



Maven a další build nástroje



Profiling Java aplikací

- Populární nástroje:
 - Zdarma (součást JDK):
 - JVisualVM (nově na GitHubu:
 - <https://visualvm.github.io/>
 - JMC (Java Mission Control)
 - Na produkci vyžaduje licenci
 - Placené
 - JProfiler
 - New Relic
 - YourKit

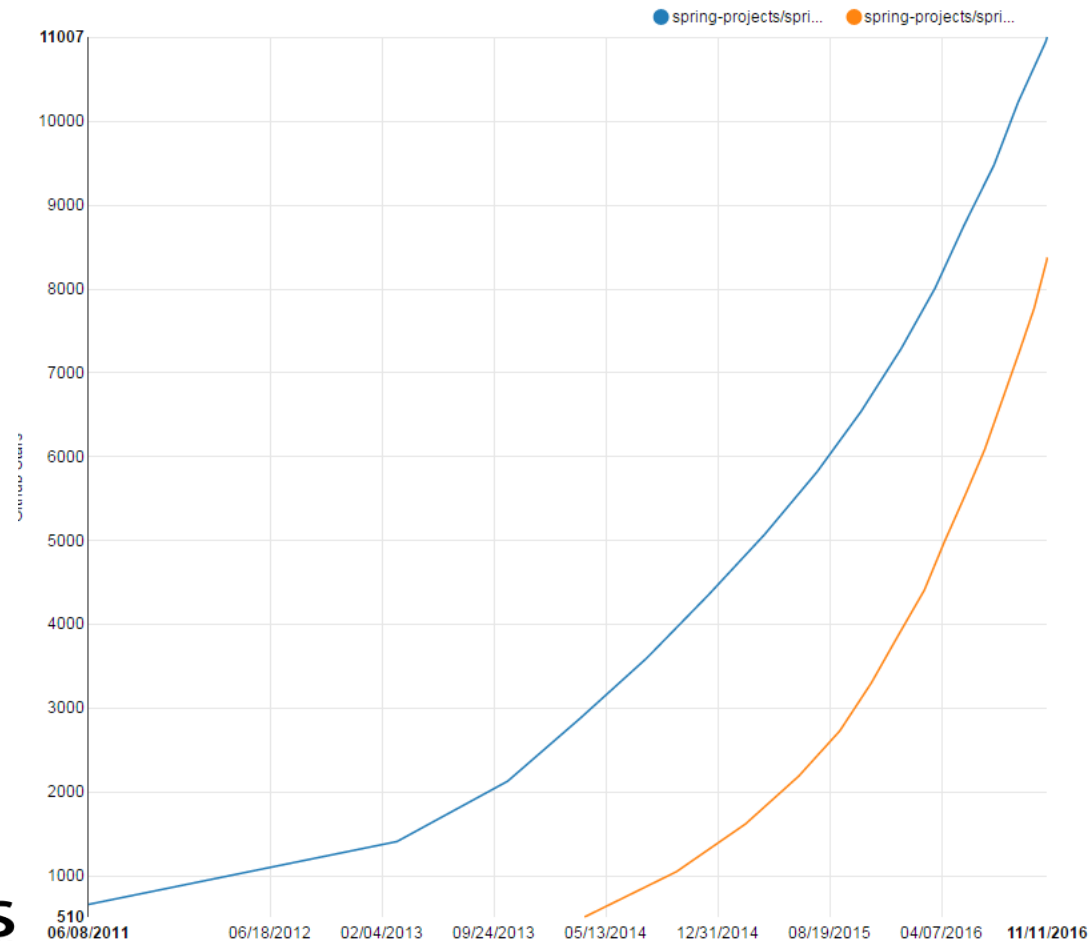


Spring vs. Java EE

- Další „populární“ otázka. Vzhledem k tomu, že Spring často kooperuje a staví nad Java EE, pak ani není moc férová.
- Nicméně díky tomu, že je Spring na GitHubu, pak se můžeme podívat jak hodně je populární tam:
 - <http://www.timqian.com/star-history/>



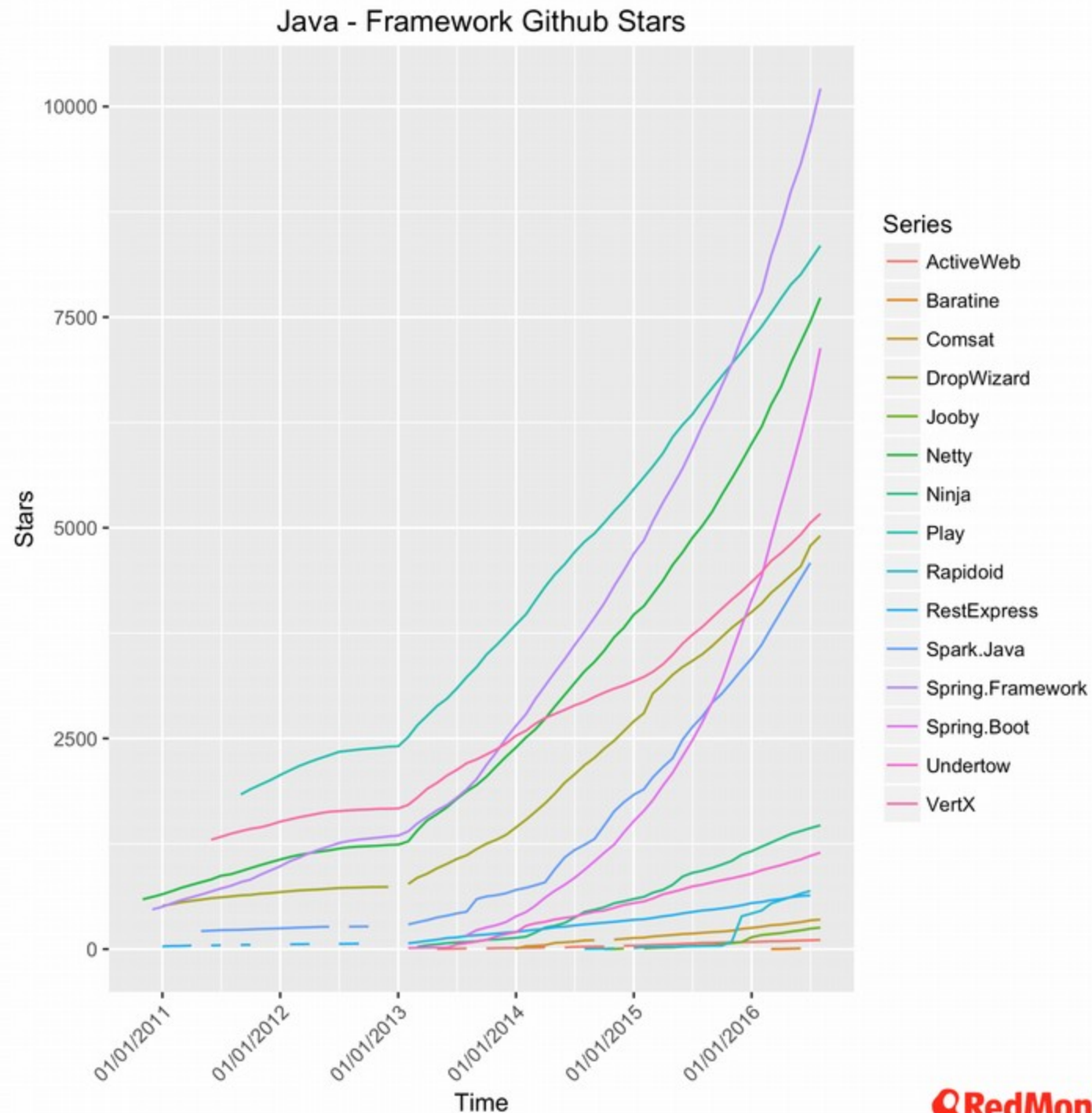
- Vývoj počtu hvězdiček v čase (kumulativní součet) pro Spring Framework a Spring Boot:



Další srovnání GitHub stars populárních Java web projektů od společnosti

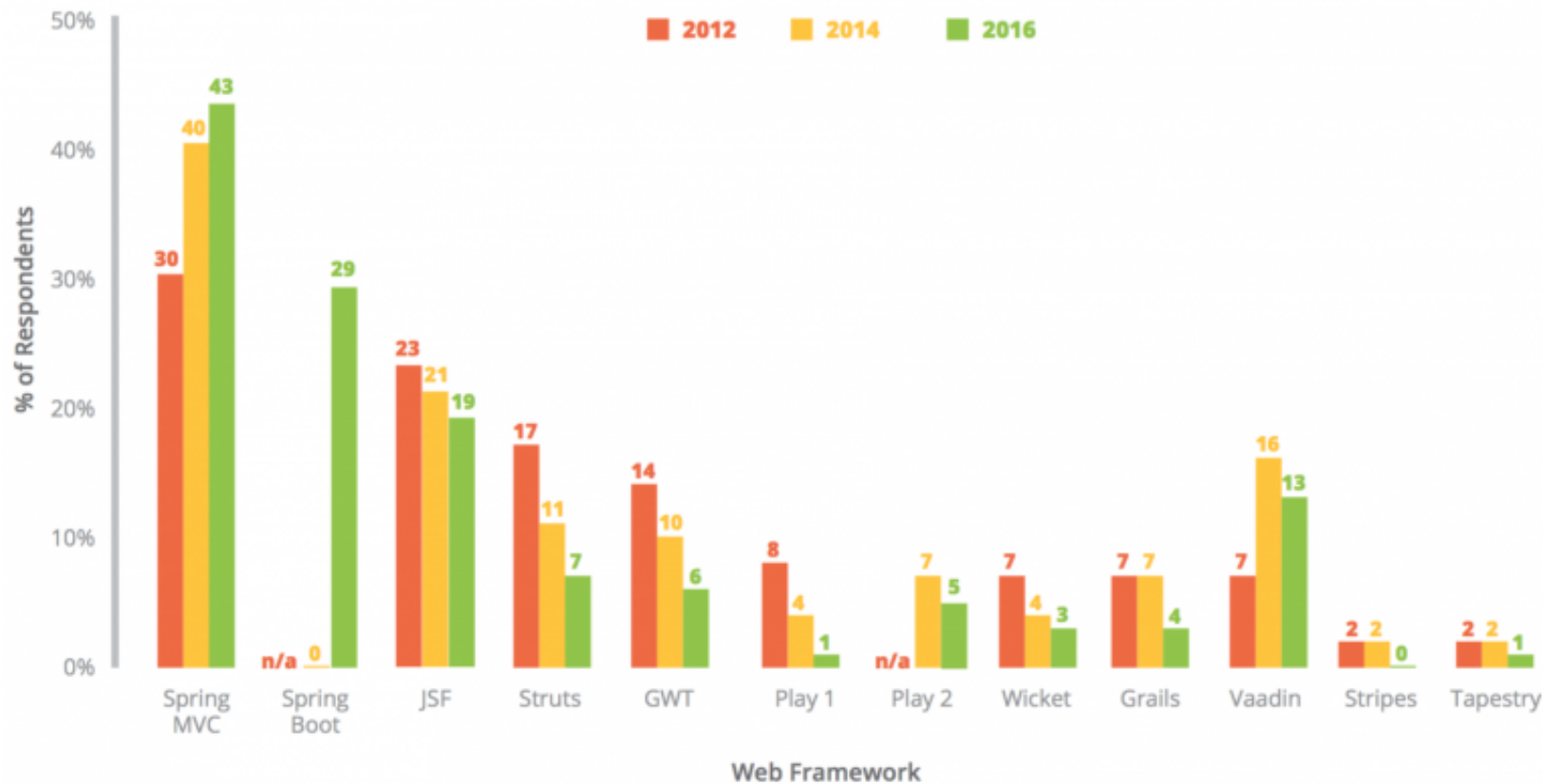
- Redmonk:

– <https://www.infoq.com/news/2016/09/redmonk-java-frameworks>

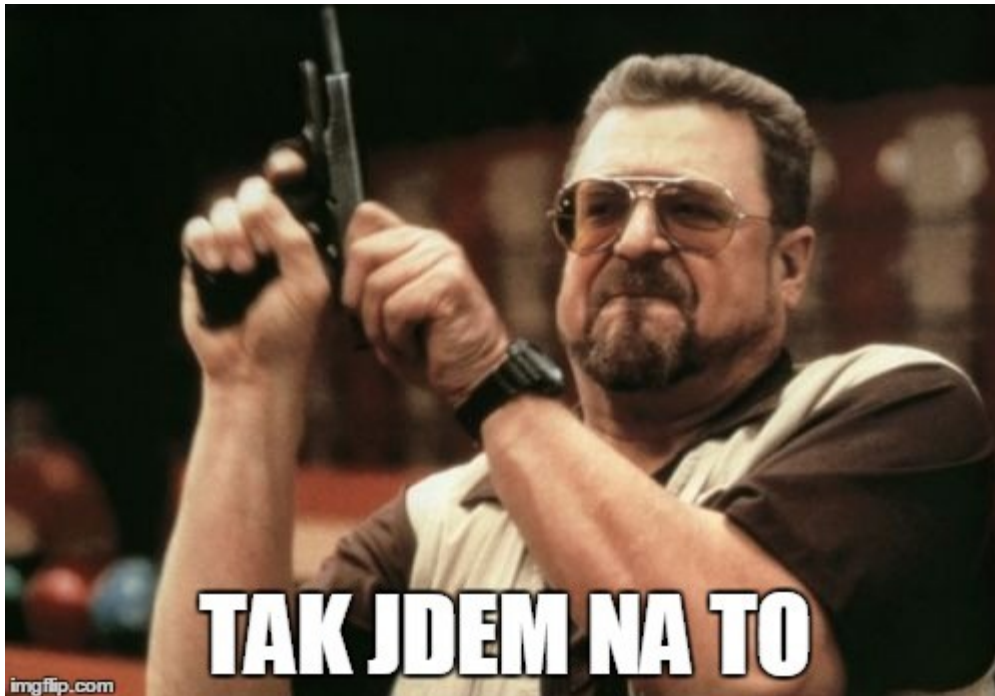


Populární web. frameworky podle RebelLabs

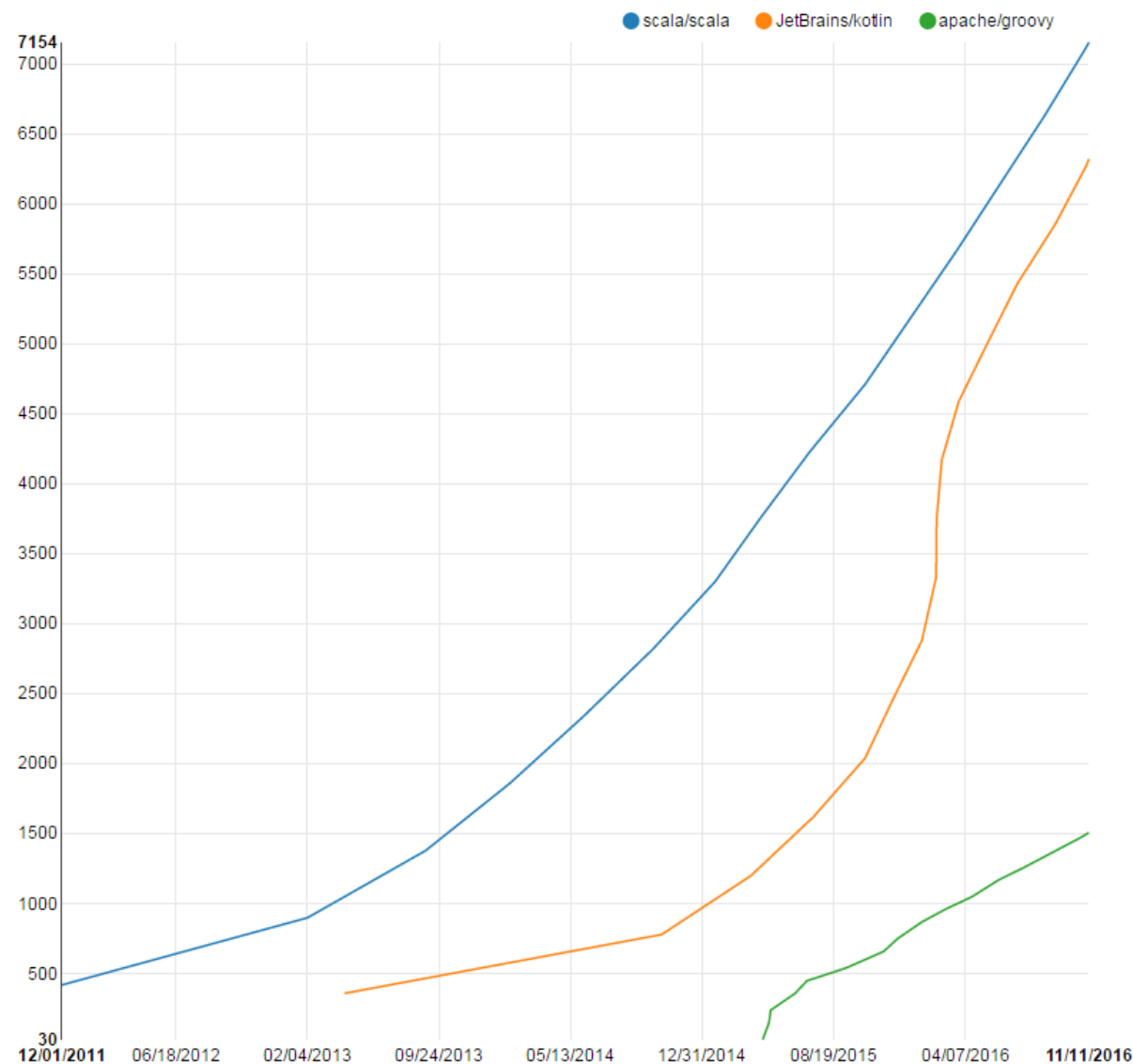
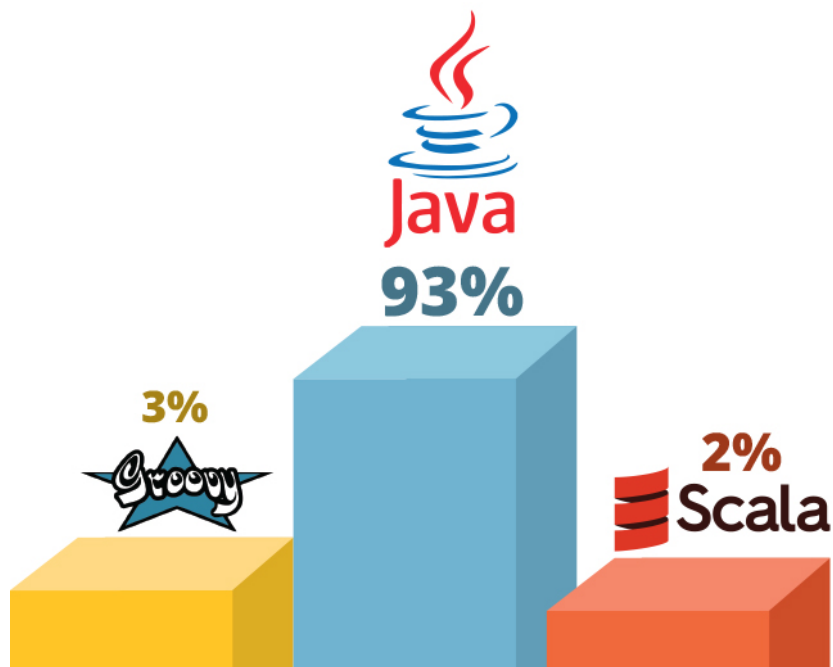
Figure 3.5 Web Framework Usage Since 2012



- GitHub data mining:
 - Java knihovny na GitHubu s největším počtem hvězdiček:
 - <https://github.com/search?l=Java&p=1&q=stars%3A%3E1&s=stars&type=Repositories>



- Java vs. Scala vs. Groovy vs. Kotlin



Proč Scala, Groovy, Kotlin, ... ?

- Mají alternativní jazyky v JVM vůbec význam? ANO, ale nemá význam je porovnávat s Javou a myslet si, že Javu nahradí. Nenahradí.
 - <https://dzone.com/articles/scala-is-dead-long-live-scala> - Scala je super pro funkcionální programování zejména ve spojení s big data (Apache Spark)
 - Kotlin v současnosti našel využití v Androidu, protože umí Lambdy a Android „umí“ Java 8 až od Android 7.0. Obrovský rozdíl Kotlinu oproti Groovy a Scale je jeho prvotřídní podpora v IntelliJ Idea (JetBrains jsou tvůrci Kotlinu).



Proč Scala, Groovy, Kotlin, ... ?

- Groovy je super pro DSL (našlo využití v Gradle, ale tam je postupně nahrazováno Kotlinem). Originální tým (glaforge, blackdrag a melix) byl před dvěma lety zlikvidován Pivotalem a Groovy už moc nevyvíjí, ale pár dalších dobrovolníků se našlo a v současnosti Groovy víceméně přežívá:
 - <https://github.com/apache/groovy/graphs/contributors>
 - <http://melix.github.io/blog/2016/05/gradle-kotlin.html>



Maven vs. Gradle

```

1 <project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
2       xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/maven-v4_0_0.xsd">
3   <modelVersion>4.0.0</modelVersion>
4   <groupId>info.solidsoft.rnd</groupId>
5   <artifactId>spock-10-groovy-24-gradle-maven</artifactId>
6   <version>0.0.1-SNAPSHOT</version>
7   <properties>
8       <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
9       <surefire.version>2.18.1</surefire.version>
10   </properties>
11   <build>
12       <plugins>
13           <plugin>
14               <groupId>org.codehaus.gmavenplus</groupId>
15               <artifactId>gmavenplus-plugin</artifactId>
16               <version>1.4</version>
17               <executions>
18                   <execution>
19                       <goals>
20                           <goal>compile</goal>
21                           <goal>testCompile</goal>
22                       </goals>
23                   </execution>
24               </executions>
25           </plugin>
26           <plugin>
27               <artifactId>maven-surefire-plugin</artifactId>
28               <version>${surefire.version}</version>
29               <configuration>
30                   <includes>
31                       <include>**/*Spec.java</include> <!-- Yes, .java extension -->
32                       <include>**/*Test.java</include> <!-- Just in case having "normal" JUnit tests -->
33                   </includes>
34               </configuration>
35           </plugin>
36       </plugins>
37   </build>
38   <dependencies>
39       <dependency>
40           <groupId>org.codehaus.groovy</groupId>
41           <artifactId>groovy-all</artifactId>
42           <version>2.4.1</version>
43       </dependency>
44       <dependency>
45           <groupId>org.spockframework</groupId>
46           <artifactId>spock-core</artifactId>
47           <version>1.0-groovy-2.4</version>
48           <scope>test</scope>
49       </dependency>
50   </dependencies>
51 </project>
52

```

pom.xml

```

1 apply plugin: 'groovy'
2
3 group = "info.solidsoft.rnd"
4 version = "0.0.1-SNAPSHOT"
5
6 repositories {
7     mavenCentral()
8 }
9
10 dependencies {
11     compile 'org.codehaus.groovy:groovy-all:2.4.1'
12     testCompile 'org.spockframework:spock-core:1.0-groovy-2.4'
13 }
14
15

```

build.gradle

```

1 rootProject.name = 'spock-10-groovy-24-gradle-maven'
2

```

settings.xml

maven

gradle

Days

Maven vs. Gradle II.

- „Trošku složitější“ Gradle build script:
 - <https://github.com/spring-projects/spring-framework/blob/master/build.gradle>



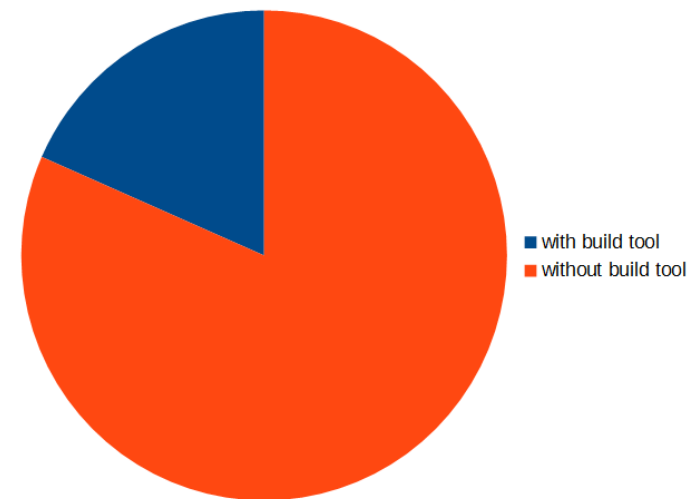
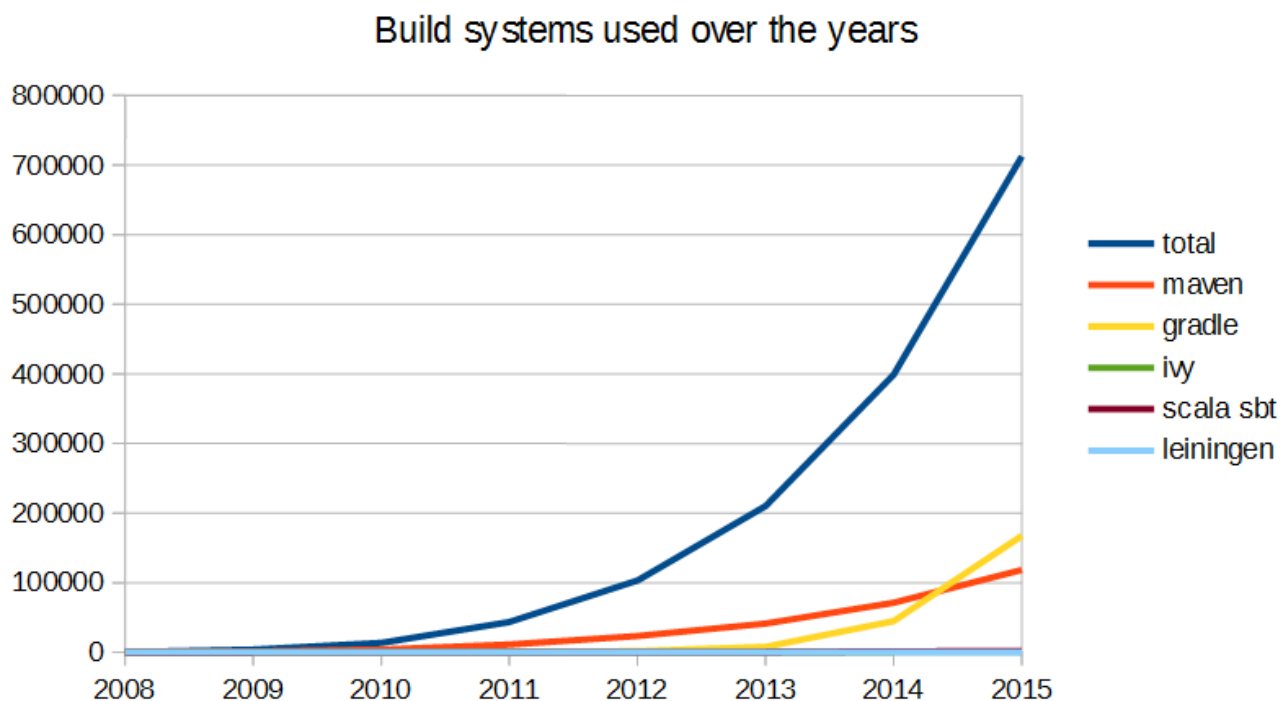
- Maven vs. Gradle

- Gradle vítězí u Android projektů (a projektů se složitým a customizovaným build procesem).
- Maven vítězí všude jinde.



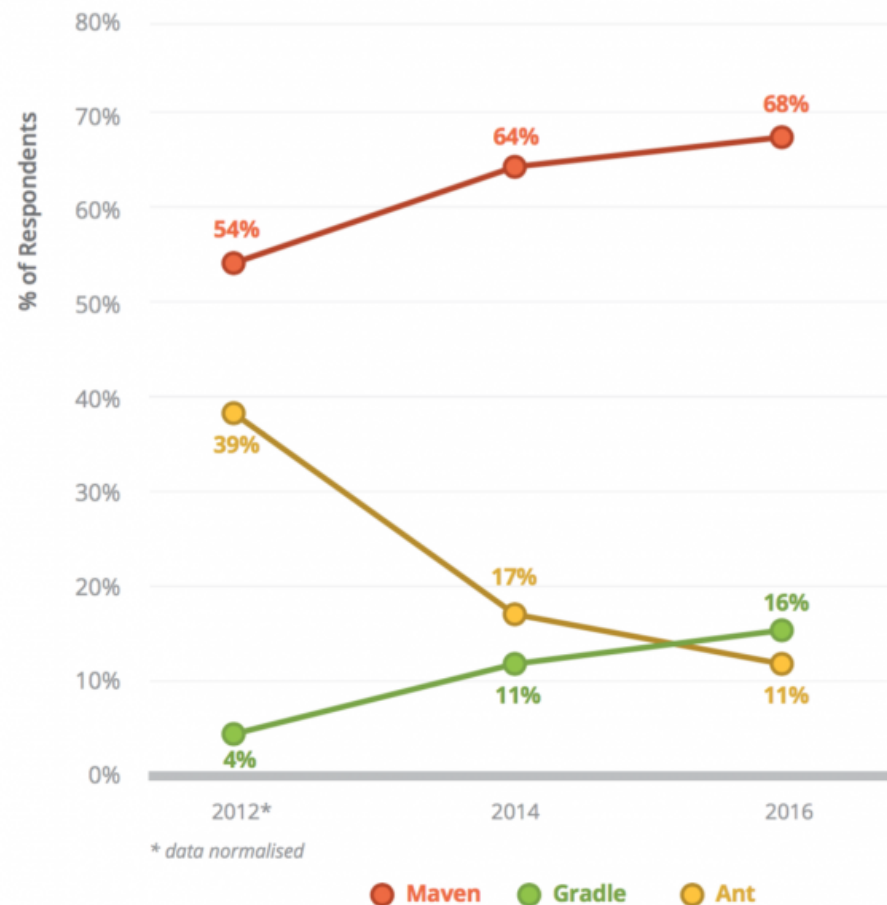
Použití build nástrojů na GitHubu

- Poznámka: graf NEZOBRAZUJE kumulativní součet hodnot, GitHub jenom exponenciálně roste) :-)
 - Zdroj: vlastní data-mining GitHub API



Použití build nástrojů podle RebelLabs

Figure 3.4 Build Tool Usage Since 2012



Poznámka: Řekl bych, že tady nebudou zahrnuti Android vývojáři. RebelLabs report se totiž vytváří dotazníkovým šetřením a k Android vývojářům se nemusí dostat (společnost ZeroTurnaround – tvůrce nástroje JRebel znají zejména Java EE vývojáři).



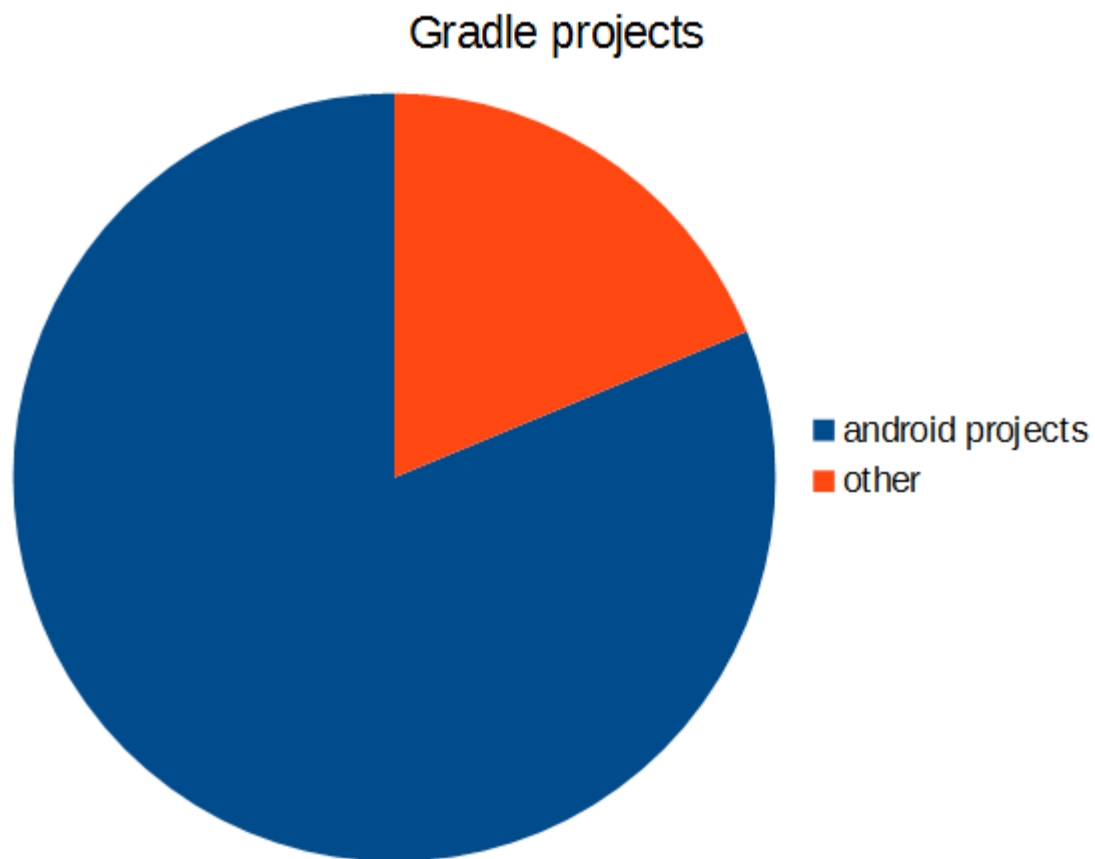
Projekty bez build nástrojů

- V čem jsou projekty bez build nástrojů naprogramovány?
 - Zdroj: vlastní data-mining GitHub API
 - Idea: 330 052
 - Eclipse: 290 236
 - NetBeans: 83 721



Kde se používá Gradle?

- Převážně na Androidu :-)



GitHub alternatives

- Spoustu statistik mám z GitHubu, protože je to zdaleka největší online služba pro publikování a sdílení aplikací. A také ho extenzivně využívám. Nicméně existuje pár alternativ:
 - <https://bitbucket.org/> - umí neomezený počet privátních repozitářů
 - <https://gitlab.com/> - takový lepší GitHub i BitBucket ... ale moc se nepoužívá

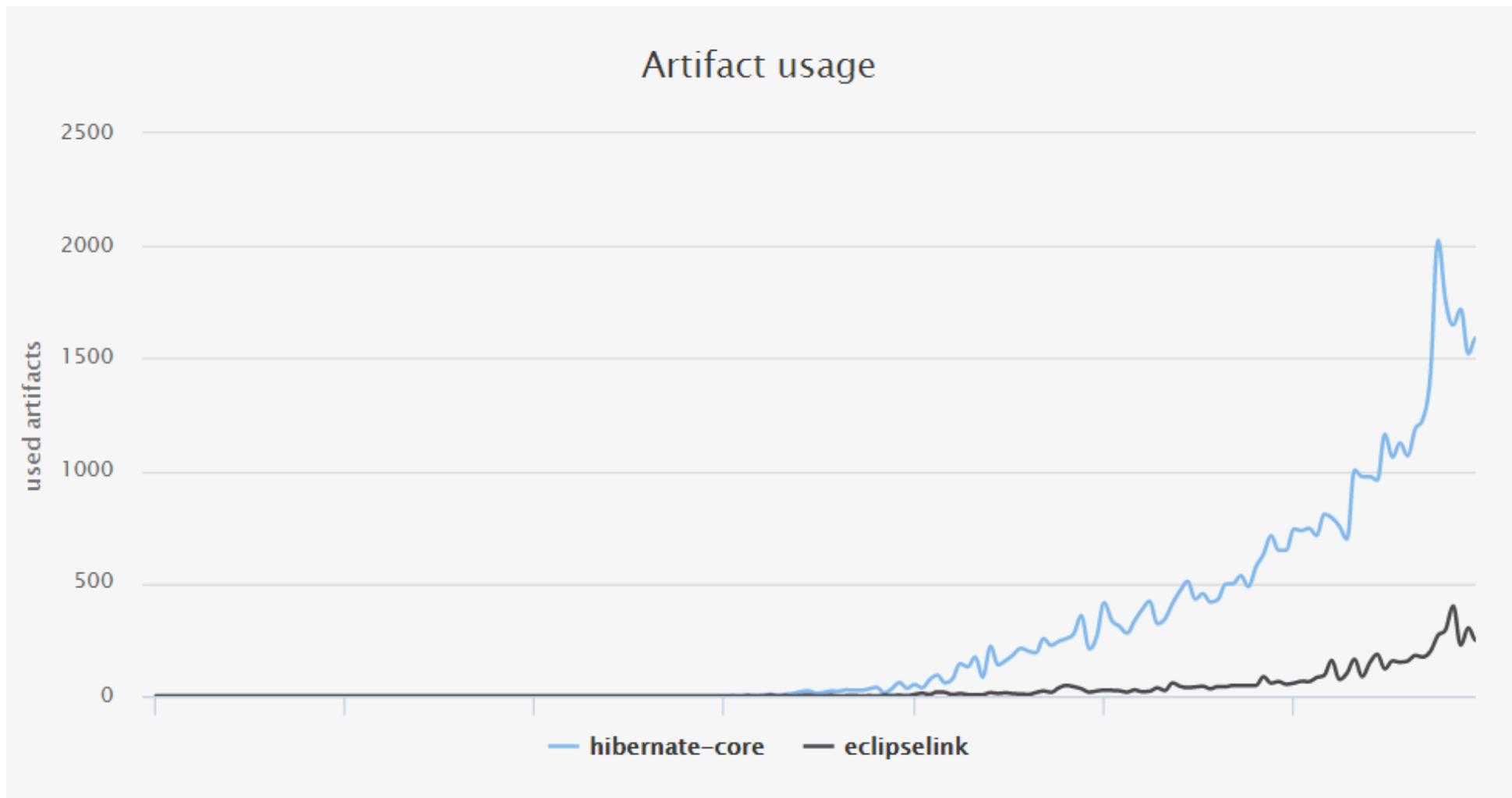


Hibernate vs. MyBatis vs. XYZ

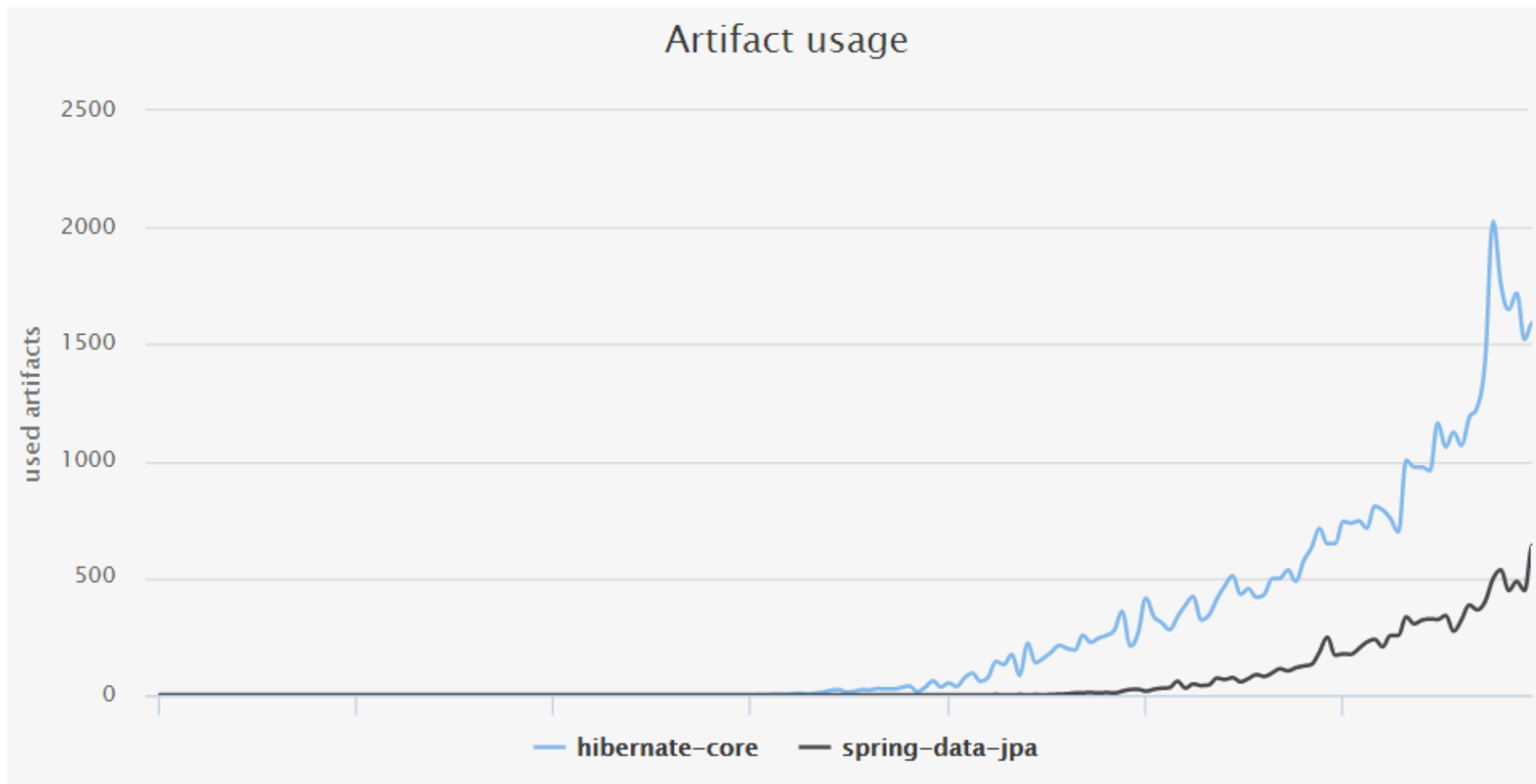
- Nejpoužívanější frameworky pro práci s databází:
 - Hibernate, Spring Data JPA
 - MyBatis
 - JOOQ
 - QueryDSL
 - Třída JdbcTemplate ze Springu
 - Následující statistiky jsou získané převážně z GitHubu a NEJEDNÁ se o kumulativní součet:
<https://javalibs.com/custom-chart>



Hibernate vs. EclipseLink

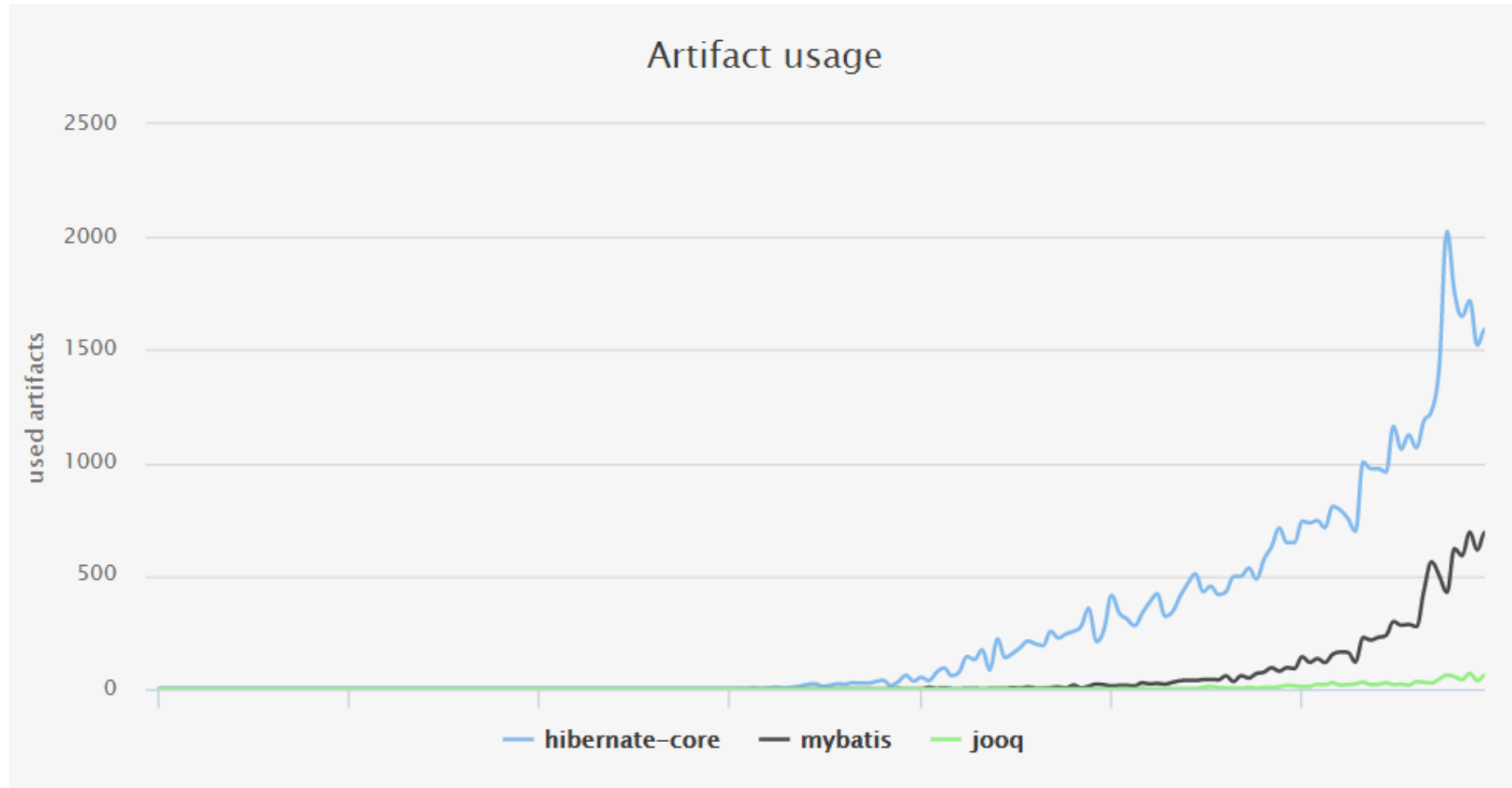


Hibernate vs. Spring Data JPA

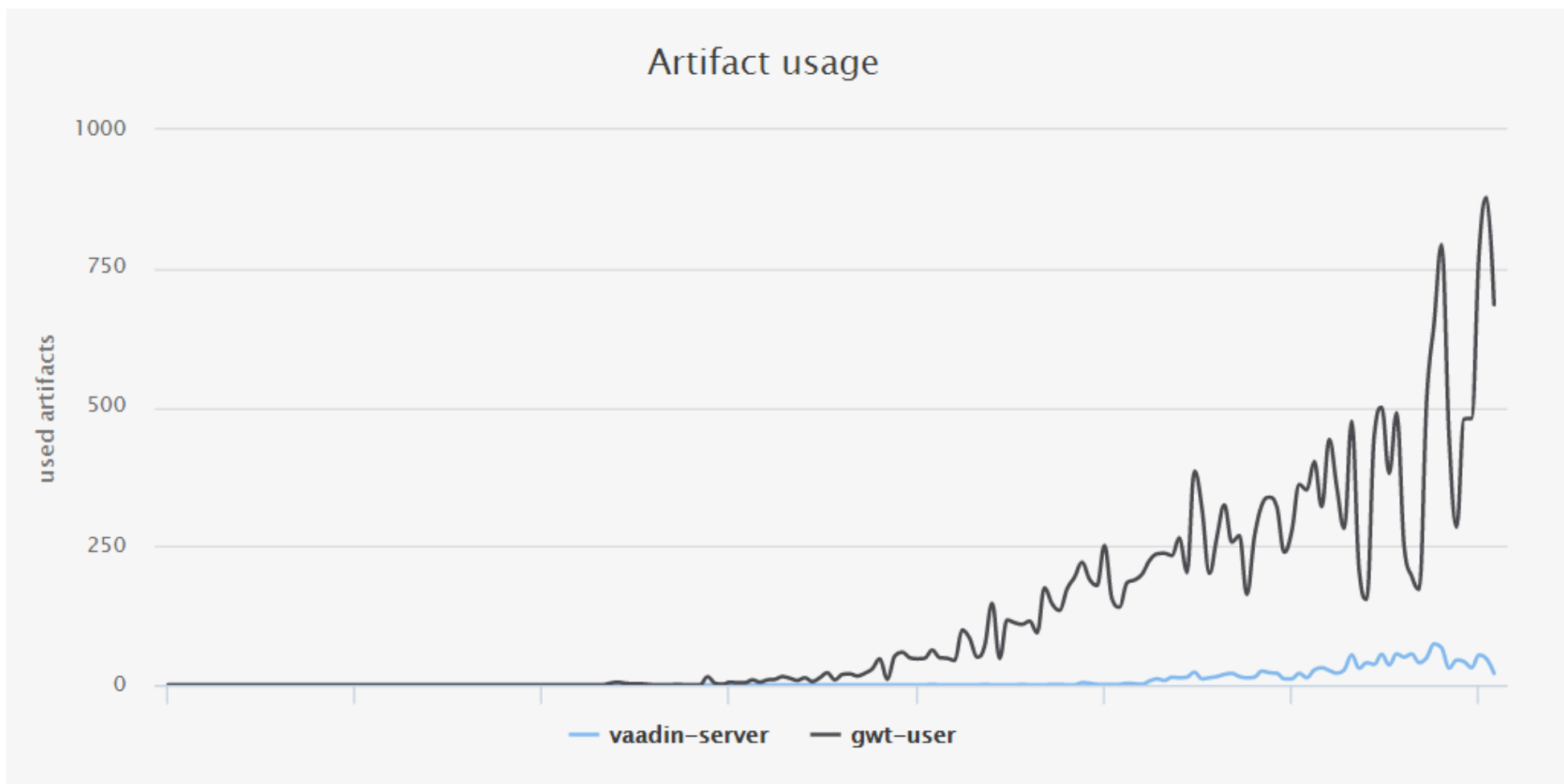


Days

Hibernate vs. MyBatis vs. JOOQ

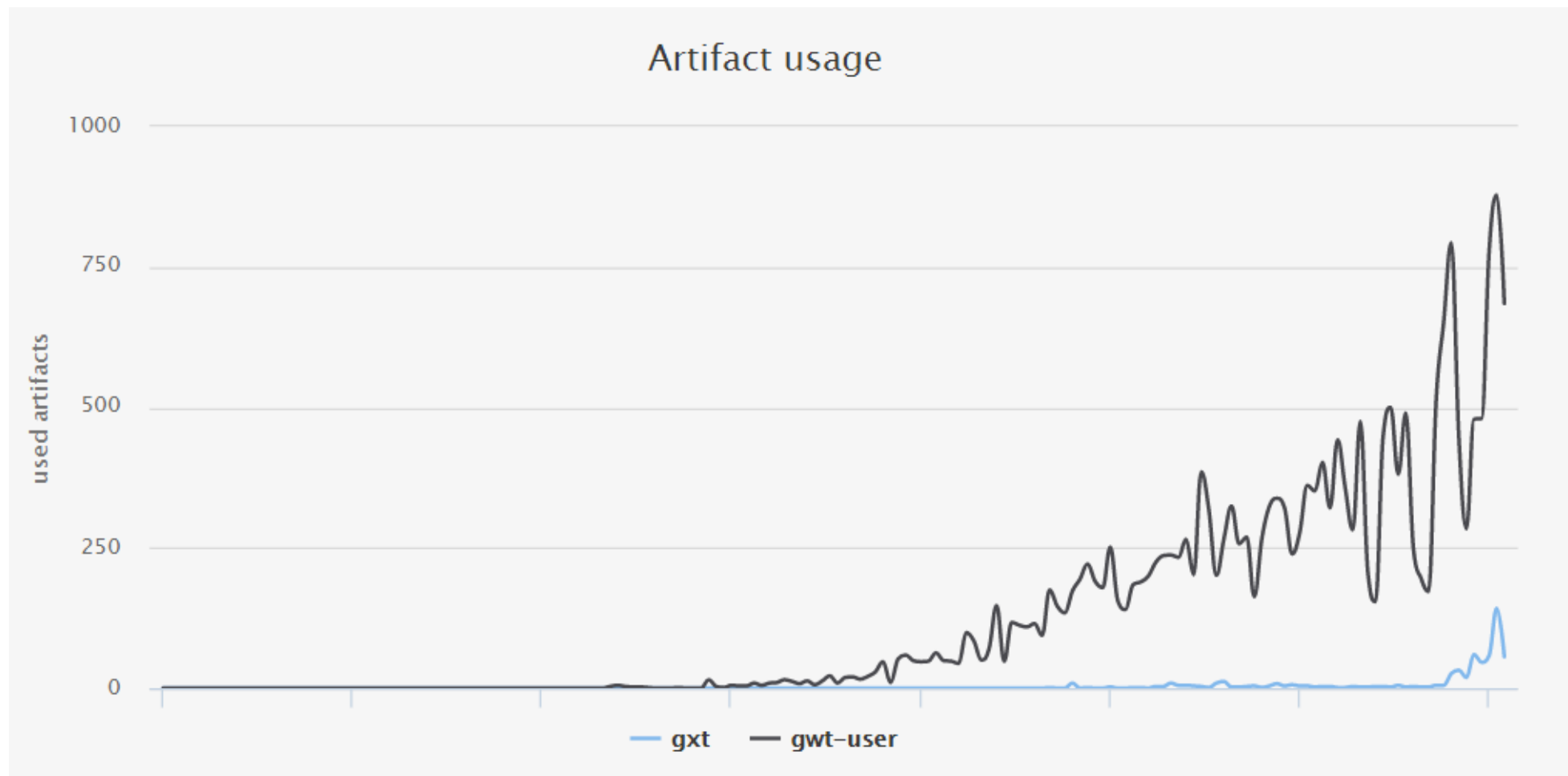


Vaadin vs. GWT



Days

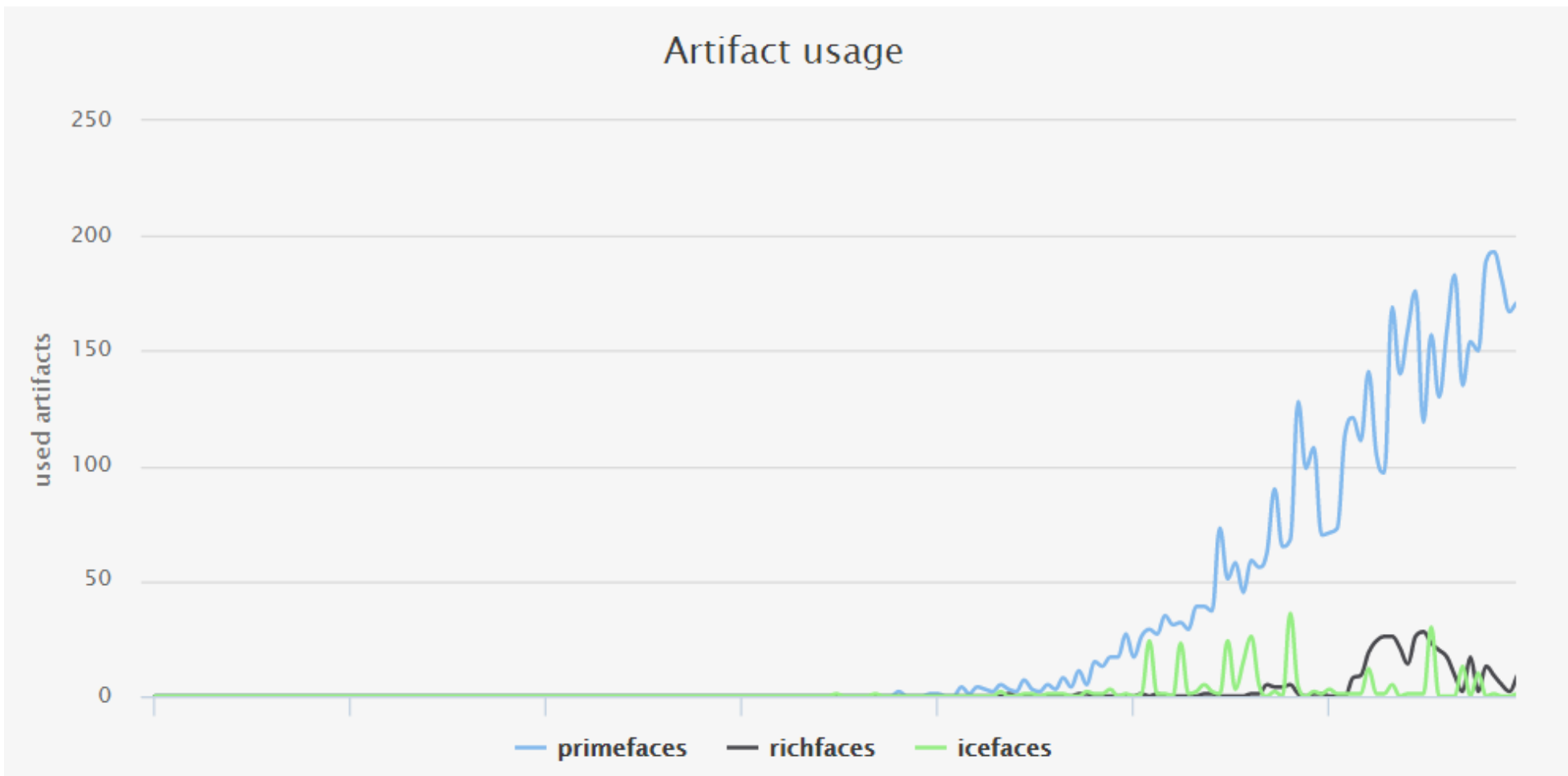
Sencha vs. GWT



Days

*Faces

Poznámka: RichFaces jsou oficiálně mrtvé,
kdyby o tom náhodou někdo nevěděl :-)



Days

Další porovnání

- Kupa zajímavých informací je zde:
 - <https://javalibs.com/charts>



Rust Central Repository I.

All artifacts:

GAV count: 1,650,497

GA count: 164,600

Maven plugins:

GAV count: 34,542

GA count: 3,452

Maven archetypes:

GAV count: 17,217

GA count: 1,216

GAV annual growth:

2015: +41%

2014: +39%

2013: +40%

2012: +42%

2011: +53%

2010: +84%

2009: +90%

2008: +86%

2007: +74%

2006: +92%

GA annual growth:

2015: +35%

2014: +35%

2013: +35%

2012: +42%

2011: +40%

2010: +51%

2009: +39%

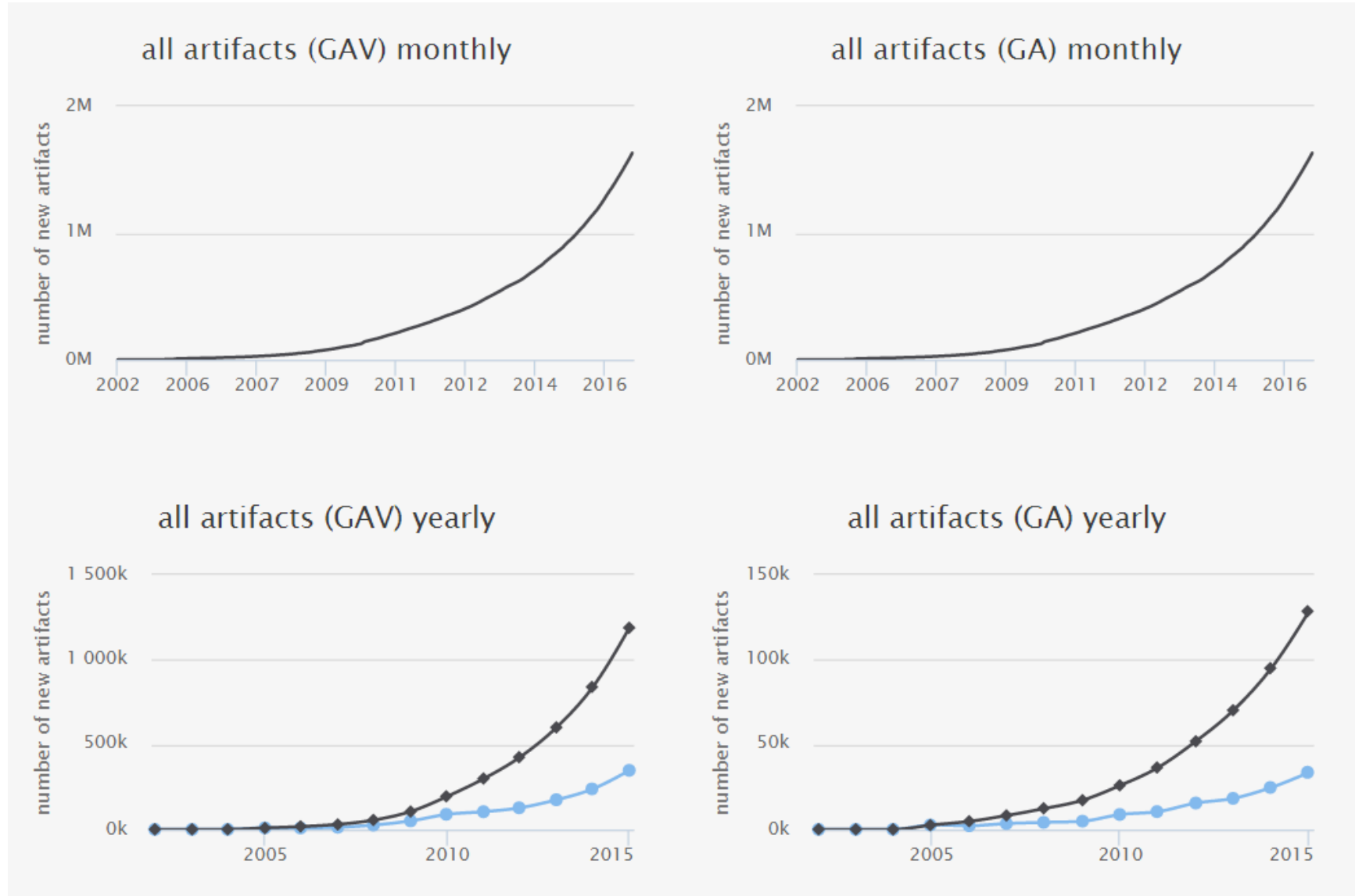
2008: +50%

2007: +73%

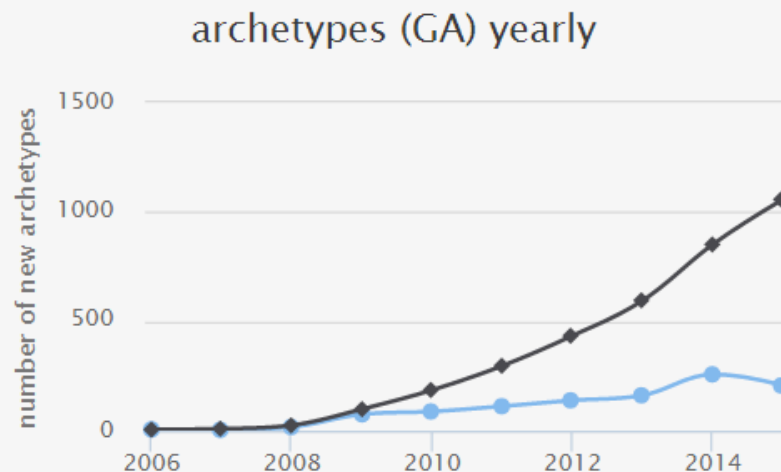
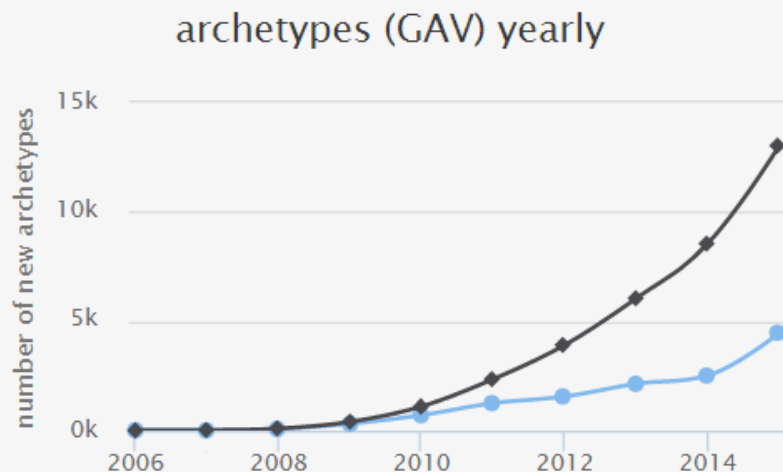
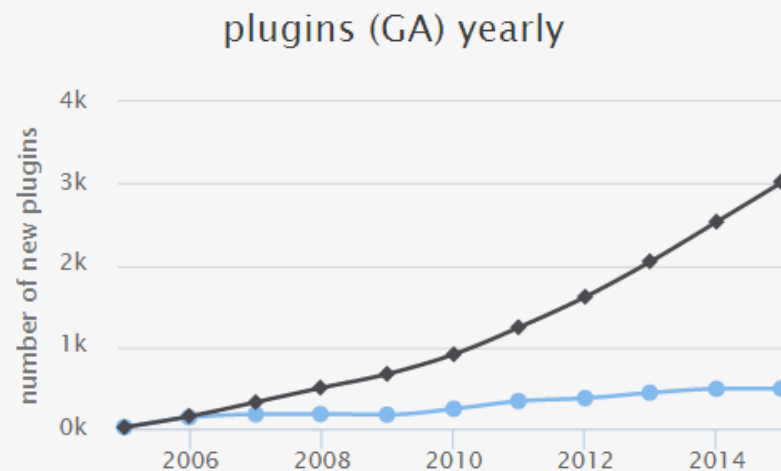
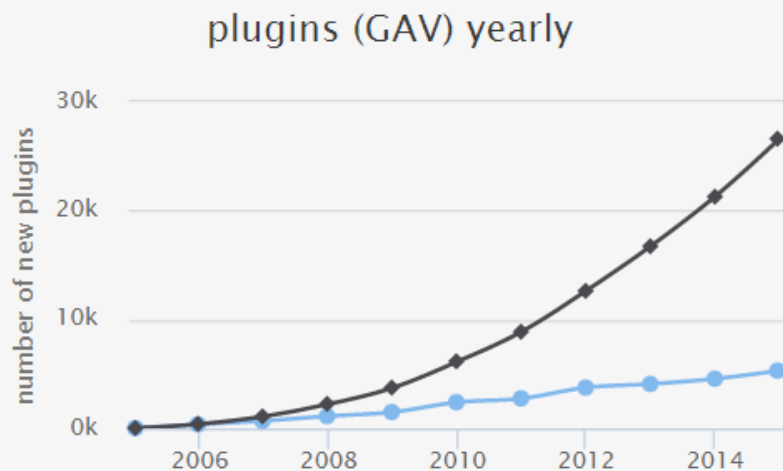
2006: +82%

 **Days**

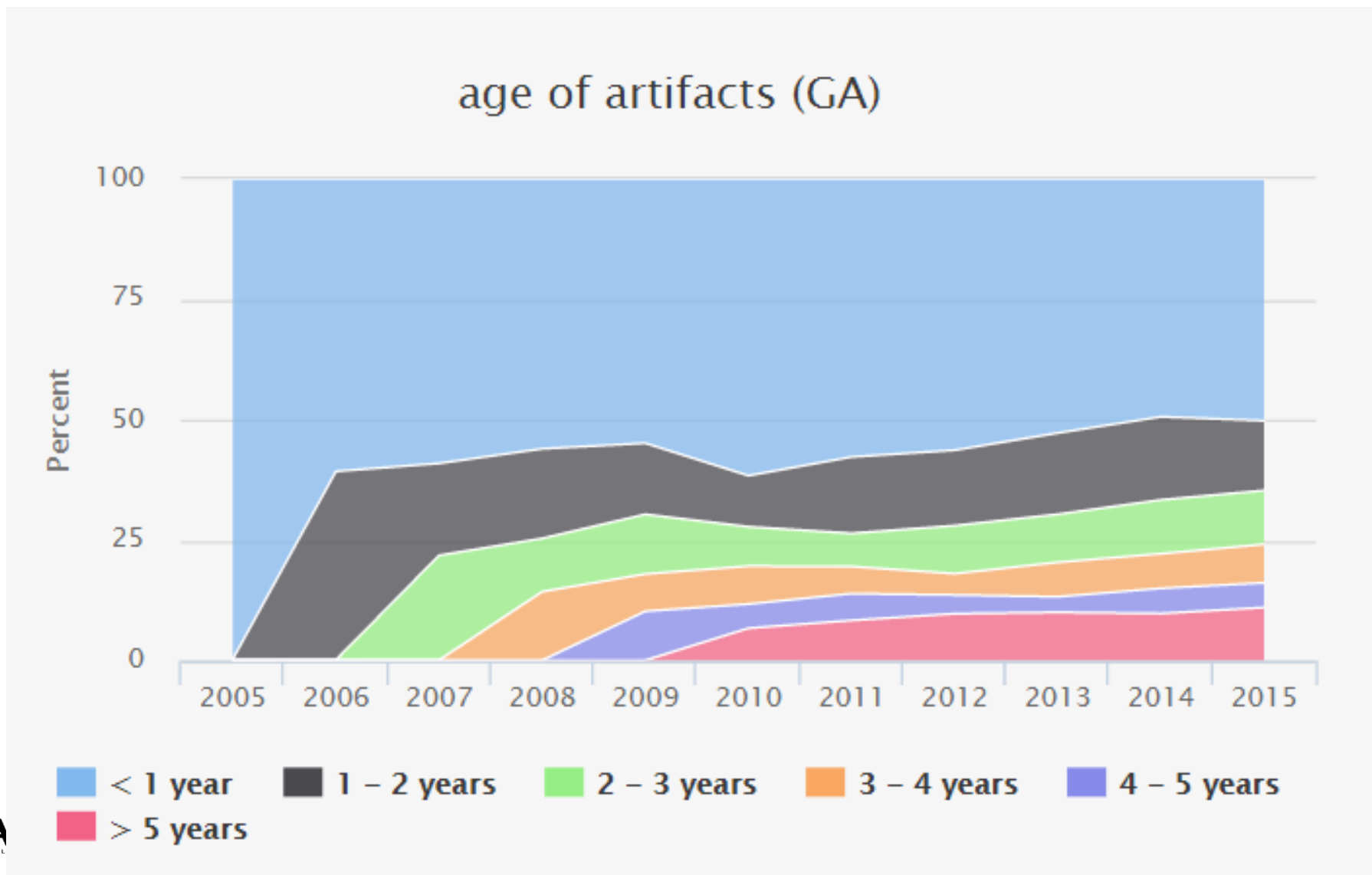
Rust Central Repository II.



Rust Central Repository III.



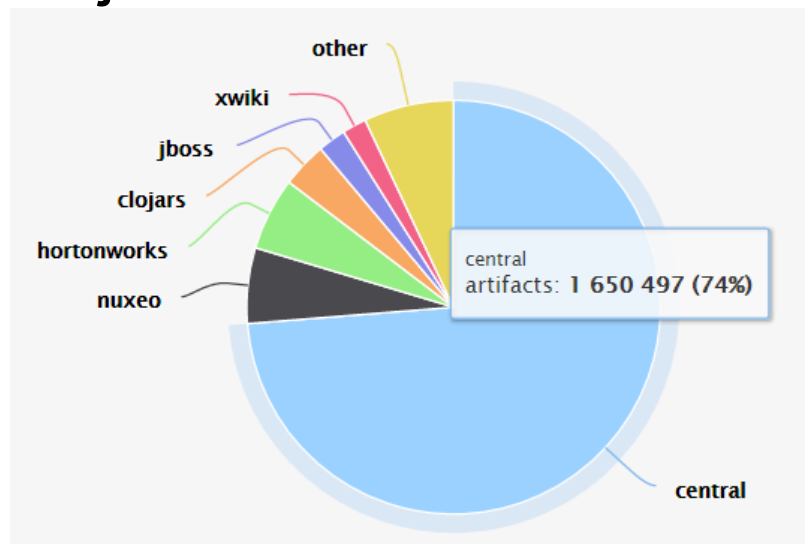
Rust Central Repository IV.



Days

Další repozitáře

- Aby to nebylo tak jednoduché, tak Central Repository není jediný repozitář (byl jenom prvním a je zároveň největší). Dalším známým repozitářem je jcenter od společnosti JFrog (mají svůj server Artifactory). Jcenter sdružuje řadu repozitářů včetně Centralu a pár knihoven je pouze v jcenter.



Nepoužívejte LATEST

- V pom.xml je možné definovat poslední verzi pomocí názvu verze LATEST. Je to deprecated a je nedoporučované abyste to používali a přesto to lidi používají.
- Proč to nemáte používat?
 - Váš build se může lehce rozbít při vydání nové verze knihovny.
 - LATEST nemusí znamenat „Latest“. Řazení verzí je definováno docela složitým algoritmem, který je postaven na tom, že budete dodržovat verzování ve stylu MAJOR.MINOR.REVISION.BUILDNUMBER
 - Jenže občas to někdo přece jenom nedodrží.
 - Například:
 - <https://javalibs.com/artifact/io.spring.platform/platform-bom>



Nepoužívejte vyhledávač v Idea / Eclipse

- V pom.xml můžete přímo z IntelliJ Idea / Eclipse vyhledávat knihovny. Tento mechanismus ale nefunguje tak, jak byste očekávali. Central Repository jednou za týden (obvykle neděle večer) zveřejňuje Lucene index se všemi verzemi všech knihoven k danému datu. A navíc to občas neprovedou vůbec a ta prodleva je třeba i měsíc. A pak vám Idea / Eclipse nabízí staré verze knihoven celý měsíc.
- Co používat? Když přesně více co hledáte, pak:
 - <http://search.maven.org>
- Jinak:
 - <http://mvnrepository.com>
 - NEBO
 - <http://javalibs.com>



Děkuji za pozornost.

www.JavaDays.cz

www.gopas.cz