

# 10 WAYS TO BE A **MORE** **PRODUCTIVE** DEVELOPER

SIMON MAPLE

@SJMAPLE



# ABOUT **ME**

---



SIMON MAPLE  
**@SJMAPLE**



**DEVELOPER** ADVOCATE

**VIRTUAL JUG** FOUNDER

**LONDON JUG** CO-LEADER

**JAVA** CHAMPION

**JAVAONE** ROCKSTAR

**REBELLABS** AUTHOR

# 10 KICK-ASS TECHNOLOGIES MODERN DEVELOPERS LOVE

GEEKS REVEAL THE INDUSTRY'S BEST TECH

Is geek love better  
than normal love?

1. Confluence
2. Git
3. Gradle
4. Groovy
5. IntelliJ IDEA
6. Jenkins
7. JIRA
8. MongoDB
9. Scala
10. Tomcat



**REBELLABS**  
by ZERO TURNAROUND

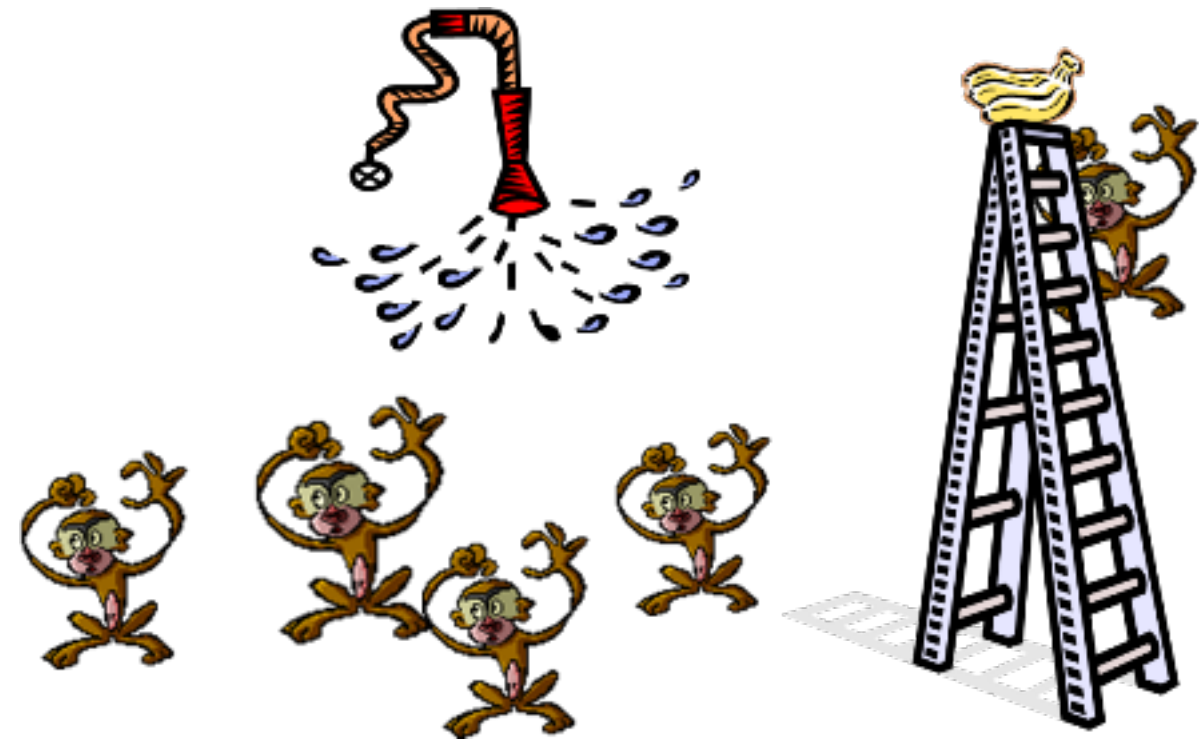
# HOW CAN WE **IMPROVE** OUR **PRODUCTIVITY?**



# PRODUCTIVITY i

---

**Removing**  
non-essential  
tasks



# PRODUCTIVITY ii

---

**Optimising** existing tasks

**2005**

:0)

**2015**

:)

# PRODUCTIVITY iii

---

**Retaining** awesome tasks

# DON'T LOSE SIGHT OF THE GOAL

---

**Speed** is **not** our goal

**Speed** is a **measure** of how fast  
we reach our goal



# PRODUCTIVITY IS A LIE





# COMMUNICATION



# COMMUNICATION

---

**Sharing** knowledge across a team

# COMMUNICATION

---

It isn't just **technology**  
that increases your productivity

# COMMUNICATION

---



# SLACK

---



Group **channels**

**Open door** chat policy

Smart **Integrations**

# ATLASSIAN **TOOLING**

---





# ATLASSIAN **TOOLING**

---





# TOOLING

---

PRODUCTIVITY ii

**OPTIMISING** EXISTING TASKS  
IN A LESS THAN  
AWESOME PROCESS



# THE GREAT JAVA APPLICATION SERVER DEBATE

TOMCAT, JBOSS, JETTY, GLASSFISH, LIBERTY  
PROFILE, WEBLOGIC & WEBSHERE



DOWNLOAD & INSTALLATION

TOOLING SUPPORT

DOCS & COMMUNITY

# PERFORMANCE METRICS

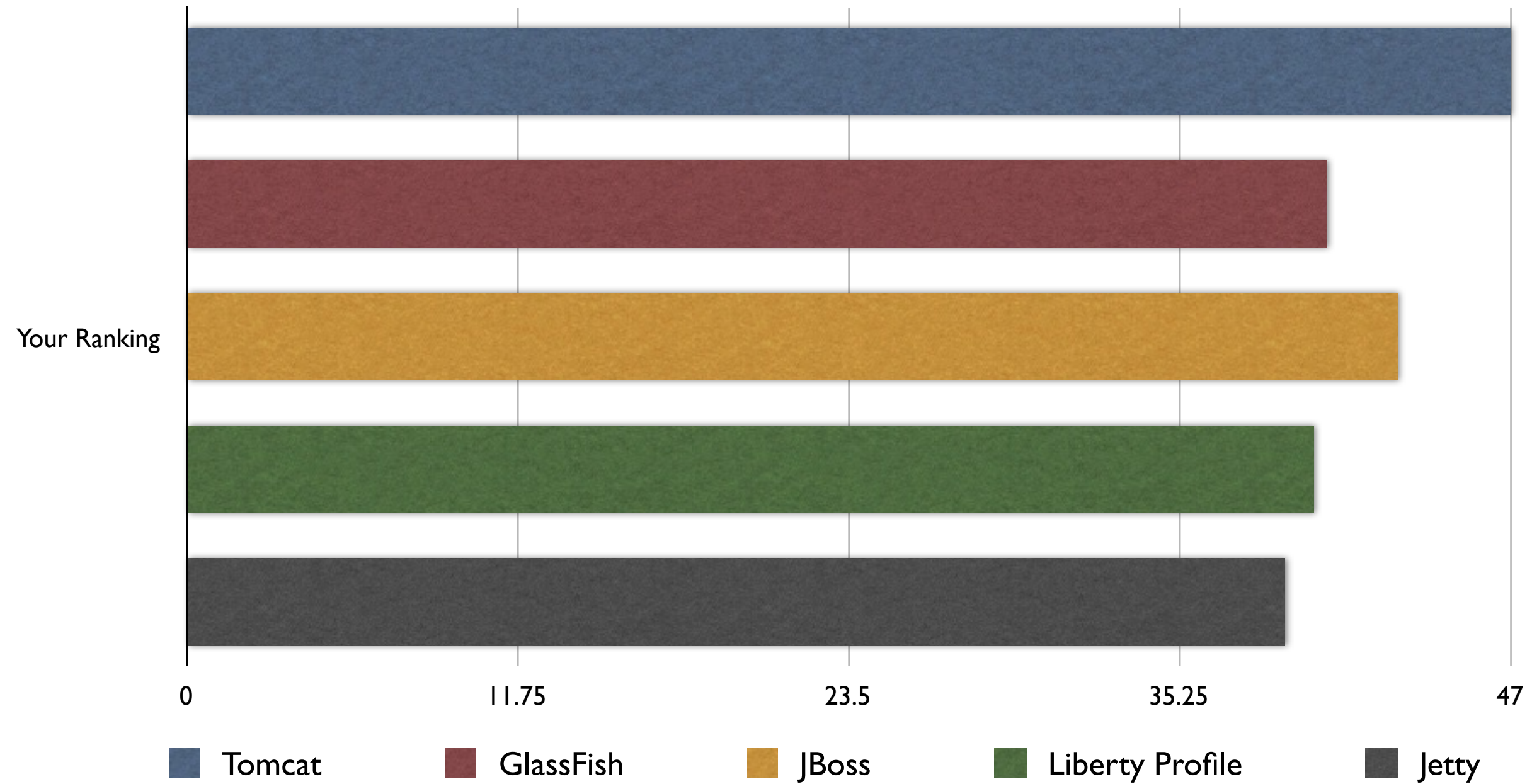
---

	LP	TC	GF	JETTY	JB	AVG
Startup with app & invoke petclinic	10.5	7.5	11	12.5	8.5	10
Startup, deploy & invoke petclinic	11	8.5	11.5	11.5	7.5	10
Startup with app & invoke jenkins	25	11.5	14	15	17	16.5
Startup, deploy & invoke jenkins	25	12.5	16.5	15	18	17.4

# PERFORMANCE METRICS

Task	Liberty Profile 8.5.5.4	Liberty Profile 8.5.0.0
Empty server startup	2	2
Server startup time with petclinic installed	3	2.5
Server startup time with jenkins installed	8	4
Petclinic app deploy time	2.5	1
Jenkins app deploy time	5.5	2
Petclinic init time (1st req)	4	8
Jenkins init time (1st req)	8	21
Server startup, deploy petclinic & invoke	10	11
Server startup, deploy jenkins & invoke	16	25

# JFokus 2015



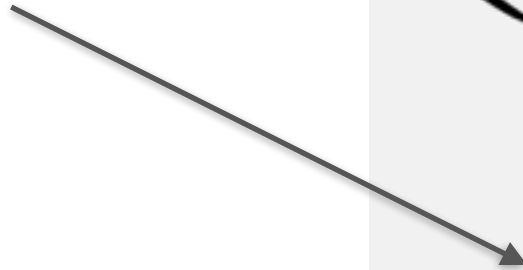
# FEATURES



---



# FEATURES

NOW  
**JAVA EE 7**  
COMPLIANT





# SPRING BOOT

---



**Just Run** Applications

**Production** Ready

Sensible **defaults**

# SPRING BOOT IN ACTION

---



*“Make JAR not WAR!”*

— Josh Long (@starbuxman)

# TOMEE

---

PRODUCTIVITY iii

---

**RETAINING AWESOME  
TASKS IN A  
AWESOME PROCESS**



<http://tomee.apache.org/>

# TOMEE

---

PRODUCTIVITY

---

**SPEED** IS NOT OUR GOAL

**SPEED** IS A MEASURE OF HOW  
FAST WE REACH OUR GOAL



<http://tomee.apache.org/>

# GRADLE

---







# JAVA BUILD TOOLS: PART 2

A DECISION MAKER'S COMPARISON OF  
MAVEN, GRADLE AND ANT + IVY

Who will win  
"Least Annoying Build Tool"?

# GRADLE

## DOING A CLEAN BUILD WITH TESTS (DOWNLOAD DEPENDENCIES)

	 <b>maven</b>	 <b>gradle</b>	  <b>ivy</b>
Command	rm -rf ~/.m2/repository && time mvn clean package	rm -rf ~/.m2/repository && rm -rf ~/.gradle/caches/ && time gradle clean build --daemon	rm -rf ~/.ivy2/cache/ && time ant clean war test
Time - Run 1 <i>(seconds)</i>	41.393	35.412	136
Time - Run 2 <i>(seconds)</i>	37.418	33.402	133
Time - Run 3 <i>(seconds)</i>	36.797	30.548	137
Time - Run 4 <i>(seconds)</i>	42.656	30.336	141
Time - Run 5 <i>(seconds)</i>	39.637	35.369	129
<b>Average</b> (min /max omitted)	<b>39.483</b>	<b>33.106</b>	<b>135.333</b>

**Maven**






**Gradle**



**Ant + Ivy\***



# GRADLE

Tasks	<b>maven</b>			 gradle			 		
IDEs	Eclipse	IDEA	NetBeans	Eclipse	IDEA	NetBeans	Eclipse	IDEA	NetBeans
Import a project build structure into IDE	yes	yes	yes	yes	yes	yes	yes	yes	yes
Submit a build from IDE	yes	yes	yes	yes	yes	yes	yes	yes	yes
Dependency management in IDE	yes	yes	yes	yes	yes	yes	yes	yes	yes
Automatic download of dependencies	yes	yes	yes	yes	yes	yes	yes	yes	yes
Support/Wizard to create build scripts	yes	yes	yes	yes	yes	yes	yes	yes	yes
Enablement	plugin	built-in	built-in	plugin	built-in	plugin	Ant: built-in Ivy: plugin	Ant: built-in Ivy: plugin	Ant: built-in Ivy: plugin



# INTELLIJ IDEA

---



# INTELLIJ IDEA

---



**Simon Maple**








@sjmaple

Could you help me write a presentation :)  
What are your favourite features in IntelliJ that help your productivity?



8:38 PM - 21 Mar 2015

# INTELLIJ IDEA

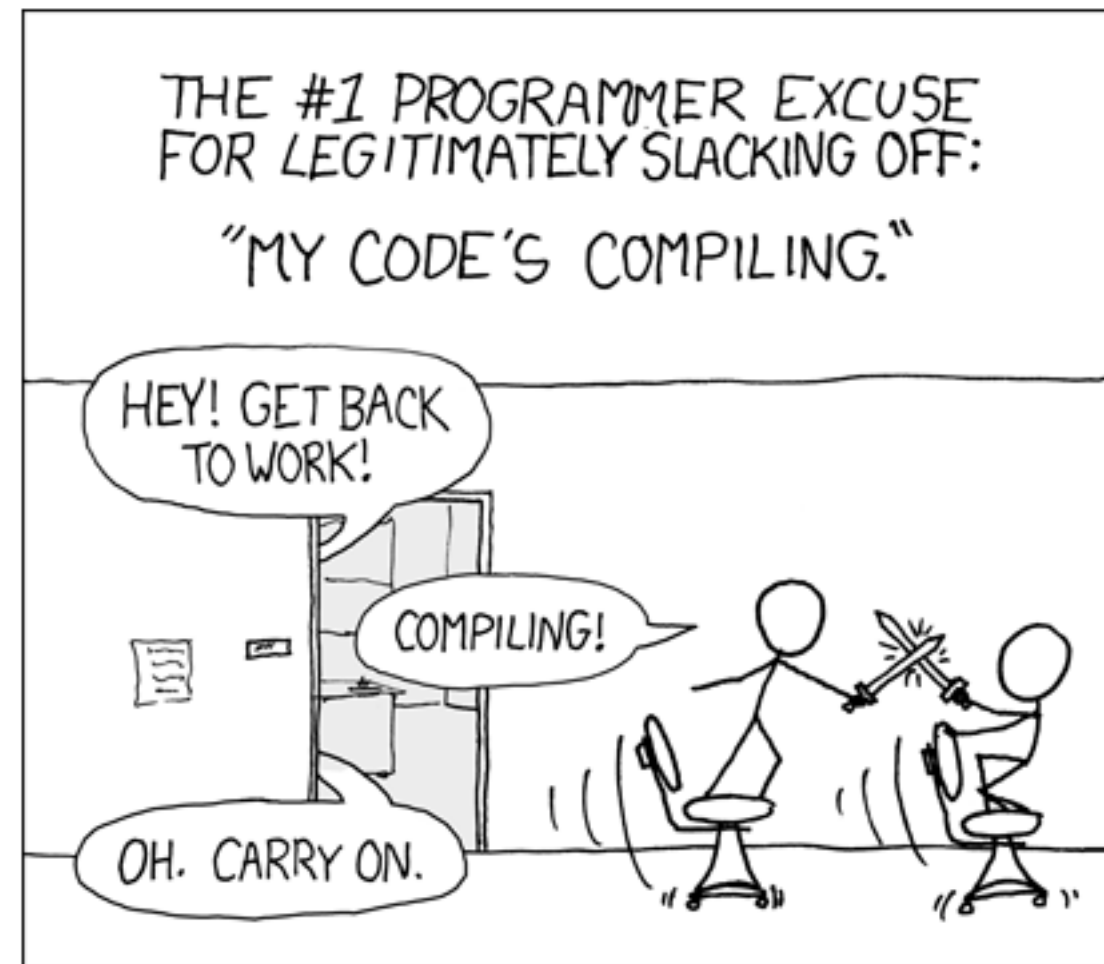
-  **Phil Standen** @phil\_standen 2d  
@sjmaple auto-gen .gitignore, JAX-RS @PathParam auto complete, JPA "make this a legit entity". Point at a git with a pom.xml and go.
-  **Benjamin Savoy** @benjamin\_savoy  
@sjmaple well, their licencing scheme and the fact I have to open a different IDE when I swich languages isn't making me more productive ;)
-  **Phil Standen** @phil\_standen 2d  
@sjmaple oh and "it looks like you are doing X click to make everything auto-config"
-  **Mert** @mertcal 2d  
@sjmaple tight integration with Spring: creating ApplicationContext configuration by identifying xml files, enabling bean profiles selection & etc.
-  **ags313** @ags313 2d  
@sjmaple suggestions without needing to bash ctrl-space, code centricism, ctrl-w, static analysis, refactorings, mac-shift-a
-  **Mert** @mertcal 2d  
@sjmaple awesome integration w/ VCS tools like git, even offers a merge UI when conflicts occur. no need to leave the IDE for 3rd party tool
-  **Mert** @mertcal 2d  
@sjmaple ability to change the default keymaps and export it to distribute for your teammates. also supports eclipse based shortcuts.

-  **Raúl Ávila** @ravila80 2d  
@sjmaple Ctrl + shift + enter to complete statement, it's amazing
-  **Sargerino** @sargerino 2d  
@sjmaple stability, seamless integration of features, i.e. no different perspectives to navigate...
-  **Benjamin Savoy** @benjamin\_savoy 2d  
@hhariri I work with modules in multiple languages. So far Komodo has been the best for that.
-  **Christopher Batey** @chbatey 2d  
@sjmaple a simple one but a goodie: generate constructor/getter/setter/equals/to string
-  **Damian Blazejewski** @damian\_bl 2d  
@sjmaple vcs operations pop up: alt + back quote, brilliant maven support
-  **Victor Bucutea** @victor\_bucutea 1d  
@sjmaple Alt+J multiple select, db tool window-indexes DB and makes brilliant auto completion on query typing. Simple yet brilliant thingz.
-  **Sebastien Arod** @seboch 1d  
@sjmaple Terminal integration (Alt+F12)

# WASTED TIME

PRODUCTIVITY i

**REMOVING NON-ESSENTIAL  
TASKS IN AN OTHERWISE  
AWESOME PROCESS**



XKCD.COM/303

# JREBEL



# WITH JREBEL

---

**To test code changes, developers need to:**

**0 CODE CHANGE**

~~1 Compile, build, package~~

~~2 SSH/SCP/FTP to target machine~~

~~3 Redeploy the application~~

~~4 Restart the application server~~

~~5 Create state required to test your code change~~

**6 TEST!**

**0 CODE CHANGE**

**Remove the  
remote access**

**Saving (avr)  
1 hr/day/dev**



# JREBEL IN ACTION

---



**FREE STUFF!**

**JRebel**

**0t.ee/jfokusjr**



*Get a free  
t-shirt!*



# **JBoss** FORGE

---



**JAVA EE** SCAFFOLDING

**RAPID** DEVELOPMENT

**PLUGIN** BASED

# **JBOSS FORGE** IN ACTION

---



# FAIL EARLY

---

PRODUCTIVITY

---

**SPEED** IS NOT OUR GOAL

**SPEED** IS A MEASURE OF HOW  
FAST WE REACH OUR GOAL



# ~~FAIL~~ LEARN EARLY

---

PRODUCTIVITY

---

**SPEED** IS NOT OUR GOAL

**SPEED** IS A MEASURE OF HOW  
FAST WE REACH OUR GOAL



# XREBEL



**Profiler for** Java Web Applications

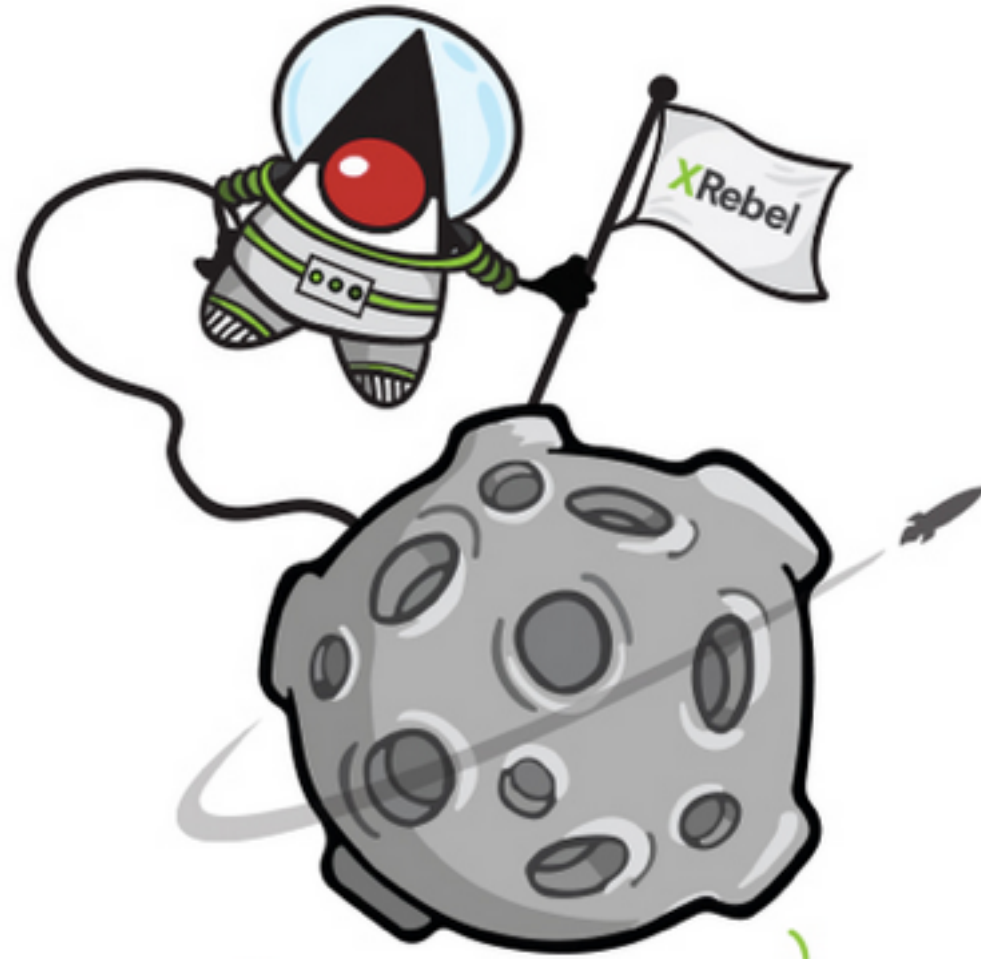
Understand what is happening  
**behind your code**

**Find** and **fix** issues



# XREBEL IN ACTION

---



*One small step for Duke,  
one giant leap for Java-kind*

**FREE STUFF!**

**XRebel**

**0t.ee/jfokusxr**



*Get a free  
t-shirt!*



# ARQUILLIAN

---





# ARQUILLIAN

---



**Take your tests** to the container

**Connect** to that container

**View** results like unit tests

# ARQUILLIAN

---

```
1  @RunWith (Arquillian.class)
2  public class ArquillianTest {
3
4
5      @Deployment
6      public static Archive<?> createTestArchive() {
7
8
9          return ShrinkWrap.create (WebArchive.class, "test-demo.war")
10             .addPackage (Manager.class.getPackage ())
11             .addPackage (SimpleProperty.class.getPackage ())
12             .addPackage (Producer.class.getPackage ())
13             .addPackage (RepositoryManager.class.getPackage ())
14             .addPackage (ServiceBean.class.getPackage ())
15             .addAsResource ("META-INF/persistence.xml")
16             .addAsWebInfResource (EmptyAsset.INSTANCE, "beans.xml")
17             // Deploy our test datasource
18             ;
19
20     }
21
```

# ARQUILLIAN

---

```
21  
22     @Inject  
23     ServiceBean service;  
24  
25     @Inject  
26     RepositoryManager query;  
27
```

# ARQUILLIAN

---

```
27
28 @Test
29 public void testApplication() throws Exception {
30
31     SimpleProperty property = new SimpleProperty();
32     property.setKey("mykey");
33     property.setValue("value");
34
35     service.put(property);
36     System.out.println("Created SimpleProperty");
37
38     SimpleProperty key = query.queryCache("mykey");
39     System.out.println("Retrieved: "+property);
40     assertNotNull(key);
41
42     service.delete(property);
43     System.out.println("Deleted: "+property);
44
45
46 }
47
```

# SUMMARY

---

**CARE** ABOUT **YOUR TIME**

**DON'T** LOSE SIGHT OF **YOUR GOAL**

**EXPLORE** **PRODUCTIVITY TOOLS**



# PRODUCTIVITY IS A LIE





# **DON'T** COMMUNICATE!





# DON'T USE SPRING BOOT!





**DON'T USE TOMEE!**





# DON'T USE INTELLIJ!





**STOP WORKING FASTER!**





**DON'T FAIL EARLY!**





# DON'T BE PRODUCTIVE!



# FREE STUFF!

## JRebel



[0t.ee/jfokusjr](http://0t.ee/jfokusjr)

## XRebel



[0t.ee/jfokusxr](http://0t.ee/jfokusxr)