



Continuous Delivery with Gradle and Jenkins

By Peter Niederwieser

Releases don't have to be painful



Continuous Delivery



Deliver software fast and frequently

principles

#1 Every commit can result in a release

principles

#1 Every commit can result in a release

#2 Automate everything!

principles

#1 Every commit can result in a release

#2 Automate everything!

#3 Automated tests are essential

principles

#1 Every commit can result in a release

#2 Automate everything!

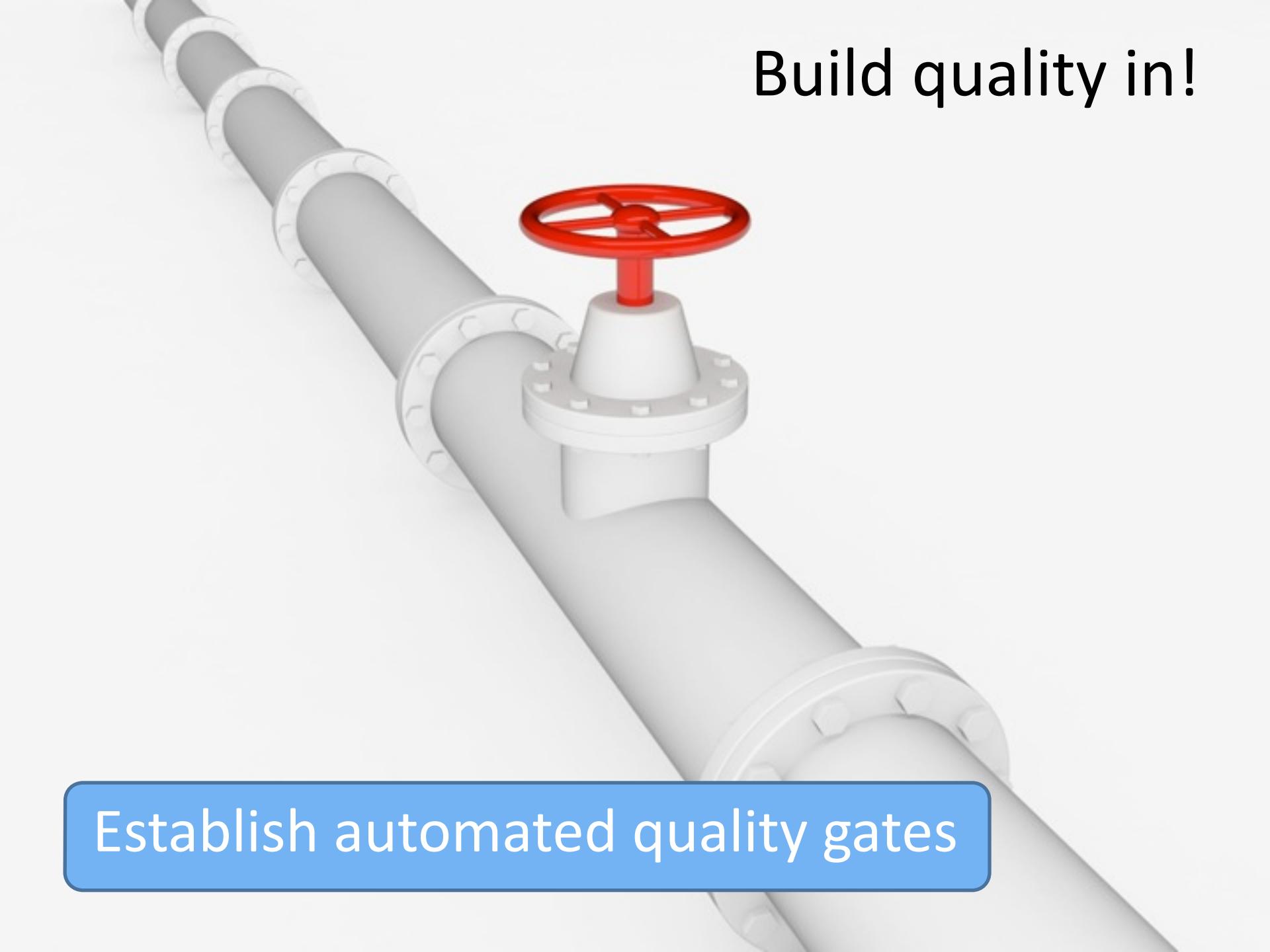
#3 Automated tests are essential

#4 Done means released

Build pipeline



Automated manifestation of delivery process



Build quality in!

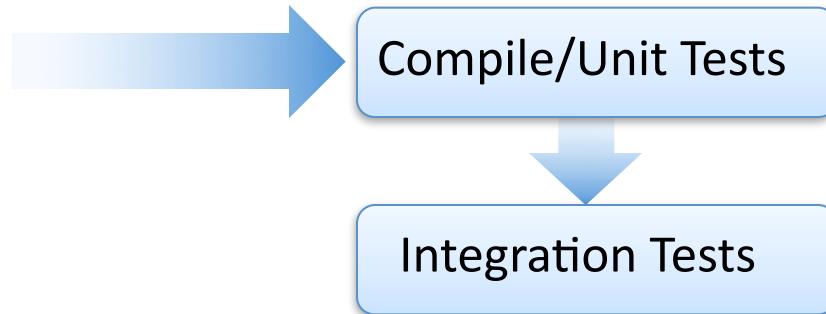
Establish automated quality gates

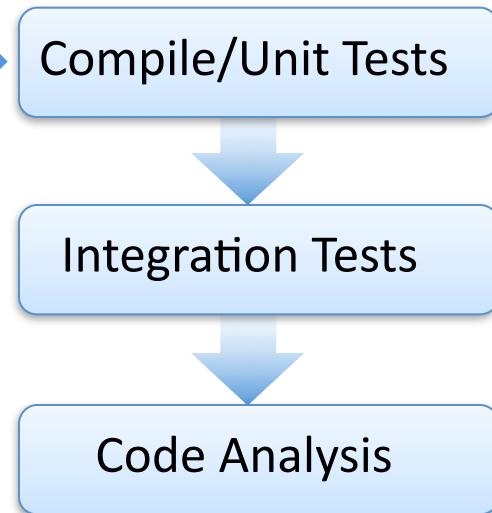


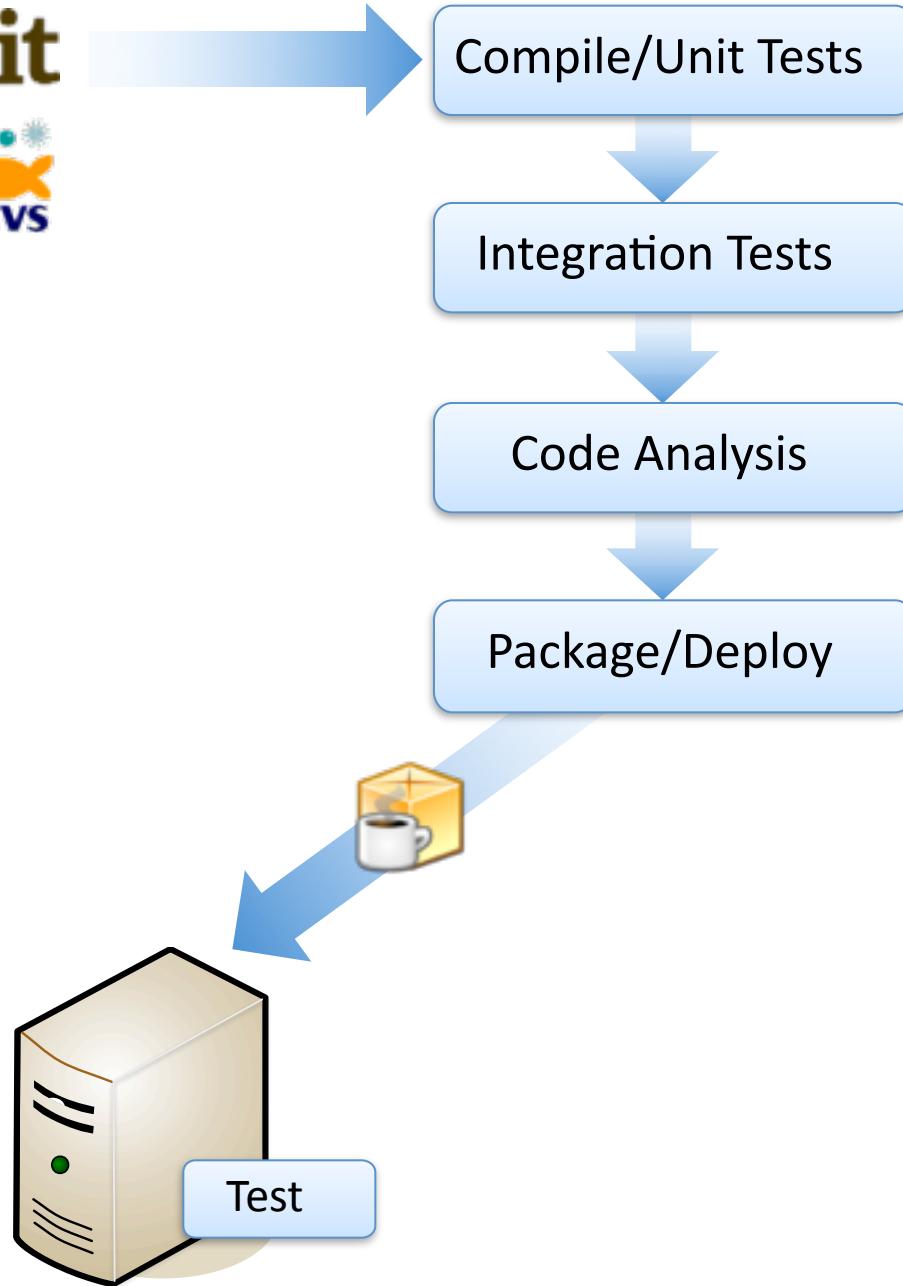
git

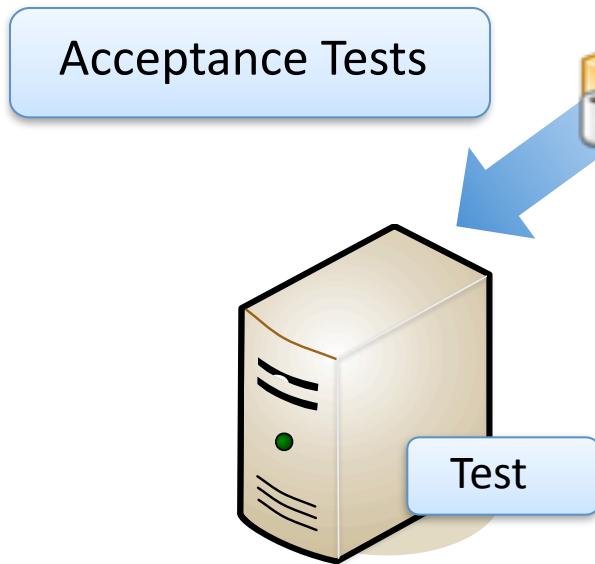
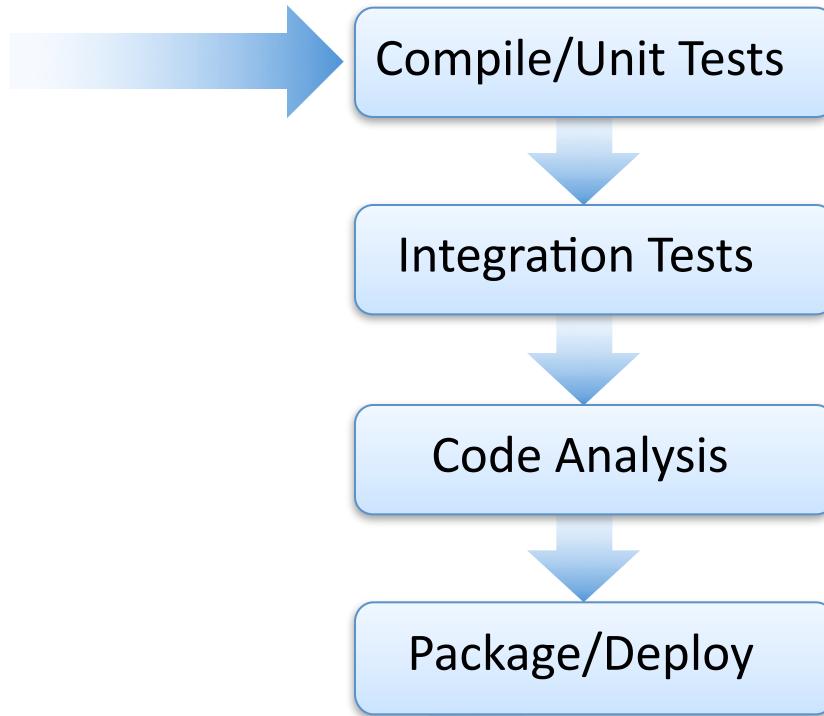


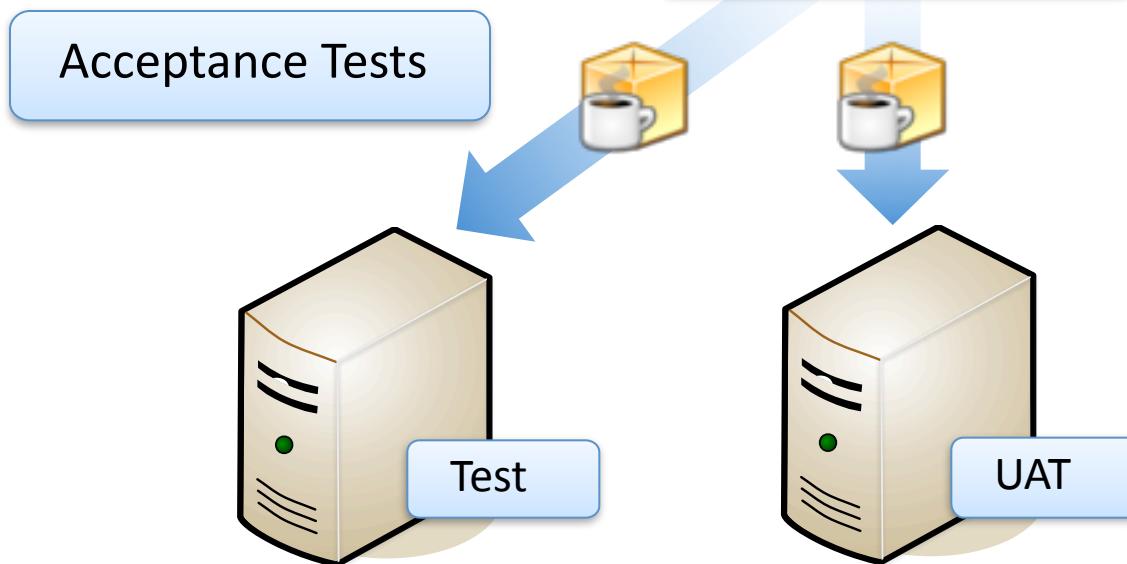
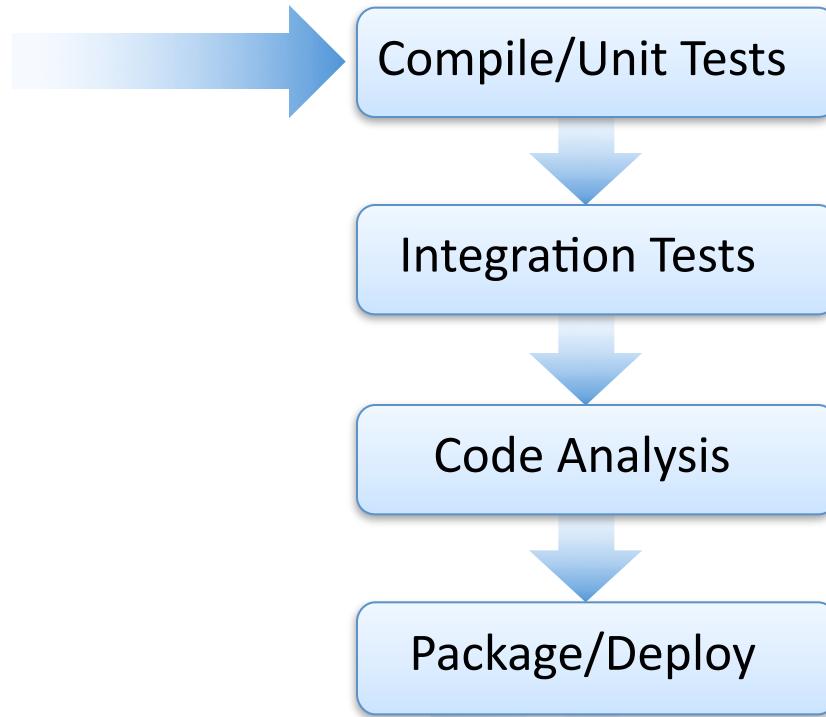
Compile/Unit Tests

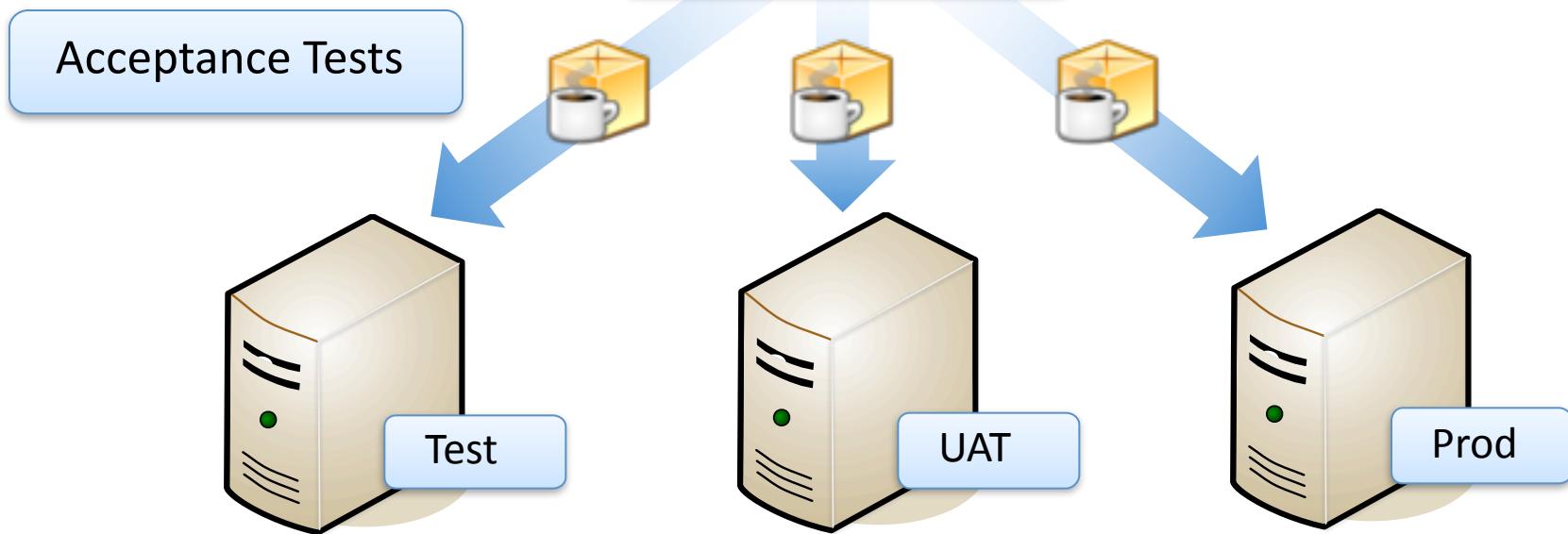
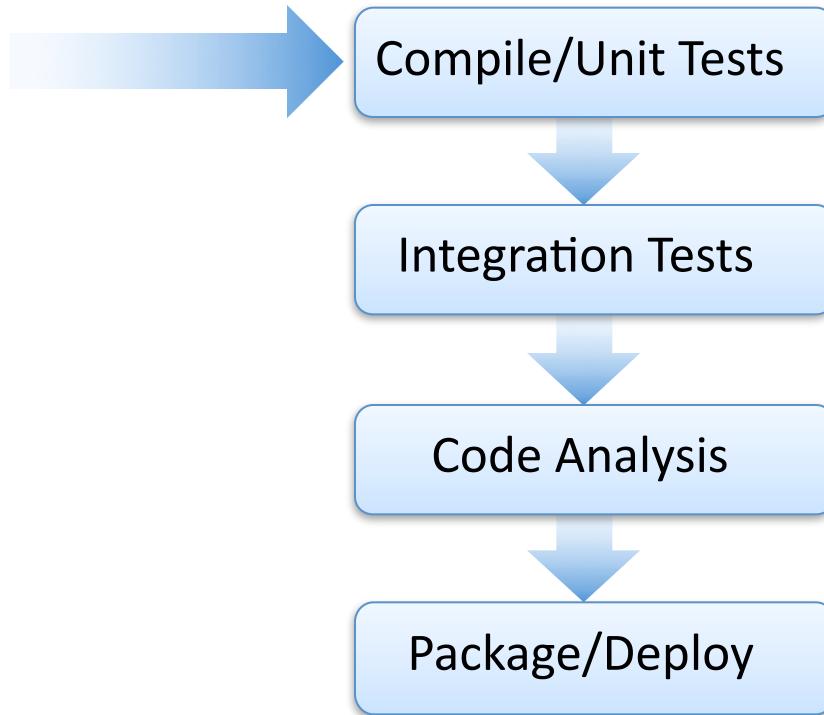








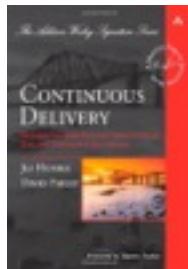






!



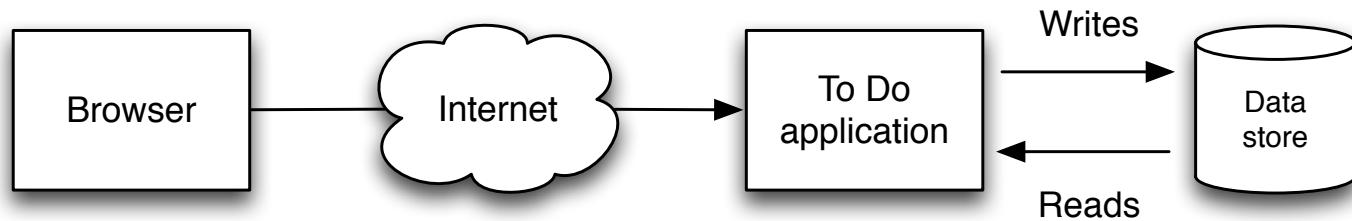
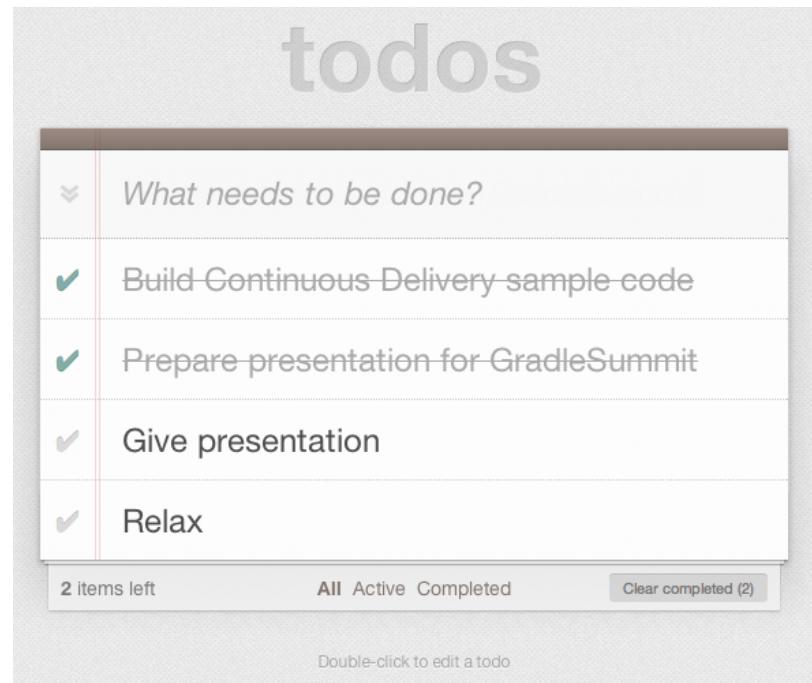
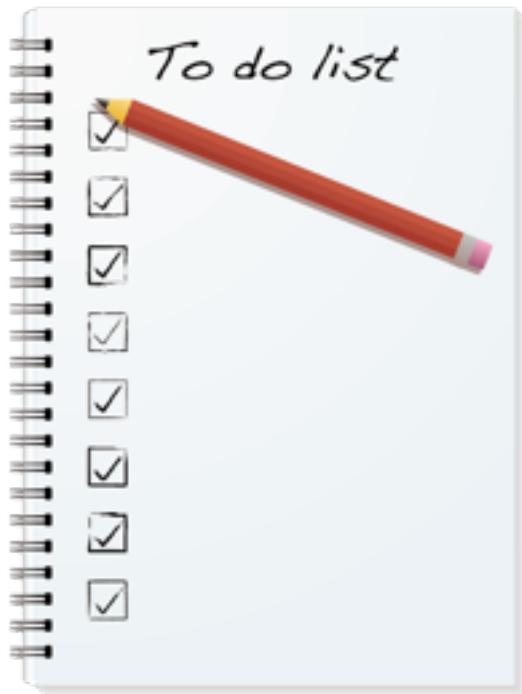


!

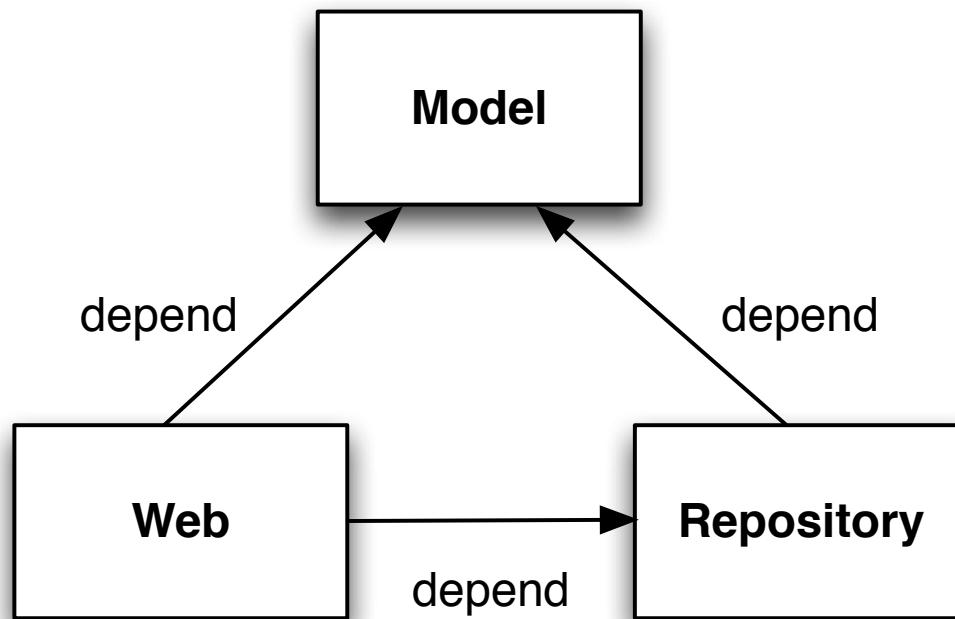
But
how?



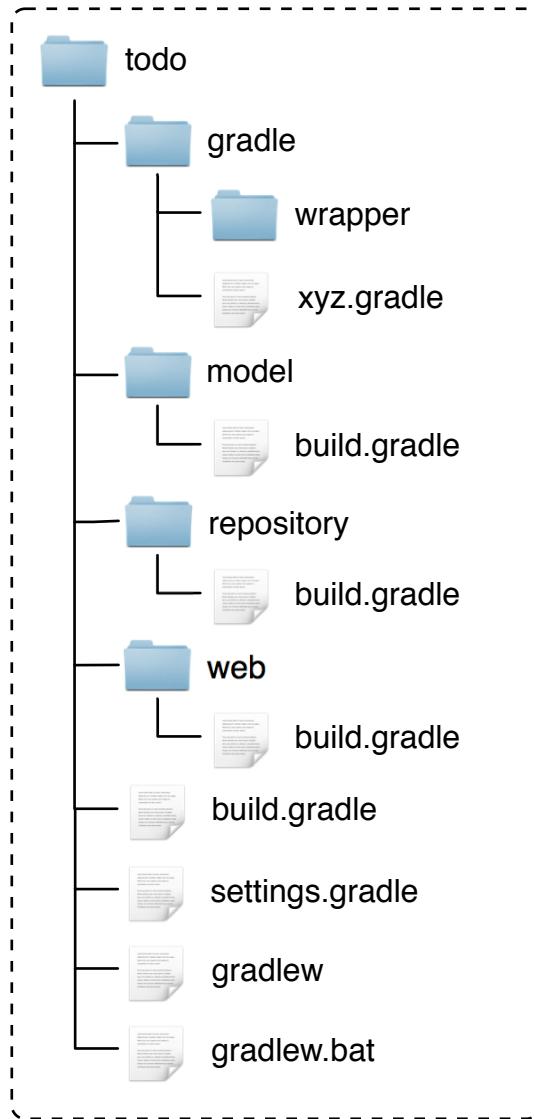
The “revolutionary” sample application



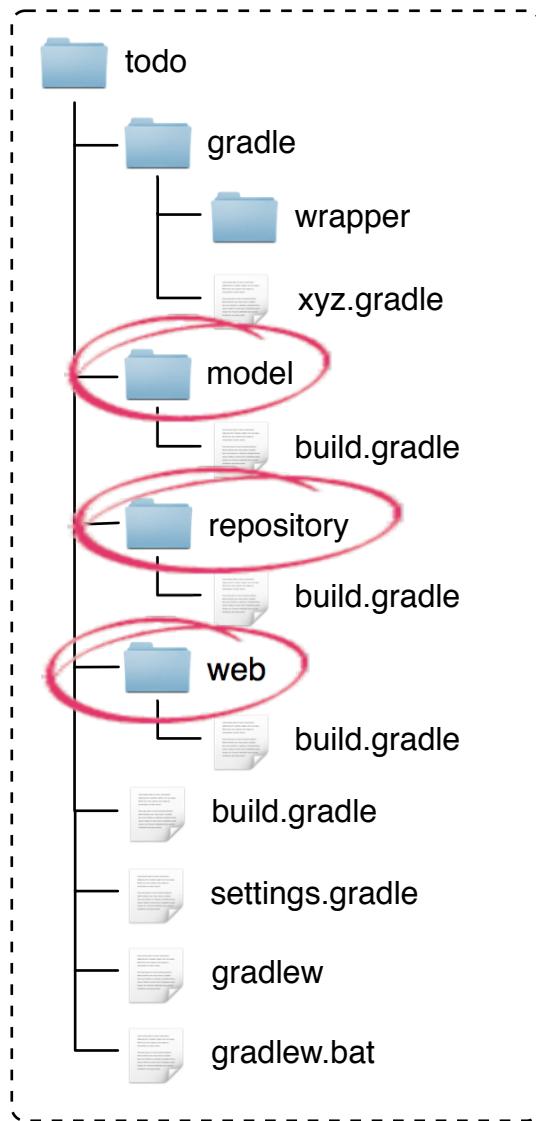
Multi-project dependencies



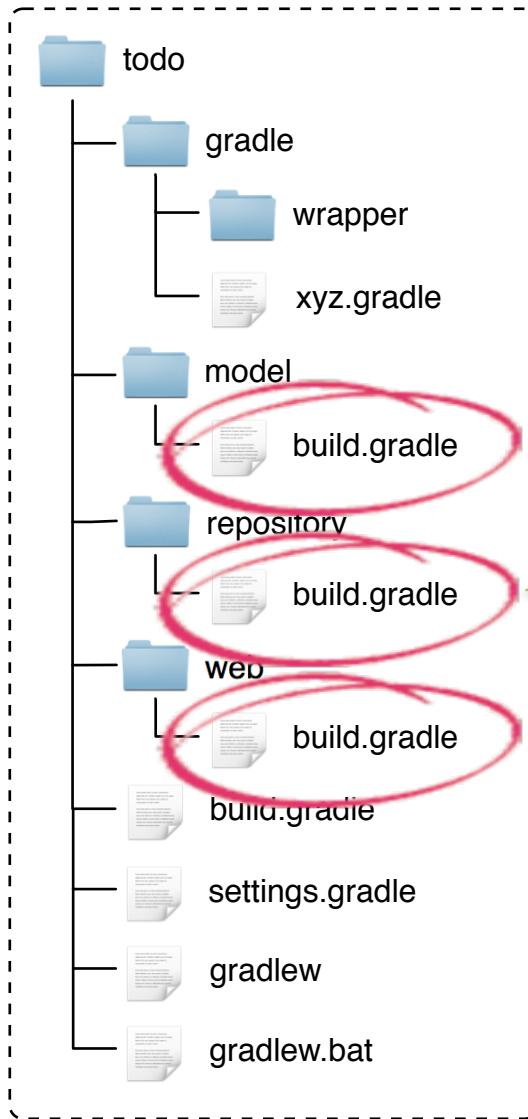
Project hierarchy



Project hierarchy

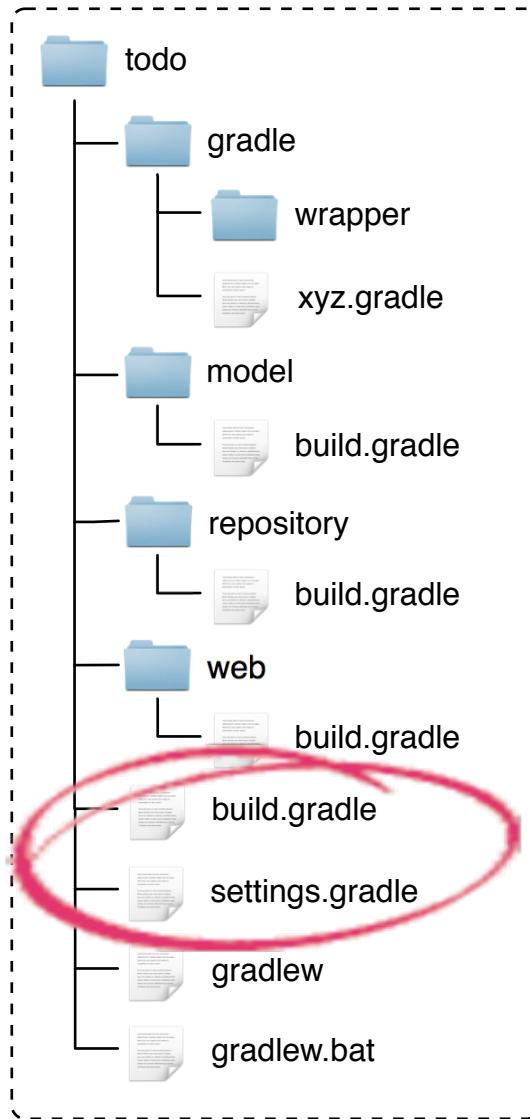


Project hierarchy



Define project-specific
behavior

Project hierarchy

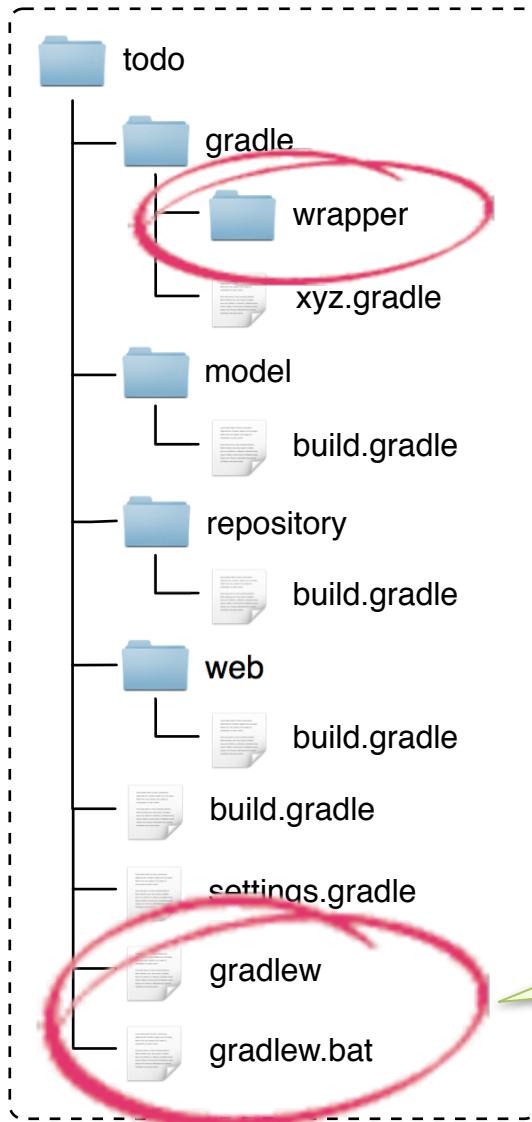


Defines which projects are taking part in the build

settings.gradle

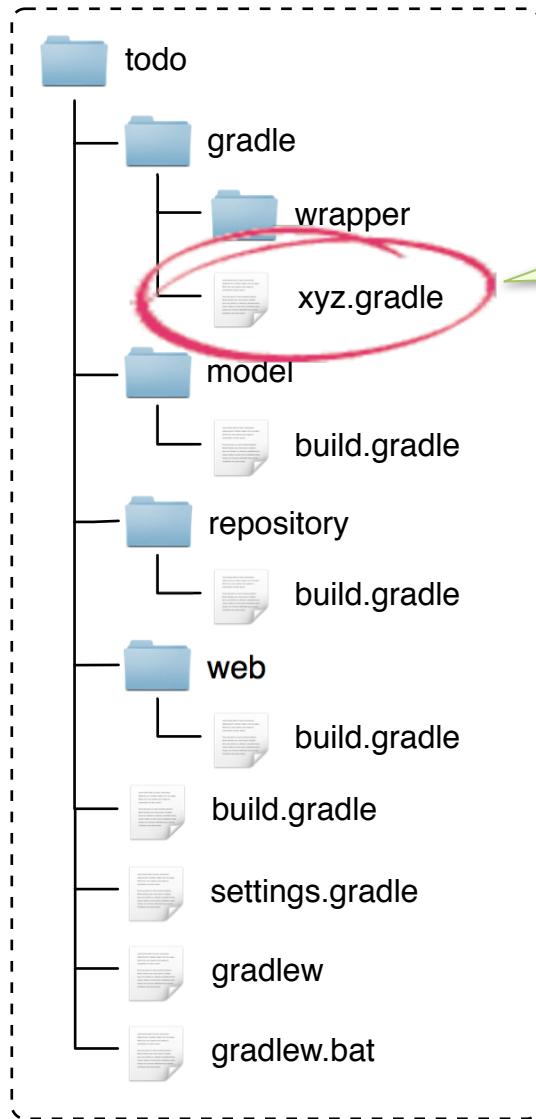
```
include 'model'  
include 'repository'  
include 'web'
```

Project hierarchy



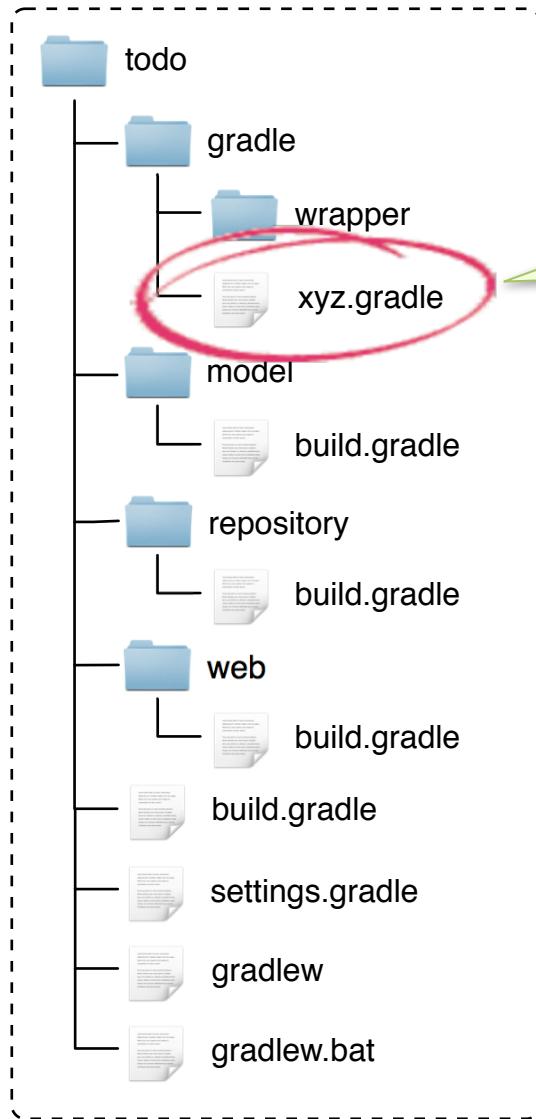
Always use Wrapper
to execute the build!

Project hierarchy



Externalize concerns into script plugins
and organize them in dedicated directory

Project hierarchy

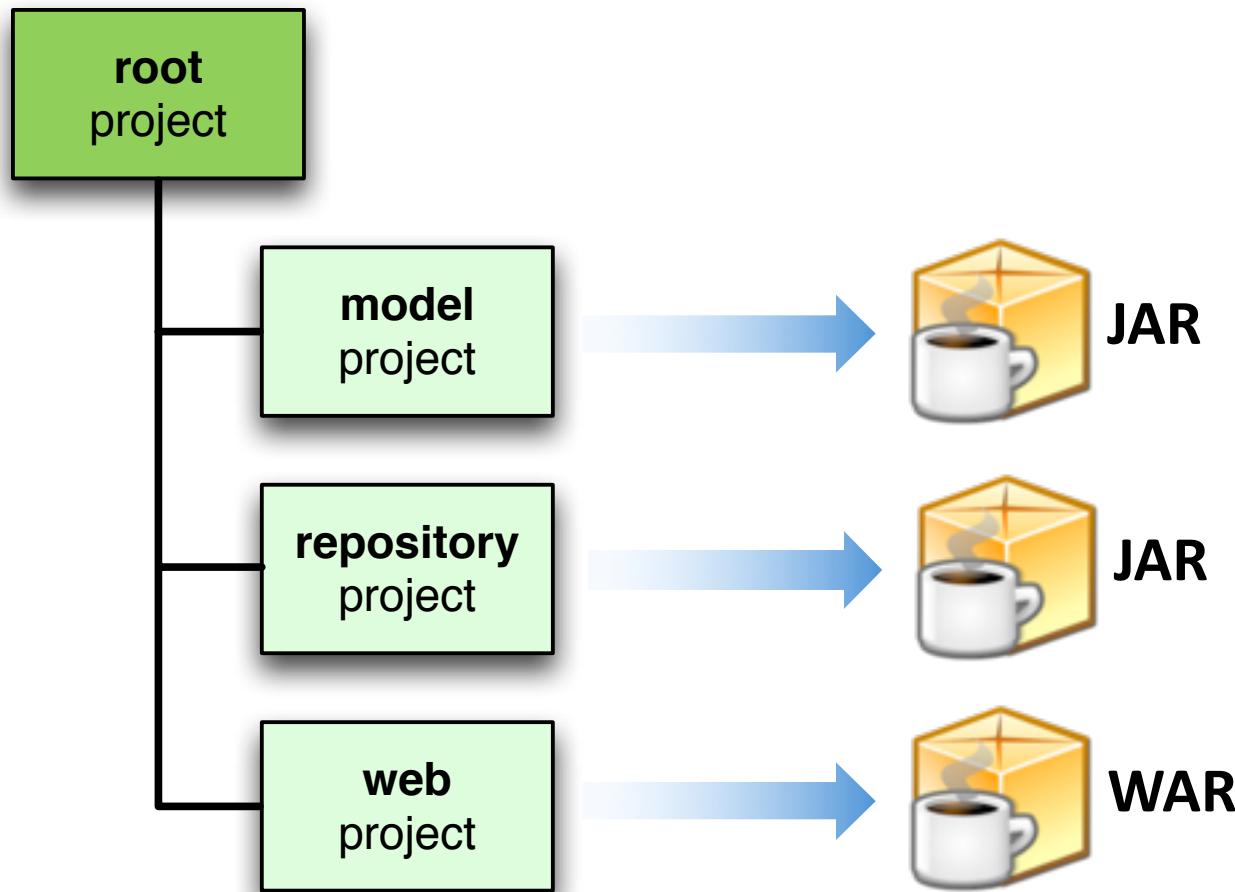


Externalize concerns into script plugins and organize them in dedicated directory

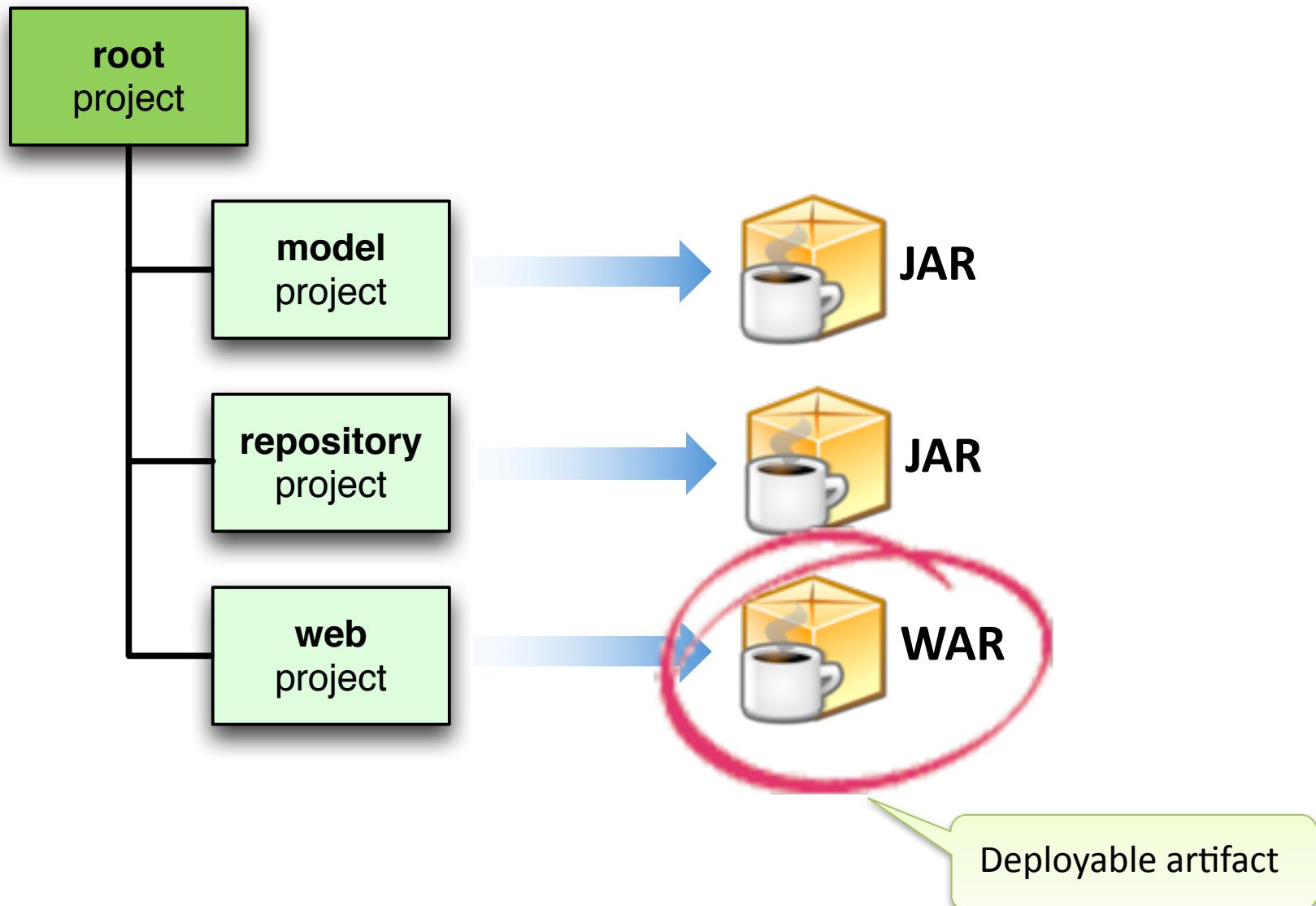
Examples:

- ✓ Versioning strategy
- ✓ Integration and functional test setup
- ✓ Deployment functionality
- ✓ ...

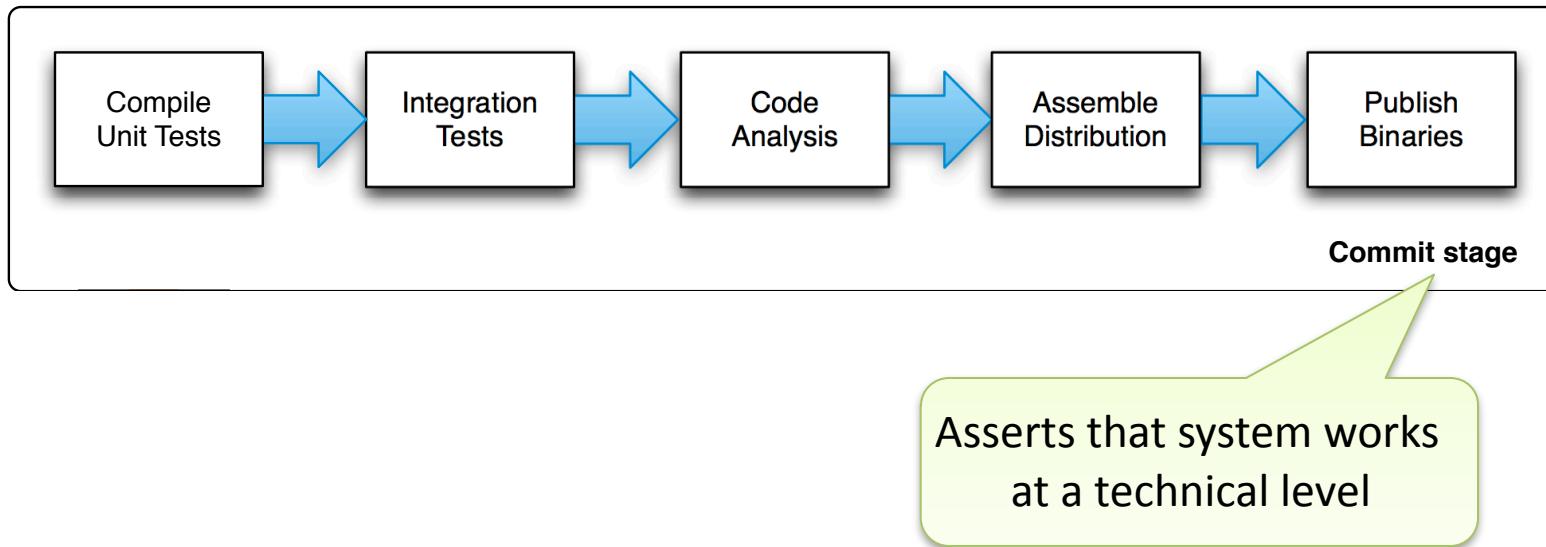
Project artifacts



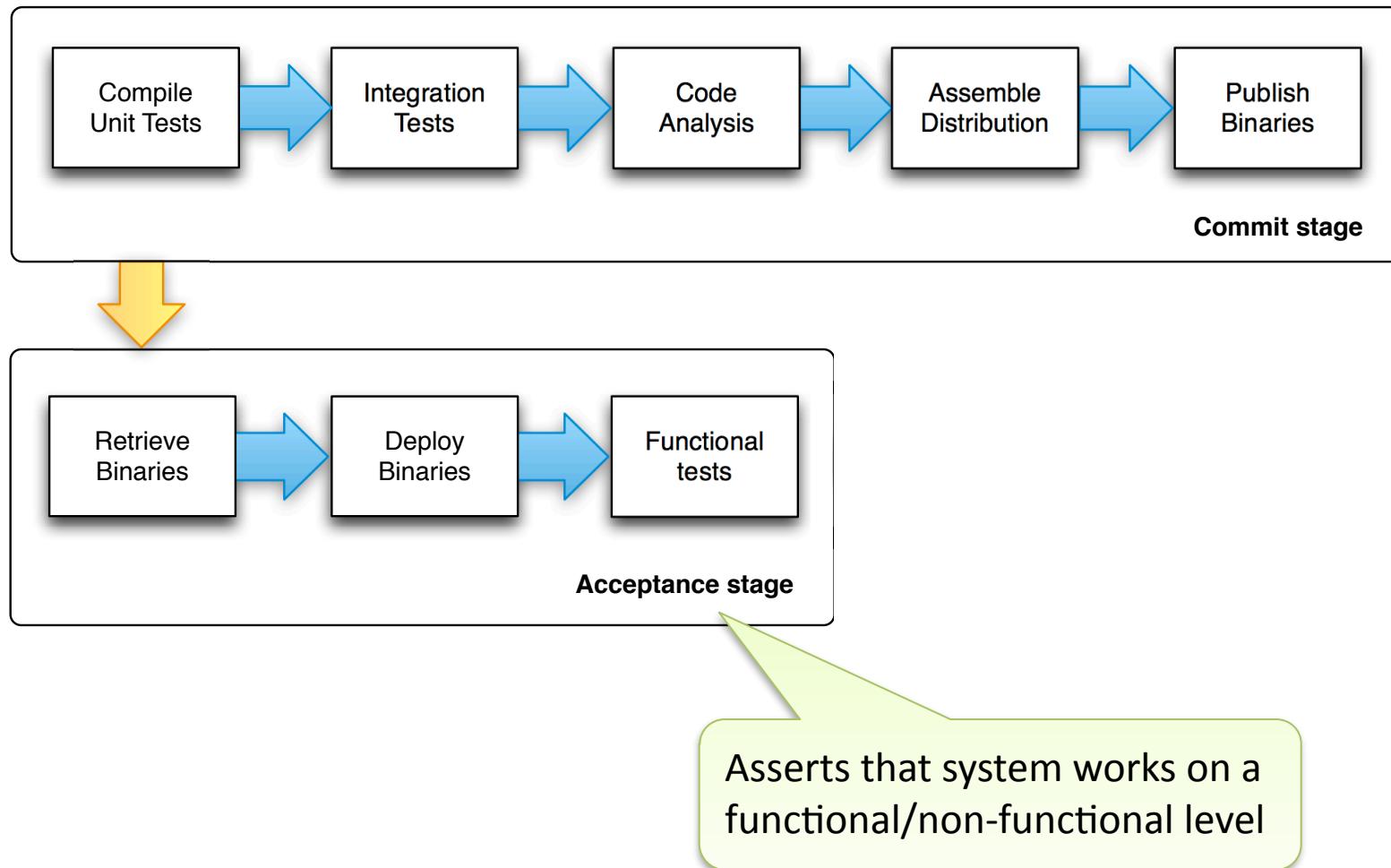
Project artifacts



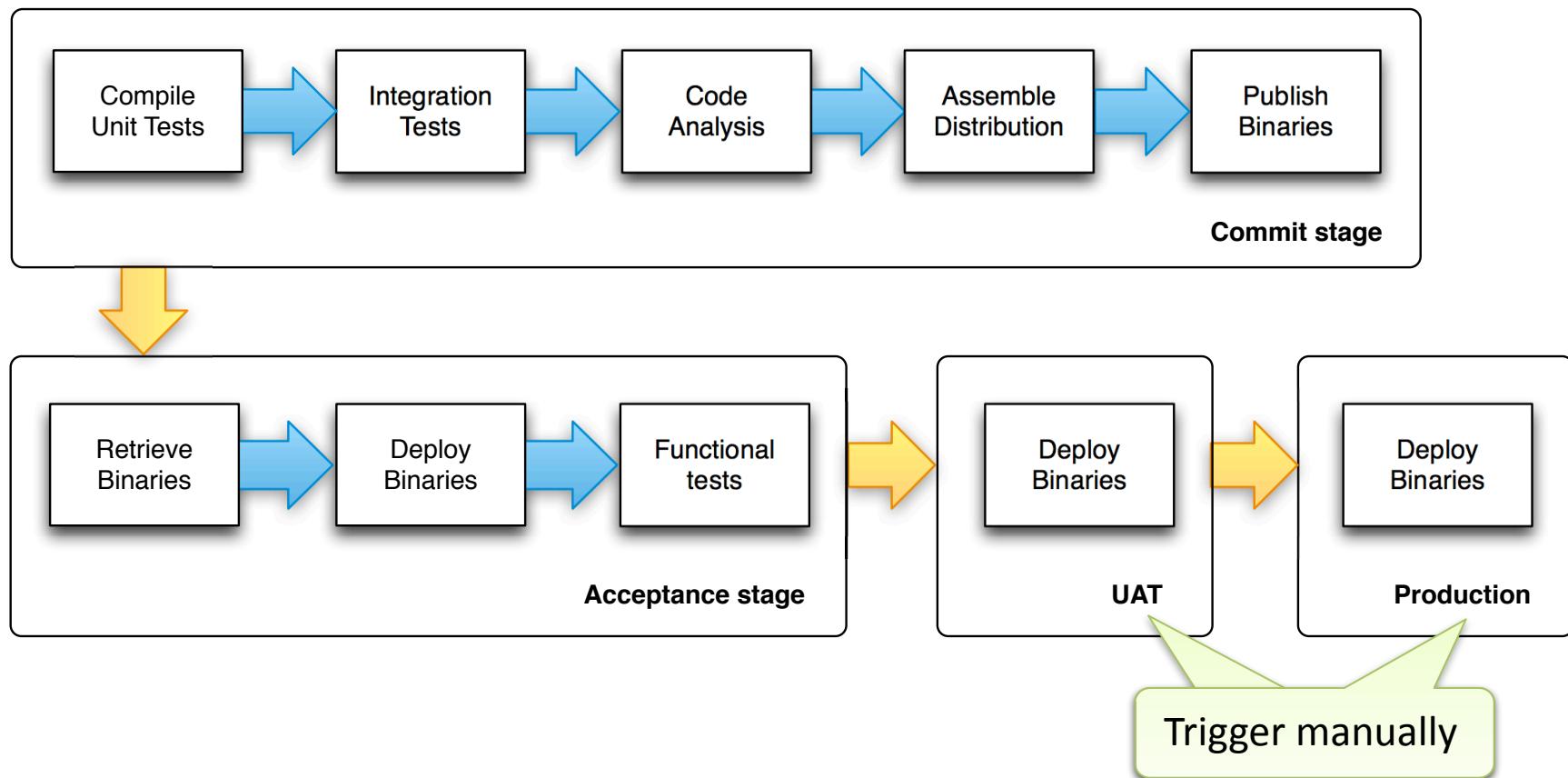
Stages in build pipeline



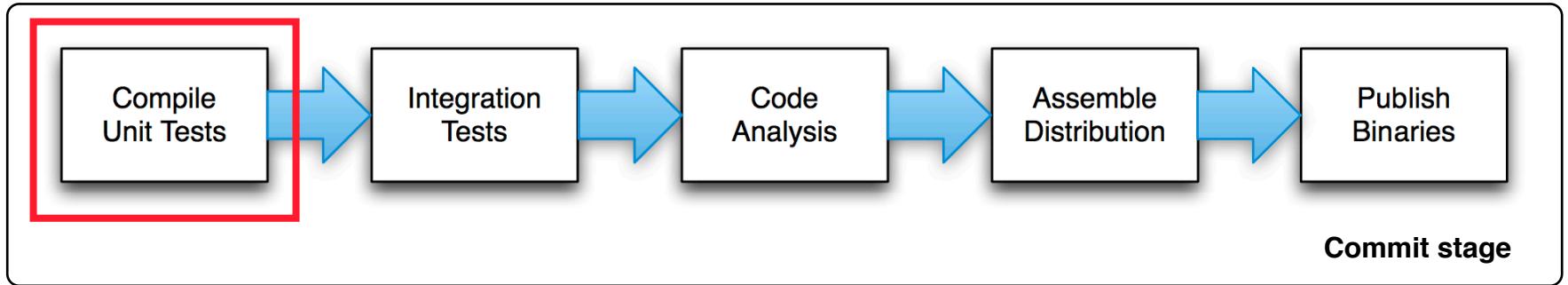
Stages in build pipeline



Stages in build pipeline



Commit stage: Compile/unit tests



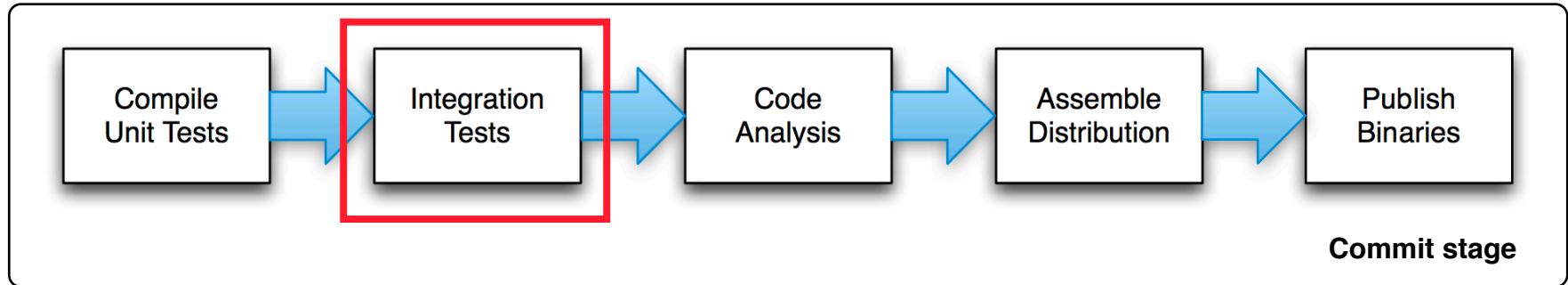
Rapid feedback (< 5 mins)

Run on every VCS check-in

Priority: fix broken build

```
String sql = "select * from stc
resultSet = statement.executeQuery();
if (resultSet.next()) {
    result = true;
    setstoreId(resultSet.getInt("s
    storeDescription = res...
    storeTypeId = re...
```

Commit stage: Integration tests



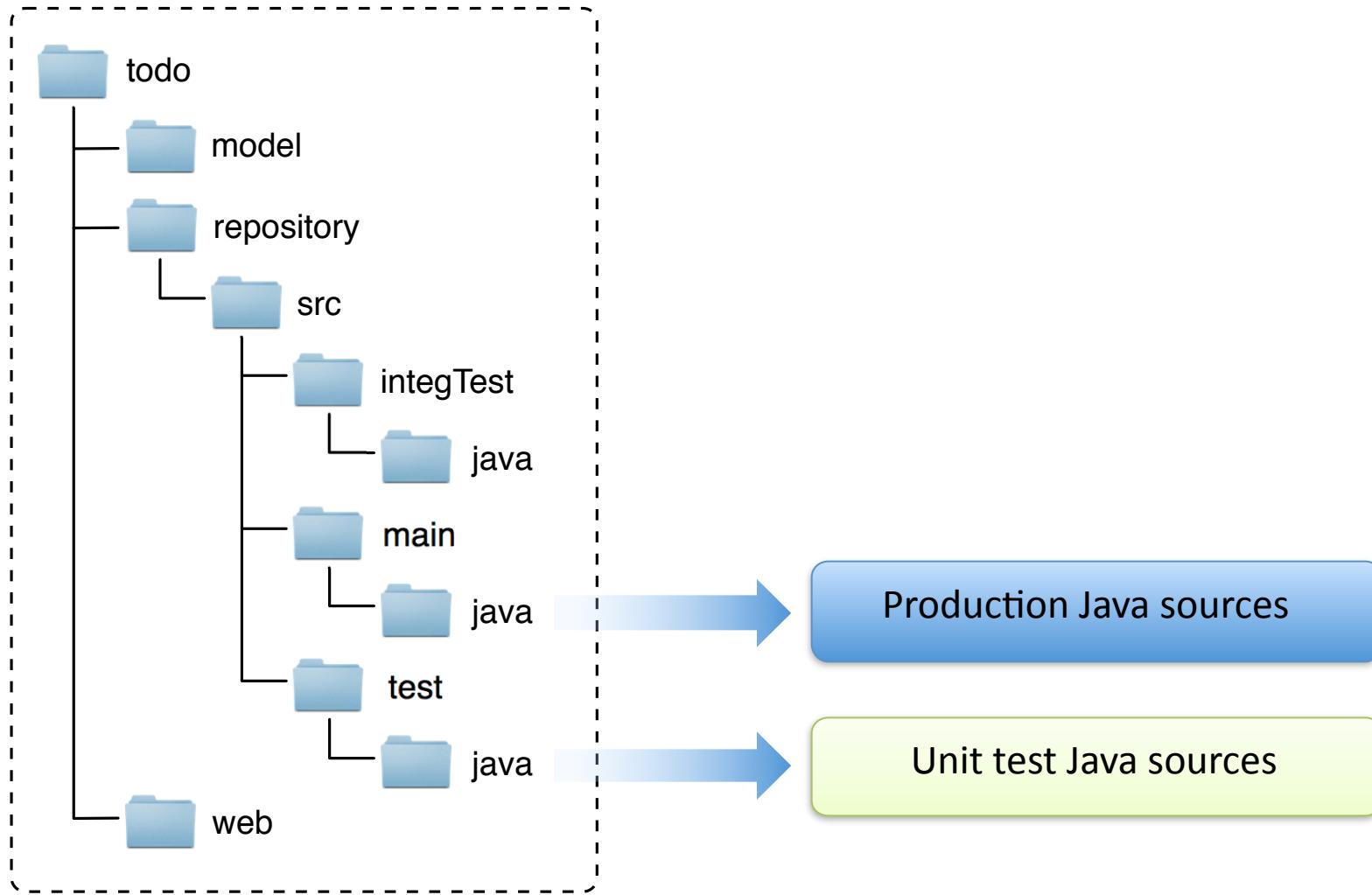
Longer running tests

Require environment setup

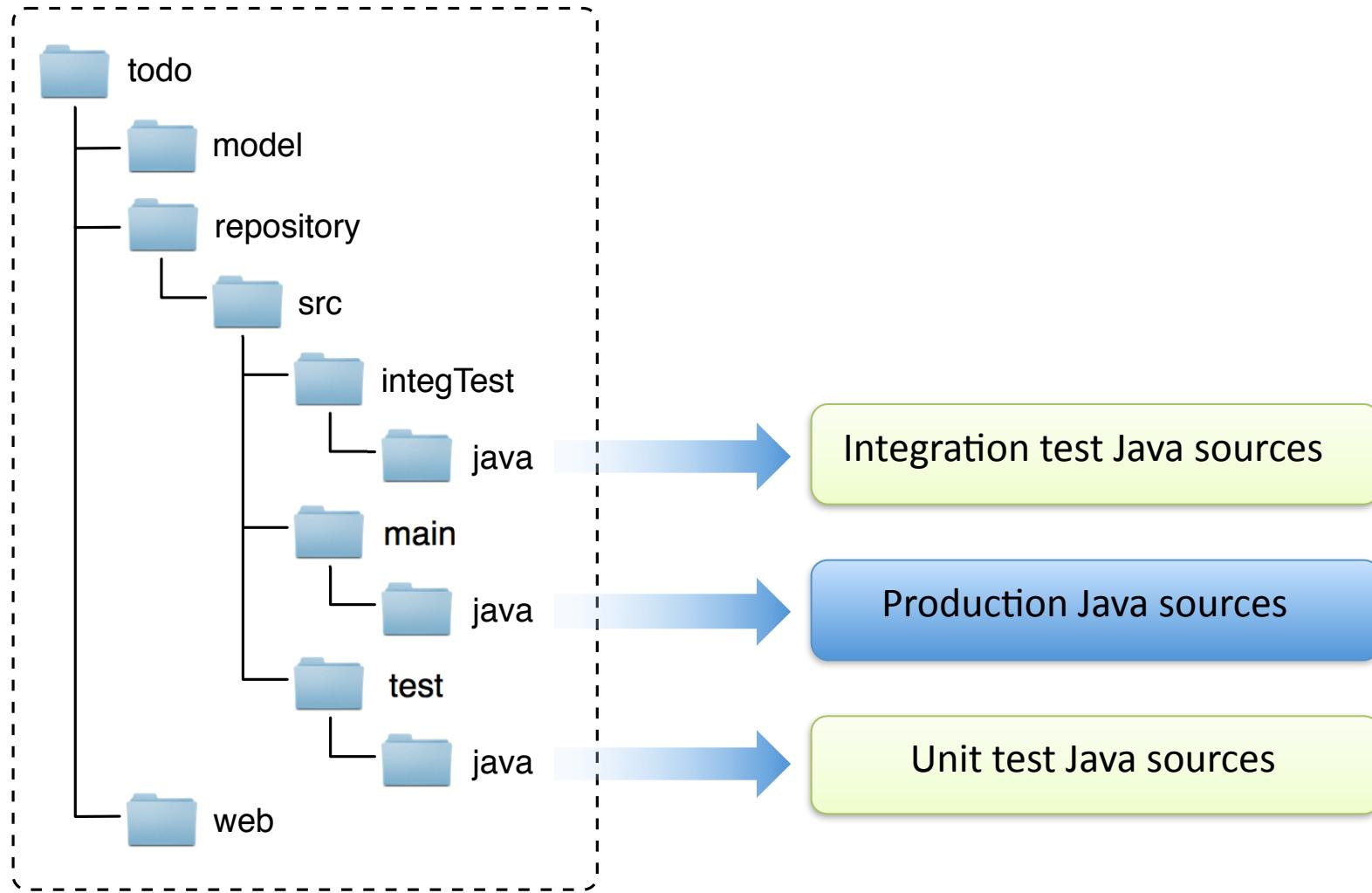
Harder to maintain



Separate tests in project layout



Separate tests in project layout



Separate tests with SourceSets

```
sourceSets {  
    integrationTest {  
        java.srcDir file('src/integTest/java')  
        resources.srcDir file('src/integTest/resources')  
        compileClasspath = sourceSets.main.output + configurations.testRuntime  
        runtimeClasspath = output + compileClasspath  
    }  
}  
  
task integrationTest(type: Test) {  
    description = 'Runs the integration tests.'  
    group = 'verification'  
    testClassesDir = sourceSets.integrationTest.output.classesDir  
    classpath = sourceSets.integrationTest.runtimeClasspath  
    testResultsDir = file("$testResultsDir/integration")  
}
```

Separate tests with SourceSets

```
sourceSets {  
    integrationTest {  
        java.srcDir file('src/integTest/java')  
        resources.srcDir file('src/integTest/resources')  
        compileClasspath = sourceSets.main.output + configurations.testRuntime  
        runtimeClasspath = output + compileClasspath  
    }  
}  
  
task integrationTest(type: Test) {  
    description = 'Runs the integration tests.'  
    group = 'verification'  
    testClassesDir = sourceSets.integrationTest.output.classesDir  
    classpath = sourceSets.integrationTest.runtimeClasspath  
    testResultsDir = file("$testResultsDir/integration")  
}
```

Set source and resources directory

Separate tests with SourceSets

```
sourceSets {  
    integrationTest {  
        java.srcDir file('src/integTest/java')  
        resources.srcDir file('src/integTest/resources')  
        compileClasspath = sourceSets.main.output + configurations.testRuntime  
        runtimeClasspath = output + compileClasspath  
    }  
}  
  
task integrationTest(type: Test) {  
    description = 'Runs the integration tests.'  
    group = 'verification'  
    testClassesDir = sourceSets.integrationTest.output.classesDir  
    classpath = sourceSets.integrationTest.runtimeClasspath  
    testResultsDir = file("$testResultsDir/integration")  
}
```

Set source and resources directory

Set compile and runtime classpath

Separate tests with SourceSets

```
sourceSets {  
    integrationTest {  
        java.srcDir file('src/integTest/java')  
        resources.srcDir file('src/integTest/resources')  
        compileClasspath = sourceSets.main.output + configurations.testRuntime  
        runtimeClasspath = output + compileClasspath  
    }  
}  
  
task integrationTest(type: Test) {  
    description = 'Runs the integration tests.'  
    group = 'verification'  
    testClassesDir = sourceSets.integrationTest.output.classesDir  
    classpath = sourceSets.integrationTest.runtimeClasspath  
    testResultsDir = file("$testResultsDir/integration")  
}
```

Set source and resources directory

Set compile and runtime classpath

Custom test results directory

Separate tests with SourceSets

```
sourceSets {  
    integrationTest {  
        java.srcDir file('src/integTest/java')  
        resources.srcDir file('src/integTest/resources')  
        compileClasspath = sourceSets.main.output + configurations.testRuntime  
        runtimeClasspath = output + compileClasspath  
    }  
}  
  
task integrationTest(type: Test) {  
    description = 'Runs the integration tests.'  
    group = 'verification'  
    testClassesDir = sourceSets.integrationTest.output.classesDir  
    classpath = sourceSets.integrationTest.runtimeClasspath  
    testResultsDir = file("$testResultsDir/integration")  
}
```

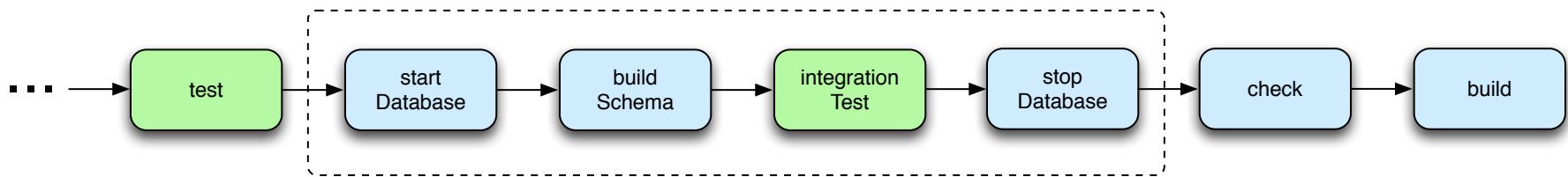
Set source and resources directory

Set compile and runtime classpath

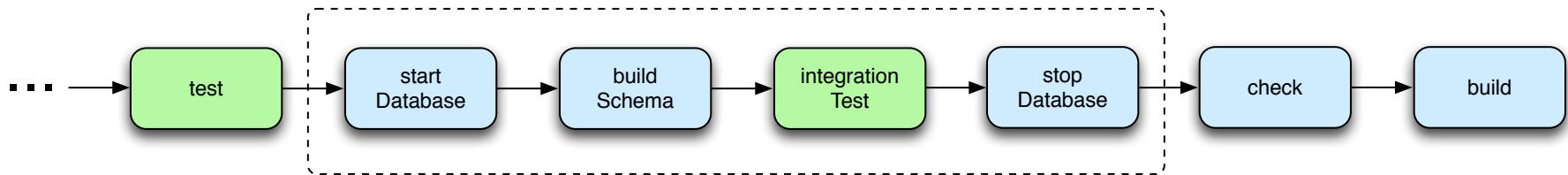
Custom test results directory

gradlew integrationTest

Database integration tests



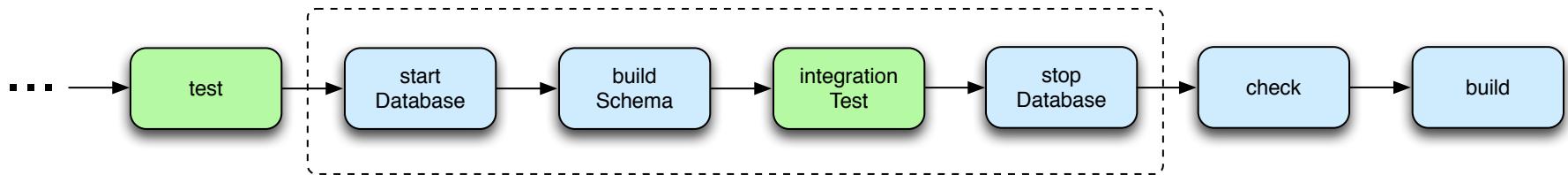
Database integration tests



LIQUI^{DB}ASE



Database integration tests

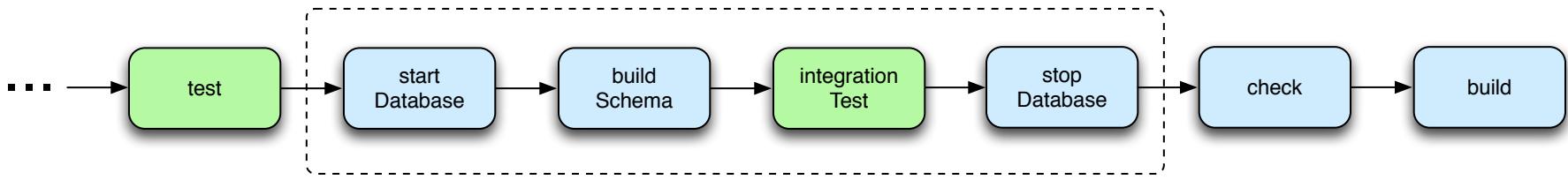


```
apply from: "$rootDir/gradle/databaseSetup.gradle"

integrationTest.dependsOn startAndPrepareDatabase
integrationTest.finalizedBy stopDatabase

check.dependsOn integrationTest
```

Database integration tests



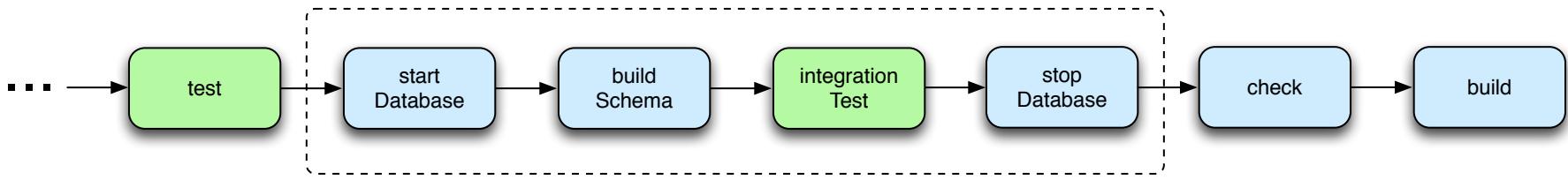
Separate complex setup logic into script plugin

```
apply from: "$rootDir/gradle/databaseSetup.gradle"

integrationTest.dependsOn startAndPrepareDatabase
integrationTest.finalizedBy stopDatabase

check.dependsOn integrationTest
```

Database integration tests



Separate complex setup logic into script plugin

```
apply from: "$rootDir/gradle/databaseSetup.gradle"

integrationTest.dependsOn startAndPrepareDatabase
integrationTest.finalizedBy stopDatabase

check.dependsOn integrationTest
```

Integrate tasks into build lifecycle

Picking the “right” code coverage tool

Picking the “right” code coverage tool

Cobertura



Offline bytecode
instrumentation

Picking the “right” code coverage tool

Cobertura



Offline bytecode
instrumentation



Offline bytecode
instrumentation

Picking the “right” code coverage tool

Cobertura



Offline bytecode
instrumentation



Offline bytecode
instrumentation

Clover



Source code
instrumentation

Picking the “right” code coverage tool

Cobertura



Offline bytecode
instrumentation



Offline bytecode
instrumentation

Clover



Source code
instrumentation



On-the-fly bytecode
instrumentation



On-the-fly bytecode instrumentation

No modification to source or bytecode

Code coverage with JaCoCo

```
apply plugin: "jacoco"

task jacocoIntegrationTestReport(type: JacocoReport) {
    sourceSets sourceSets.main
    executionData integTest
}
```

Code coverage with JaCoCo

Apply JaCoCo plugin

```
apply plugin: "jacoco"

task jacocoIntegrationTestReport(type: JacocoReport) {
    sourceSets sourceSets.main
    executionData integTest
}
```

Code coverage with JaCoCo

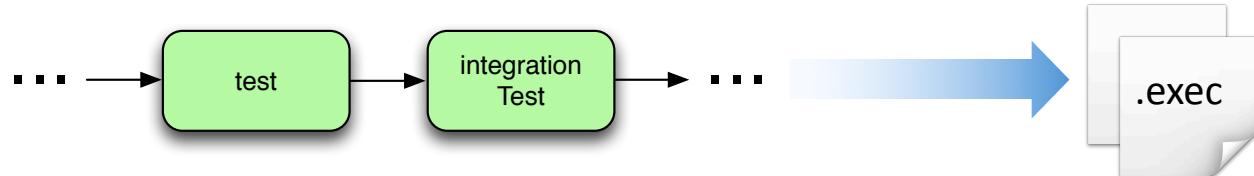
Apply JaCoCo plugin

```
apply plugin: "jacoco"

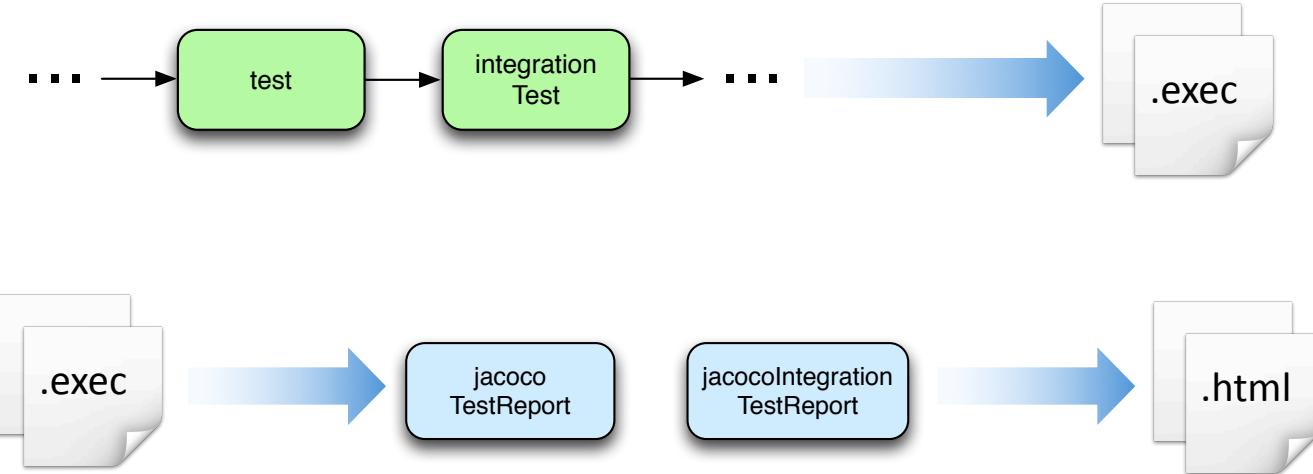
task jacocoIntegrationTestReport(type: JacocoReport) {
    sourceSets sourceSets.main
    executionData integTest
}
```

Also report code coverage
for integration tests

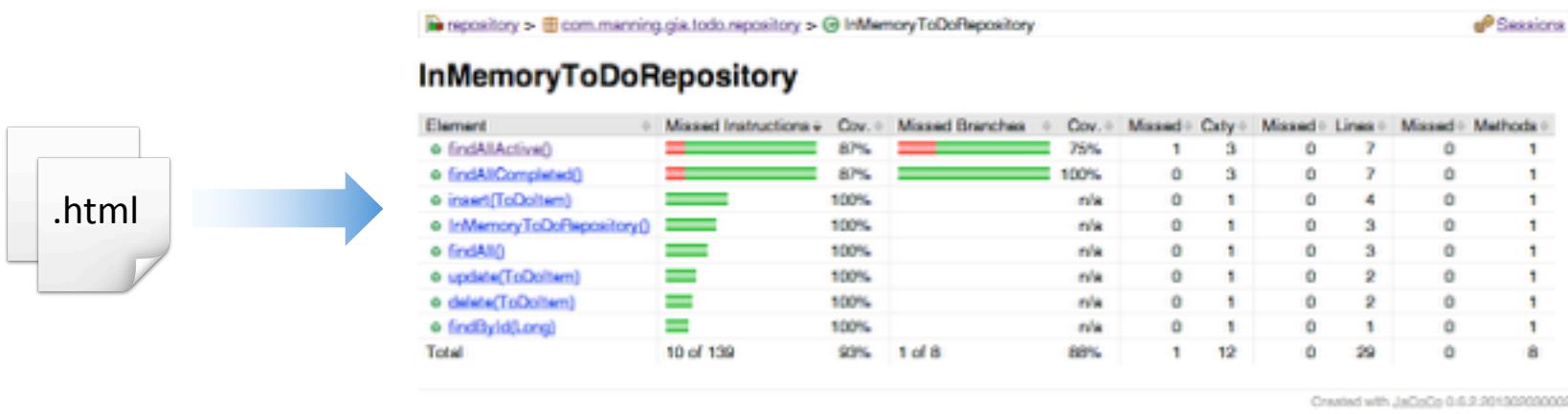
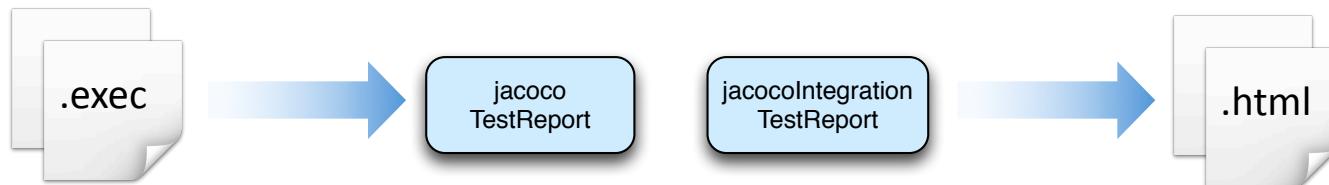
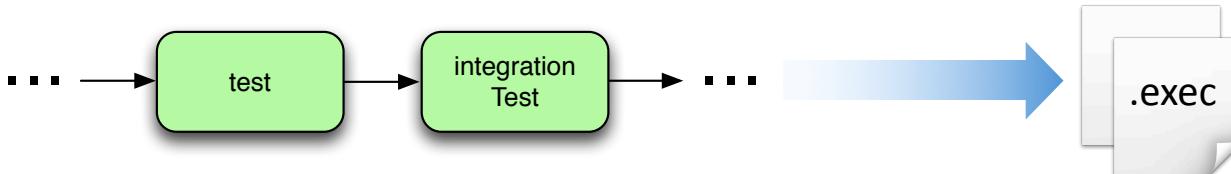
Generating coverage reports



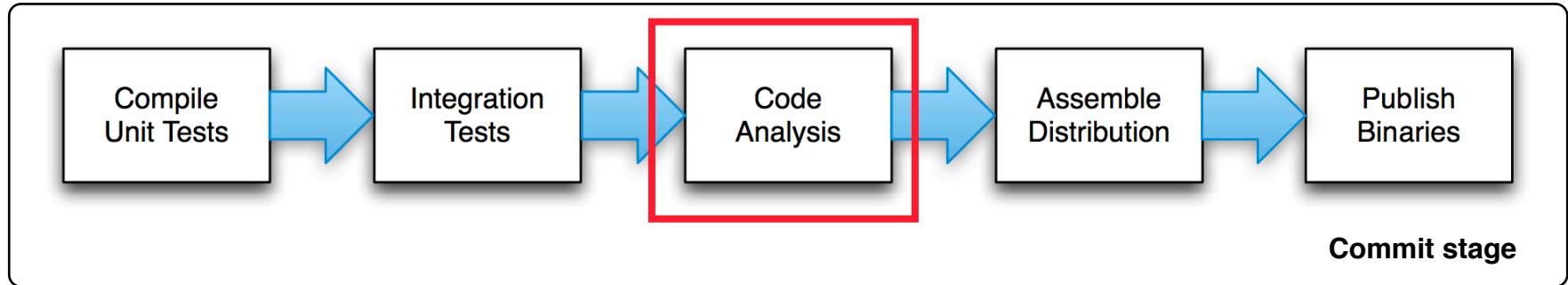
Generating coverage reports



Generating coverage reports



Commit stage: Code analysis



Perform code health check

Fail build for low quality

Record progress over time



Static code analysis tools



Checkstyle



FindBugs

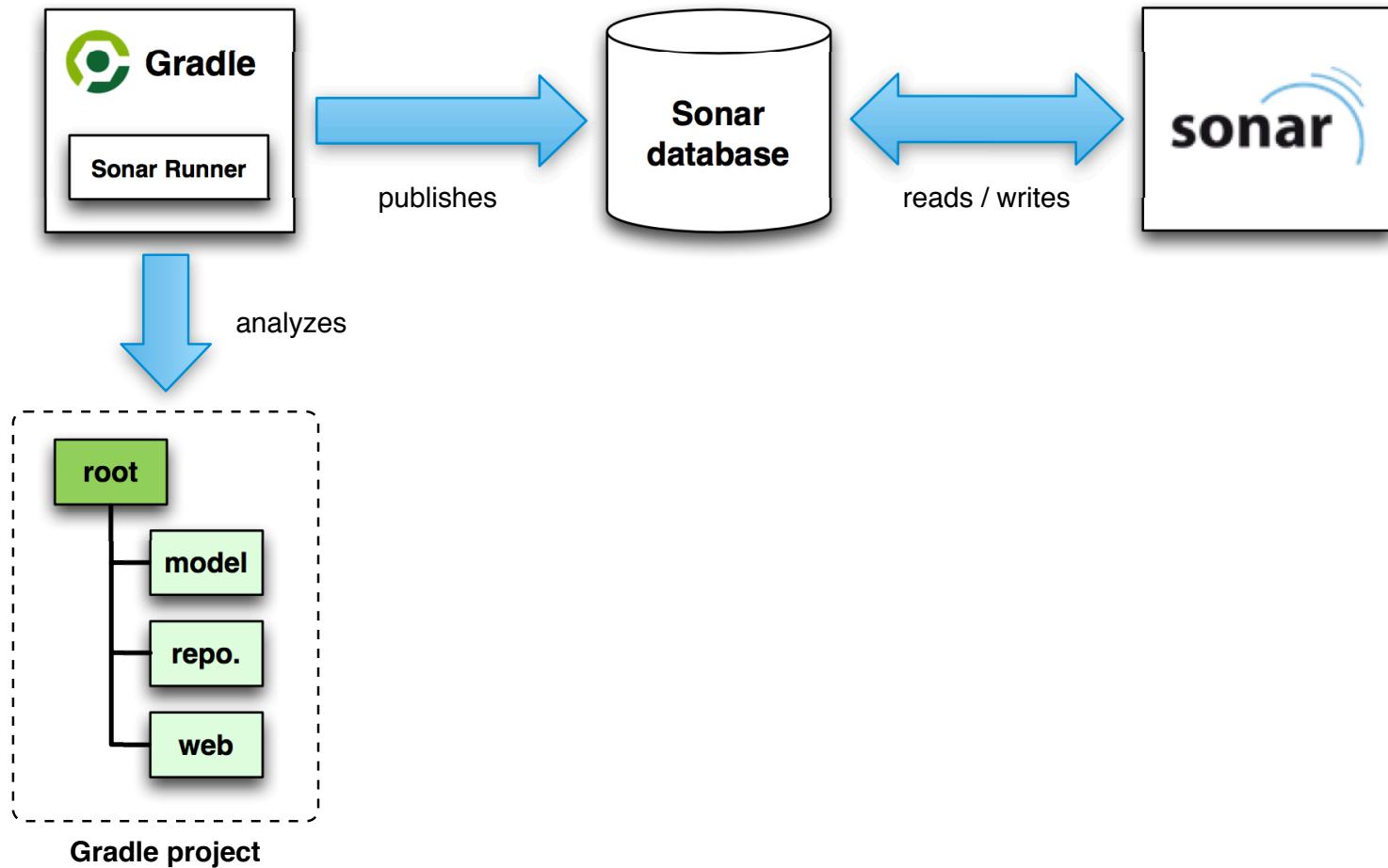
```
apply plugin: 'pmd'  
  
pmd {  
    ignoreFailures = true  
}  
  
tasks.withType(Pmd) {  
    reports {  
        xml.enabled = false  
        html.enabled = true  
    }  
}
```



```
apply plugin: 'jdepend'  
  
jdepend {  
    toolVersion = '2.9.1'  
    ignoreFailures = true  
}
```

gradlew check

Measure quality over time with Sonar



Applying the Sonar Runner plugin

```
apply plugin: "sonar-runner"

sonarRunner {
    sonarProperties {
        property "sonar.host.url", "http://my.server.com"
        property "sonar.jdbc.url", "jdbc:mysql://my.server.com/sonar"
        property "sonar.jdbc.driverClassName", "com.mysql.jdbc.Driver"
        property "sonar.jdbc.username", "Fred Flintstone"
        property "sonar.jdbc.password", "very clever"
    }
}

subprojects {
    sonarRunner {
        sonarProperties {
            property "sonar.sourceEncoding", "UTF-8"
        }
    }
}
```

gradlew sonarRunner

Safari File Edit View History Bookmarks Develop Window Help

Thu 2:14 PM Benjamin Muschko

localhost:9000

Sonar

Dashboards Projects Measures Reviews Settings Administrator Search

Home

TOOLS

Dependencies Compare

sonar

Welcome to Sonar Dashboard

Since you are able to read this, it means that you have successfully started your Sonar server. Well done!

If you have not removed this text, it also means that you have not yet played much with Sonar. So here are a few pointers for your next step:

- » Do you now want to [run analysis](#) on a project?
- » Maybe start [customizing dashboards](#)?
- » Or simply browse the [complete documentation](#)?
- » If you have a question or an issue, please report it on the [mailing list](#).

My favourites

A Name - Date

No data

Projects

A Name - Version LOCs RCI Date

No data

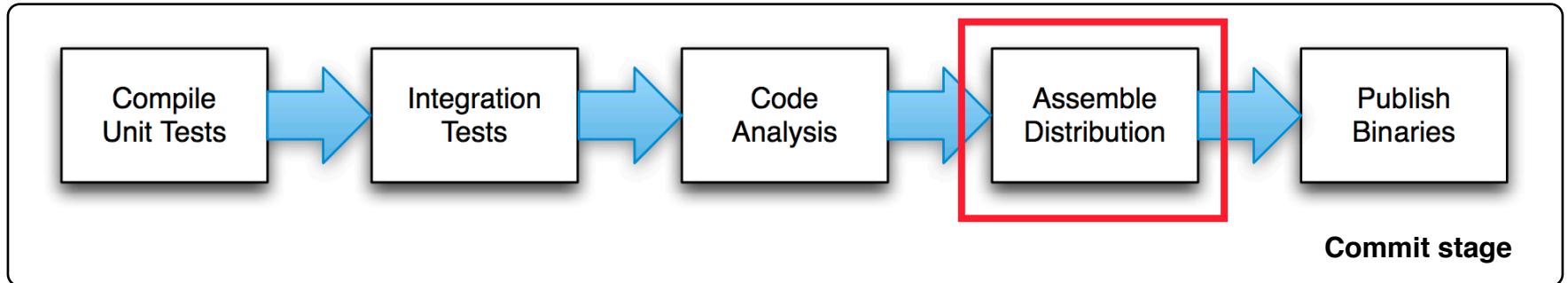
Projects

Size: Lines of code Color: Rules compliance 0.0% 100.0%

No data

Powered by SonarSource - Open Source LGPL - v.3.5 - Plugins - Documentation - Ask a question
Embedded database should be used for evaluation purpose only

Commit stage: Assemble distribution



Exclude env. configuration

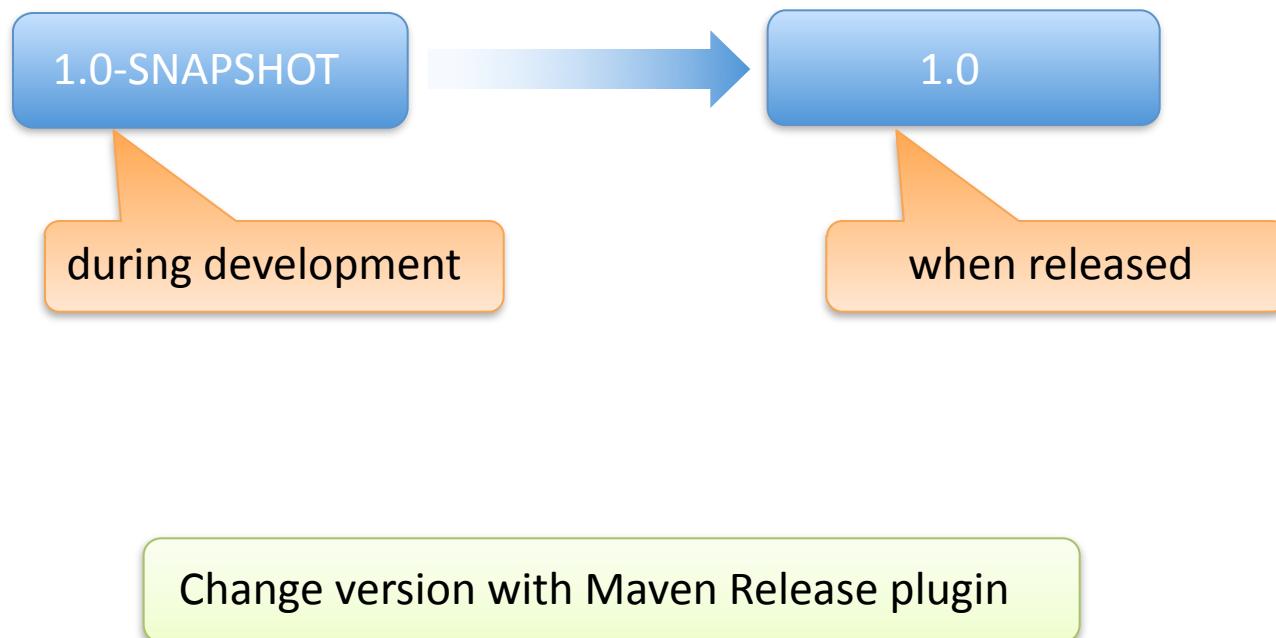
Include build information

Choose versioning strategy



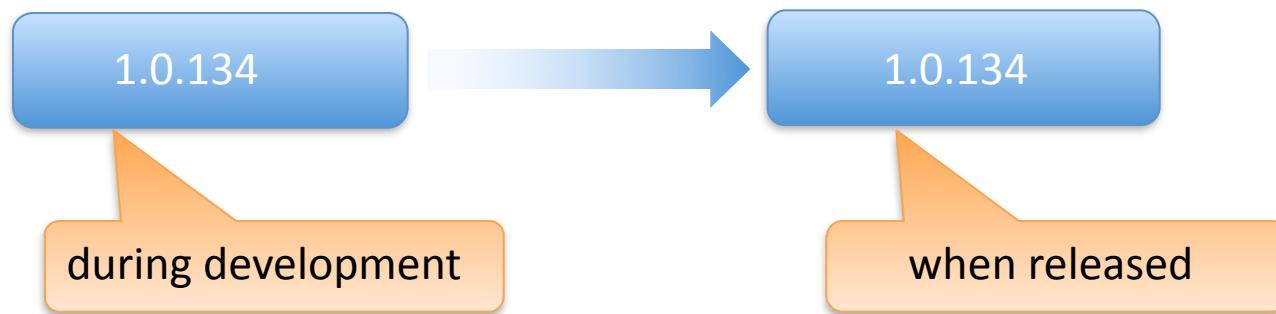
Versioning strategy

...the Maven way



Versioning strategy

...the Continuous Delivery way



1.0.134

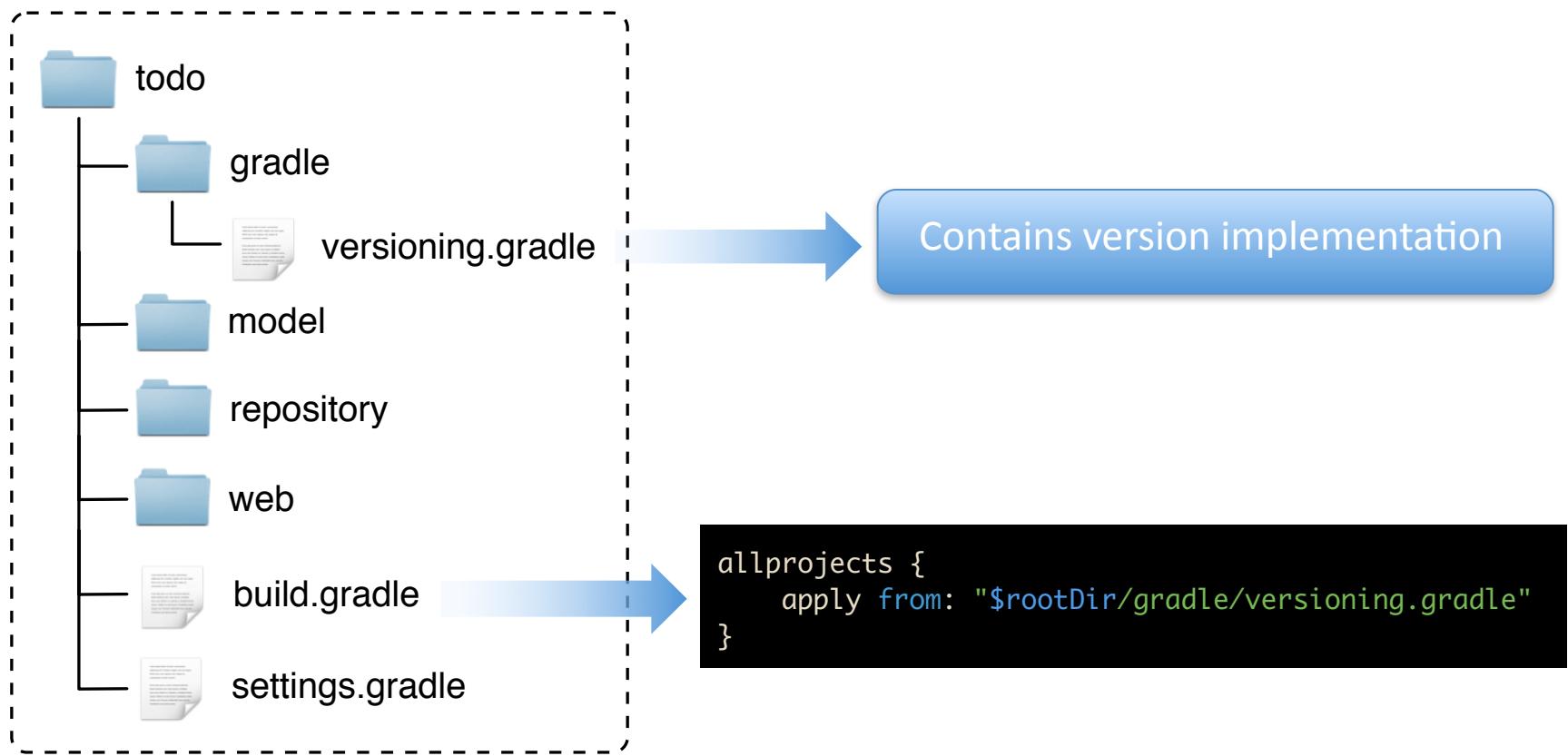
Project version number

Jenkins build number



Versioning strategy

...implemented with Gradle



Versioning strategy

...implemented with Gradle

```
ext.buildTimestamp = new Date().format('yyyy-MM-dd HH:mm:ss')

version = new ProjectVersion(1, 0, System.env.SOURCE_BUILD_NUMBER)

class ProjectVersion {
    Integer major
    Integer minor
    String build

    ProjectVersion(Integer major, Integer minor, String build) {
        this.major = major
        this.minor = minor
        this.build = build
    }

    @Override
    String toString() {
        String fullVersion = "$major.$minor"

        if(build) {
            fullVersion += ".$build"
        }

        fullVersion
    }
}
```

Versioning strategy

...implemented with Gradle

```
ext.buildTimestamp = new Date().format('yyyy-MM-dd HH:mm:ss')

version = new ProjectVersion(1, 0, System.env.SOURCE_BUILD_NUMBER)

class ProjectVersion {
    Integer major
    Integer minor
    String build

    ProjectVersion(Integer major, Integer minor, String build) {
        this.major = major
        this.minor = minor
        this.build = build
    }

    @Override
    String toString() {
        String fullVersion = "$major.$minor"

        if(build) {
            fullVersion += ".$build"
        }

        fullVersion
    }
}
```

Jenkins Build Number

Versioning strategy

...implemented with Gradle

```
ext.buildTimestamp = new Date().format('yyyy-MM-dd HH:mm:ss')

version = new ProjectVersion(1, 0, System.env.SOURCE_BUILD_NUMBER)

class ProjectVersion {
    Integer major
    Integer minor
    String build

    ProjectVersion(Integer major, Integer minor, String build) {
        this.major = major
        this.minor = minor
        this.build = build
    }

    @Override
    String toString() {
        String fullVersion = "$major.$minor"

        if(build) {
            fullVersion += ".$build"
        }

        fullVersion
    }
}
```

Jenkins Build Number

Builds version
String representation

Packaging the deployable artifact



```
project(':web') {
    apply plugin: 'war'

    task createBuildInfoFile << {
        def buildInfoFile = new File("$buildDir/build-info.properties")
        Properties props = new Properties()
        props.setProperty('version', project.version.toString())
        props.setProperty('timestamp', project.buildTimestamp)
        props.store(buildInfoFile.newWriter(), null)
    }

    war {
        dependsOn createBuildInfoFile
        baseName = 'todo'

        from(buildDir) {
            include 'build-info.properties'
            into('WEB-INF/classes')
        }
    }
}
```

Packaging the deployable artifact



```
project(':web') {
    apply plugin: 'war'

    task createBuildInfoFile << {
        def buildInfoFile = new File("$buildDir/build-info.properties")
        Properties props = new Properties()
        props.setProperty('version', project.version.toString())
        props.setProperty('timestamp', project.buildTimestamp)
        props.store(buildInfoFile.newWriter(), null)
    }

    war {
        dependsOn createBuildInfoFile
        baseName = 'todo'

        from(buildDir) {
            include 'build-info.properties'
            into('WEB-INF/classes')
        }
    }
}
```

Creates file containing
build information

Packaging the deployable artifact



```
project(':web') {
    apply plugin: 'war'

    task createBuildInfoFile << {
        def buildInfoFile = new File("$buildDir/build-info.properties")
        Properties props = new Properties()
        props.setProperty('version', project.version.toString())
        props.setProperty('timestamp', project.buildTimestamp)
        props.store(buildInfoFile.newWriter(), null)
    }

    war {
        dependsOn createBuildInfoFile
        baseName = 'todo'

        from(buildDir) {
            include 'build-info.properties'
            into('WEB-INF/classes')
        }
    }
}
```

Creates file containing
build information

Include build info file
Into WAR distribution

Packaging the deployable artifact



```
project(':web') {
    apply plugin: 'war'

    task createBuildInfoFile << {
        def buildInfoFile = new File("$buildDir/build-info.properties")
        Properties props = new Properties()
        props.setProperty('version', project.version.toString())
        props.setProperty('timestamp', project.buildTimestamp)
        props.store(buildInfoFile.newWriter(), null)
    }

    war {
        dependsOn createBuildInfoFile
        baseName = 'todo'

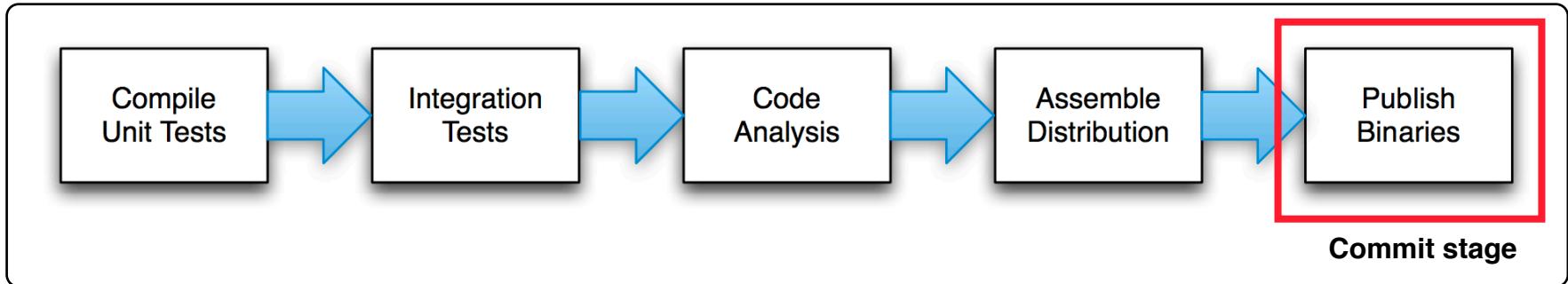
        from(buildDir) {
            include 'build-info.properties'
            into('WEB-INF/classes')
        }
    }
}
```

Creates file containing
build information

Include build info file
Into WAR distribution

gradlew assemble

Commit stage: Publish binaries



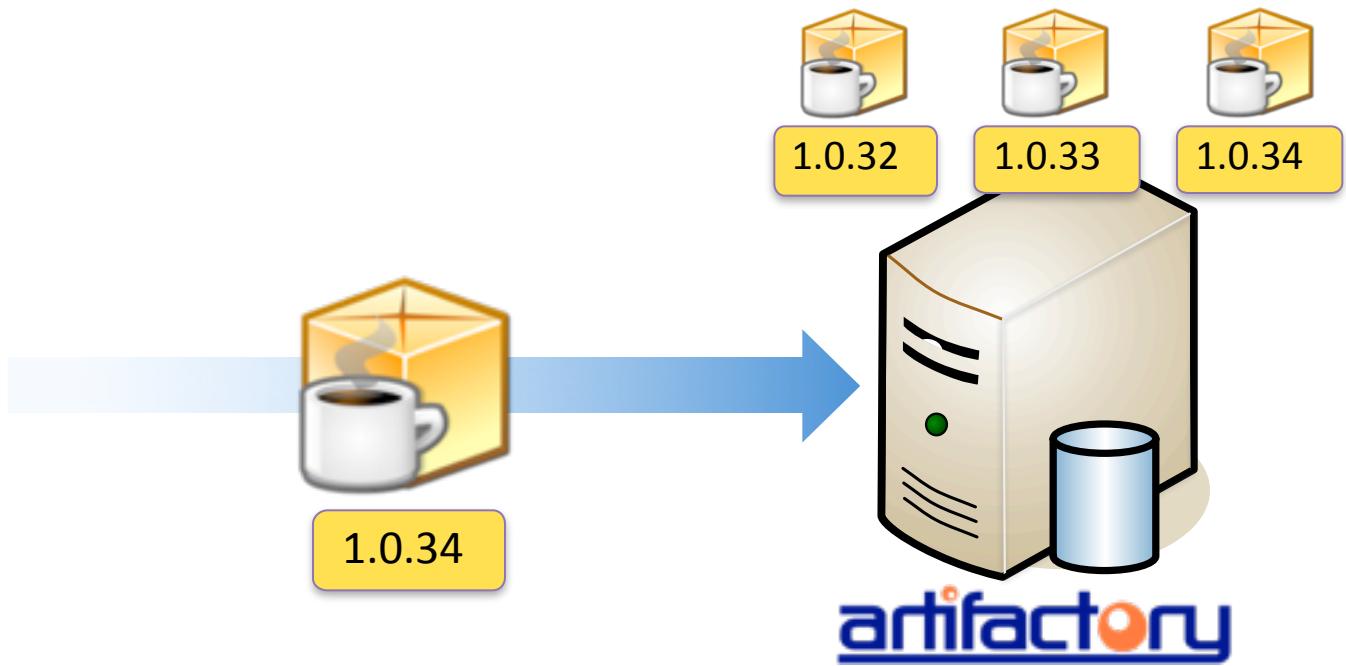
Version artifact(s)

Use binary repository

Publish once, then reuse



Publishing the deployable artifact



Defining build configuration

```
binaryRepository {  
    url = 'http://mycompany.bin.repo:8081/artifactory'  
    username = 'admin'  
    password = 'password'  
    name = 'libs-release-local'  
}  
  
environments {  
    test {  
        server {  
            hostname = 'mycompany.test'  
            port = 8099  
            context = 'todo'  
            username = 'manager'  
            password = 'manager'  
        }  
    }  
  
    uat {  
        server {  
            hostname = 'mycompany.uat'  
            port = 8199  
            context = 'todo'  
            username = 'manager'  
            password = 'manager'  
        }  
    }  
}
```

Defining build configuration

Common
configuration

```
binaryRepository {  
    url = 'http://mycompany.bin.repo:8081/artifactory'  
    username = 'admin'  
    password = 'password'  
    name = 'libs-release-local'  
}  
  
environments {  
    test {  
        server {  
            hostname = 'mycompany.test'  
            port = 8099  
            context = 'todo'  
            username = 'manager'  
            password = 'manager'  
        }  
    }  
  
    uat {  
        server {  
            hostname = 'mycompany.uat'  
            port = 8199  
            context = 'todo'  
            username = 'manager'  
            password = 'manager'  
        }  
    }  
}
```

Defining build configuration

Common
configuration

Environment-specific
configuration

```
binaryRepository {  
    url = 'http://mycompany.bin.repo:8081/artifactory'  
    username = 'admin'  
    password = 'password'  
    name = 'libs-release-local'  
}  
  
environments {  
    test {  
        server {  
            hostname = 'mycompany.test'  
            port = 8099  
            context = 'todo'  
            username = 'manager'  
            password = 'manager'  
        }  
    }  
  
    uat {  
        server {  
            hostname = 'mycompany.uat'  
            port = 8199  
            context = 'todo'  
            username = 'manager'  
            password = 'manager'  
        }  
    }  
}
```

Defining build configuration

Common configuration

Environment-specific configuration

```
binaryRepository {  
    url = 'http://mycompany.bin.repo:8081/artifactory'  
    username = 'admin'  
    password = 'password'  
    name = 'libs-release-local'  
}
```

Read credentials from gradle.properties

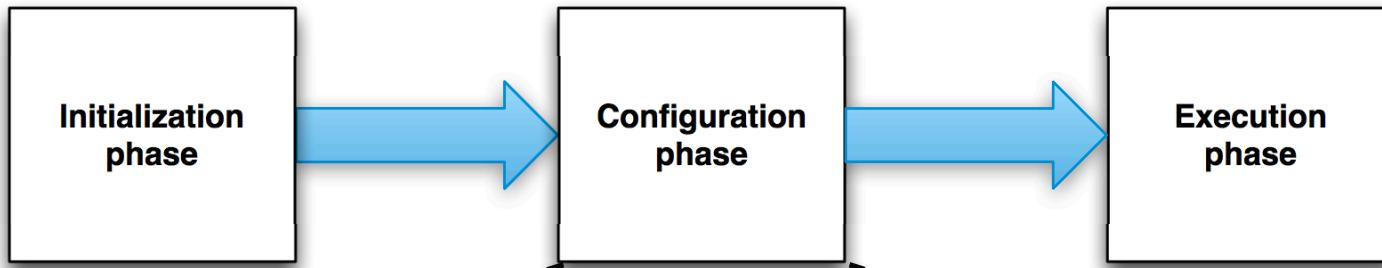
```
environments {  
    test {  
        server {  
            hostname = 'mycompany.test'  
            port = 8099  
            context = 'todo'  
            username = 'manager'  
            password = 'manager'  
        }  
    }  
}
```

Read credentials from gradle.properties

```
uat {  
    server {  
        hostname = 'mycompany.uat'  
        port = 8199  
        context = 'todo'  
        username = 'manager'  
        password = 'manager'  
    }  
}
```

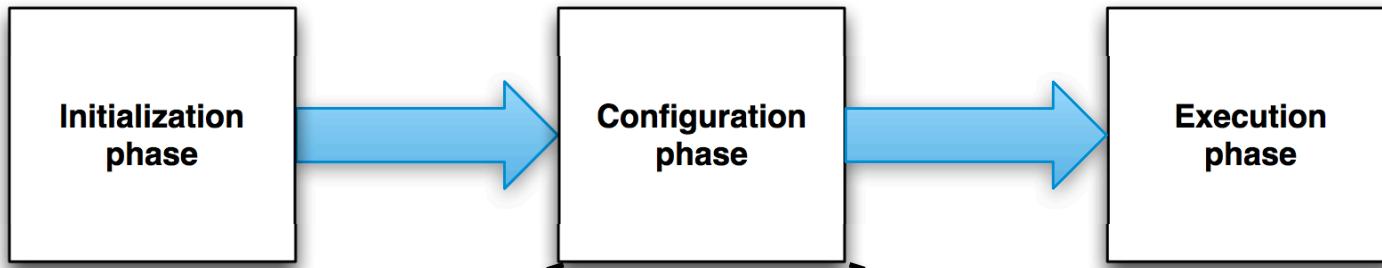
Read credentials from gradle.properties

Reading build configuration



```
def env = project.hasProperty('env') ? project.getProperty('env') : 'test'  
logger.quiet "Loading configuration for environment '$env'."  
  
def configFile = file("$rootDir/gradle/config/buildConfig.groovy")  
def parsedConfig = new ConfigSlurper(env).parse(configFile.toURL())  
  
ext.config = parsedConfig
```

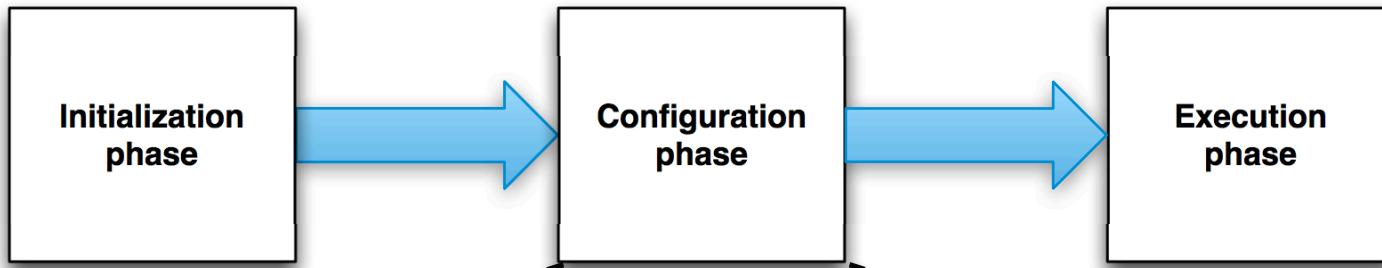
Reading build configuration



```
def env = project.hasProperty('env') ? project.getProperty('env') : 'test'  
logger.quiet "Loading configuration for environment '$env'."  
  
def configFile = file("$rootDir/gradle/config/buildConfig.groovy")  
def parsedConfig = new ConfigSlurper(env).parse(configFile.toURL())  
  
ext.config = parsedConfig
```

Assign configuration to extra property

Reading build configuration



```
def env = project.hasProperty('env') ? project.getProperty('env') : 'test'  
logger.quiet "Loading configuration for environment '$env'."  
  
def configFile = file("$rootDir/gradle/config/buildConfig.groovy")  
def parsedConfig = new ConfigSlurper(env).parse(configFile.toURL())  
  
ext.config = parsedConfig
```

Assign configuration to extra property

gradlew -Penv=uat ...

Using the Maven Publishing plugin

```
apply plugin: 'maven-publish'

def fullRepoUrl = "$config.binaryRepository.url/$config.binaryRepository.name"

publishing {
    publications {
        webApp(MavenPublication) {
            from components.web
        }
    }

    repositories {
        maven {
            url fullRepoUrl

            credentials {
                username = config.binaryRepository.username
                password = config.binaryRepository.password
            }
        }
    }
}
```

Using the Maven Publishing plugin

Build repository URL
from configuration

```
apply plugin: 'maven-publish'

def fullRepoUrl = "$config.binaryRepository.url/$config.binaryRepository.name"

publishing {
    publications {
        webApp(MavenPublication) {
            from components.web
        }
    }

    repositories {
        maven {
            url fullRepoUrl

            credentials {
                username = config.binaryRepository.username
                password = config.binaryRepository.password
            }
        }
    }
}
```

Using the Maven Publishing plugin

```
apply plugin: 'maven-publish'

def fullRepoUrl = "$config.binaryRepository.url/$config.binaryRepository.name"

publishing {
    publications {
        webApp(MavenPublication) {
            from components.web
        }
    }

    repositories {
        maven {
            url fullRepoUrl

            credentials {
                username = config.binaryRepository.username
                password = config.binaryRepository.password
            }
        }
    }
}
```

Build repository URL
from configuration

Assign publication name
and component

Using the Maven Publishing plugin

```
apply plugin: 'maven-publish'

def fullRepoUrl = "$config.binaryRepository.url/$config.binaryRepository.name"

publishing {
    publications {
        webApp(MavenPublication) {
            from components.web
        }
    }

    repositories {
        maven {
            url fullRepoUrl

            credentials {
                username = config.binaryRepository.username
                password = config.binaryRepository.password
            }
        }
    }
}
```

Build repository URL
from configuration

Assign publication name
and component

gradlew publish

Safari File Edit View History Bookmarks Develop Window Help Thu 2:58 PM Benjamin Muschko

localhost:8081/artifactory/webapp/browserrepo.html?10

Logged In as admin | Log Out

Home Artifacts Deploy Admin

Repository Browser

Tree Browser Simple Browser Search Results Builds

Search Quick Search Class Search GAVC Search Property Search Checksum Search Remote Search

ext-release-local ext-snapshot-local libs-release-local libs-snapshot-local plugins-release-local plugins-snapshot-local codehaus-cache google-code-cache gradle-libs-cache gradle-plugins-cache java.net.m1-cache java.net.m2-cache jboss-cache jcenter-cache jfrog-libs-cache jfrog-plugins-cache repot-cache spring-milestone-cache spring-release-cache

General Effective Permissions Properties Watchers

Info

Name: **libs-release-local** [?]
Description: Local repository for in-house libraries
Created: 21-05-13 13:48:17 EDT
Artifact Count: [Show...](#)
Repository Path: **libs-release-local** [?]
Repository Layout: maven-2-default

Actions

Refresh Delete Versions... Copy Content...
Delete Content Move Content...

Virtual Repository Associations

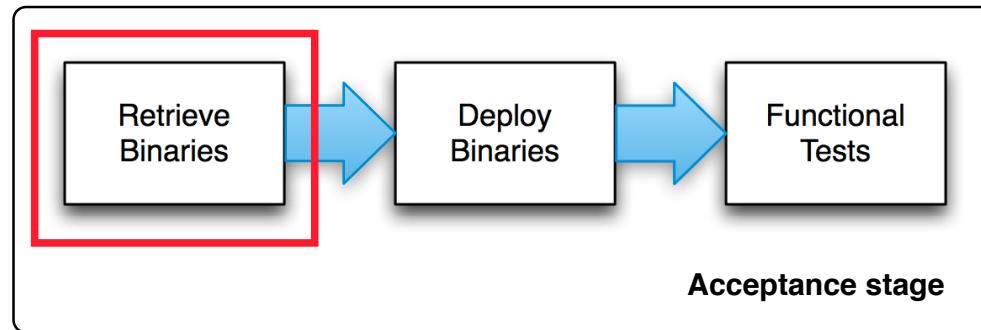
repo libs-release

Distribution Management

```
<distributionManagement>
    <repository>
        <id>Benjamins-MacBook-Pro.local</id>
        <name>Benjamins-MacBook-Pro.local-releases</name>
        <url>http://localhost:8081/artifactory/libs-release-local</url>
    </repository>
</distributionManagement>
```

Compact empty folders

Acceptance stage: Retrieve binaries

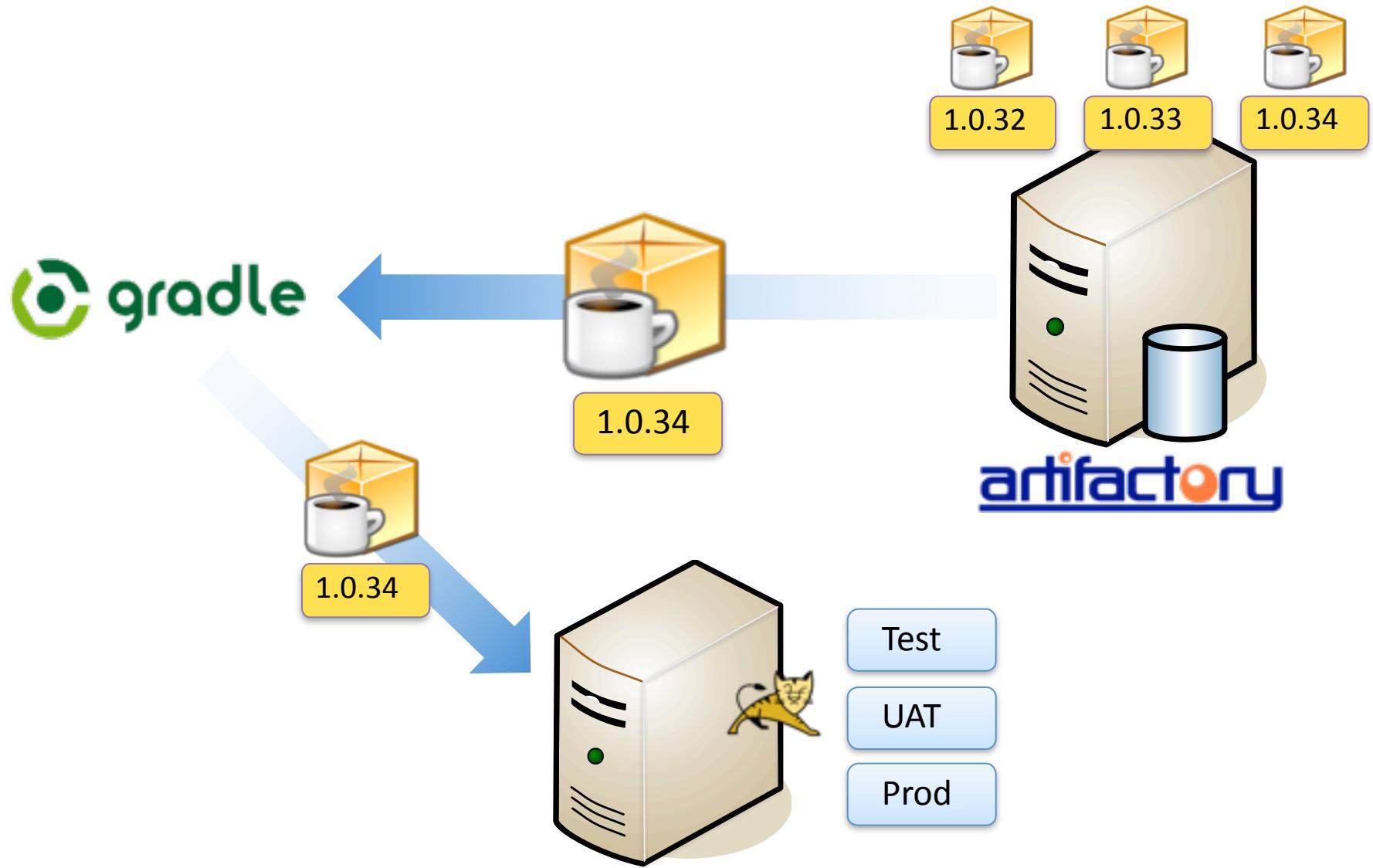


Request versioned artifact

Store in temp. directory



Downloading the deployable artifact



Task for downloading artifact

```
repositories {  
    maven { url fullRepoUrl }  
}  
  
configurations {  
    war  
}  
  
dependencies {  
    war "$project.group:$project.name:$project.version"  
}  
  
task downloadBinaryArchive(type: Copy) {  
    from configurations.war  
    into "$buildDir/download"  
}
```

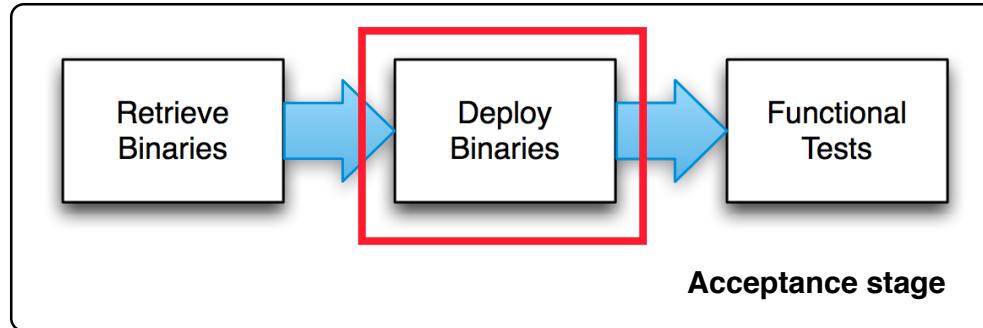
Task for downloading artifact

Target location for
downloaded artifact

```
repositories {  
    maven { url fullRepoUrl }  
}  
  
configurations {  
    war  
}  
  
dependencies {  
    war "$project.group:$project.name:$project.version"  
}  
  
task downloadBinaryArchive(type: Copy) {  
    from configurations.war  
    into "$buildDir/download"  
}
```

gradlew downloadBinaryArchive

Acceptance stage: Deploy binaries



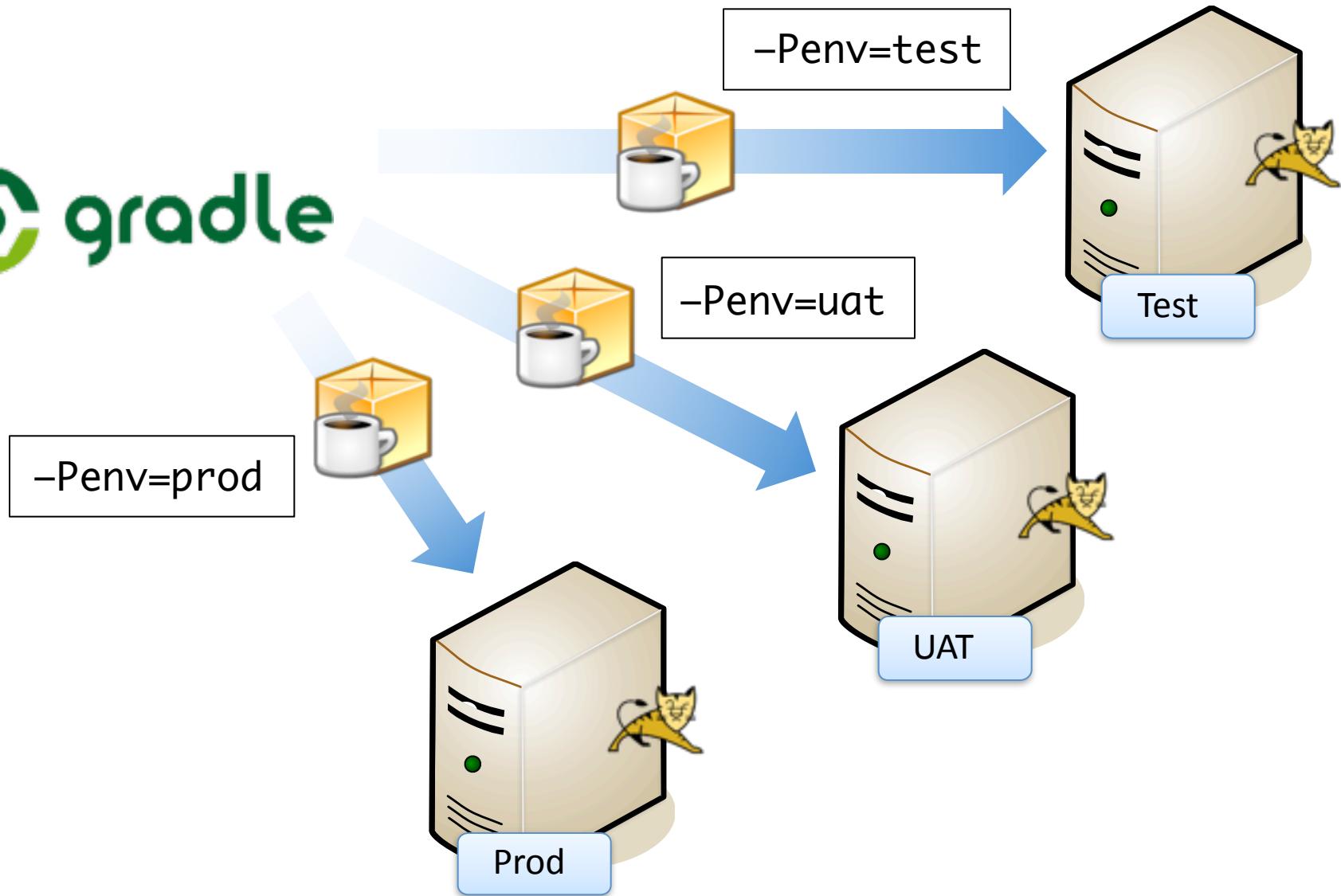
Deployment on request

Make it a reliable process

Use process for all envs



Deploying to multiple environments



Deployment with the Cargo plugin

```
cargoDeployRemote.dependsOn downloadBinaryArchive, cargoUndeployRemote

cargoUndeployRemote {
    onlyIf applicationContextStatus
}

cargo {
    containerId = 'tomcat7x'
    port = config.server.port

    deployable {
        file = downloadedArtifact
        context = config.server.context
    }

    remote {
        hostname = config.server.hostname
        username = config.server.username
        password = config.server.password
    }
}
```

Deployment with the Cargo plugin

Download artifact from binary repository and undeploy existing one

```
cargoDeployRemote.dependsOn downloadBinaryArchive, cargoUndeployRemote

cargoUndeployRemote {
    onlyIf applicationContextStatus
}

cargo {
    containerId = 'tomcat7x'
    port = config.server.port

    deployable {
        file = downloadedArtifact
        context = config.server.context
    }

    remote {
        hostname = config.server.hostname
        username = config.server.username
        password = config.server.password
    }
}
```

Deployment with the Cargo plugin

Download artifact from binary repository and undeploy existing one

```
cargoDeployRemote.dependsOn downloadBinaryArchive, cargoUndeployRemote

cargoUndeployRemote {
    onlyIf applicationContextStatus
}

cargo {
    containerId = 'tomcat7x'
    port = config.server.port

    deployable {
        file = downloadedArtifact
        context = config.server.context
    }

    remote {
        hostname = config.server.hostname
        username = config.server.username
        password = config.server.password
    }
}
```

Only undeploy if URL context exists

Deployment with the Cargo plugin

Download artifact from binary repository and undeploy existing one

```
cargoDeployRemote.dependsOn downloadBinaryArchive, cargoUndeployRemote
```

```
cargoUndeployRemote {  
    onlyIf applicationContextStatus  
}
```

Only undeploy if URL context exists

```
cargo {  
    containerId = 'tomcat7x'  
    port = config.server.port
```

```
    deployable {  
        file = downloadedArtifact  
        context = config.server.context  
    }
```

Use environment-specific configuration

```
    remote {  
        hostname = config.server.hostname  
        username = config.server.username  
        password = config.server.password  
    }  
}
```

Deployment with the Cargo plugin

Download artifact from binary repository and undeploy existing one

```
cargoDeployRemote.dependsOn downloadBinaryArchive, cargoUndeployRemote
```

```
cargoUndeployRemote {  
    onlyIf applicationContextStatus  
}
```

Only undeploy if URL context exists

```
cargo {  
    containerId = 'tomcat7x'  
    port = config.server.port
```

```
    deployable {  
        file = downloadedArtifact  
        context = config.server.context  
    }
```

Use environment-specific configuration

```
    remote {  
        hostname = config.server.hostname  
        username = config.server.username  
        password = config.server.password  
    }
```

```
}
```

gradlew -Penv=uat cargoDeploy

Safari File Edit View History Bookmarks Develop Window Help

Thu 3:28 PM Benjamin Muschko

localhost:8099 Apache Tomcat/7.0.40 Reader

Home Documentation Configuration Examples Wiki Mailing Lists Find Help

Apache Tomcat/7.0.40

The Apache Software Foundation <http://www.apache.org/>

If you're seeing this, you've successfully installed Tomcat. Congratulations!

 Recommended Reading:

- [Security Considerations HOW-TO](#)
- [Manager Application HOW-TO](#)
- [Clustering/Session Replication HOW-TO](#)

[Server Status](#)

[Manager App](#)

[Host Manager](#)

Developer Quick Start

[Tomcat Setup](#) [Realms & AAA](#) [Examples](#) [Servlet Specifications](#)

[First Web Application](#) [JDBC DataSources](#) [Tomcat Versions](#)

Managing Tomcat

For security, access to the [manager webapp](#) is restricted. Users are defined in: `CATALINA_HOME/conf/tomcat-users.xml`. In Tomcat 7.0 access to the manager application is split between different users. [Read more...](#)

[Release Notes](#)

[Changelog](#)

[Migration Guide](#)

[Security Notices](#)

Documentation

[Tomcat 7.0 Documentation](#)

[Tomcat 7.0 Configuration](#)

[Tomcat Wiki](#)

Find additional important configuration information in: `CATALINA_HOME/RUNNING.txt`. Developers may be interested in:

[Tomcat 7.0 Bug Database](#)

[Tomcat 7.0 JavaDocs](#)

[Tomcat 7.0 SVN Repository](#)

Getting Help

[FAQ and Mailing Lists](#)

The following mailing lists are available:

announce@tomcat.apache.org
Important announcements, releases, security vulnerability notifications. (Low volume).

users@tomcat.apache.org
User support and discussion.

taglibs-user@tomcat.apache.org
User support and discussion for Apache Taglibs.

dev@tomcat.apache.org
Development mailing list, including commit messages.

Other Downloads [Tomcat Connectors](#)

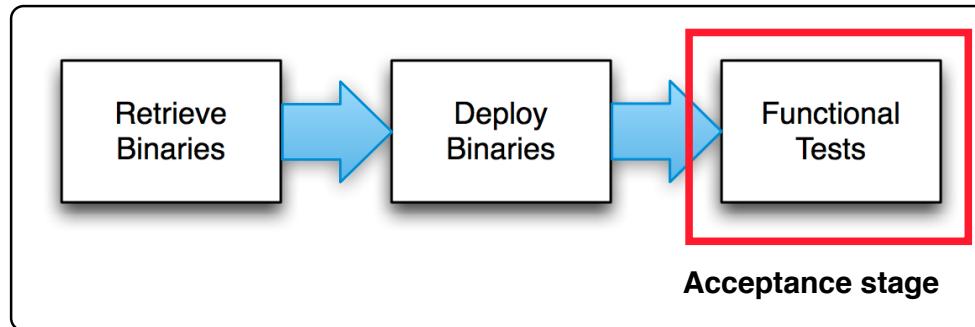
Other Documentation [Tomcat Connectors](#)

Get Involved [Overview](#)

Miscellaneous [Contact](#)

Apache Software Foundation [Who We Are](#)

Acceptance stage: Functional tests



Test UI permutations

Test important use cases

Run against different envs



Executing functional tests

```
def functionalTestReportDir = file("$testReportDir/functional")
def functionalTestResultsDir = file("$testResultsDir/functional")
def functionalCommonSystemProperties =
    ['geb.env': 'firefox',
     'geb.build.reportsDir': reporting.file("$name/geb")]

task functionalTest(type: Test) {
    testClassesDir = sourceSets.functionalTest.output.classesDir
    classpath = sourceSets.functionalTest.runtimeClasspath
    testReportDir = functionalTestReportDir
    testResultsDir = functionalTestResultsDir
    systemProperties functionalCommonSystemProperties
    systemProperty 'geb.build.baseUrl',
        "http://$config.server.hostname:$config.server.port/$config.server.context/"
}
```

Executing functional tests

```
def functionalTestReportDir = file("$testReportDir/functional")
def functionalTestResultsDir = file("$testResultsDir/functional")
def functionalCommonSystemProperties =
    ['geb.env': 'firefox',
     'geb.build.reportsDir': reporting.file("$name/geb")]
task functionalTest(type: Test) {
    testClassesDir = sourceSets.functionalTest.output.classesDir
    classpath = sourceSets.functionalTest.runtimeClasspath
    testReportDir = functionalTestReportDir
    testResultsDir = functionalTestResultsDir
    systemProperties functionalCommonSystemProperties
    systemProperty 'geb.build.baseUrl',
    "http://$config.server.hostname:$config.server.port/$config.server.context/"
}
```

Reuse setup properties

Executing functional tests

```
def functionalTestReportDir = file("$testReportDir/functional")
def functionalTestResultsDir = file("$testResultsDir/functional")
def functionalCommonSystemProperties =
    ['geb.env': 'firefox',
     'geb.build.reportsDir': reporting.file("$name/geb")]
task functionalTest(type: Test) {
    testClassesDir = sourceSets.functionalTest.output.classesDir
    classpath = sourceSets.functionalTest.runtimeClasspath
    testReportDir = functionalTestReportDir
    testResultsDir = functionalTestResultsDir
    systemProperties functionalCommonSystemProperties
    systemProperty 'geb.build.baseUrl',
    "http://$config.server.hostname:$config.server.port/$config.server.context/"
}
```

Reuse setup properties

Build URL from env. configuration

Executing functional tests

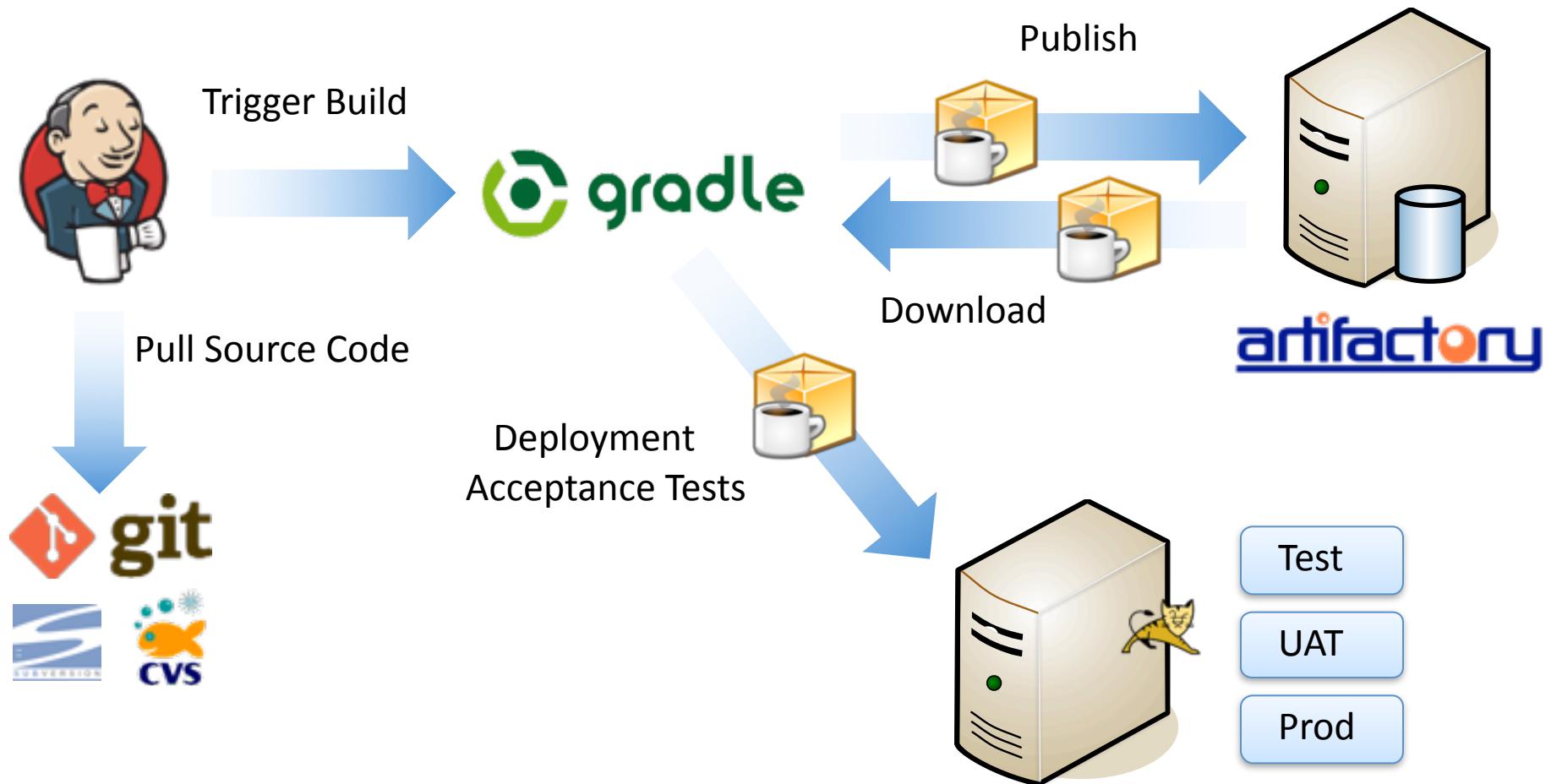
```
def functionalTestReportDir = file("$testReportDir/functional")
def functionalTestResultsDir = file("$testResultsDir/functional")
def functionalCommonSystemProperties =
    ['geb.env': 'firefox',
     'geb.build.reportsDir': reporting.file("$name/geb")]
task functionalTest(type: Test) {
    testClassesDir = sourceSets.functionalTest.output.classesDir
    classpath = sourceSets.functionalTest.runtimeClasspath
    testReportDir = functionalTestReportDir
    testResultsDir = functionalTestResultsDir
    systemProperties functionalCommonSystemProperties
    systemProperty 'geb.build.baseUrl',
        "http://$config.server.hostname:$config.server.port/$config.server.context/"
}
```

Reuse setup properties

Build URL from env. configuration

gradlew -Penv=test remoteFunctionalTest

Let's bring Jenkins into play!



Model pipeline as series of jobs

		todo-acceptance-tests	15 days (todo#76)	N/A	1 min 56 sec	
		todo-code-quality	15 days (todo#76)	N/A	1 min 6 sec	
		todo-deploy-production	N/A	N/A	N/A	
		todo-deploy-test	15 days (todo#76)	N/A	51 sec	
		todo-deploy-uat	20 days (todo#65)	16 days (todo#67)	47 sec	
		todo-distribution	15 days (todo#76)	29 days (todo#41)	45 sec	
		todo-initial	15 days (todo#76)	N/A	56 sec	
		todo-integ-tests	15 days (todo#76)	N/A	46 sec	
		todo-performance-tests	15 days (todo#76)	15 days (todo#74)	1 min 28 sec	



Jenkins

Model pipeline as series of jobs

		todo-acceptance-tests	15 days (todo#76)	N/A	1 min 56 sec	
		todo-code-quality	15 days (todo#76)	N/A	1 min 6 sec	
		todo-deploy-production	N/A	N/A	N/A	
		todo-deploy-test	15 days (todo#76)	N/A	51 sec	
		todo-deploy-uat	20 days (todo#65)	16 days (todo#67)	47 sec	
		todo-distribution	15 days (todo#76)	29 days (todo#41)	45 sec	
		todo-initial	15 days (todo#76)	N/A	56 sec	
		todo-integ-tests	15 days (todo#76)	N/A	46 sec	
		todo-performance-tests	15 days (todo#76)	15 days (todo#74)	1 min 28 sec	

Triggered job when change
to SCM is detected



Jenkins

Initial Jenkins Build Job

Jenkins / todo-initial

ENABLE AUTO REFRESH

[Back to Dashboard](#)

[Status](#)

[Changes](#)

[Workspace](#)

[Build Now](#)

[Delete Project](#)

[Configure](#)

[Coverage Trend](#)

[Git Polling Log](#)

Project todo-initial

[Workspace](#)

[Recent Changes](#)

[Latest Test Result \(no failures\)](#)

Downstream Projects

[todo-integ-tests](#)

Permalinks

- [Last build \(todo#78\), 1 hr 24 min ago](#)
- [Last stable build \(todo#78\), 1 hr 24 min ago](#)
- [Last successful build \(todo#78\), 1 hr 24 min ago](#)

[RSS for all](#) [RSS for failures](#)

Test Result Trend

This chart shows the count of test results over five builds. The count remains constant at 7 across builds 74, 75, 76, 77, and 78.

Build	Count
todo#74	7
todo#75	7
todo#76	7
todo#77	7
todo#78	7

[\(just show failures\) enlarge](#)

Code Coverage Trend

This chart tracks code coverage over five builds. The 'lineCovered' metric (green line) stays at approximately 115%, while the 'lineMissed' metric (red line) stays at 250%.

Build	lineCovered (%)	lineMissed (%)
todo#74	~115	250
todo#75	~115	250
todo#76	~115	250
todo#77	~115	250
todo#78	~115	250

Legend: [lineCovered](#) [lineMissed](#)



Build Name Setter Plugin

Jenkins Jenkins / todo-initial search ENABLE AUTO REFRESH

[Back to Dashboard](#) [Status](#) [Changes](#) [Workspace](#) [Build Now](#) [Delete Project](#) [Configure](#) [Coverage Trend](#) [Git Polling Log](#)

Project todo-initial

[Workspace](#) [Recent Changes](#) [Latest Test Result \(no failures\)](#)

Test Result Trend

Build	Count
todo#74	0
todo#75	1
todo#76	0
todo#77	0
todo#78	0

[add description](#) [Disable Project](#)

Code Coverage Trend

Build	lineCovered	lineMissed
todo#74	120	250
todo#75	120	250
todo#76	120	250
todo#77	120	250
todo#78	120	250

[\[just show failures\]](#) [enlarge](#)

Downstream Projects

- [todo-integ-tests](#)

Permalinks

- Last build (todo#78), 1 hr 24 min ago
- Last stable build (todo#78), 1 hr 24 min ago
- Last successful build (todo#78), 1 hr 24 min ago

[RSS for all](#) [RSS for failures](#)



JaCoCo Plugin

Jenkins / todo-initial

ENABLE AUTO REFRESH

[Back to Dashboard](#)

[Status](#)

[Changes](#)

[Workspace](#)

[Build Now](#)

[Delete Project](#)

[Configure](#)

[Coverage Trend](#)

[Git Polling Log](#)

Project todo-initial

[Workspace](#)

[Recent Changes](#)

[Latest Test Result \(no failures\)](#)

Test Result Trend

Count

todo#74 todo#75 todo#76 todo#77 todo#78

Code Coverage Trend

Just show failures) enlarge

lineCovered
lineMissed

Downstream Projects

[todo-integ-tests](#)

Permalinks

- Last build (todo#78), 1 hr 24 min ago
- Last stable build (todo#78), 1 hr 24 min ago
- Last successful build (todo#78), 1 hr 24 min ago

[RSS for all](#) [RSS for failures](#)



Parameterized Trigger Plugin

Jenkins / todo-initial

ENABLE AUTO REFRESH

[Back to Dashboard](#)

[Status](#)

[Changes](#)

[Workspace](#)

[Build Now](#)

[Delete Project](#)

[Configure](#)

[Coverage Trend](#)

[Git Polling Log](#)

Project todo-initial

[Workspace](#)

[Recent Changes](#)

[Latest Test Result \(no failures\)](#)

Downstream Projects

[todo-integ-tests](#)

Permalinks

- [Last build \(todo#78\), 1 hr 24 min ago](#)
- [Last stable build \(todo#78\), 1 hr 24 min ago](#)
- [Last successful build \(todo#78\), 1 hr 24 min ago](#)

[RSS for all](#) [RSS for failures](#)

Test Result Trend

Build	Count
todo#74	7
todo#75	7
todo#76	7
todo#77	7
todo#78	7

[\(just show failures\) enlarge](#)

Code Coverage Trend

Build	lineCovered	lineMissed
todo#74	~120	~250
todo#75	~120	~250
todo#76	~120	~250
todo#77	~120	~250
todo#78	~120	~250

[lineCovered](#) [lineMissed](#)



Gradle Plugin

Build

Invoke Gradle script



Invoke Gradle

Use Gradle Wrapper

Make gradlew executable

From Root Build Script Dir

Build step description



Switches



Tasks



Root Build script



Build File



Specify Gradle build file to run. Also, [some environment variables are available to the build script](#)



Gradle Plugin

Build

Invoke Gradle script

Invoke Gradle

Use Gradle Wrapper

Make gradlew executable

From Root Build Script Dir

Build step description



Switches



Tasks

 clean test

Root Build script



Build File



Always use the Wrapper!

Specify Gradle build file to run. Also, [some environment variables are available to the build script](#)



Gradle Plugin

Build

Always use the Wrapper!

Invoke Gradle script

Invoke Gradle

Use Gradle Wrapper

Make gradlew executable

From Root Build Script Dir

Build step description

Switches

Tasks

clean test

Root Build script

Build File

Run clean task to remove existing artifacts

variables are available to the build



Build Name Setter Plugin

Build History [\(trend\)](#)

- todo#78 [May 16, 2013 6:59:27 AM](#)
- todo#77 [May 16, 2013 6:56:37 AM](#)
- todo#76 [Apr 28, 2013 3:09:08 PM](#)
- todo#75 [Apr 28, 2013 11:16:37 AM](#)
- todo#74 [Apr 28, 2013 11:02:41 AM](#)

[RSS for all](#) [RSS for failures](#)

Build Environment

Set Build Name

Build Name

todo#\${BUILD_NUMBER}



Build Name Setter Plugin

Clearly identify a build through an expressive build name

Build History [\(trend\)](#)

- todo#78 [May 16, 2013 6:59:27 AM](#)
- todo#77 [May 16, 2013 6:56:37 AM](#)
- todo#76 [Apr 28, 2013 3:09:08 PM](#)
- todo#75 [Apr 28, 2013 11:16:37 AM](#)
- todo#74 [Apr 28, 2013 11:02:41 AM](#)

[RSS for all](#) [RSS for failures](#)

Build Environment

Set Build Name

Build Name

todo#\${BUILD_NUMBER}



Build Name Setter Plugin

Clearly identify a build through an expressive build name

Build History [\(trend\)](#)

- todo#78 [May 16, 2013 6:59:27 AM](#)
- todo#77 [May 16, 2013 6:56:37 AM](#)
- todo#76 [Apr 28, 2013 3:09:08 PM](#)
- todo#75 [Apr 28, 2013 11:16:37 AM](#)
- todo#74 [Apr 28, 2013 11:02:41 AM](#)

[RSS for all](#) [RSS for failures](#)

Build Environment

Set Build Name

Build Name

todo#\${BUILD_NUMBER}

Use Jenkins environment variable



Parameterized Trigger Plugin

Trigger parameterized build on other projects

Build Triggers

Projects to build	<input type="text" value="todo-integ-tests"/>	
Trigger when build is	<input type="text" value="Stable"/>	
Trigger build without parameters	<input type="checkbox"/>	

Predefined parameters

Parameters



Parameterized Trigger Plugin

Trigger parameterized build on other projects

Build Triggers

Projects to build: todo-integ-tests

Trigger when build is: Stable

Trigger build without parameters:

Predefined parameters

Parameters: SOURCE_BUILD_NUMBER=\${BUILD_NUMBER}

Next job to trigger if build is stable



Parameterized Trigger Plugin

Trigger parameterized build on other projects

Build Triggers

Projects to build: todo-integ-tests

Trigger when build is: Stable

Trigger build without parameters:

Predefined parameters

Parameters: SOURCE_BUILD_NUMBER=\${BUILD_NUMBER}

Next job to trigger if build is stable

Defines build number parameter provided to subsequent jobs



Clone Workspace SCM Plugin

Archive for Clone Workspace SCM

Files to include in cloned workspace

**/*



Files to exclude from cloned workspace



Criteria for build to be archived

Most Recent Completed Build



Archive method

Gzipped tar



Override Default Ant Excludes





Clone Workspace SCM Plugin

Archive all files

Archive for Clone Workspace SCM

Files to include in cloned workspace

**/*



Files to exclude from cloned workspace



Criteria for build to be archived

Most Recent Completed Build



Archive method

Gzipped tar



Override Default Ant Excludes





Clone Workspace SCM Plugin

Archive for Clone Workspace SCM

Files to include in cloned workspace

Files to exclude from cloned workspace

Criteria for build to be archived

Archive method

Override Default Ant Excludes

Archive all files

Only archive if build was successful



JaCoCo Plugin

Publish JUnit test result report

Test report XMLs (?)

Fileset "includes" setting that specifies the generated raw XML report files, such as 'myproject/target/test-reports/*.xml'. Basedir of the fileset is [the workspace root](#).

Retain long standard output/error (?)

Delete

Record JaCoCo coverage report

Path to exec files (e.g.:
`**/target/**.exec, **/jacoco.exec`) Path to class directories (e.g.:
`**/target/classDir, **/classes`) Path to source directories (e.g.:
`**/mySourceFiles`)

Inclusions (e.g.: `**/*.class`)
Exclusions (e.g.: `**/*Test*`)

	Instruction	% Branch	% Complexity	% Line	% Method	% Class
	100	70	70	70	70	70
	0	0	0	0	0	0



JaCoCo Plugin

Point to separated test results

Publish JUnit test result report

Test report XMLs `**/build/test-results/unit/*.xml`

Fileset "includes" setting that specifies the generated raw XML report files, such as `'myproject/target/test-reports/*.xml'`. Basedir of the fileset is [the workspace root](#).

Retain long standard output/error

Delete

Record JaCoCo coverage report

Path to exec files (e.g.:
`**/target/**.exec, **/jacoco.exec`)

Path to class directories (e.g.:
`**/target/classDir, **/classes`)

Path to source directories (e.g.:
`**/mySourceFiles`)

`**/build/jacoco/test.exec` `**/build/classes` `**/src/main/java`

Inclusions (e.g.: `**/*.class`)

Exclusions (e.g.: `**/*Test*`)

	Instruction	% Branch	% Complexity	% Line	% Method	% Class
	100	70	70	70	70	70
	0	0	0	0	0	0



JaCoCo Plugin

Point to separated test results

Publish JUnit test result report

Test report XMLs `**/build/test-results/unit/*.xml`

Fileset 'includes' setting that specifies the generated raw XML report files, such as 'myproject/target'

Retain long standard output/error

Point to JaCoCo files as well as source and class files

Record JaCoCo coverage report

Path to exec files (e.g.:
`**/target/**.exec, **/jacoco.exec`) `**/build/jacoco/test.exec`

Path to class directories (e.g.:
`**/target/classDir, **/classes`) `**/build/classes`

Path to source directories (e.g.:
`**/mySourceFiles`) `**/src/main/java`

Inclusions (e.g.: `**/*.class`)

Exclusions (e.g.: `**/*Test*`)

Instruction	% Branch	% Complexity	% Line	% Method	% Class
	100	70	70	70	70
	0	0	0	0	0



JaCoCo Plugin

Point to separated test results

Publish JUnit test result report

Test report XMLs `**/build/test-results/unit/*.xml`

Fileset 'includes' setting that specifies the generated raw XML report files, such as 'myproject/target'

Retain long standard output/error

Record JaCoCo coverage report

Path to exec files (e.g.:
`**/target/**.exec, **/jacoco.exec`)

Path to class directories (e.g.:
`**/target/classDir, **/classes`)

Path to source directories (e.g.:
`**/mySourceFiles`)

`**/build/jacoco/test.exec` `**/build/classes` `**/src/main/java`

Inclusions (e.g.: `**/*.class`)

Exclusions (e.g.: `**/*Test*`)

Instruction	% Branch	% Complexity	% Line	% Method	% Class
	100	70	70	70	70
	0	0	0	0	0

Point to JaCoCo files as well
as source and class files

Fail build of quality gate
criteria are not met



Clone Workspace SCM Plugin



Build Name Setter Plugin

Source Code Management

CVS

Clone Workspace

Parent Project

todo-initial



Criteria for parent build

Most Recent Completed Build



Git

None

Subversion

Build Environment

Set Build Name



Build Name

todo#\${ENV,var="SOURCE_BUILD_NUMBER"}





Clone Workspace SCM Plugin



Build Name Setter Plugin

Source Code Management

CVS
 Clone Workspace
Parent Project

Criteria for parent build

Git
 None
 Subversion

Build Environment

Set Build Name
Build Name

Reuse initial workspace



Clone Workspace SCM Plugin



Build Name Setter Plugin

Source Code Management

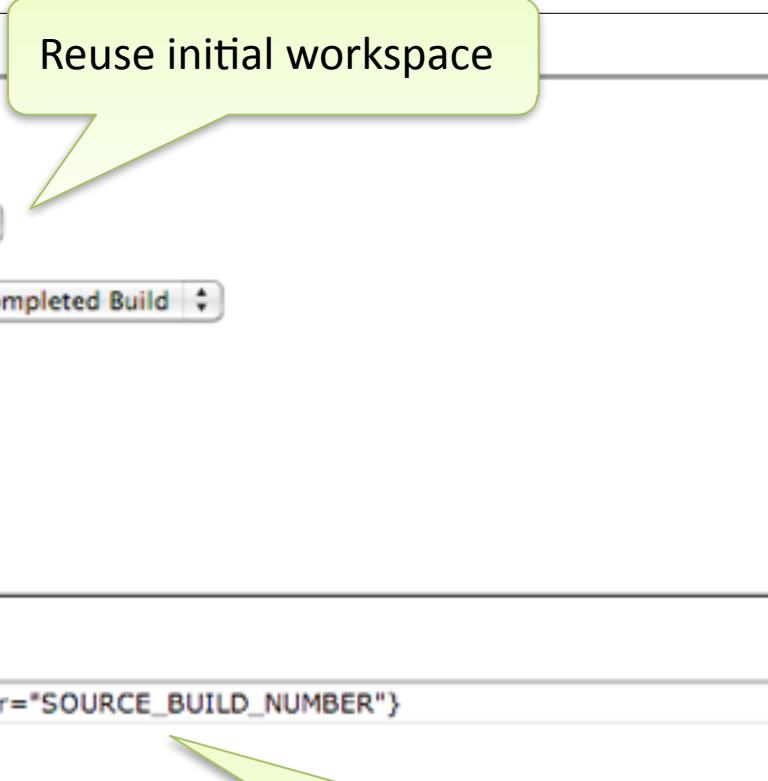
CVS
 Clone Workspace
Parent Project

Criteria for parent build

Git
 None
 Subversion

Build Environment

Set Build Name
Build Name



Reuse initial build number



Build Pipeline Plugin

Invoke Gradle script

Invoke Gradle

Use Gradle Wrapper

Make gradlew executable

From Root Build Script Dir

Build step description

▼

Switches

 -Penv=uat▼ ?

Tasks

 cargoDeployRemote▼ ?

Root Build script

?

Build File

?

Post-build Actions

Build Pipeline Plugin -> Manually Execute Downstream Project



Downstream Project Names todo-deploy-production



Build Pipeline Plugin

Invoke Gradle script

Invoke Gradle

Use Gradle Wrapper

Make gradlew executable

From Root Build Script Dir

Build step description

Switches

-Penv=uat

Tasks

cargoDeployRemote

Root Build script

Build File

Define the target environment

Post-build Actions

Build Pipeline Plugin -> Manually Execute Downstream Project

Downstream Project Names todo-deploy-production



Build Pipeline Plugin

Invoke Gradle script

Invoke Gradle

Use Gradle Wrapper

Make gradlew executable

From Root Build Script Dir

Build step description

Switches

-Penv=uat

Tasks

cargoDeployRemote

Root Build script

Build File

Define the target environment

Post-build Actions

Build Pipeline Plugin -> Manually Execute Downstream Project

Downstream Project Names todo-deploy-production

Downstream job that requires manual execution



Build Pipeline Plugin

Build Pipeline: To Do application



Visualization of chained pipeline jobs

Safari File Edit View History Bookmarks Develop Window Help

Thu 4:58 PM Benjamin Muschko Q

localhost:8080

Dashboard [Jenkins]

Reader +

Jenkins

search

ENABLE AUTO REFRESH

+ add description

New Job

People

Build History

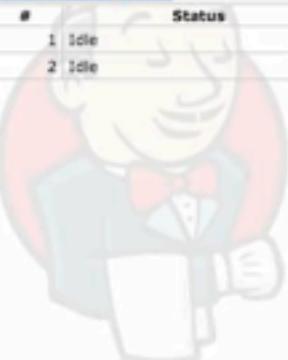
Manage Jenkins

Build Queue

No builds in the queue.

Build Executor Status

#	Status
1	Idle
2	Idle



All todo-pipeline +

S	W	Name	Last Success	Last Failure	Last Duration	Line Coverage	
●	●	todo-acceptance-tests	2 min 35 sec - todo#11	N/A	42 sec		N/A%
●	●	todo-code-quality	4 min 22 sec - todo#11	N/A	33 sec		N/A%
●	●	todo-deploy-production	N/A	N/A	N/A		N/A%
●	●	todo-deploy-test	3 min 15 sec - todo#11	N/A	35 sec		N/A%
●	●	todo-deploy-uat	N/A	N/A	N/A		N/A%
●	●	todo-distribution	3 min 43 sec - todo#11	N/A	22 sec		N/A%
●	●	todo-initial	5 min 27 sec - todo#11	N/A	18 sec		31.12%
●	●	todo-integ-tests	5 min 3 sec - todo#11	N/A	36 sec		20.1%
●	●	todo-performance-tests	N/A	N/A	N/A		N/A%

Icon: S M L

Legend RSS for all RSS for failures RSS for just latest builds

Help us localize this page

Page generated: May 30, 2013 4:58:20 PM BEST API Jenkins ver. 1.516

```
> gradle qa  
:askQuestions
```

BUILD SUCCESSFUL

Total time: 300 secs