

The background image shows a massive construction project, likely a skyscraper, under a clear blue sky. Multiple yellow tower cranes are positioned around the structure, some with their booms extended. The building's steel framework is visible, showing complex trusses and beams.

# Gradle time for a new build!

Igor Khotin

E-mail: khotin@gmx.com

# Background

- 12+ years in the IT industry
- 7+ years with Java
- Flexible design promoter
- Agile-junkie



# What do we use?

- Ant?
- Maven?
- Buildr?
- Gradle?
- ...

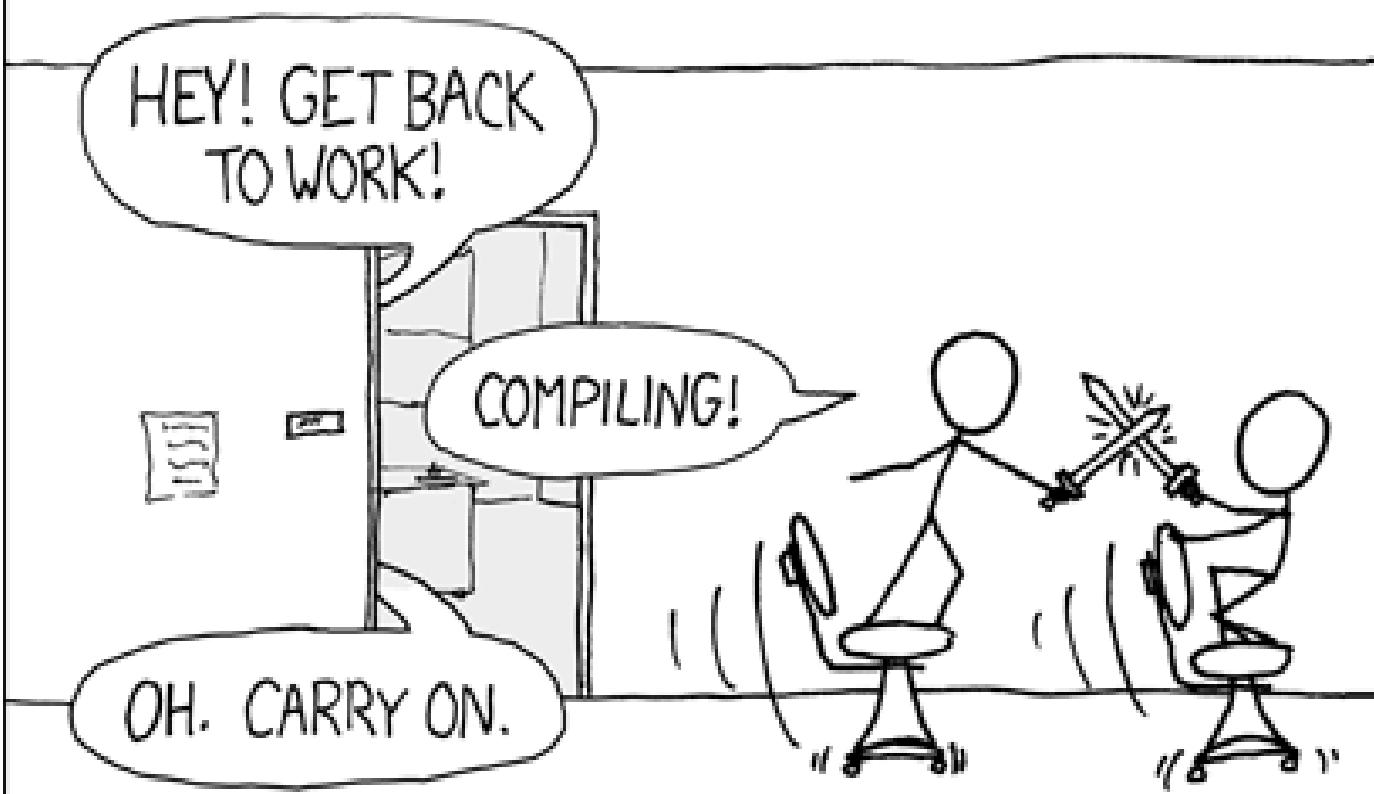


# Any problems?

- heavy builds?
- long integration cycles?
- build projects support?
- build projects
- integration?
- ...



# THE #1 PROGRAMMER EXCUSE FOR LEGITIMATELY SLACKING OFF: "MY CODE'S COMPILING."



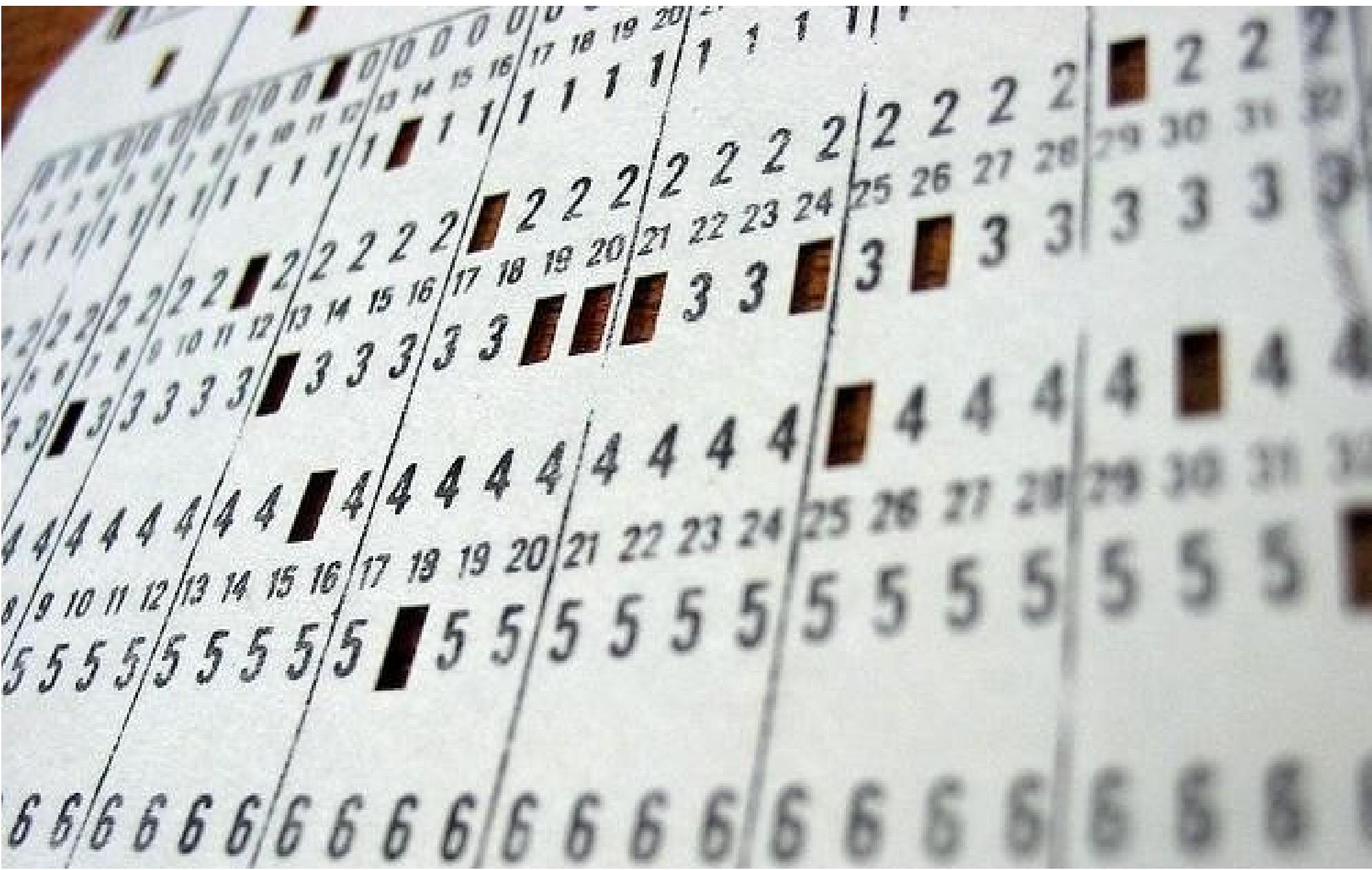
# How we used to build?



# punchcards...



# give them to an operator



# waiting for result...



# take the printouts...



# debug?



# 1977 – make

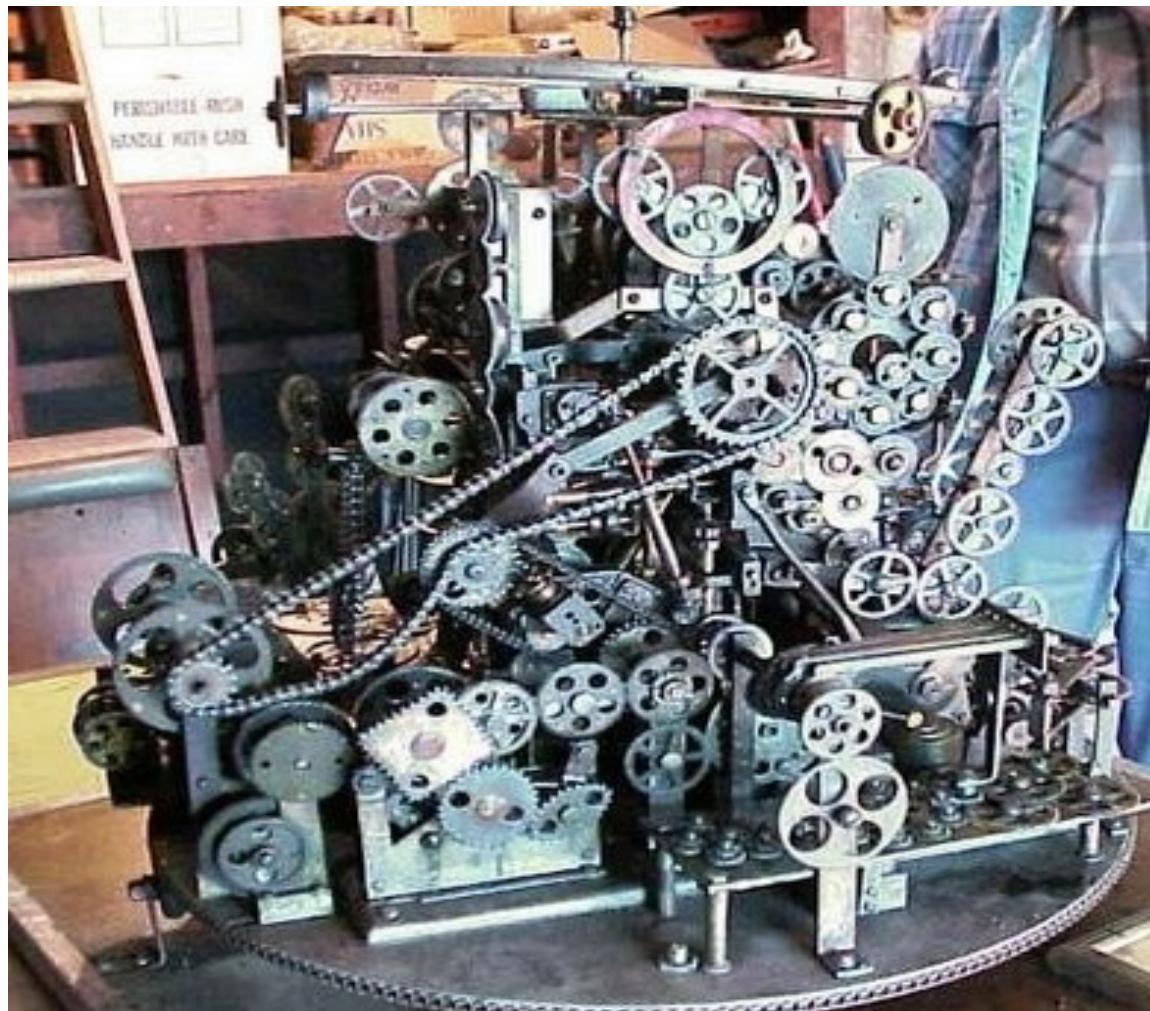


2000



ant

Complexity



ant

Complexity

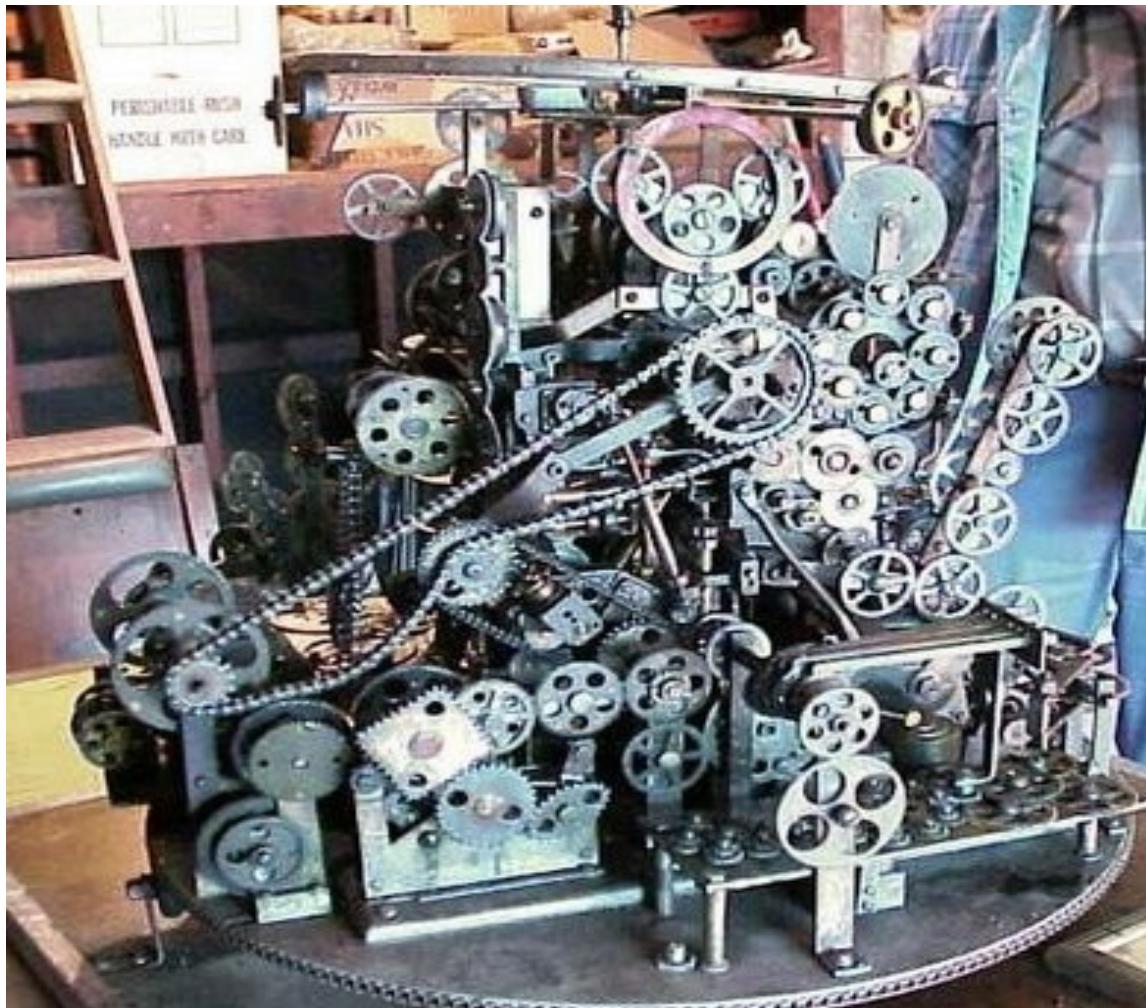
hard to reuse



ant

Complexity

hard to reuse



heavy build.xml's

ant

Complexity

hard to reuse



heavy build.xml's

Total Control

2001

maven



maven

convention over configuration

maven

maven central

convention over configuration

maven

maven central

convention over configuration

dependency management

maven

heavy xml's

maven central

convention over configuration

dependency management

maven

heavy xml's

hard to implement  
custom behavior

maven central

convention over configuration

dependency management

maven

heavy xml's

hard to implement  
custom behavior

maven central

convention over configuration

dependency management

lost control



# Gradle yesterday

- Founder - Hans Dockter
- 2008 – early releases
- 2010 Springy Innovation Award

# Gradle today

- 1.0-rc2 released on April 25, 2012
- Active community
- Gradleware
- Apache License, Version 2.0

*Understanding Next-Generation Builds*



# Building and Testing with Gradle

O'REILLY®

*Tim Berglund & Matthew McCullough*  
*Foreword by Gradle Founder Hans Dockter*

## UPCOMING GRADLE EVENTS

The following are upcoming conferences, trade shows, user group meetings or any kind of event that Gradle will be involved in.

Past events can be found [here](#).

Know of an event that's not listed here? Let us know via [info@gradleware.com](mailto:info@gradleware.com).

April, 2012



### [Northern Virginia Software Symposium](#)

**When:** 27th - 29th April, 2012

**Where:** Reston, VA, United States

[Matthew McCullough](#) is delivering his excellent and much recommended [Gradle workshop](#). In [another Gradle related talk](#) of Matthew, he talks about 'Build Lifecycle Craftsmanship Tools' (Git, Maven 3, Gradle, Jenkins, and Sonar).

May, 2012



### [CONFESS 2012](#)

**When:** 7th - 9th May, 2012

**Where:** Leogang, Austria

Gradle Core Developer [Peter Niederwieser](#) will provide a [Gradle Workshop](#) and a talk about [Gradle in the Enterprise](#). Furthermore he'll be speaking about [Smarter Testing With Spock](#).

# Gradle in the wild



ZEPPELIN®



Farm Credit Canada



# Who is that Gradle?

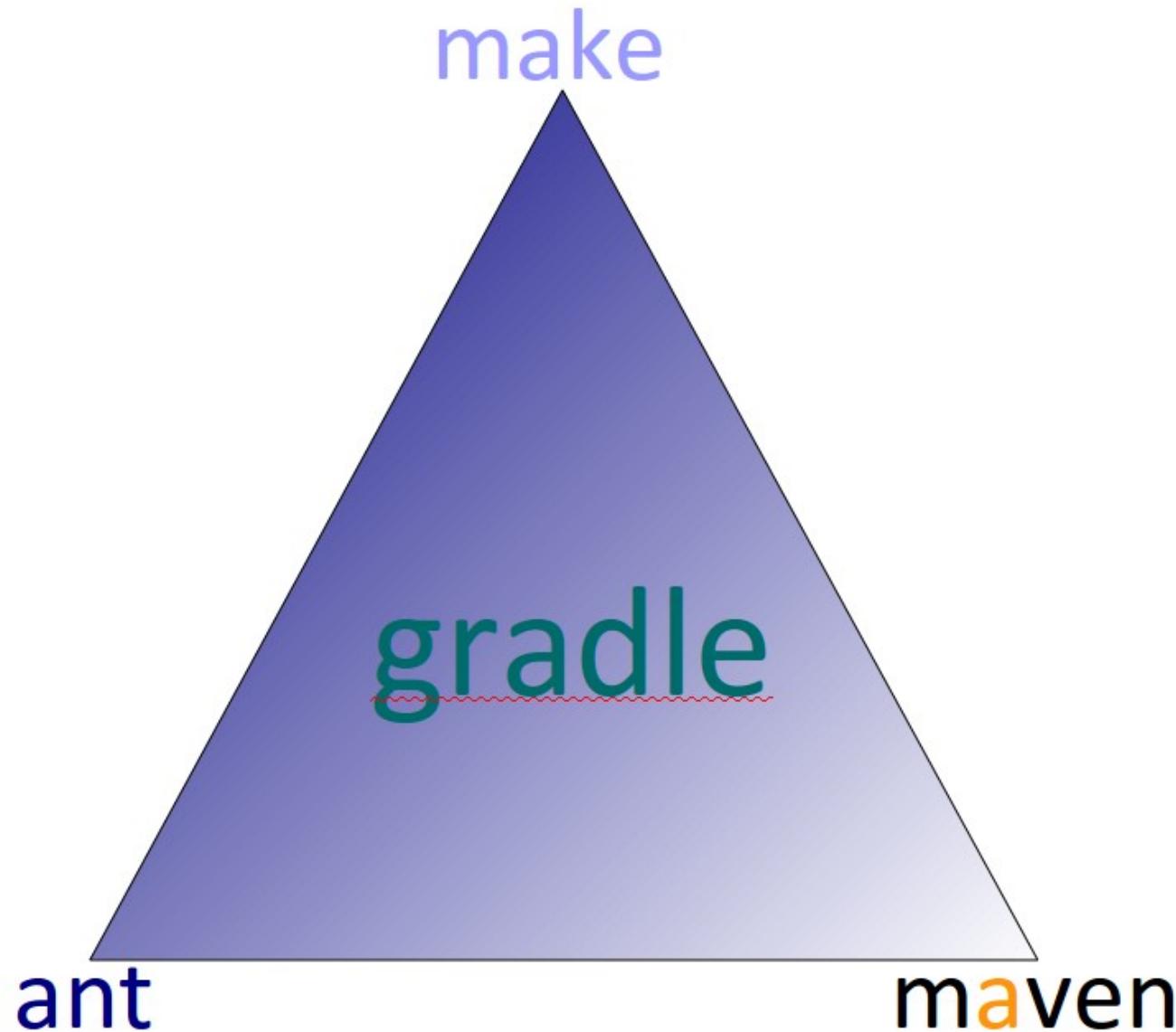
- Build integration tool
- Declarative builds
- Groovy-based build DSL
- Build-by-convention

# Who is that Gradle?

- Scalable – multi-project builds
- Dependency management
- Ease of migration
- Embeddable
- Deep API



# Gradle positioning



# Groovy and Gradle

## **build.gradle**

```
task count << {
    4.times { print "$it " }
}
```

```
> gradle -q count
0 1 2 3
```

gradle task != ant task

gradle task == ant target



# Build tree

Gradle

Task Tree Favorites Command Line Setup

Description

grails

- assemble *Assembles all Jar, War, Zip, and Tar archives.*
- build *Assembles and tests this project.*
- buildDependents *Assembles and tests this project and all projects that depend on it.*
- buildNeeded *Assembles and tests this project and all projects it depends on.*
- check *Runs all checks.*
- classes *Assembles the main classes.*
- clean *Deletes the build directory.*
- compileJava *Compiles the main Java source.*
- compileTestJava *Compiles the test Java source.*
- jar *Assembles a jar archive containing the main classes.*
- javadoc *Generates Javadoc API documentation for the main source code.*
- processResources *Processes the main resources.*
- processTestResources *Processes the test resources.*
- test *Runs the unit tests.*
- testClasses *Assembles the test classes.*

Refresh ×

Completed successfully at 2:50:24 PM

BUILD SUCCESSFUL

Total time: 3.324 secs

Completed Successfully

# Dependencies

```
task build << {
    println 'building...'
}
task count(dependsOn: build) << {
    4.times { print "$it " }
}
```

```
> gradle -q count
building...
0 1 2 3
```

# Lazy dependencies

```
task build(dependsOn: 'lazy') << {
    println 'building...'
}
task lazy << {
    println 'so lazy...'
}
```

```
> gradle -q hello
so lazy...
building...
```

# Rules

## **build.gradle**

```
tasks.addRule("Pattern: ping<ID>") { String taskName ->
    if (taskName.startsWith("ping")) {
        task(taskName) << {
            println "Pinging: " + (taskName - 'ping')
        }
    }
}
task groupPing {
    dependsOn pingServer1, pingServer2
}
```

```
> gradle -q groupPing
Pinging: Server1
Pinging: Server2
```

# Gradle Plugins

tasks – objects – conventions



# Gradle Plugins

java scala groovy cpp antlr  
checkstyle findbugs pmd sonar  
ear war osgi jetty maven



# Java Plugin

**build.gradle**

```
apply plugin: 'java'
```

```
> gradle build
```

```
...
```

# Java Plugin Folder Layout

src/main/java

src/main/resources

src/test/java

src/test/resources

Production Java source

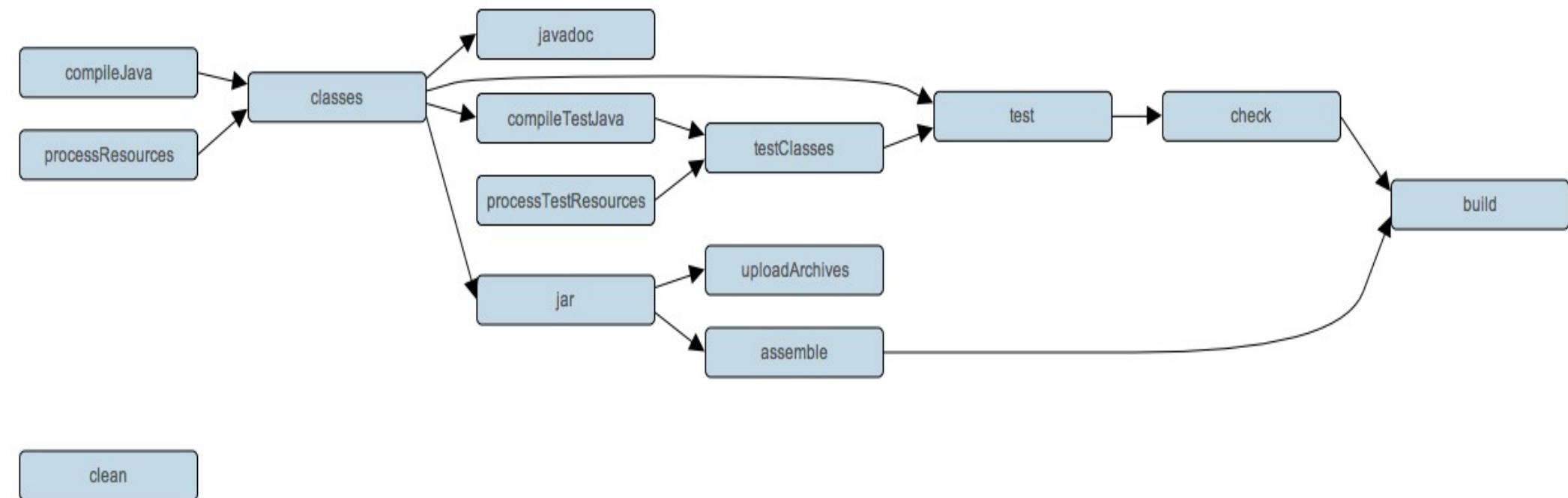
Production resources

Test Java source

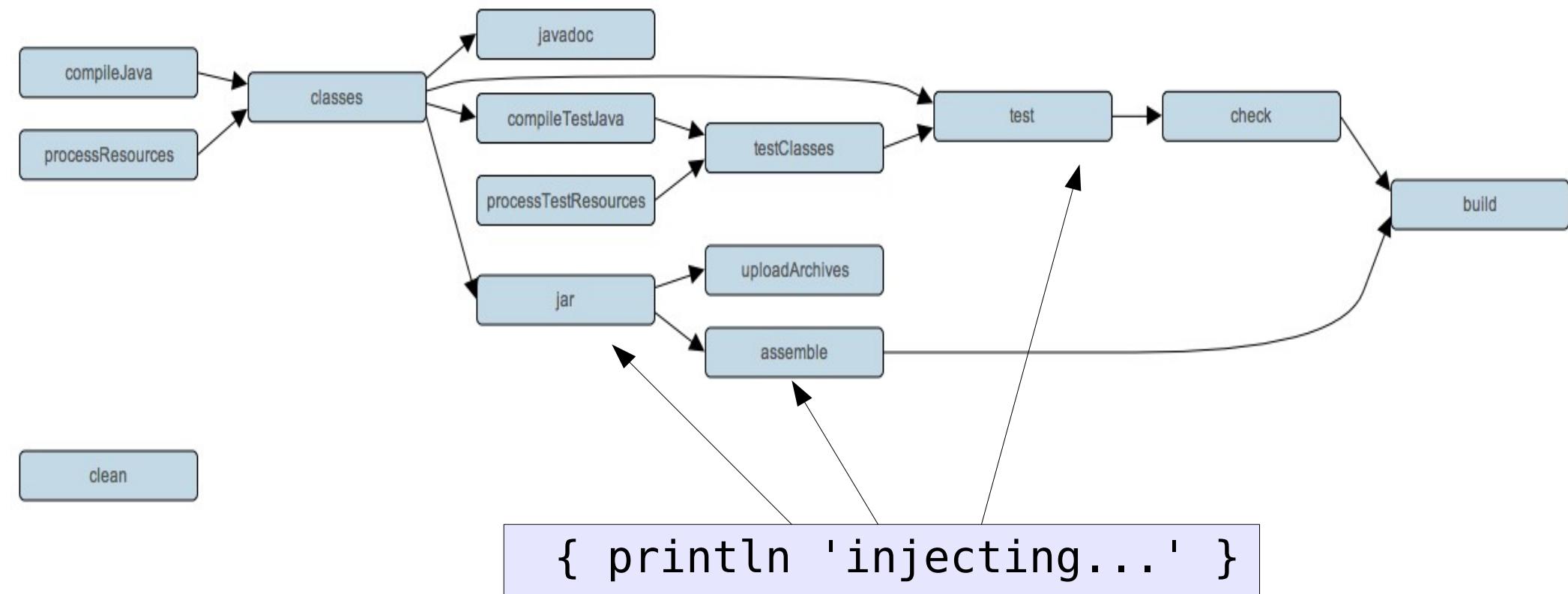
Test resources

just like maven...

# Java Plugin Tasks Flow



# Java Plugin Tasks Flow



Injectable with Groovy closures

# Java build-cycle customization

```
apply plugin: 'java'

test.doFirst {
    println 'Before testing...'
}
test.doLast {
    println '...after testing.'
}
```

# Dependency management

- You can define rules for dependencies
- Flexible repository handling
- Works with Ivy and Maven repositories
- Dynamic properties
- and more...



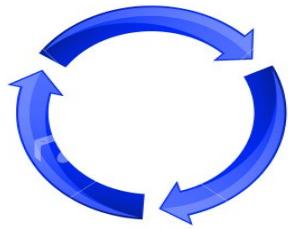
# Java & jars

```
apply plugin: 'java'  
repositories {  
    mavenCentral()  
}  
dependencies {  
    compile 'commons-lang:commons-lang:2.5'  
    testCompile 'junit:junit:4.8.1'  
}
```

# Ant

```
ant.importBuild 'build.xml'

task ant << {
    ant.echo(message: 'hello from Ant')
    ant.zip(destfile: 'archive.zip') {
        fileset(dir: 'src') {
            include(name: '**.xml')
            exclude(name: '**.java')
        }
    }
}
```



# Build Lifecycle



Initialization



Configuration



Execution

# Multi-project builds

- Arbitrary multi-project layout
- Configuration injection
- Separate config/execution tree
- Partial builds



# Multi-project builds

```
include 'project1', 'project2'
```

```
root-project/  
build.gradle  
settings.gradle  
project1/  
    build.gradle  
project2/  
    build.gradle
```

# Gradle daemon

- To reduce the startup time
- Used in STS Gradle plugin for Eclipse
- Used in IntelliJ IDEA plugin (IDEA>10)
- Used by default in Tooling API



# Apache Ant vs. Apache Maven vs. Gradle



# ant

```
<?xml version="1.0"?>
<project name="simple" default="dist" basedir=".">
  <property name="src" location="src/main/java"/>
  <property name="srcTest" location="src/test/java"/>
  <property name="build" location="build"/>
  <property name="dist" location="${build}/lib"/>
  <property name="version" value="1.0-SNAPSHOT" />
  <path id="classpath.compile">
    <pathelement location="libs/commons-lang-2.5.jar"/>
  </path>
  <path id="classpath.test">
    <pathelement location="libs/junit-4.8.2.jar"/>
    <pathelement location="libs/commons-lang-2.5.jar"/>
    <pathelement location="${srcTest}"/>
    <pathelement location="${build}/classes"/>
    <pathelement location="${build}/test-classes"/>
  </path>
...

```



# ant

```
...  
<target name="init">  
  <mkdir dir="${build}/classes"/>  
  <mkdir dir="${build}/test-classes"/>  
</target>  
<target name="compile" depends="init">  
  <javac srcdir="${src}" destdir="${build}/classes">  
    <classpath refid="classpath.compile"/>  
  </javac>  
</target>  
<target name="testCompile" depends="compile">  
  <javac srcdir="${srcTest}" destdir="${build}/test-classes">  
    <classpath refid="classpath.test"/>  
  </javac>  
</target>  
...
```



# ant

```
...  
<target name="test" depends="testCompile">  
  <junit fork="yes" haltonfailure="yes">  
    <batchtest fork="yes">  
      <fileset dir="${srcTest}">  
        <include name="**/*Test.java"/>  
      </fileset>  
    </batchtest>  
    <classpath refid="classpath.test"/>  
    <formatter type="plain"/>  
  </junit>  
</target>  
<target name="dist" depends="test">  
  <mkdir dir="${dist}" />  
  <jar jarfile="${dist}/coc-comparison-${version}.jar"  
      basedir="${build}/classes" />  
</target>  
<target name="clean"><delete dir="${build}" /></target>  
</project>
```



DON'T  
PANIC

# maven

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
  http://maven.apache.org/maven-v4_0_0.xsd">
<modelVersion>4.0.0</modelVersion>
<groupId>grId</groupId>
<artifactId>coc-comparison</artifactId>
<packaging>jar</packaging>
<version>1.0-SNAPSHOT</version>
```

...



# maven

```
...  
<dependencies>  
  <dependency>  
    <groupId>commons-lang</groupId>  
    <artifactId>commons-lang</artifactId>  
    <version>2.5</version>  
  </dependency>  
  <dependency>  
    <groupId>junit</groupId>  
    <artifactId>junit</artifactId>  
    <version>4.8.1</version>  
    <scope>test</scope>  
  </dependency>  
</dependencies>  
</project>
```



# polyglot maven

```
project {  
    modelVersion '4.0.0'  
    artifactId 'coc-comparison'  
    groupId 'grid'  
    version '1.0-SNAPSHOT'  
  
    dependencies {  
        dependency('commons-lang:commons-lang:2.5')  
        dependency('junit:junit:4.8.1')  
    }  
  
    properties {  
        'maven.compiler.target' '1.6'  
        'maven.compiler.source' '1.6'  
    }  
}
```

# gradle

```
apply plugin: 'java'

version="1.0-SNAPSHOT"
group="grId"
archivesBaseName="coc-comparison"

repositories {
    mavenCentral()
}

dependencies {
    compile 'commons-lang:commons-lang:2.5'
    testCompile 'junit:junit:4.8.1'
}
```



# Gradle problems



# Gradle bugs

gradle

Log In! ▾ Quick Search

Dashboards ▾ Projects ▾ Issues ▾

## Issue Navigator

Summary Edit New Manage

You are currently using a new, unsaved search.

Summary

Project: Gradle  
Status: Open  
Sorted by: Issue Type ascending, then Priority descending

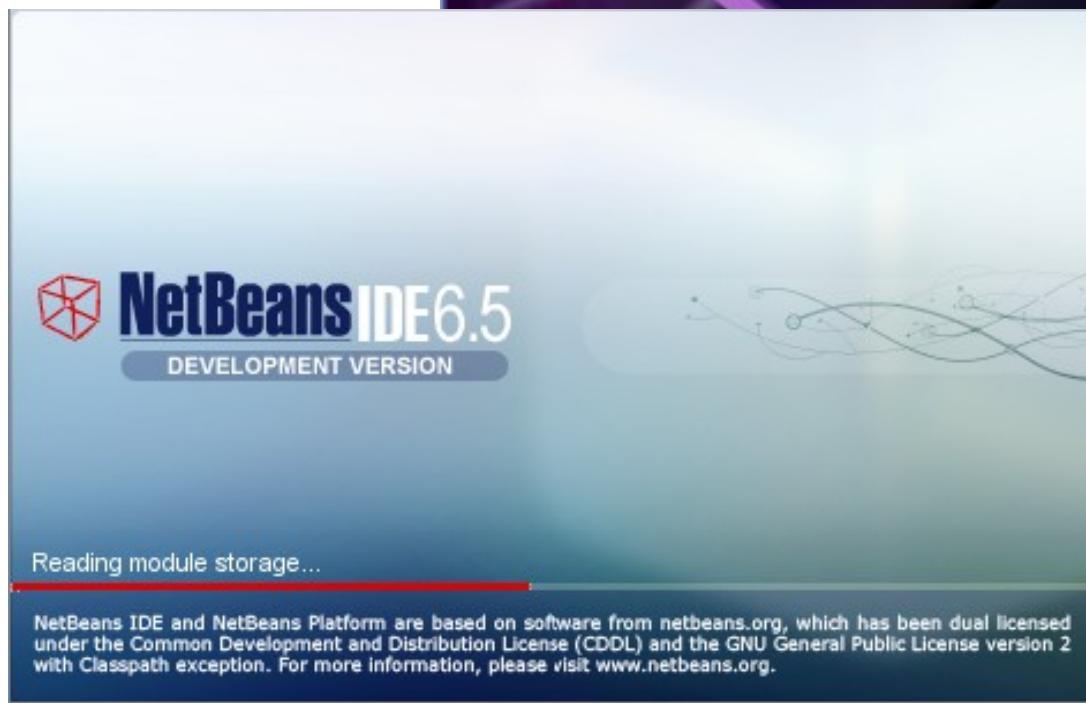
Operations

Save it as a filter

Displaying issues 1 to 50 of 977 matching issues.

T	Key	Summary	Assignee	Reporter	P	Status	Resolution	Created	Updated	Due
▶	GRADLE-2005	Received a failure event for test with unknown id	Unassigned	Gradle Forums	✖	Open	Unresolved	17/Dec/11	17/Dec/11	
▶	GRADLE-2092	Gradle misbehaves when there is a cycle in the chain of parent POMs	Unassigned	Gradle Forums	✖	Open	Unresolved	09/Feb/12	09/Feb/12	
▶	GRADLE-2115	Up-To-Date check ignores @Input-annotated boolean properties	Unassigned	Gradle Forums	✖	Open	Unresolved	22/Feb/12	22/Feb/12	
▶	GRADLE-2133	Eclipse plugin doesn't attach source for the gradleApi() dependency	Unassigned	Gradle Forums	✖	Open	Unresolved	28/Feb/12	28/Feb/12	
▶	GRADLE-2087	Project status always reverting to integration	Unassigned	Gradle Forums	✖	Open	Unresolved	08/Feb/12	06/Mar/12	
▶	GRADLE-2141	Gradle downloads changing modules every build when using an ivy DependencyResolver	Unassigned	Gradle Forums	✖	Open	Unresolved	04/Mar/12	04/Mar/12	
▶	GRADLE-2172	gradle cleanClasses base plugin added rule task always results in 'UP-TO-DATE'	Unassigned	Gradle Forums	✖	Open	Unresolved	14/Mar/12	14/Mar/12	
▶	GRADLE-2200	Cycle detected when setting up gradle's own source in eclipse	Unassigned	Gradle Forums	✖	Open	Unresolved	28/Mar/12	28/Mar/12	
▶	GRADLE-2206	Sources jar not downloaded after missing cache populated	Unassigned	Gradle Forums	✖	Open	Unresolved	02/Apr/12	02/Apr/12	
▶	GRADLE-2136	Third-party plugins cannot be applied by ID from external build scripts	Unassigned	Gradle Forums	✖	Open	Unresolved	03/Mar/12	05/Apr/12	
▶	GRADLE-2171	Add an option to avoid duplicate entries when creating a zip file	Unassigned	Gradle Forums	✖	Open	Unresolved	14/Mar/12	03/Apr/12	
▶	GRADLE-2235	Programmatically configuring Jetty Plugin causes ClassCastException	Unassigned	Gradle Forums	✖	Open	Unresolved	14/Apr/12	14/Apr/12	
▶	GRADLE-1982	Should disabling a task automatically remove any artifact it might publish?	Unassigned	Gradle Forums	✖	Open	Unresolved	06/Dec/11	16/Apr/12	
▶	GRADLE-2249	Sporadic failures due to "Could not load compiled classes for initialization script 'init.gradle' from cache"	René Grösche	Gradle Forums	✖	Open	Unresolved	24/Apr/12	06/May/12	
▶	GRADLE-1597	Custom configuration transitive dependency omitted due to project naming	Luke Daley	Daniel Gredler	✖	Open	Unresolved	07/Jun/11	21/Aug/11	
▶	GRADLE-2202	-Dhttp.nonProxyHosts is not respected	Unassigned	Gradle Forums	✖	Open	Unresolved	28/Mar/12	28/Mar/12	
▶	GRADLE-1912	Project Dependency for ear / war not working	Unassigned	Georg Fleischer	✖	Open	Unresolved	10/Nov/11	02/Mar/12	
▶	GRADLE-2184	Regression: Spring application context creation fails with org.springframework.aop.aspectj.AspectJAdviceParameterNameDiscoverer\$AmbiguousBindingException: Still 2 unbound args at this(),target(),args() binding stage, with no way to determine between them	Unassigned	mech	✖	Open	Unresolved	20/Mar/12	20/Mar/12	

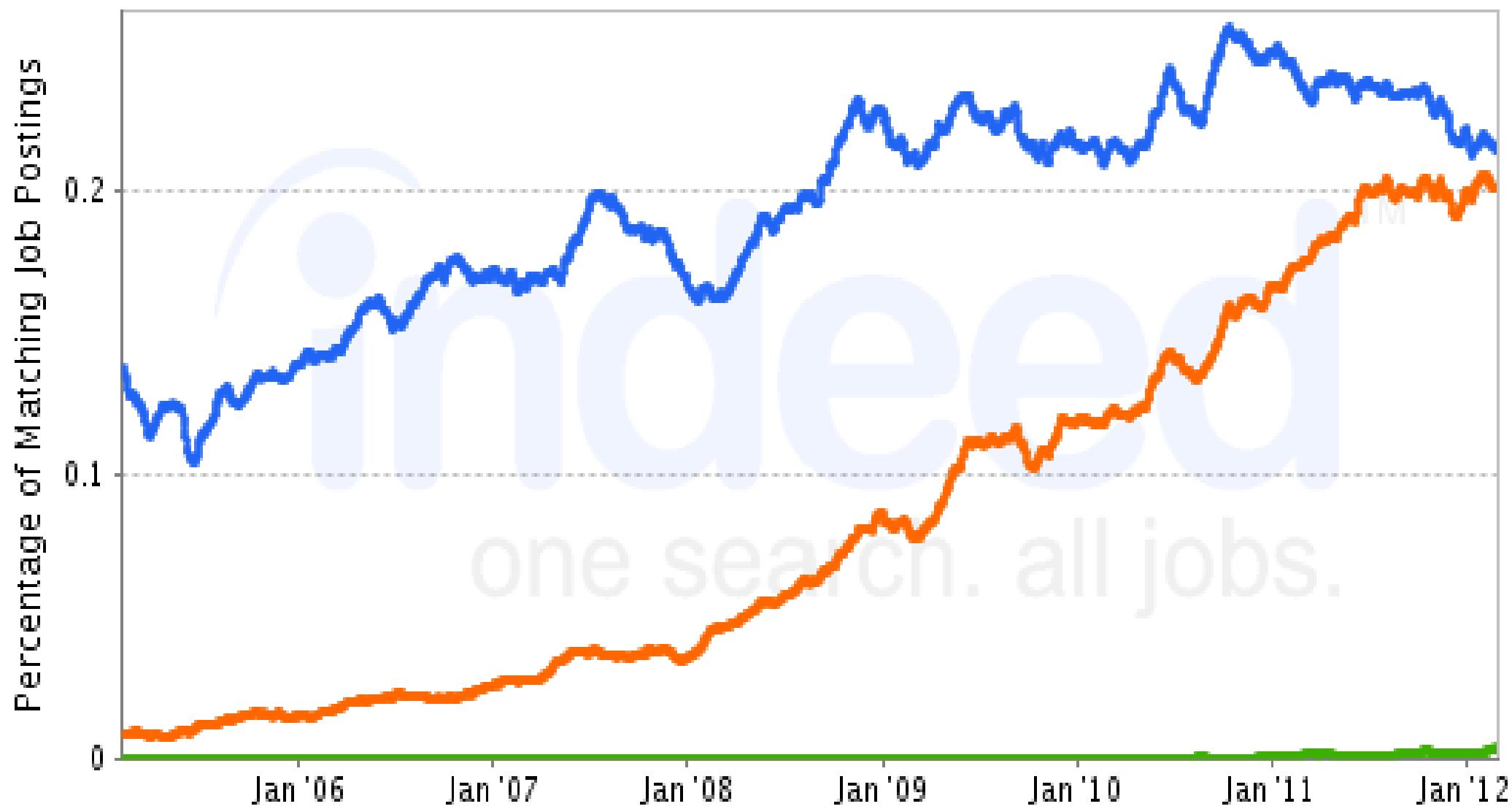
# IDE integration



# market penetration

Job Trends from Indeed.com

maven ant gradle



# Gradle tomorrow

- Release 1.0 till the end of 2012
  - Deep import of maven projects
  - Release management
  - Smart testing
  - Archetypes?



THANK  
YOU!



# Resources

- gradle.org
- groovy.codehaus.org
- sonatype.org
- ant.apache.org/ivy



# Questions



# Contacts

Igor Khotin

E-mail: [khotin@gmx.com](mailto:khotin@gmx.com)

Blog: [www.ikhotin.com](http://www.ikhotin.com)

Twitter: [chaostarter](https://twitter.com/chaostarter)

