



Raul Heinrich

source: https://commons.wikimedia.org/wiki/File:Toronto_Evening_Skyline_HDR.jpg

Maven 102

Best Practices

Brian Demers
Open Source Developer

 BrianDemers  bdemers

 Gradle

Who is this guy?



Java™
Champions



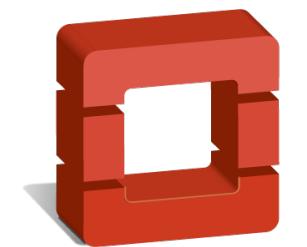
Warner River
Bees



codemonkey.fm



Apache Directory



openstack™



Maven™

Sonatype
Nexus







VS

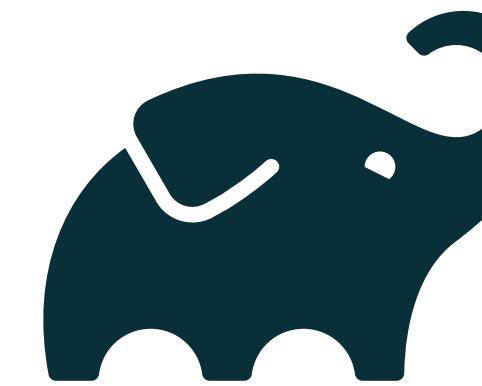




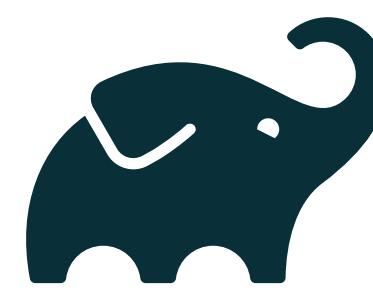
Gradle Build Tool

VS

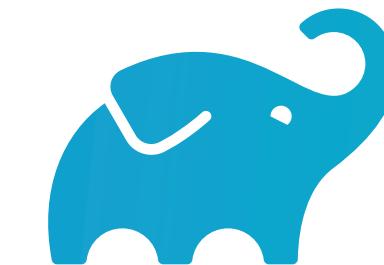
Maven™



Gradle

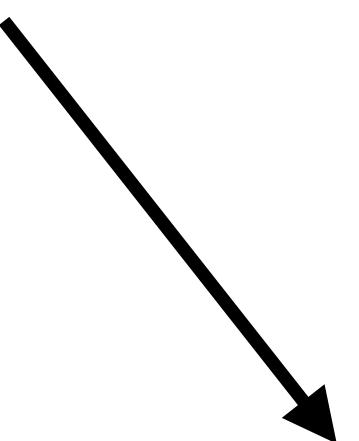
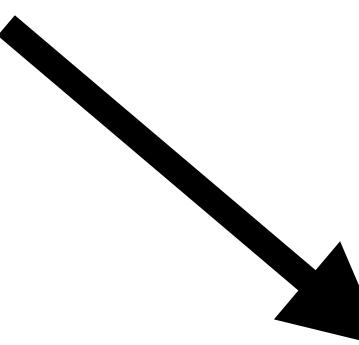
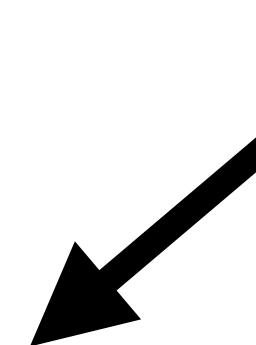


Gradle Build Tool



Gradle Enterprise

Developer
Productivity
Engineering



Gradle Build Tool

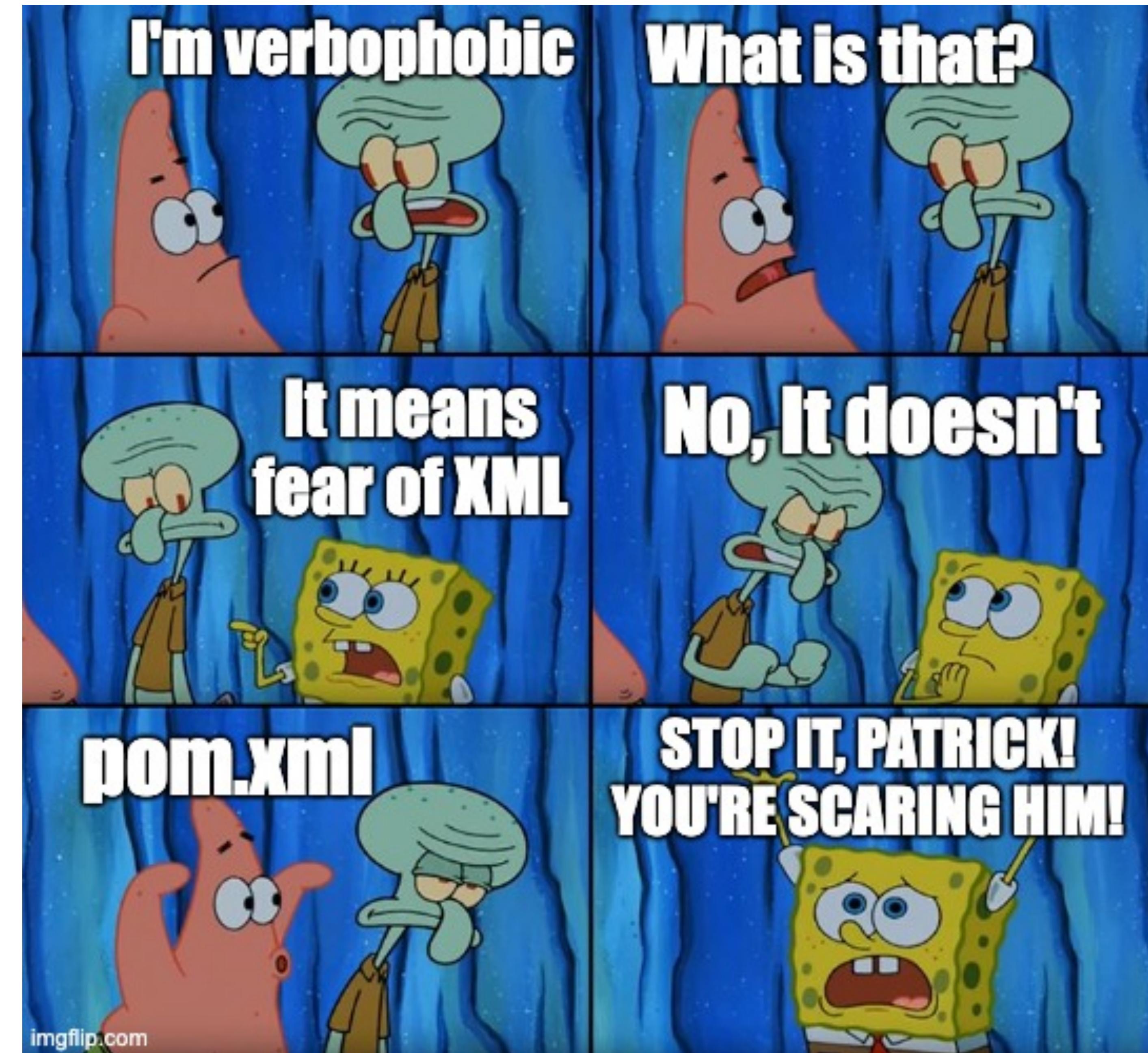
MavenTM

Bazel

sbt

Topics

- Maven Demystification
- Speed up your build
- Things to Commit or Omit
- Memes





Dependencies are a tree



```
$ mvn dependency:tree
junit-jupiter
+- junit-jupiter-api
|   +- opentest4j
|   +- junit-platform-commons
|   \- apiguardian-api
+- junit-jupiter-params
\- junit-jupiter-engine
  \- junit-platform-engine
```

Your Application





Andres Almiray
@aalmiray



#maven



3:00 AM · Mar 16, 2020

Maven Build Lifecycle

- validate
- initialize
- generate-sources
- process-sources
- generate-resources
- process-resources
- compile
- process-classes
- generate-test-sources
- process-test-sources
- generate-test-resources
- process-test-resources
- test-compile
- process-test-classes
- test
- prepare-package
- package
- pre-integration-test
- integration-test
- post-integration-test
- verify
- install
- deploy

Maven Build Lifecycle

Phase	Plugin Goal
process-resources	resources:resources
compile	compiler:compile
process-test-resources	resources:testResources
test-compile	compiler:testCompile
test	surefire:test
package	jar:jar
install	install:install
deploy	deploy:deploy

Maven Build Lifecycle

Phase	Plugin Goal
process-resources	resources:resources
compile	compiler:compile
process-test-resources	resources:testResources
test-compile	compiler:testCompile
test	surefire:test
package	jar:jar
verify	pmd:check, check style:check, jacoco:check
install	install:install
deploy	deploy:deploy

Plugin Configuration

```
<plugin>
  <groupId>org.codehaus.mojo</groupId>
  <artifactId>buildnumber-maven-plugin</artifactId>
  <version>3.0.0</version>
  <executions>
    <execution>
      <phase>validate</phase>
      <goals>
        <goal>create</goal>
      </goals>
    </execution>
  </executions>
  <configuration>
    <shortRevisionLength>7</shortRevisionLength>
    <doCheck>false</doCheck>
    <doUpdate>false</doUpdate>
  </configuration>
</plugin>
```

Plugin Configuration

```
<plugin>
  <groupId>org.codehaus.mojo</groupId>
  <artifactId>buildnumber-maven-plugin</artifactId>
  <version>3.0.0</version>
  <executions>
    <execution>
      <phase>validate</phase>
      <goals>
        <goal>create</goal>
      </goals>
      <configuration>
        <shortRevisionLength>7</shortRevisionLength>
        <doCheck>false</doCheck>
        <doUpdate>false</doUpdate>
      </configuration>
    </execution>
  </executions>
</plugin>
```

Plugin Configuration (example)

```
$ mvn buildnumber:create # global config
```

```
[INFO] Storing buildNumber: 6231815
```

```
$ mvn buildnumber:create # execution config
```

```
[INFO] Storing buildNumber: 623181597908f815cb9af206a389e739d80435a5
```

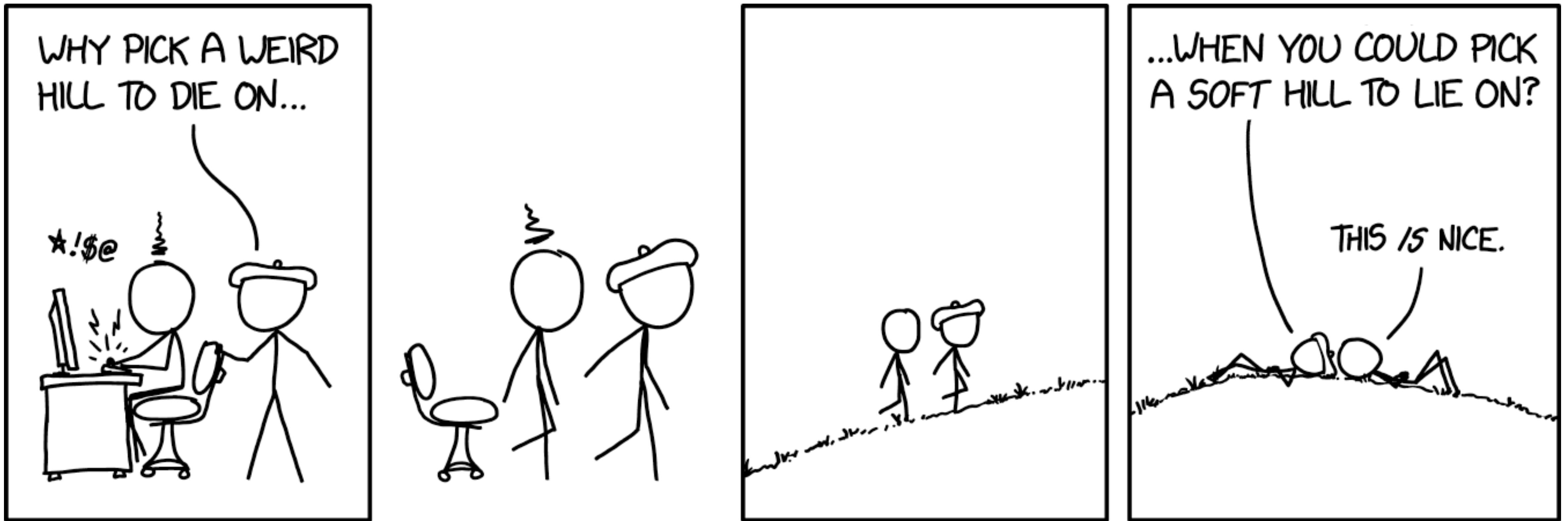
Custom Maven Plugins

```
@Mojo(name = "roll", defaultPhase = LifecyclePhase.VERIFY)
public class DiceMojo extends org.apache.maven.plugin.AbstractMojo {

    @Parameter(property = "expression", defaultValue = "1d20")
    private String expression;

    public void execute() throws MojoExecutionException,
                               MojoFailureException {
        getLog().info("Dice eval: " + Dice.roll(expression));
    }
}
```

Dependency Ranges

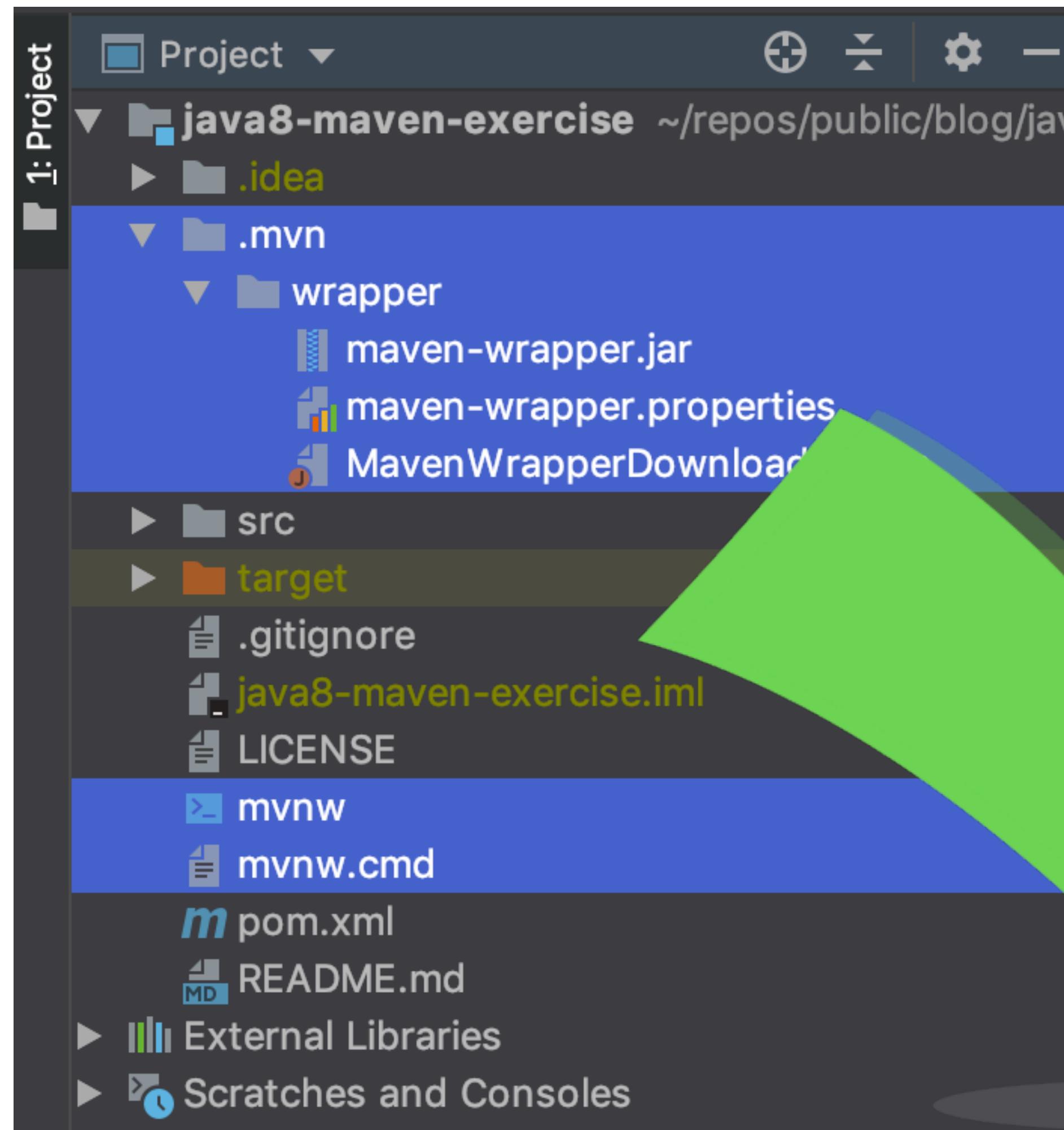


source: xkcd.com/2247

Dependency Ranges

Range	Meaning
(,1.0]	$x < 1.0$
(,1.0)	$x < 1.0$
[1.0]	$x = 1.0$
[1.0,)	$x = 1.0$
(1.0,)	$x > 1.0$
(1.0,2.0)	$1.0 < x < 2.0$
[1.0,2.0]	$1.0 \leq x \leq 2.0$
(,1.0],[1.2,)	$x < 1.0 \text{ or } x > 1.2$ Multiple ranges are comma-separated
(,1.1),(1.1,)	$x \neq 1.1$

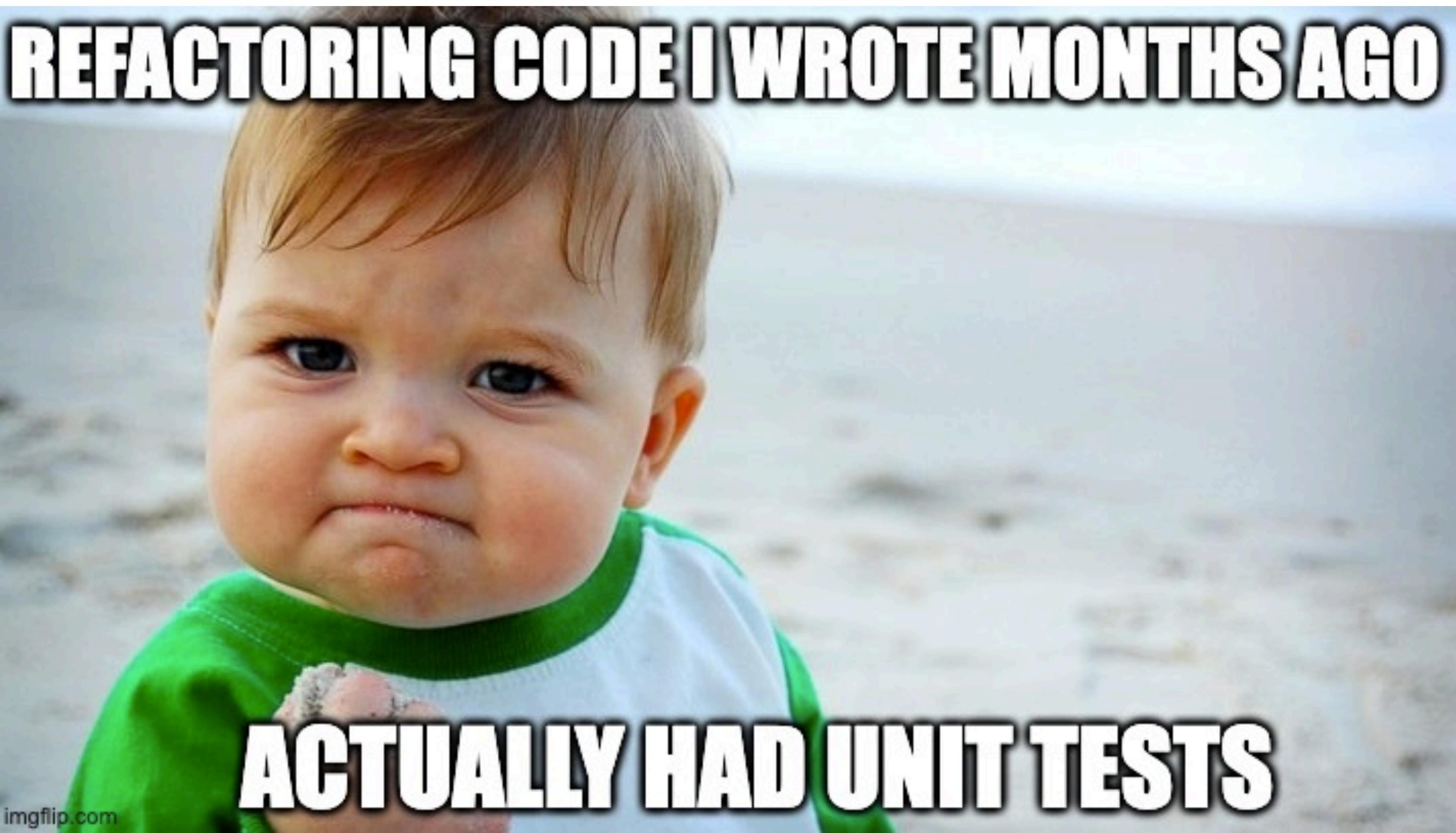
Pin Versions



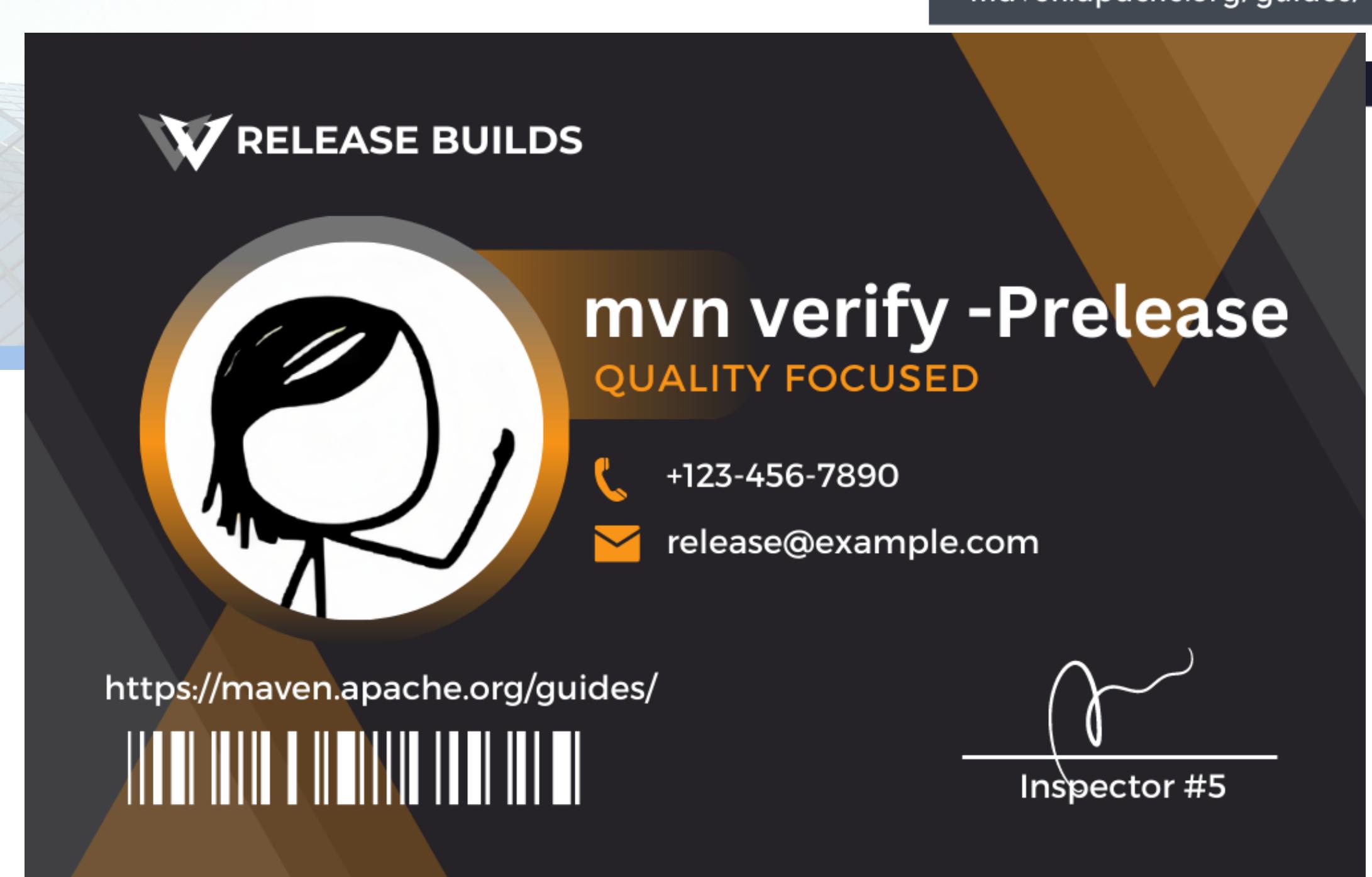
<https://area-b.com/blog/2019/05/04/1210/>



Separate Unit and Integration Tests



Use Profiles



Email : quick@example.com
Address: 10 Second Build St.
Phone : +123-456-7890

mvn verify -Pquick
Fast Build

headshots: xkcd.com

Test Jars



Uberjars



Source jars



A screenshot of an IDE interface, likely IntelliJ IDEA, displaying the Java `Assert` class source code. The window title is "repro-jwe". The left sidebar shows a file tree with "CompressionExceptionTest.groovy" and "Assert.java" selected. The right sidebar includes icons for Current File, Reader Mode, and a search bar with the letter "m". The main editor area shows the following code:

```
package org.junit;
import ...;

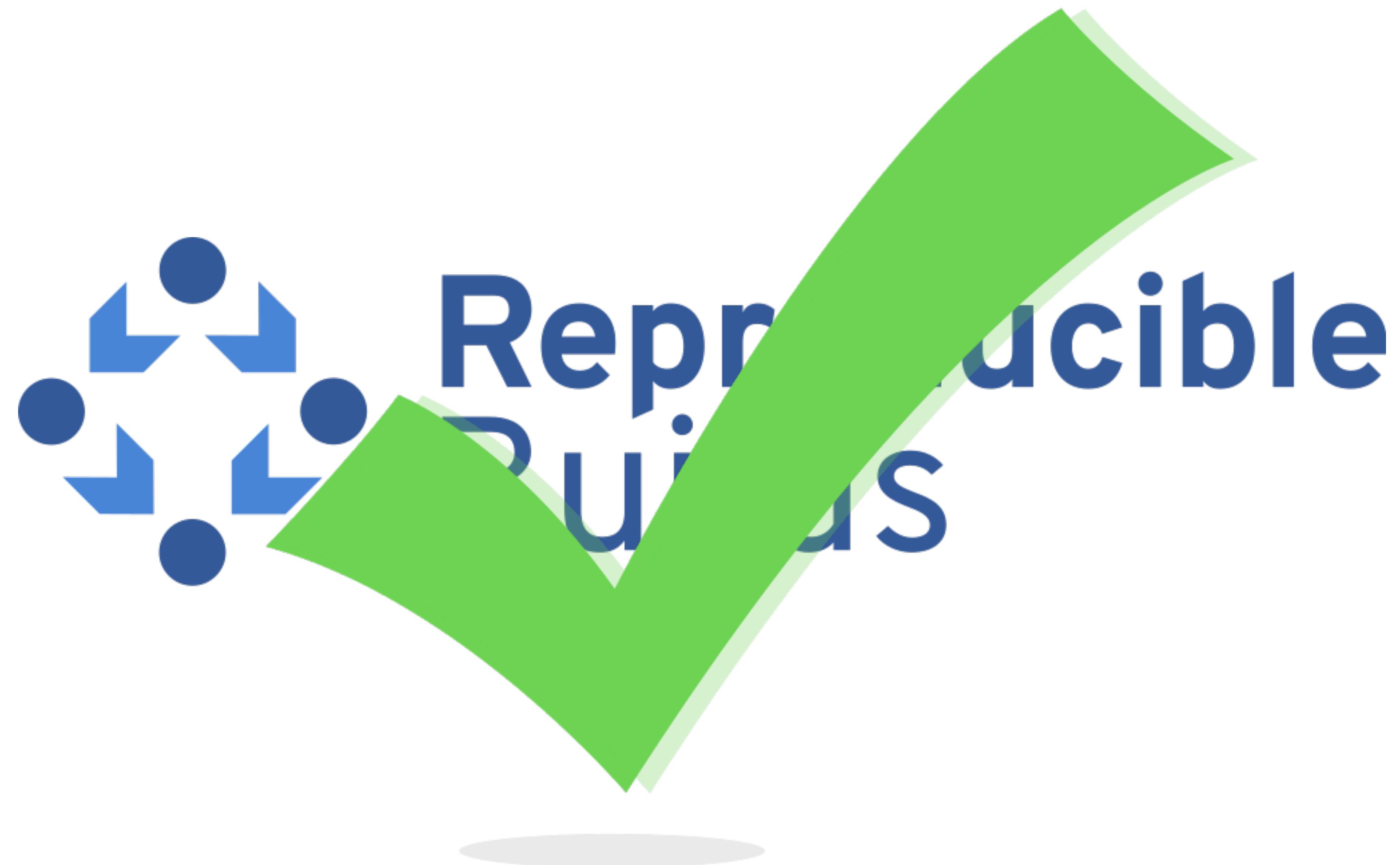
A set of assertion methods useful for writing tests. Only failed assertions are recorded. These can be used directly: Assert.assertEquals(...), however, they read better if they are imported through static import:
import static org.junit.Assert.*;
...
assertEquals(...);

Since: 4.0
See Also: AssertionError

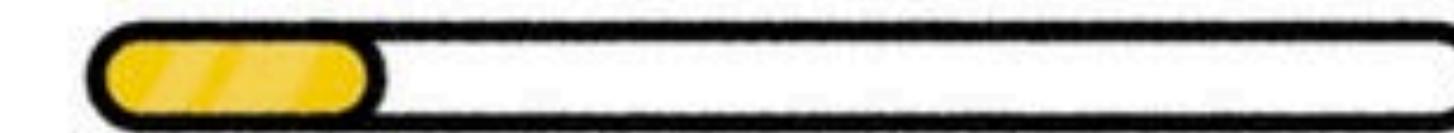
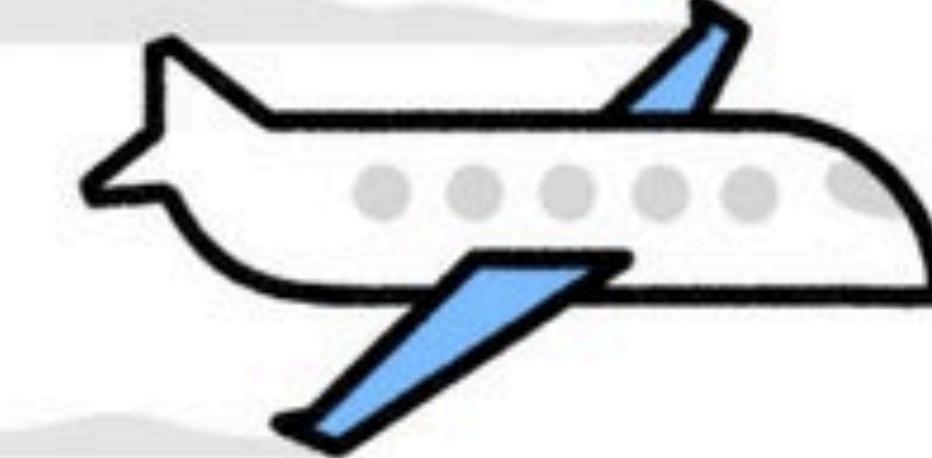
public class Assert {
    // Protect constructor
    protected Assert() {
    }
}

Asserts that a condition is true. If the condition is false, then an AssertionError is thrown with the given message.
Params: message – the identifying message for the error (null okay)
        condition – condition to be checked
public static void assertEquals(String message, Object expected, Object actual, boolean condition) {
    if (!condition) {
        fail(message);
    }
}
```

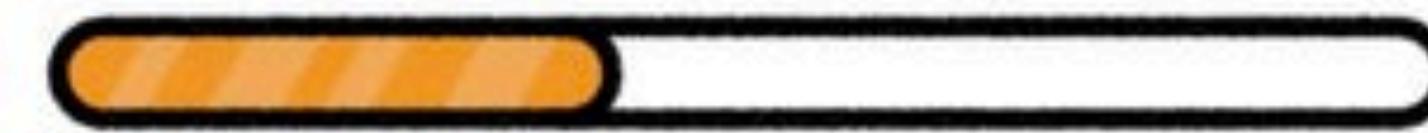
The code editor shows line numbers 1 through 43. A tooltip for the `assertEquals` method is visible, providing its documentation and parameters. The status bar at the bottom indicates the file is "junit-4.13.1.jar > org > junit > © Assert", with encoding "UTF-8" and "4 spaces".



THE FASTEST THINGS ON EARTH



CHEETAH



AIRPLANE



Maven™

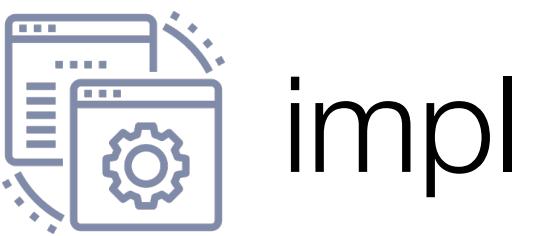


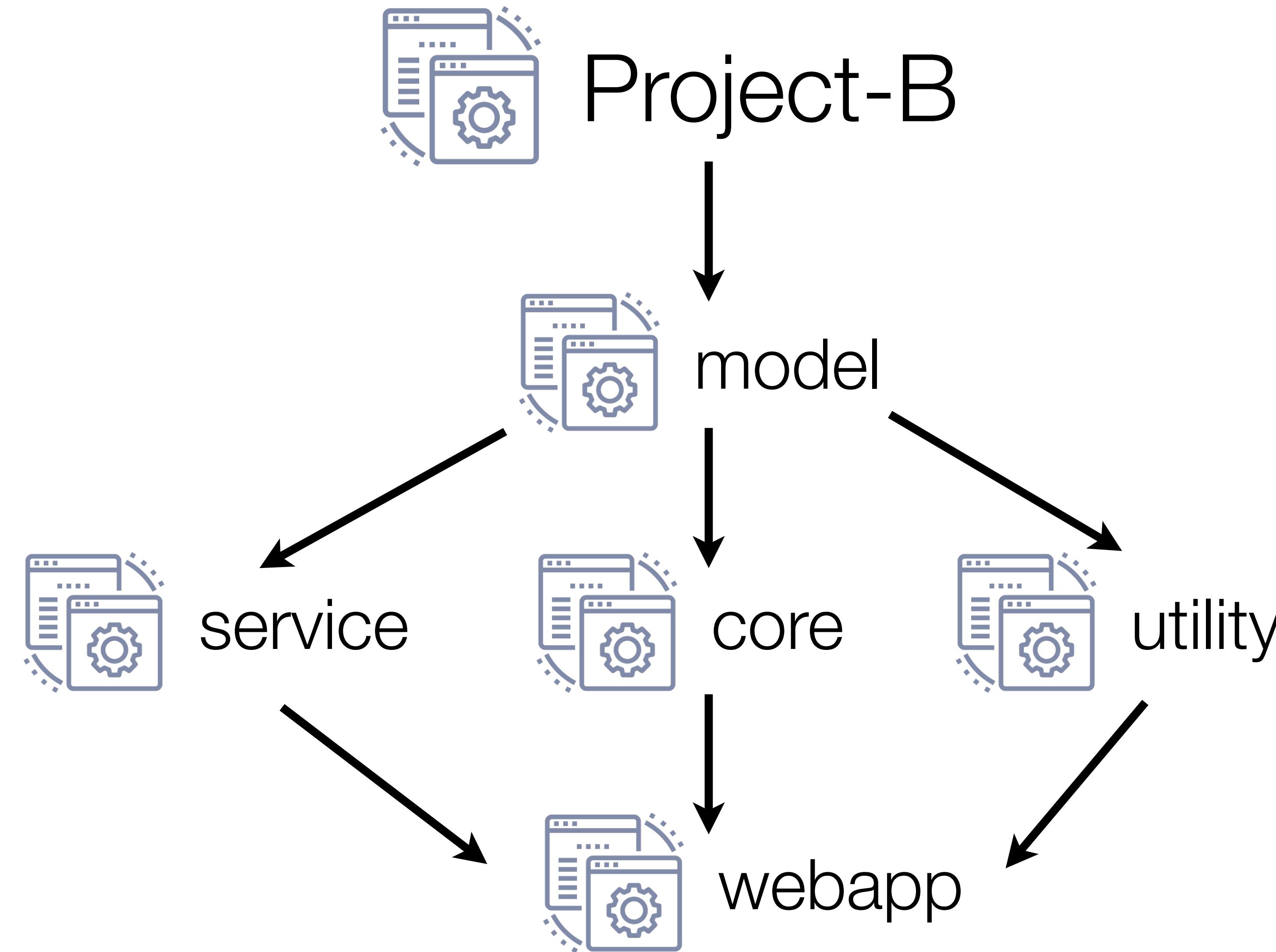
SPEED OF LIGHT



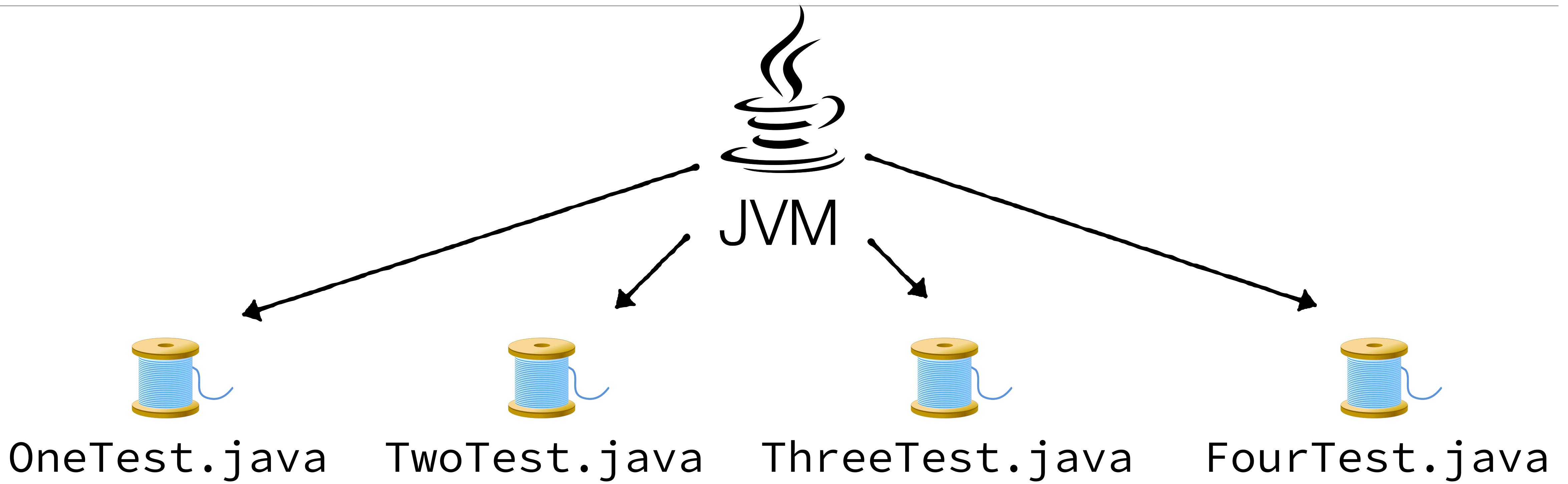
Parallel Builds

Multiple Threads

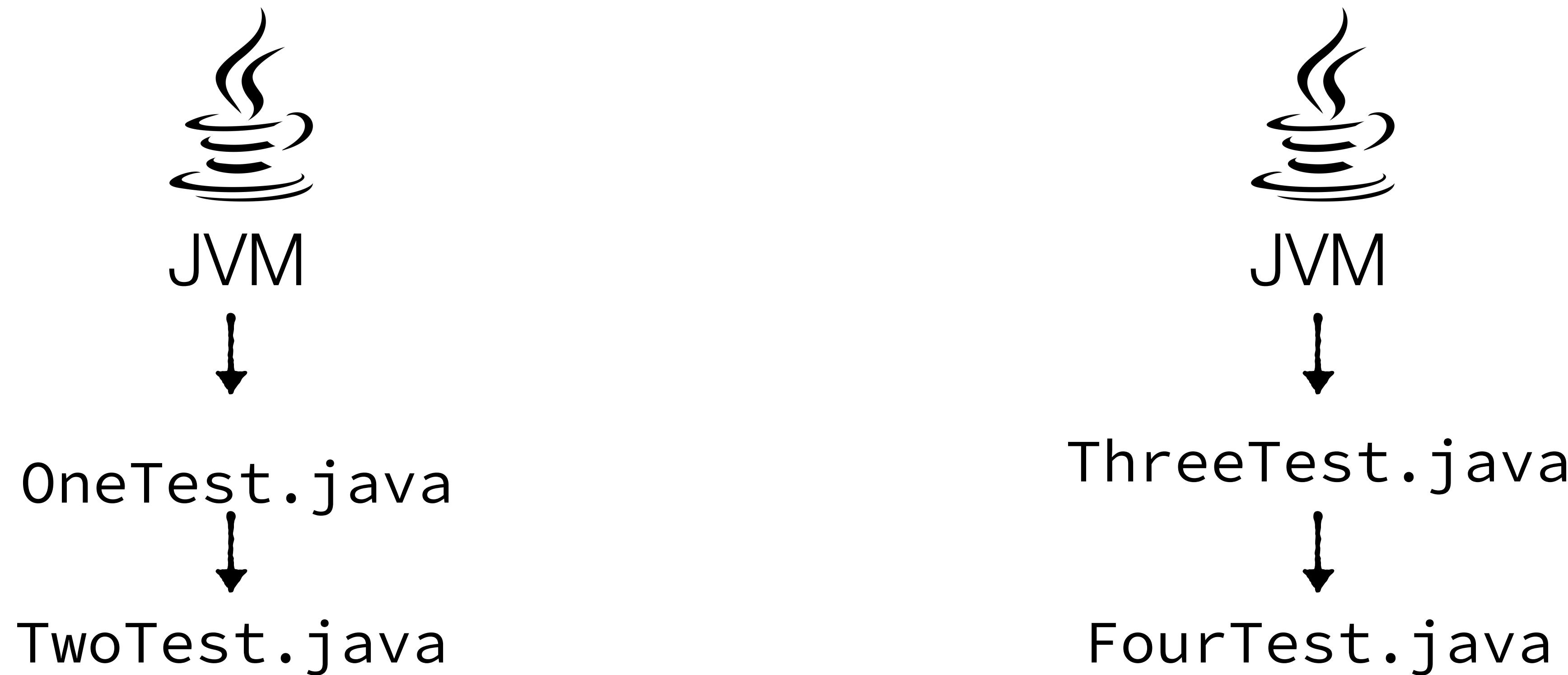




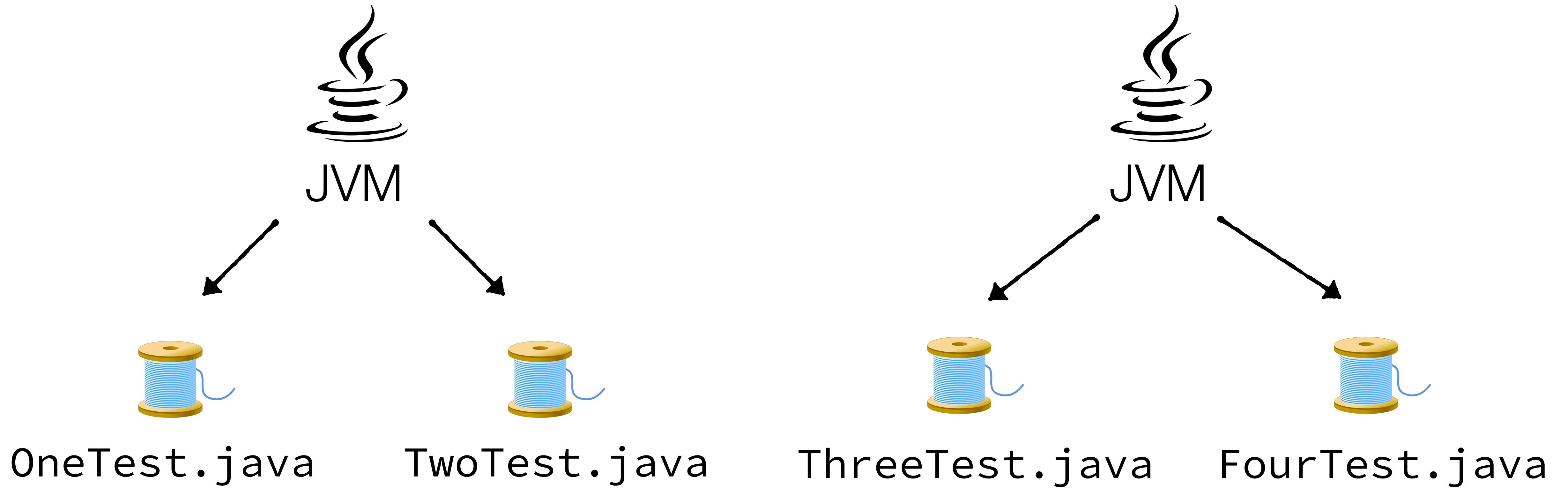
Test w/ Multiple Threads



Test w/ Multiple JVMs (forkCount)



Test w/ Multiple JVMs & Multiple Threads



Just add .parallel()
and everything will be all right



source: <https://levelup.gitconnected.com/be-careful-with-java-parallel-streams-3ed0fd70c3d0>

“The future is always beginning now.”

-Mark Strand



Daemon



MavenTM

4.0.0

Questions?



Speed Challenge
(Free Swag)



Slides