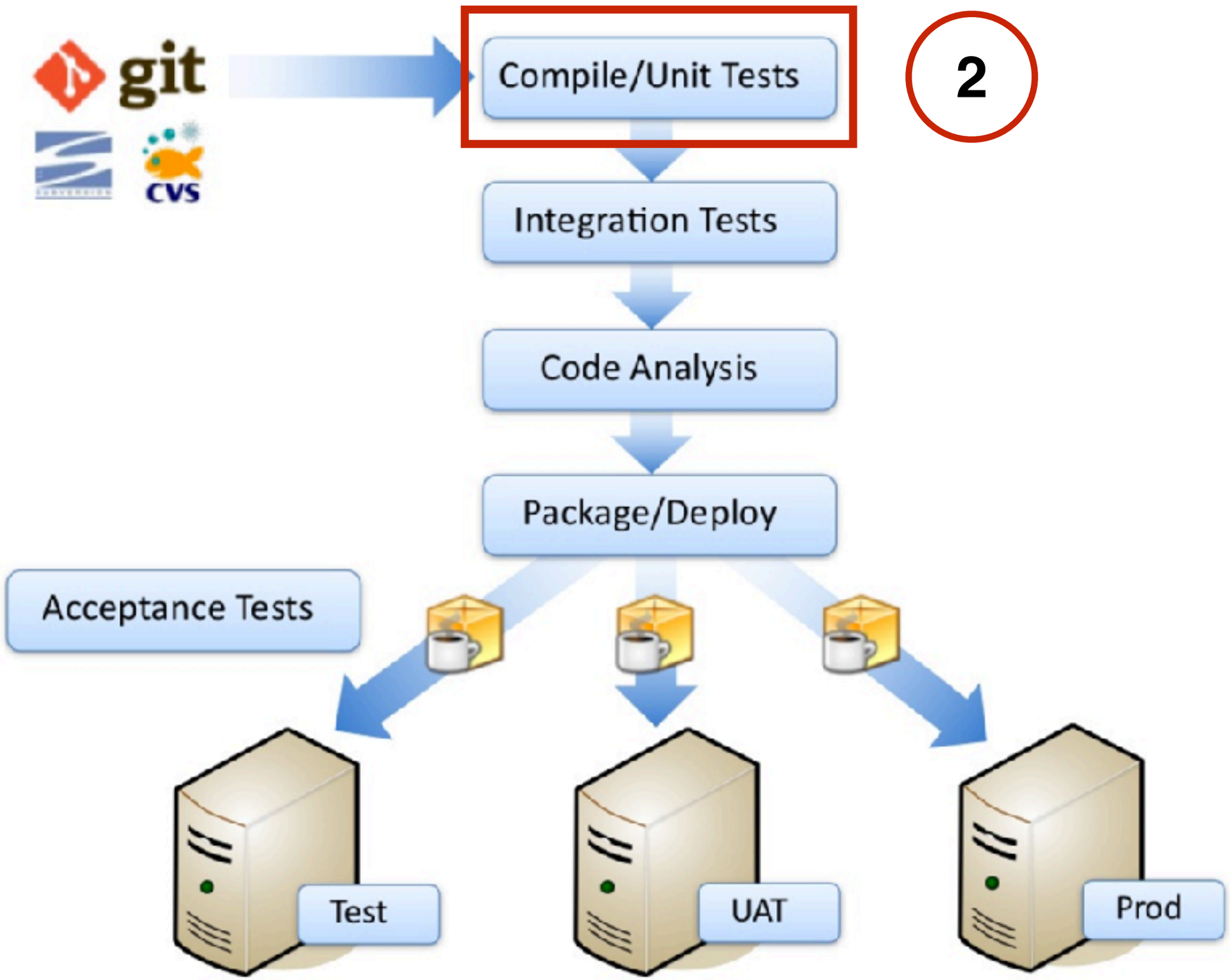


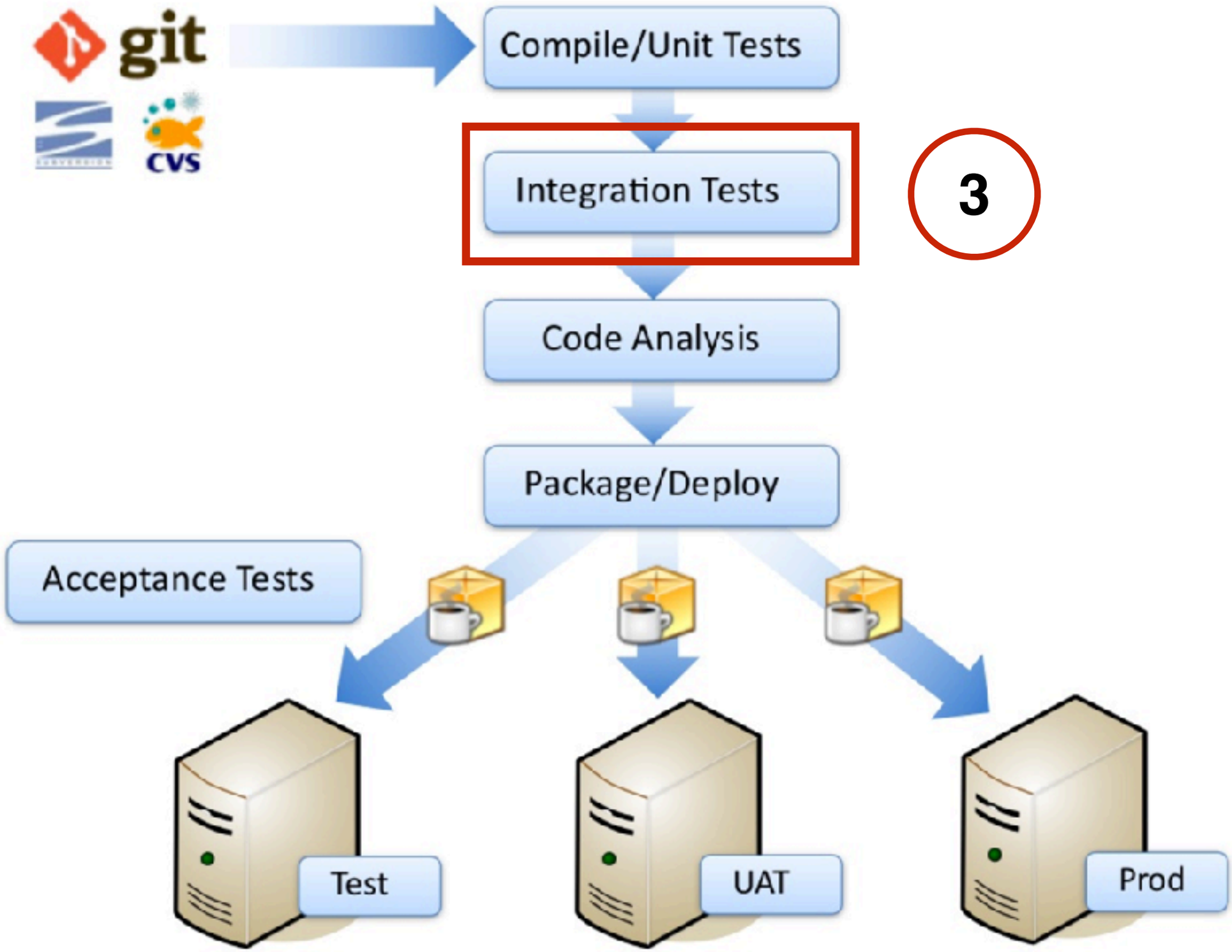
Tools for development

Technology	Description
Java	Main programming language
Apache Maven	Build tool
JUnit	Unit and integration test tool
Cobertura	Code coverage tool
Robotframework	Acceptance test tool
Apache Tomcat	Java Web Server
Git	Version Control System
SonarQube	Static code analysis tool
Frog Artifactory	Keep artifact files
Jenkins	Continuous Integration Server
IntelliJ IDEA	IDE for Java development

Let’s create pipeline with Jenkins

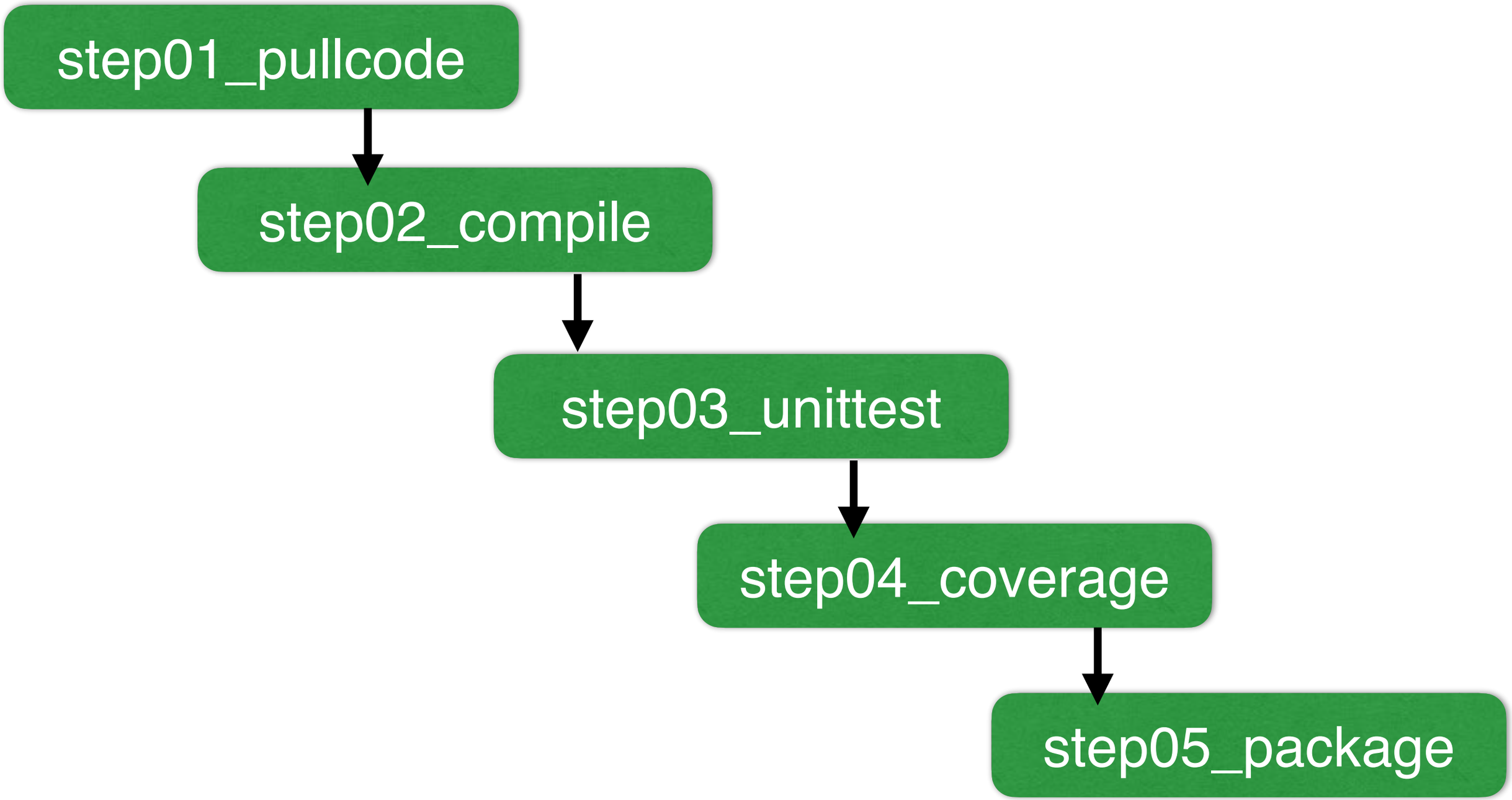


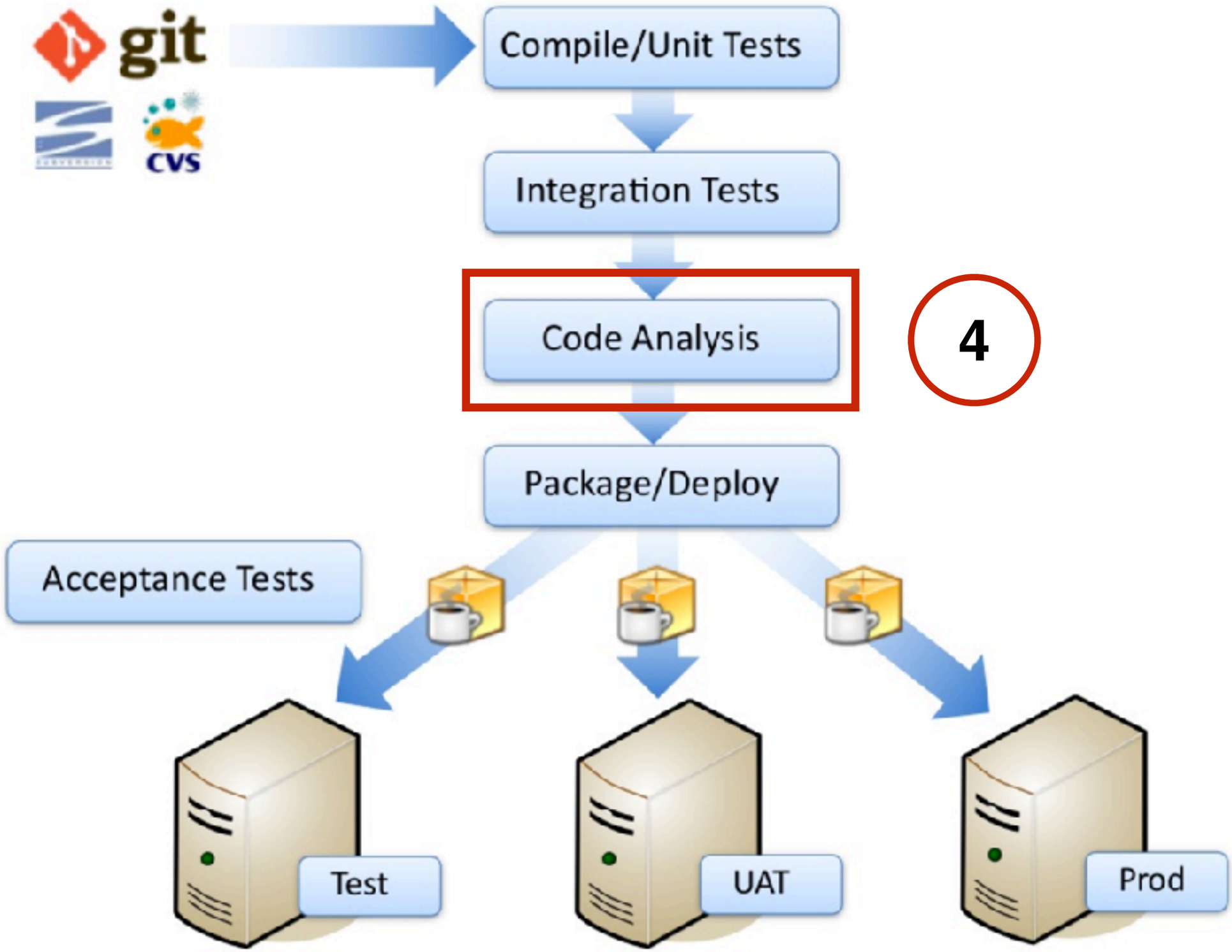


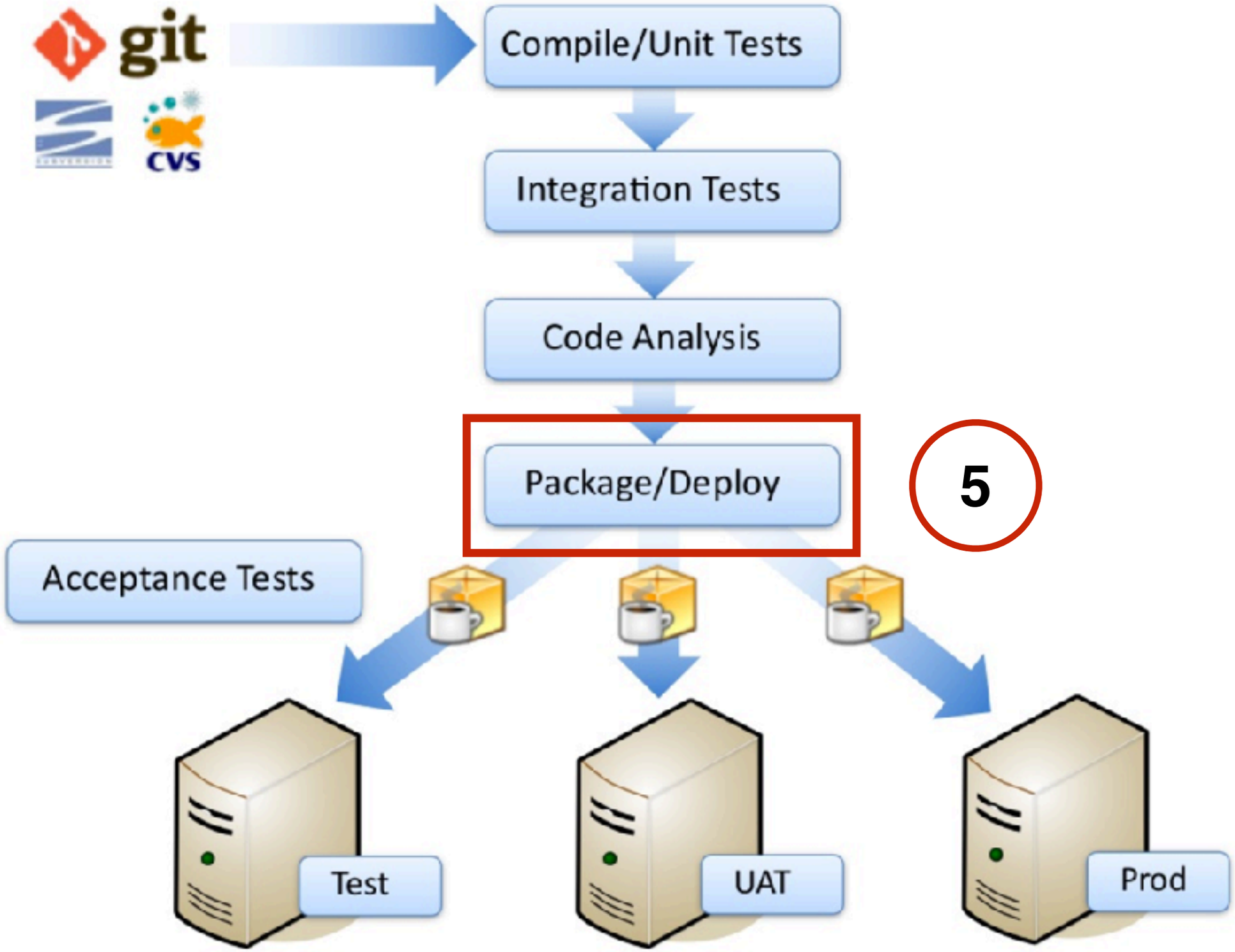


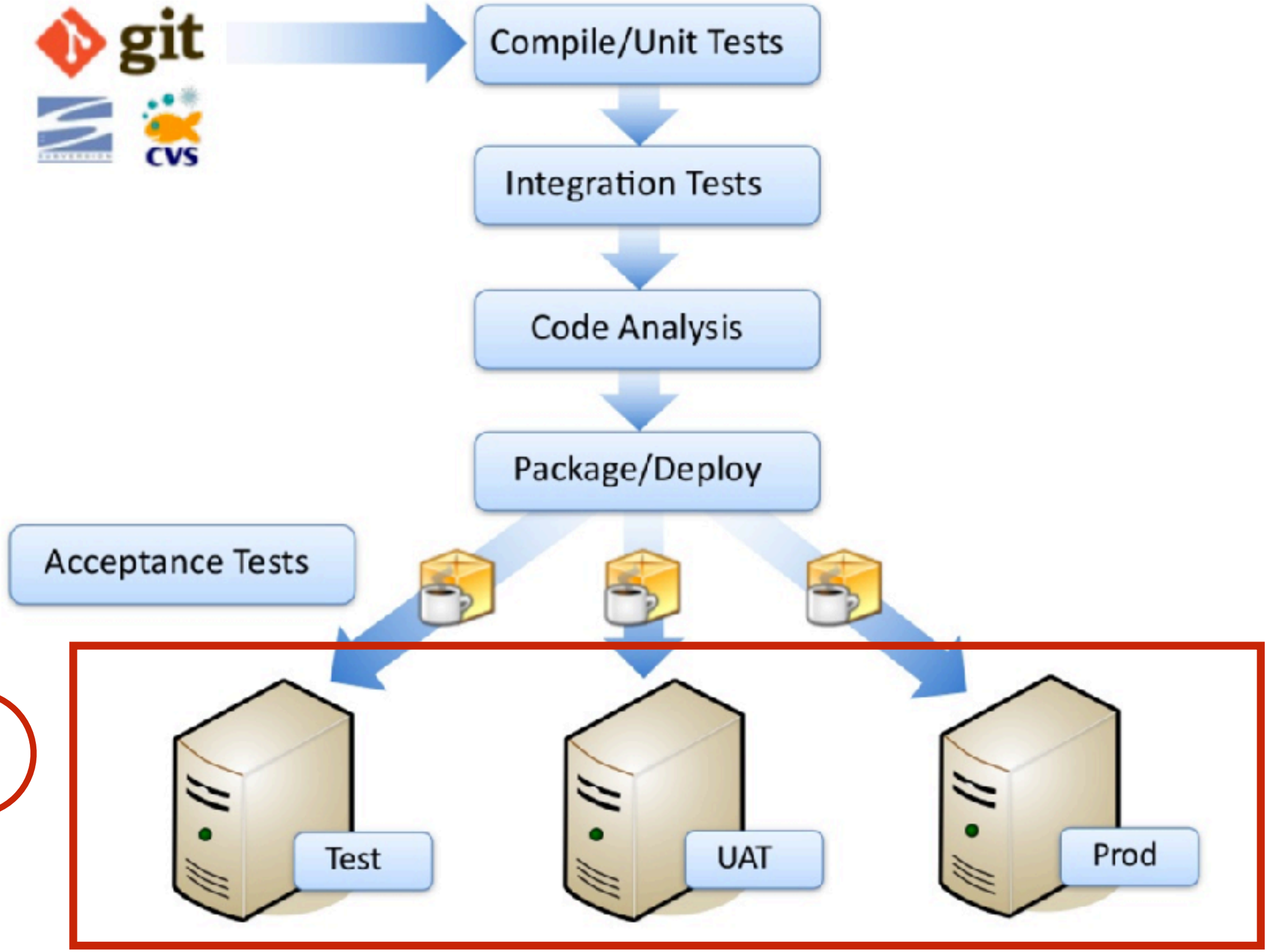
Suggestion

We need to separate to more jobs

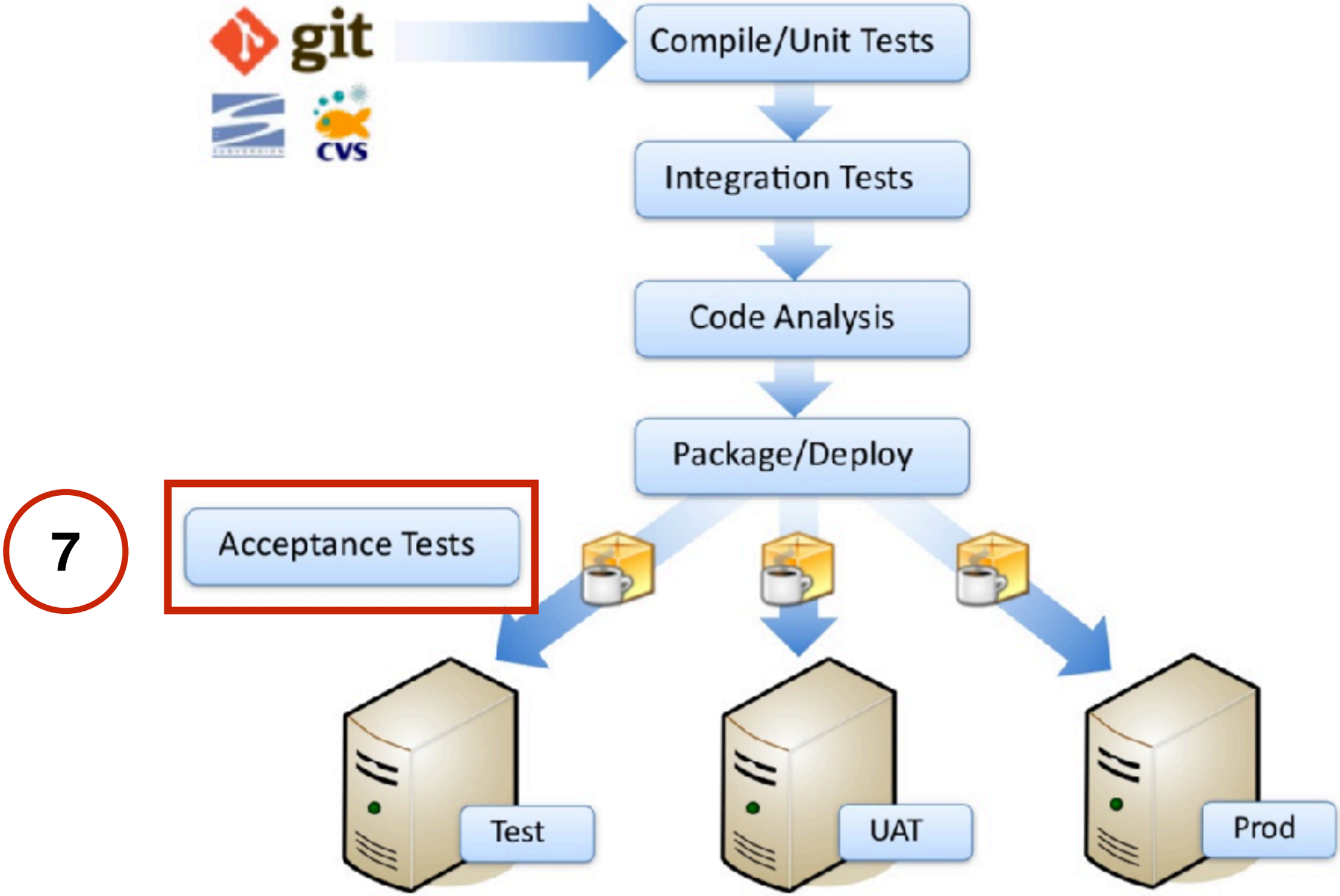








6



Working with Pipeline as a Code

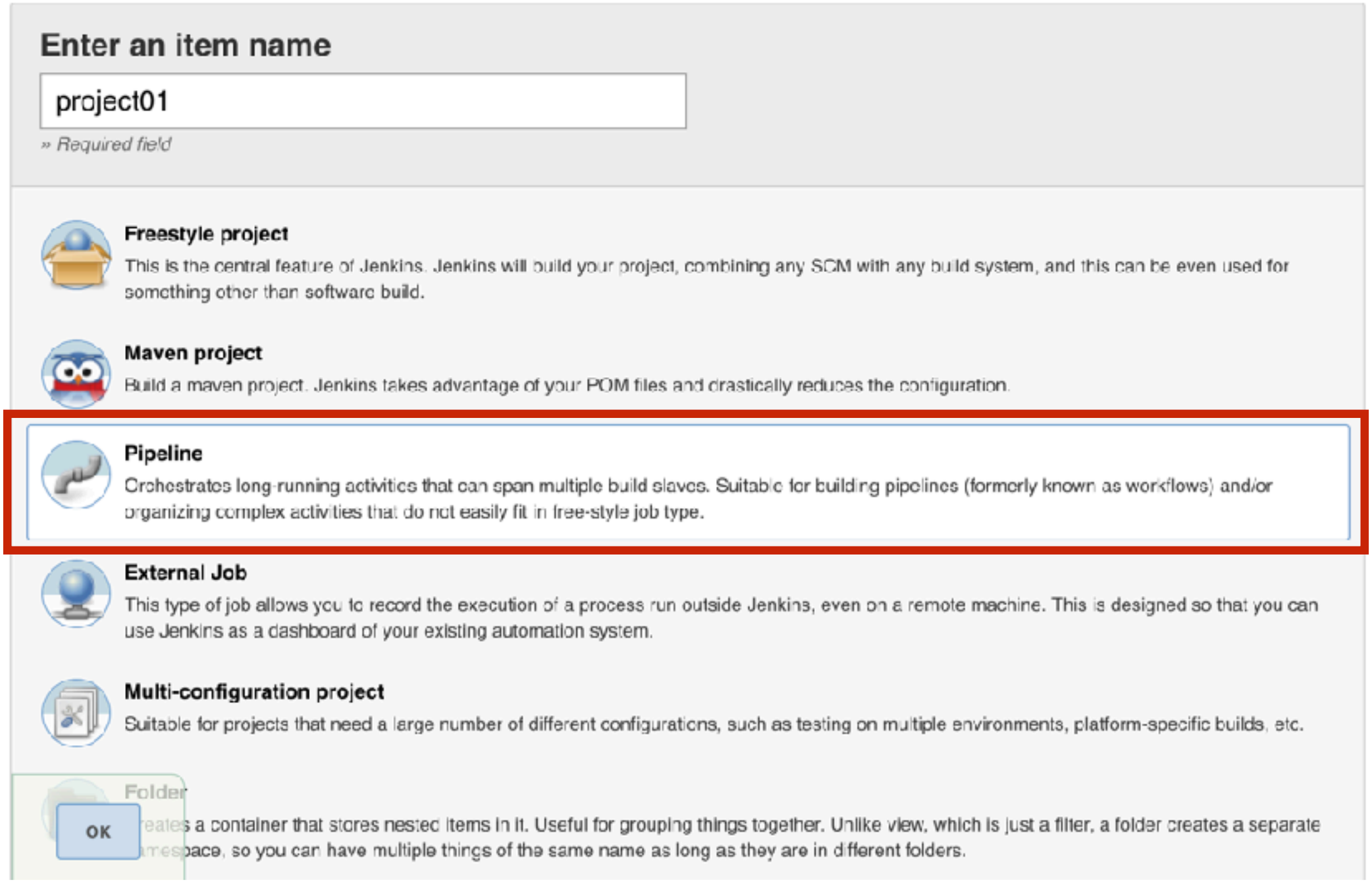
Pipeline as a Code

Declarative Pipeline

Scripted Pipeline

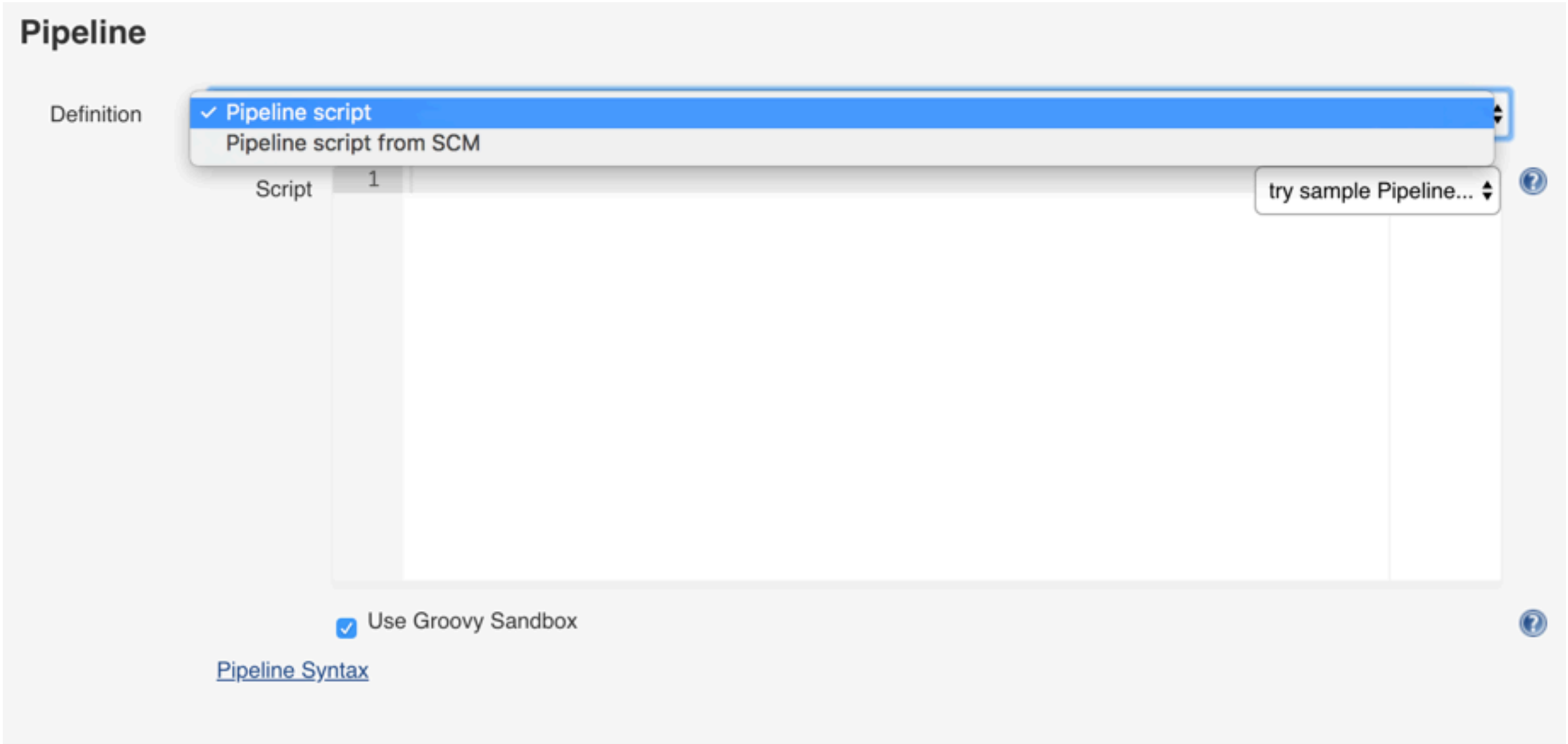
https://jenkins.io/doc/book/pipeline/

Create a new job with pipeline

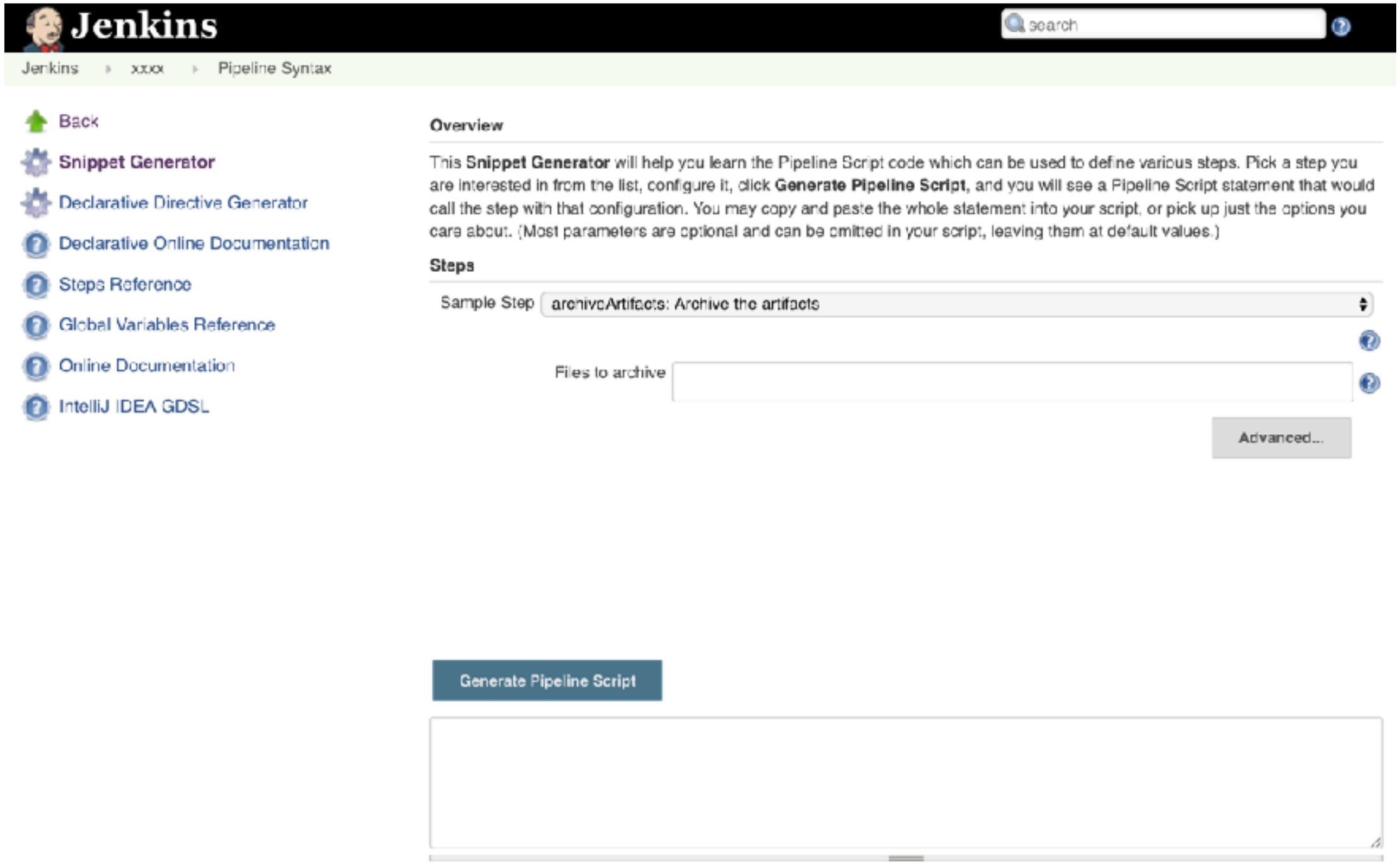


Write your pipeline

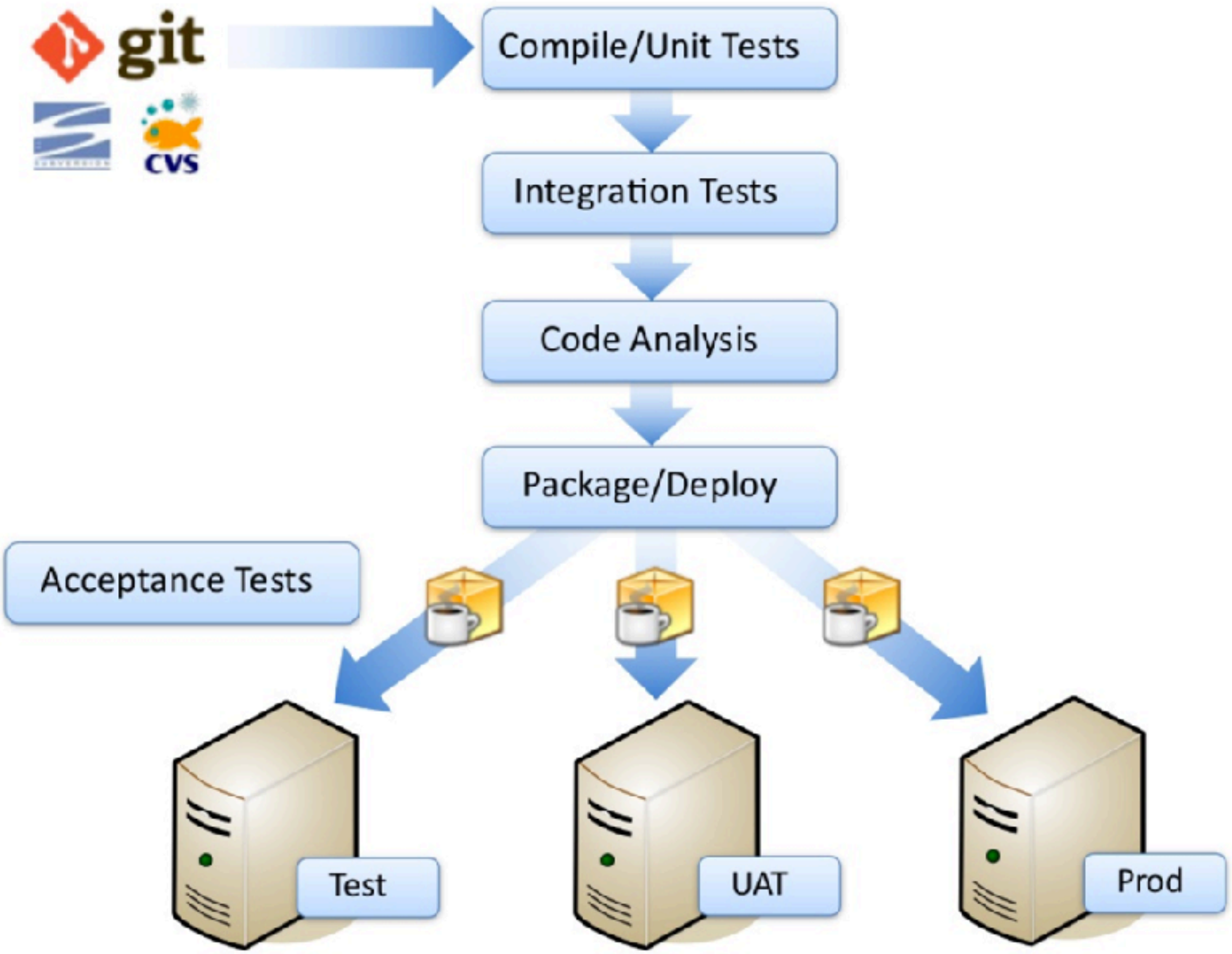
Pipeline script or from .Jenkinsfile



Pipeline syntax



Build pipeline



Create pipeline as code

```
node {
    stage('Pullcode') {
        git 'https://github.com/up1/workshop-java-web-tdd.git'
    }
    stage('Testing') {
        sh "mvn clean test"
        junit 'target/surefire-reports/*.xml'
    }
    stage('Package') {
        sh "mvn package"
    }
    stage('Code coverage') {
        cobertura autoUpdateHealth: false, autoUpdateStability: false
    }
}
```

Result

	Declarative: Checkout SCM	Pullcode	Testing	Package	Code coverage	Declarative: Post Actions
Average stage times: (Average full run time: ~33s)	2s	1s	7s	15s	55ms	33ms
#18 Feb 08 00:34 — 5 commits	2s	1s	7s	15s	55ms	33ms
#17 Feb 08 00:33 — No Changes						

Jenkinsfile

```
pipeline {
      agent any
      stages {
        stage('Pullcode') {
          steps {
              git 'https://github.com/up1/workshop-java-web-tdd.git'
          }
        }
    stage('Testing') {
       steps {
         sh "mvn clean test"
         junit 'target/surefire-reports/*.xml'
       }
     }
```

Jenkinsfile

```
stage('Package') {
  steps {
    sh "mvn package"
  }
}
stage('Code coverage') {
  steps {
      cobertura autoUpdateHealth: false, autoUpdateStability: fa
    }
  }
}
post {
  always {
      junit 'target/surefire-reports/*.xml'
  }
}
```

Try it by yourself

Useful plug-ins

Useful plug-ins

Build Pipeline
Build Monitor View
Cobertura Report
Jacoco Report
Robotframework Report
Sound
Green Ball
SonarQube Scanner
Blue Ocean

Jenkins Best Practices

https://wiki.jenkins.io/display/JENKINS/Jenkins+Best+Practices

Jenkins Best Practices

Always secure Jenkins

In large system, don’t build on the master

Backup JENKINS_HOME regularly

Limit project name to the sane character set

https://wiki.jenkins.io/display/JENKINS/Administering+Jenkins

Jenkins Best Practices

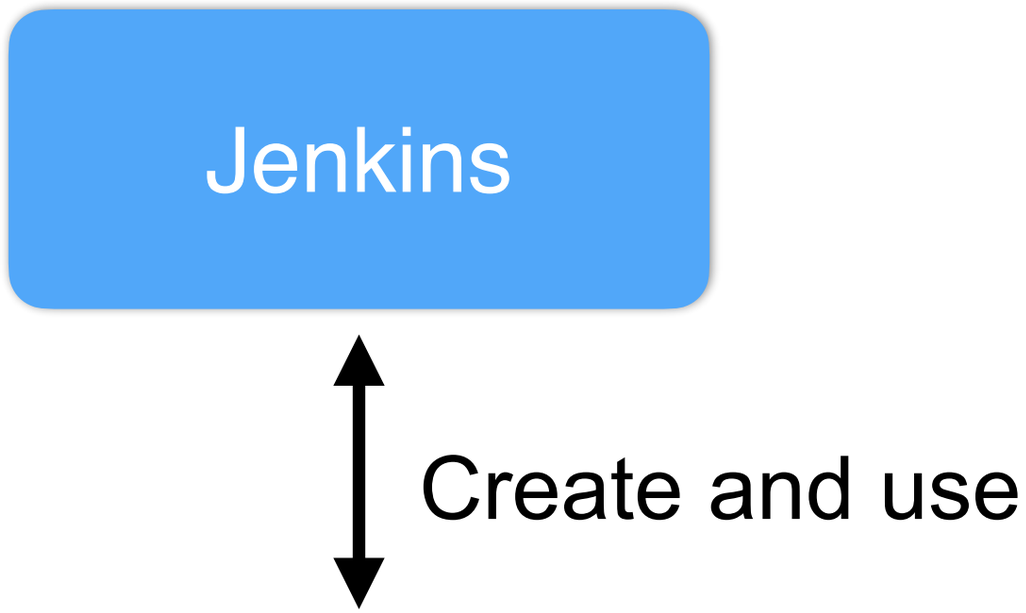
Always config your job to generate report
Archive unused jobs before removing them
Setting difference job for each branch
Prevent resource collisions in job (parallel)
Fail fast

Recap Day 1

Recap day 1

CI and CD concepts
CI principles and practices
Install and config Jenkins
Workshop with Java Web application

Recap day 1 :: Jenkins Home



${JENKINS_HOME} = [Home directory]/.jenkins

Recap day 1 :: Jenkins Home

set JENKINS_HOME=<>

Jenkins

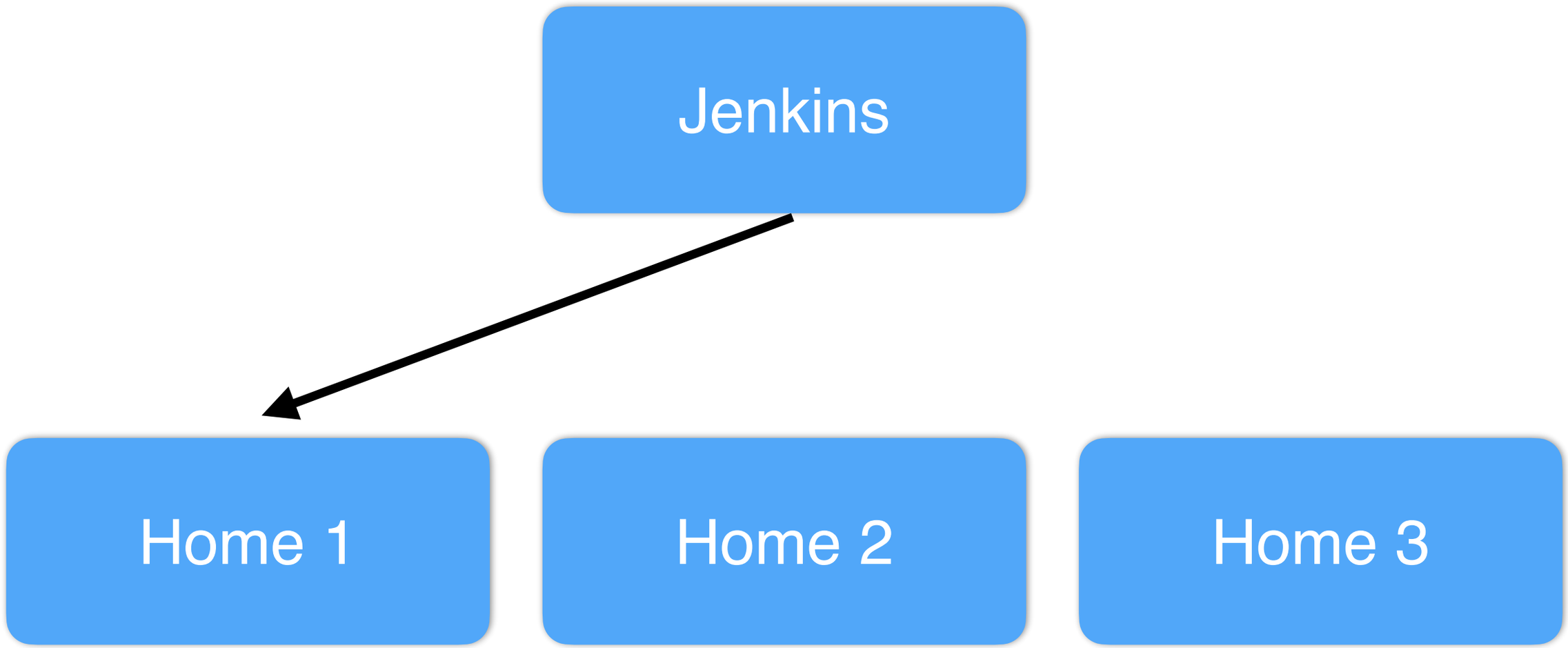
Home 1

Home 2

Home 3

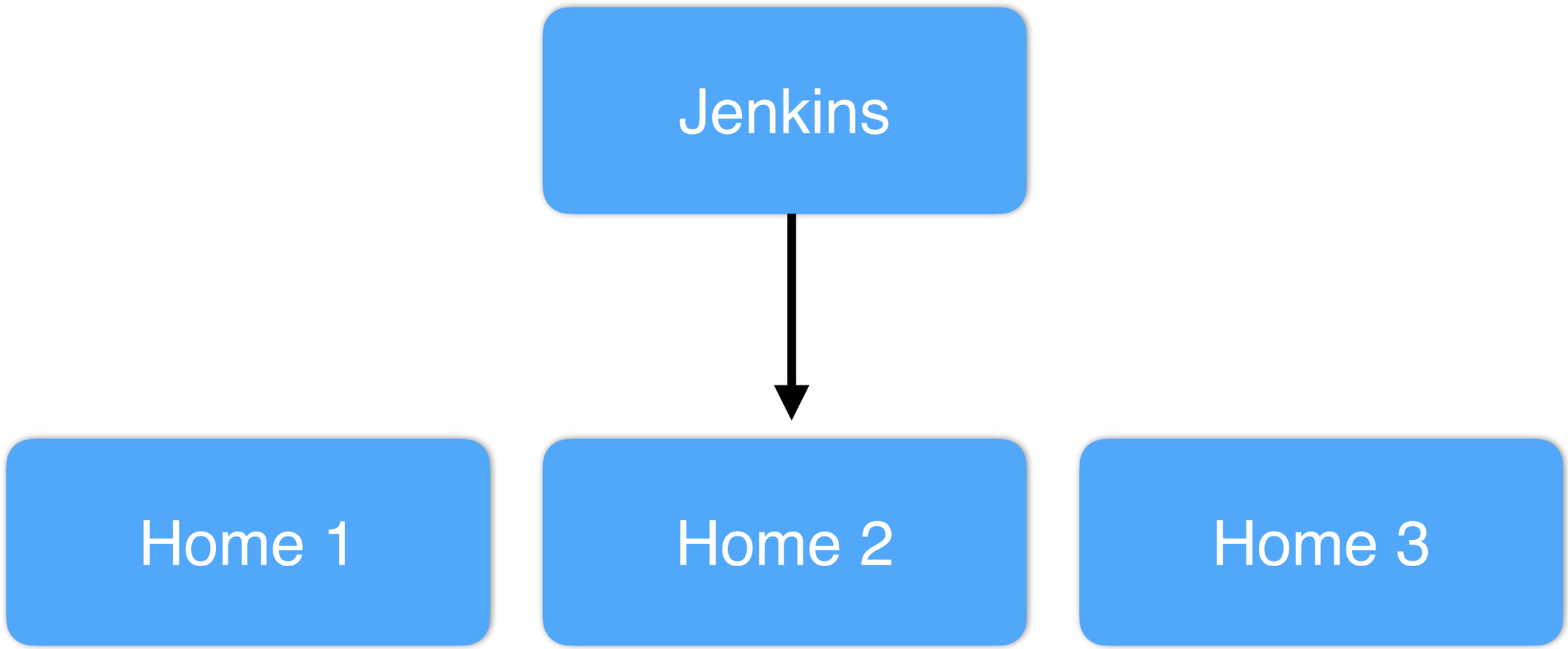
Recap day 1 :: Jenkins Home

set JENKINS_HOME=Home 1



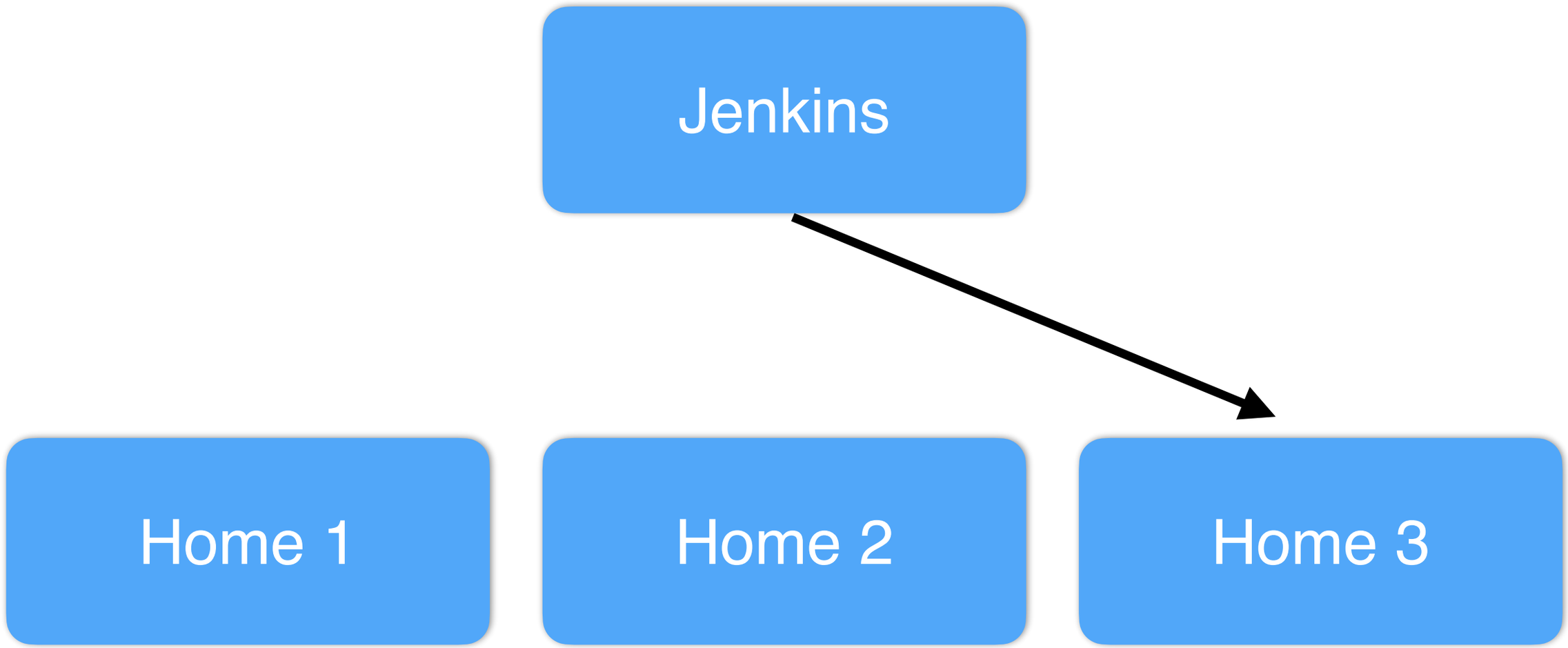
Recap day 1 :: Jenkins Home

set JENKINS_HOME=Home 2

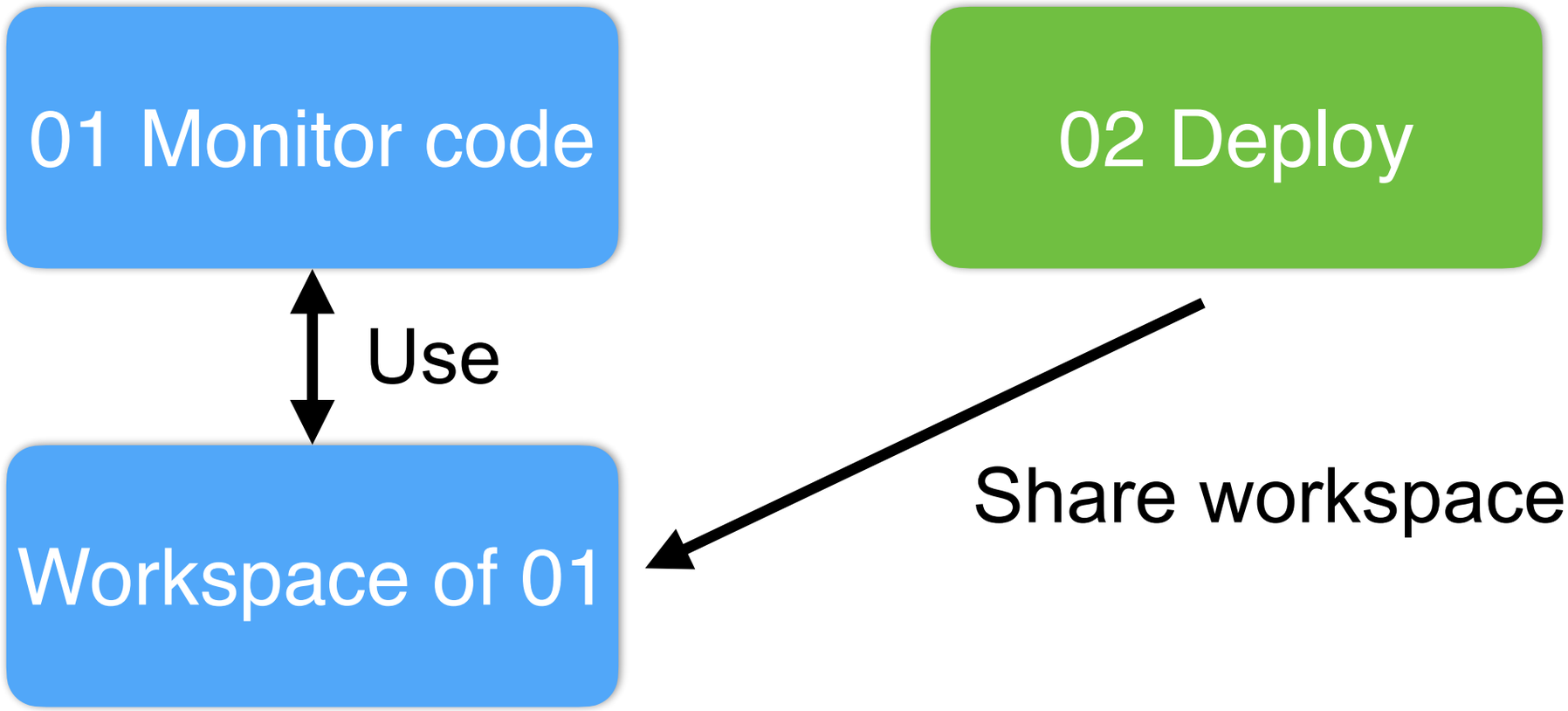


Recap day 1 :: Jenkins Home

set JENKINS_HOME=Home 3

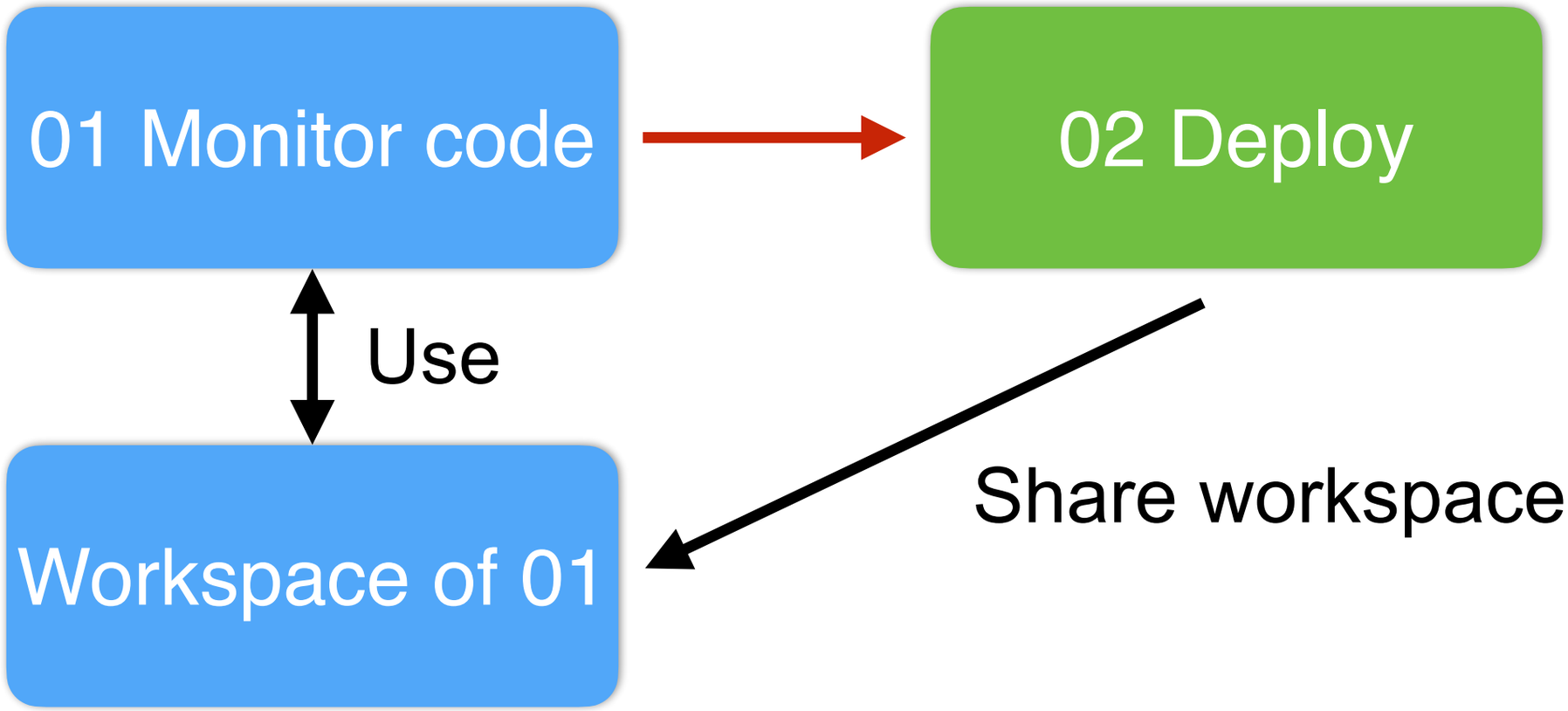


Recap day 1 :: pipeline



${JENKINS_HOME}/workspace

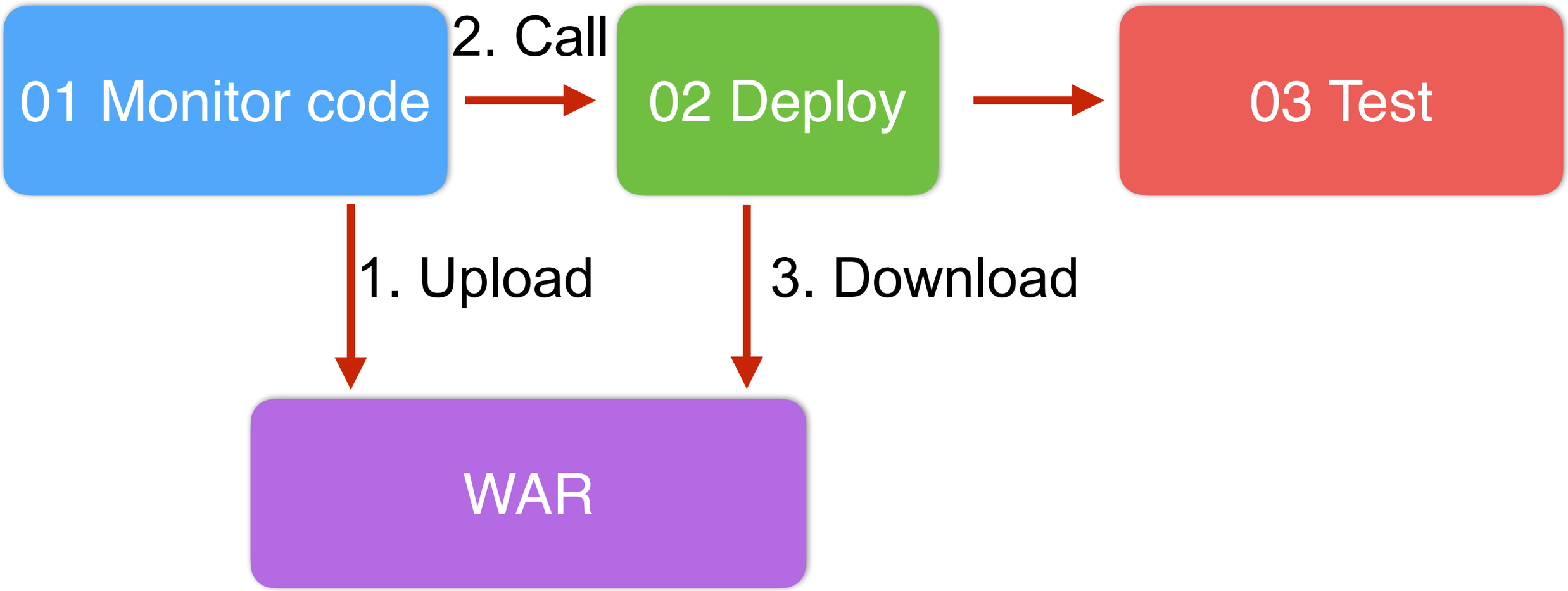
Recap day 1 :: pipeline



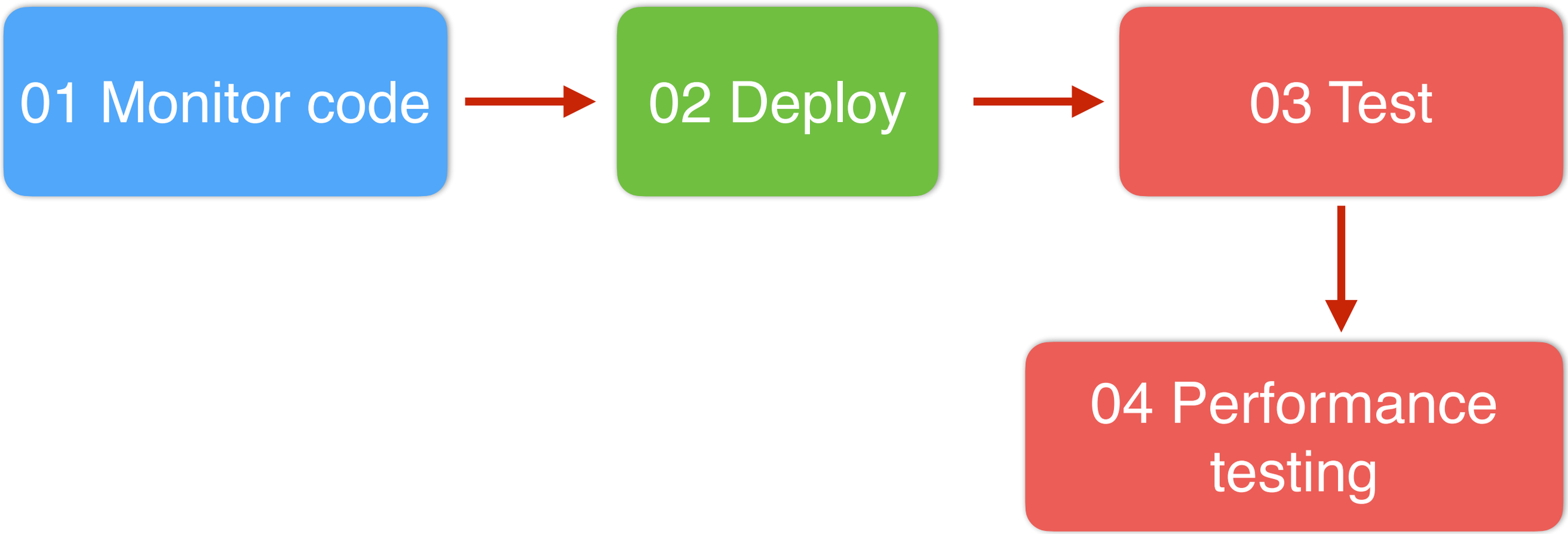
${JENKINS_HOME}/workspace

UI testing

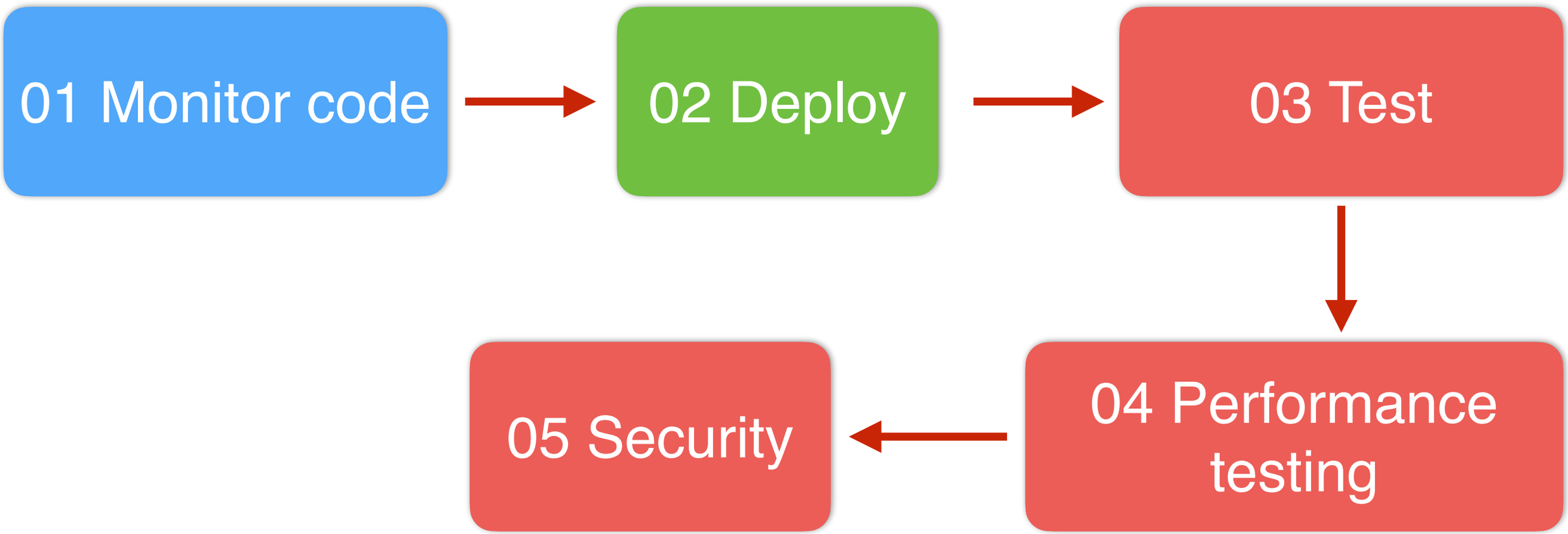
Share Software Package (Artifact)



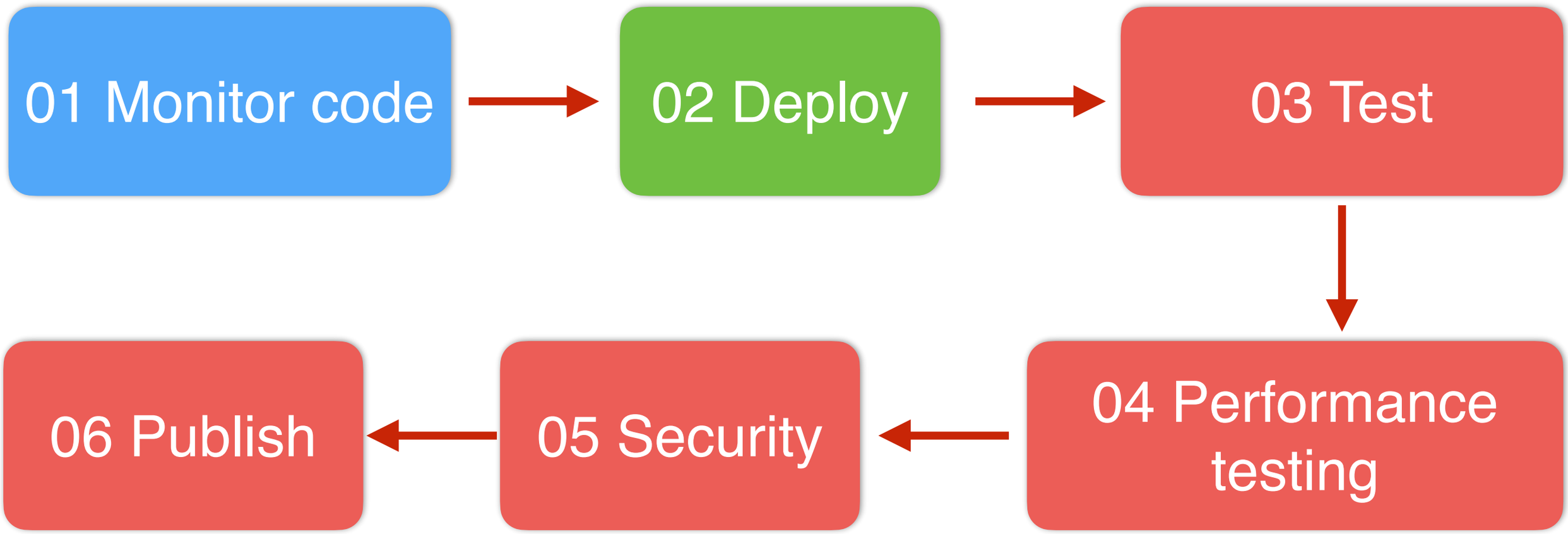
Performance testing



Security testing



Publish Artifact



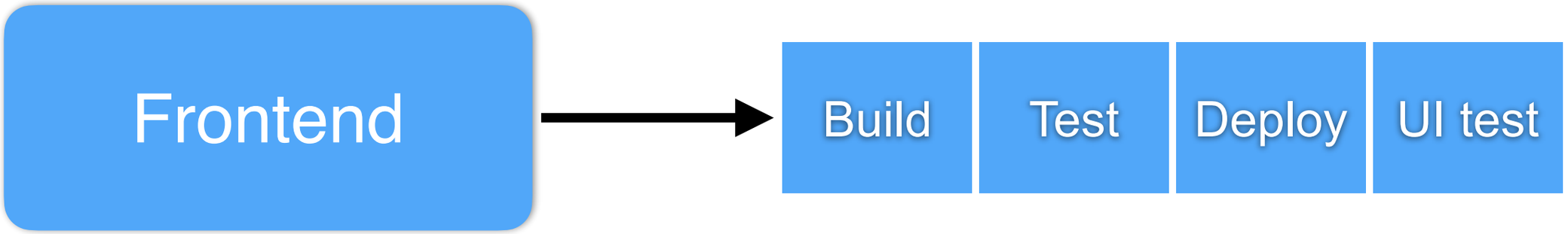
Design your pipeline

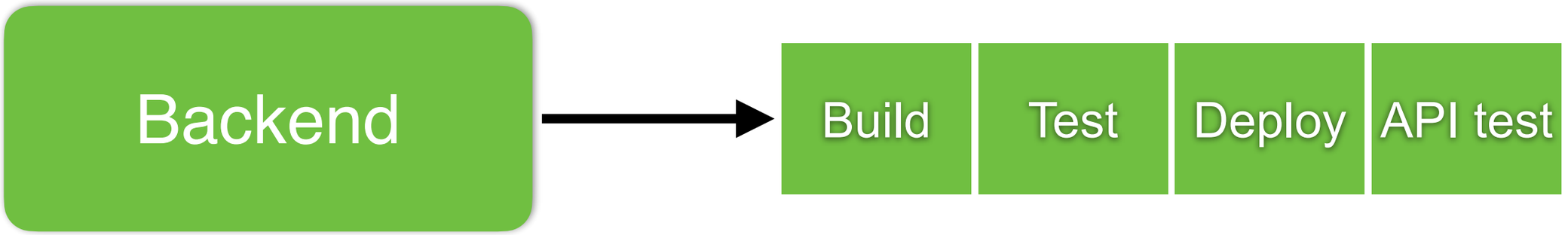
Use cases

Use cases

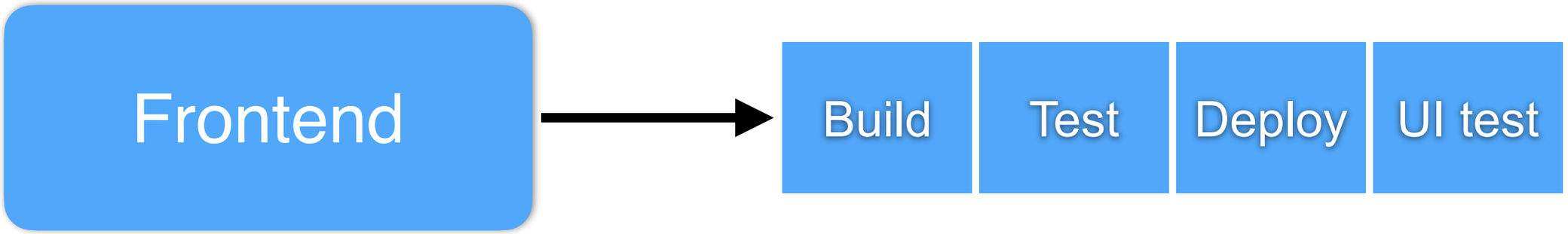
Build	Test	Deploy	Test Frontend
Build	Test	Deploy	Mock backend
Build	Test	Deploy	Real backend

Pipeline

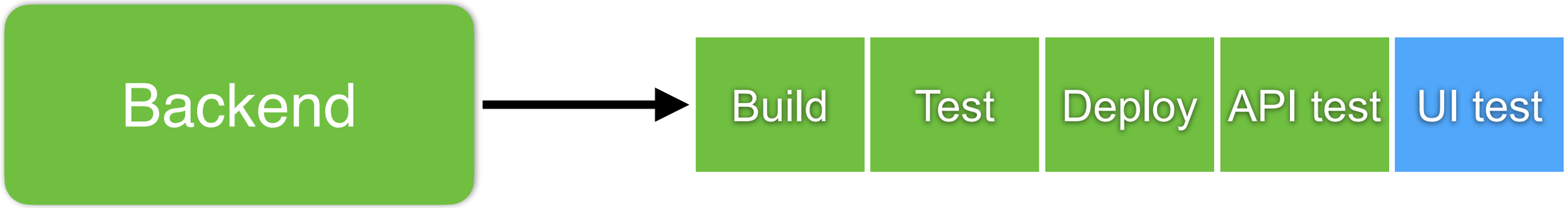




When frontend is changed

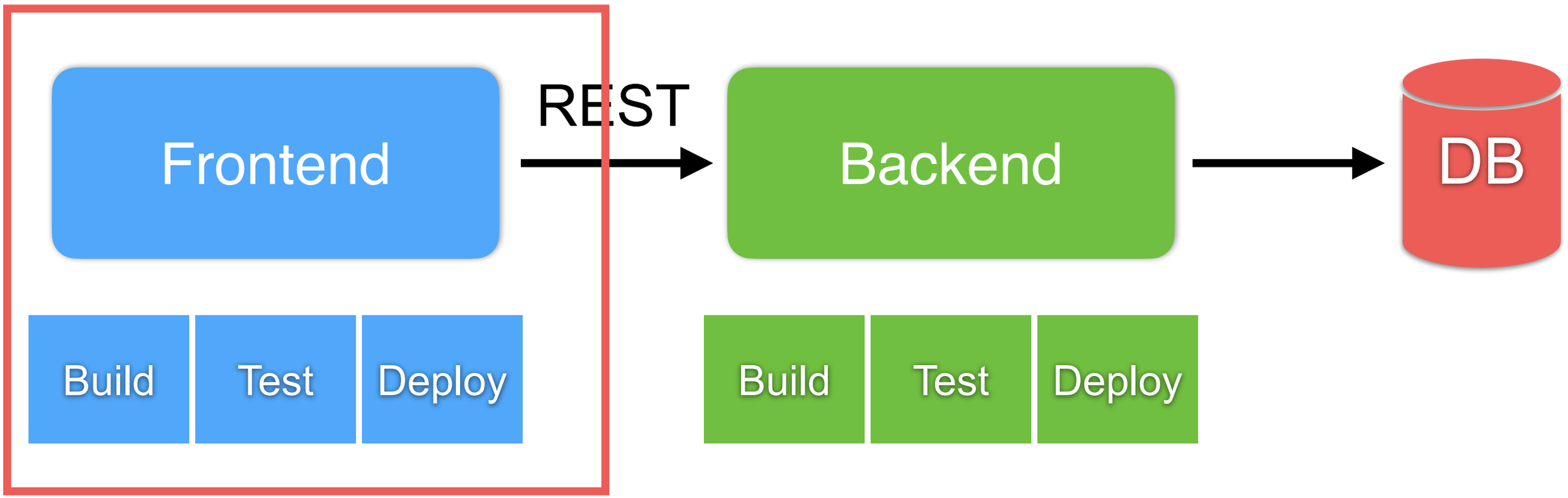


When backend is changed

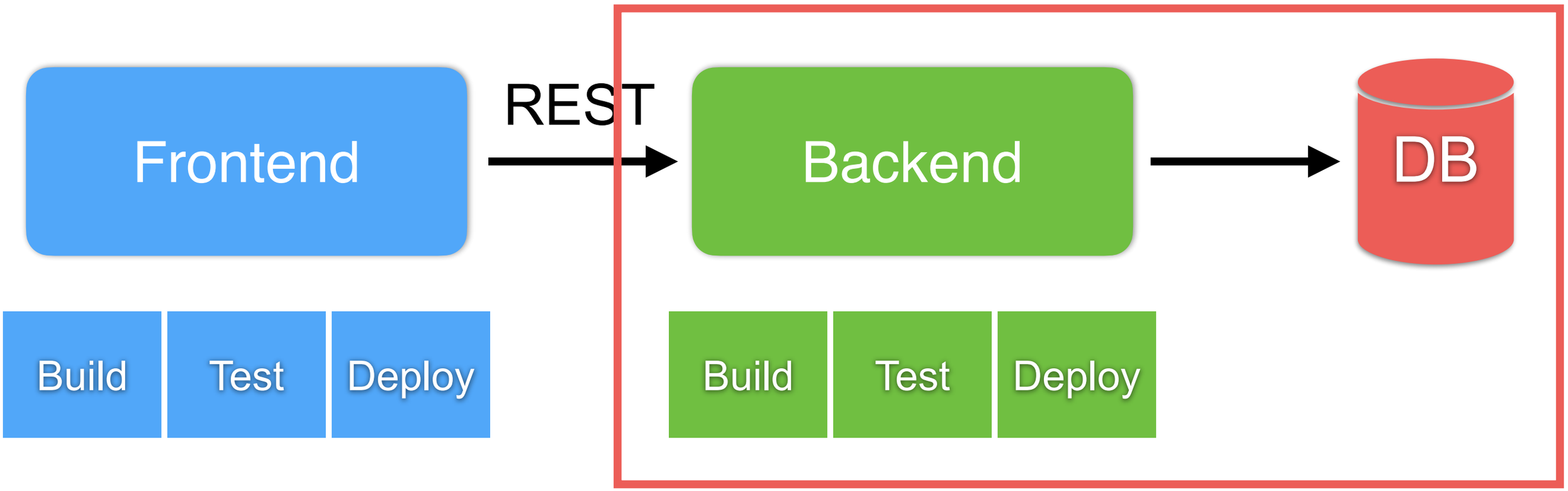


Flow

How to test frontend ?



How to test backend ?



Next ...