

COMPARING JVM WEB FRAMEWORKS



Matt Raible

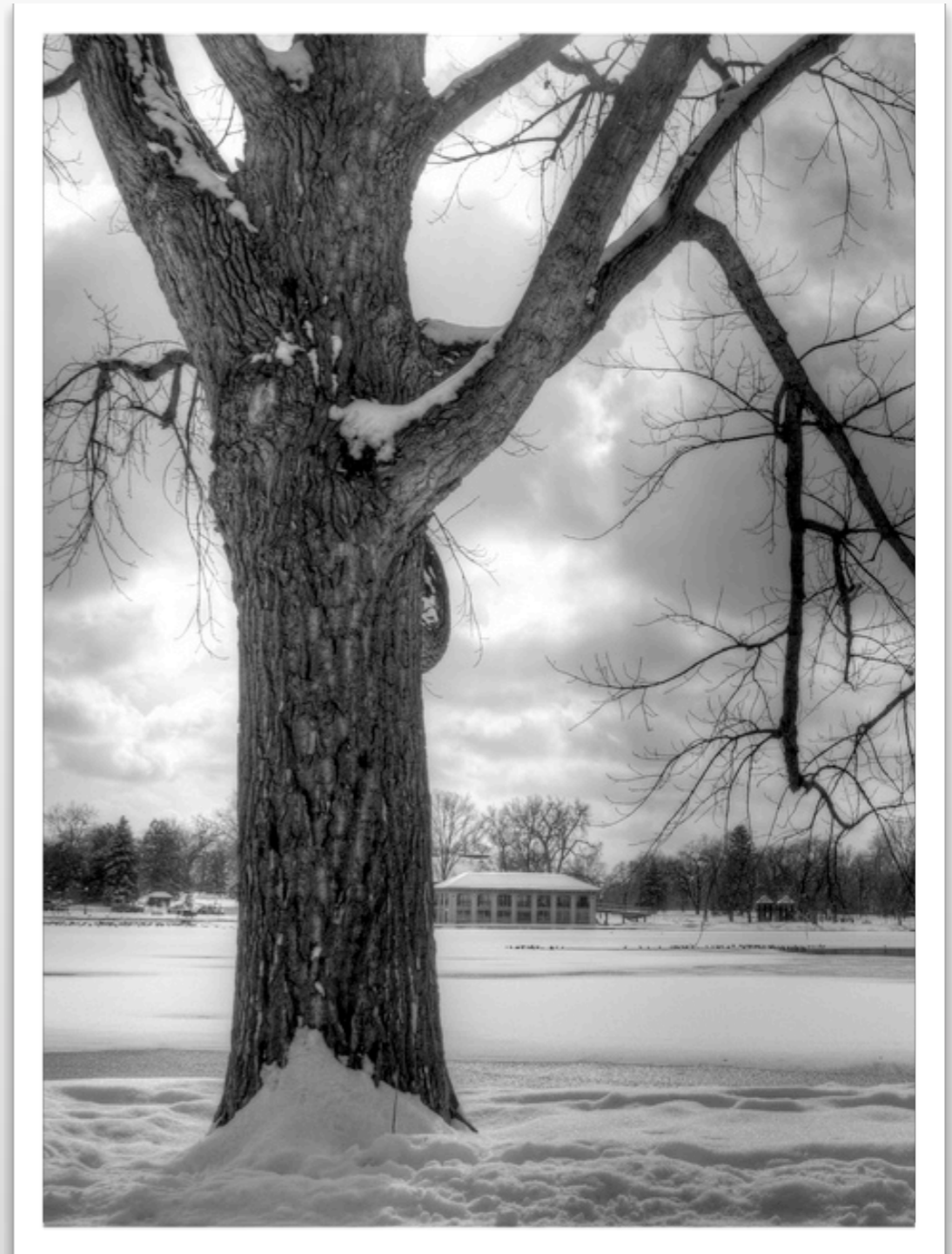
<http://raibledesigns.com>

Photos by **Trish** - <http://mcginityphoto.com>

Introductions



- ▶ Your experience with web development?
- ▶ Your experience with Java EE development?
- ▶ What do you want to get from this session?
- ▶ Experience with Grails, GWT, Rails, Spring MVC, Wicket, Tapestry or Play?



A scenic landscape at sunrise or sunset. The sun is low on the horizon, partially obscured by a mountain peak, creating a bright glow and lens flare. The sky is a gradient of blue and orange. In the foreground, a calm lake reflects the sun and the surrounding mountains. The overall mood is peaceful and serene.

Matt

Raible

Clients and Open Source



Struts²



AppFuse

Roller







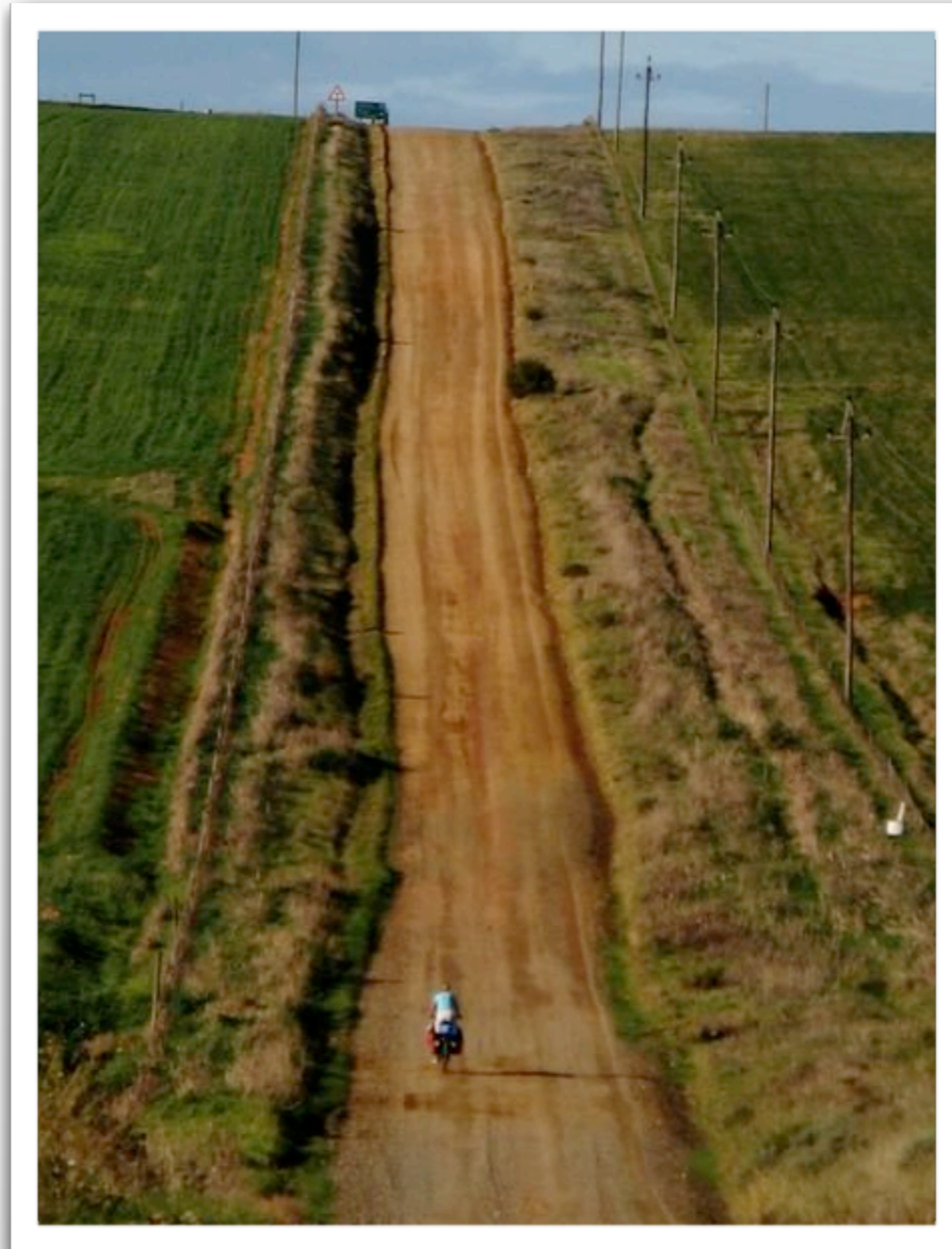
<http://flickr.com/photos/leecullivan/122271605/>



<http://flickr.com/photos/crowleymr/2530170585/>



<http://www.flickr.com/photos/mraible/2644737051/>



<http://www.travelblog.org/Photos/1597321.html>







VW Trends
www.vwtrendsweb.com



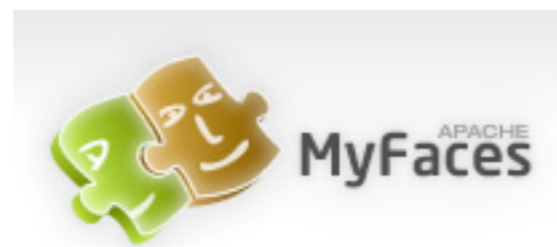
Session Agenda



- ▶ The Problem with Web Frameworks
- ▶ The Candidates
- ▶ Comparison Points
- ▶ The Matrix
- ▶ Conclusion
- ▶ Q and A



The Problem



The Real Problem



The Real Problem



The Real Problem



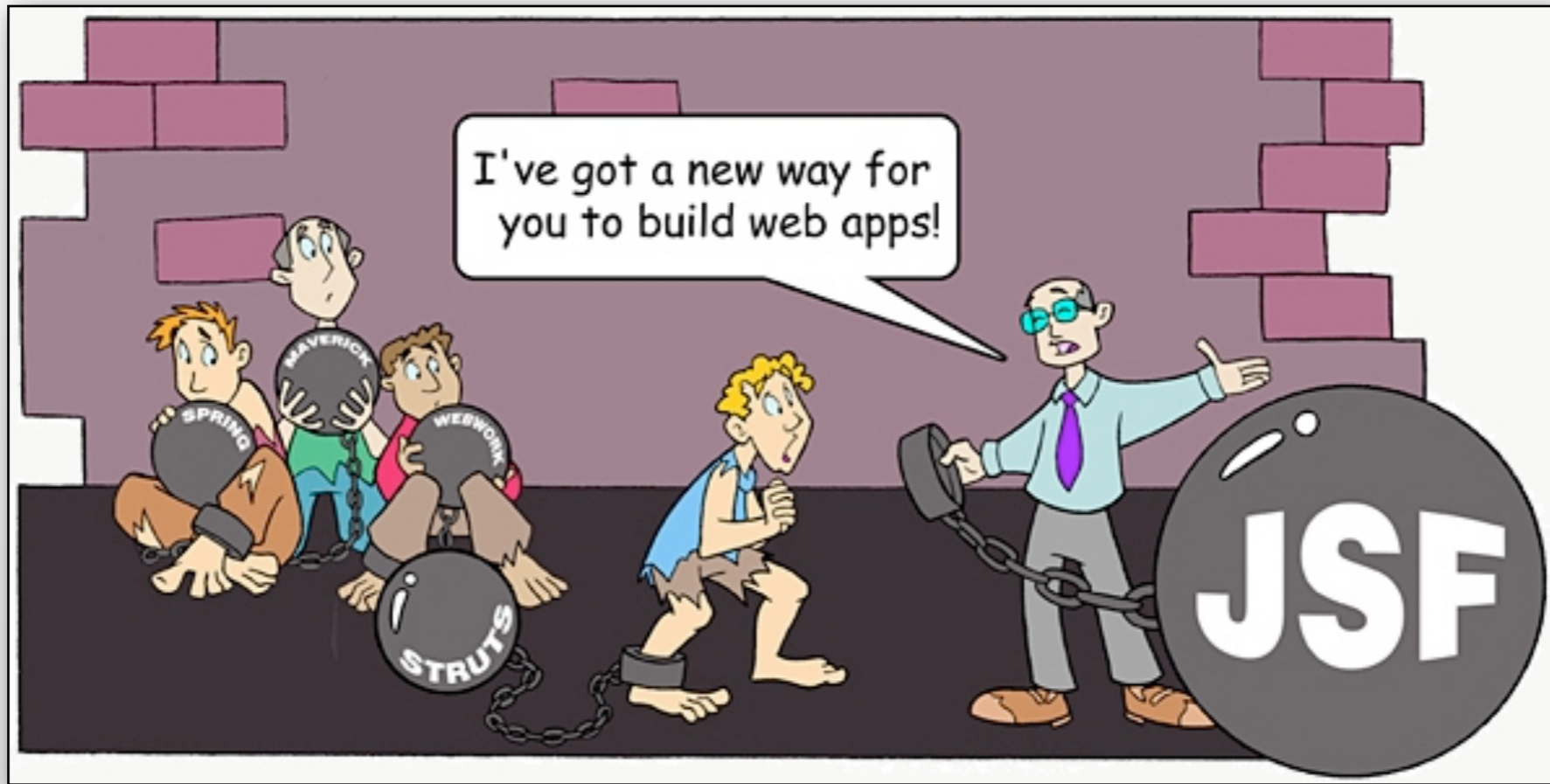
How do you choose?





Eliminate, Don't Include





Craig McClanahan on Rails



... while I'd *love* to see life made simpler for Java web developers, and a lot of the things happening in Struts2 are going that way -- it won't be me doing it.

I've gone over to the dark side :-)) and **much prefer to develop in Rails** -- for the conciseness mentioned above, but also **because I don't ever have to do a "build" or "deploy" step** during my development cycle any more. But you guys and gals need to be reminded that *this* is the kind of thing you are competing against if you expect to attract Rails developers ... or to avoid even more "previously Java web developer" defectors like me :-)).

-- Craig McClanahan, 10/23/2007

<http://markmail.org/thread/qfb5sekad33eobh2>

James Gosling on JSF

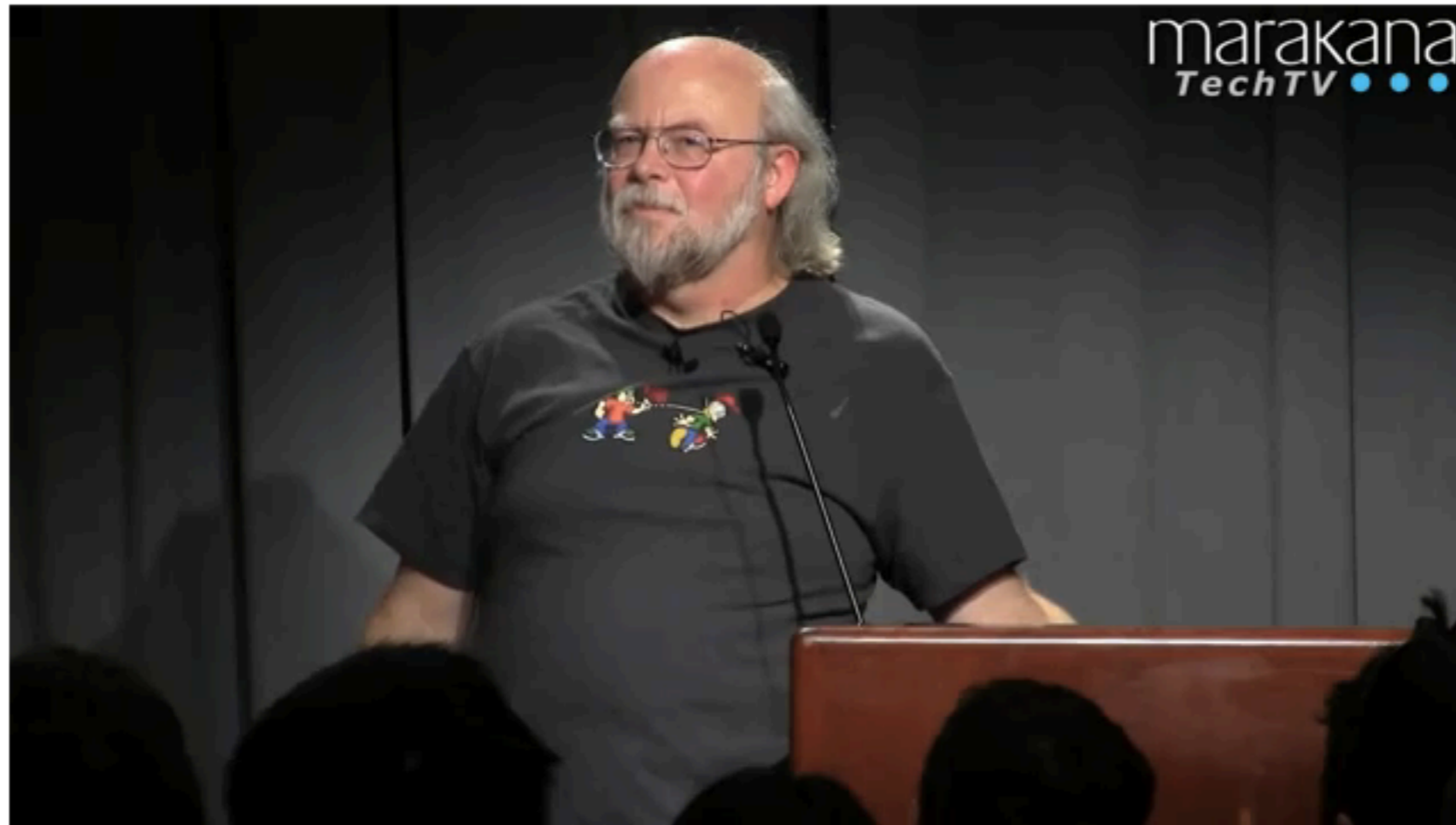


James Gosling on Apple, Apache, Google, Oracle and the Future of Java

UserGroupsatGoogle

62 videos

Subscribe



<http://www.youtube.com/watch?v=9ei-rbULWoA#t=47m>

2010: Comparison Points



- ▶ Developer Productivity
- ▶ Developer Perception
- ▶ Learning Curve
- ▶ Project Health
- ▶ Developer Availability
- ▶ Job Trends



2010: Comparison Points



- ▶ Templating
- ▶ Components
- ▶ Ajax
- ▶ Plugins or Add-Ons
- ▶ Scalability
- ▶ Testing Support



2010: Comparison Points

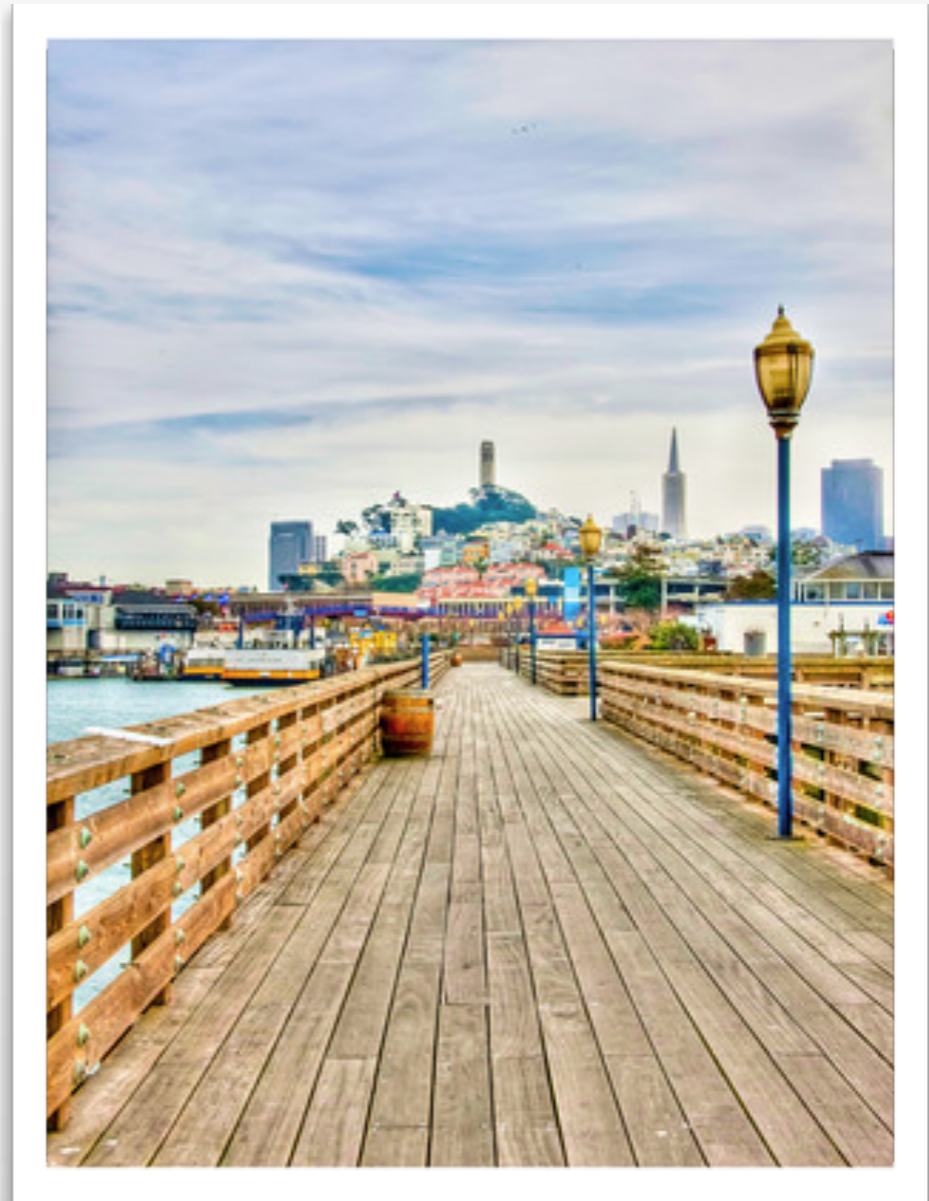


- ▶ i18n and l10n
- ▶ Validation
- ▶ Multi-language Support (Groovy / Scala)
- ▶ Quality of Documentation/Tutorials
- ▶ Books Published
- ▶ REST Support (client and server)

2010: Comparison Points



- ▶ Mobile / iPhone Support
- ▶ Degree of Risk



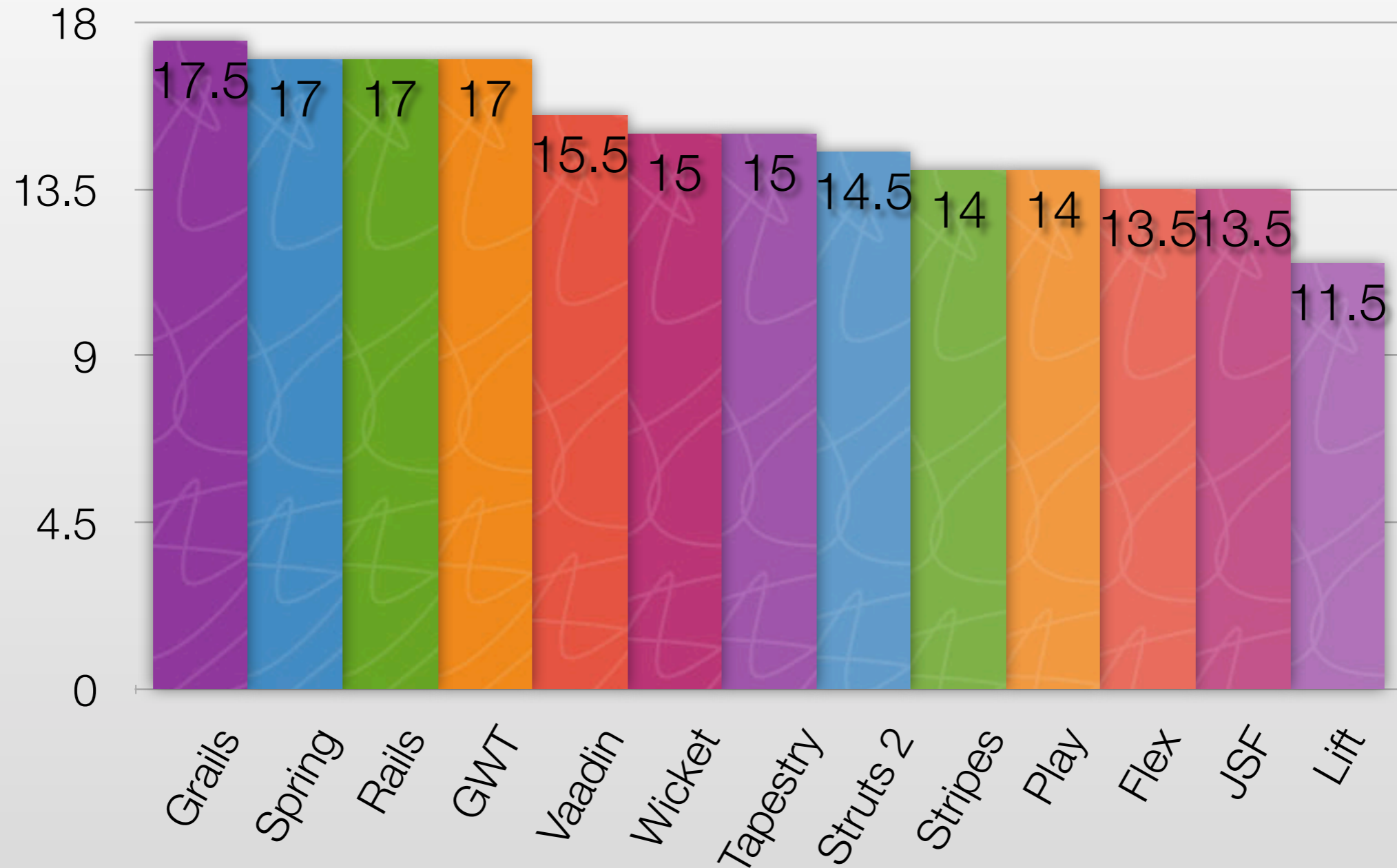
Comparison Matrix



Criteria	Struts 2	Spring MVC	Wicket	JSF 2	Tapestry	Stripes	GWT	Grails	Rails	Flex	Vaadin	Lift	Play
Developer Productivity	0.50	0.50	0.50	0.50	1.00	0.50	1.00	1.00	1.00	0.00	1.00	0.50	1.00
Developer Perception	0.50	1.00	1.00	0.00	0.50	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Learning Curve	1.00	1.00	0.50	0.50	0.50	1.00	1.00	1.00	1.00	1.00	1.00	0.50	1.00
Project Health	0.50	1.00	1.00	1.00	0.50	0.50	1.00	1.00	1.00	0.50	1.00	1.00	1.00
Developer Availability	0.50	1.00	0.50	1.00	1.00	0.50	1.00	0.50	1.00	1.00	0.50	0.00	0.50
Job Trends	1.00	1.00	0.50	1.00	0.50	0.00	1.00	0.50	1.00	1.00	0.00	0.00	0.50
Templating	1.00	1.00	1.00	0.50	1.00	1.00	0.50	1.00	1.00	0.50	0.50	0.50	0.50
Components	0.00	0.00	1.00	1.00	1.00	0.00	0.50	0.50	0.50	1.00	1.00	0.00	0.00
Ajax	0.50	1.00	0.50	0.50	0.50	0.50	1.00	0.50	0.50	0.50	1.00	1.00	0.50
Plugins or Add-Ons	0.50	0.00	1.00	1.00	0.50	0.00	1.00	1.00	1.00	1.00	1.00	0.50	1.00
Scalability	1.00	1.00	0.50	0.50	0.50	1.00	1.00	0.50	0.50	0.50	0.50	1.00	1.00
Testing	1.00	1.00	0.50	0.50	1.00	1.00	0.50	1.00	1.00	0.00	0.50	0.50	1.00
i18n and l10n	1.00	1.00	1.00	0.50	1.00	1.00	1.00	1.00	0.50	0.50	1.00	1.00	1.00
Validation	1.00	1.00	1.00	0.50	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.50	0.50
Multi-language Support (Groovy / Scala)	0.50	0.50	1.00	1.00	1.00	1.00	0.00	1.00	0.00	0.00	1.00	0.00	0.50
Quality of Documentation/Tutorials	0.50	1.00	0.50	0.50	0.50	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Books Published	1.00	1.00	0.50	1.00	0.50	0.50	1.00	1.00	1.00	1.00	0.50	0.50	0.00
REST Support (client and server)	0.50	1.00	0.50	0.00	0.50	0.50	0.50	1.00	1.00	0.50	0.50	0.50	0.50
Mobile / iPhone Support	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.50	1.00	1.00	1.00
Degree of Risk	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.50	0.50	0.50
Totals	14.5	17	15	13.5	15	14	17	17.5	17	13.5	15.5	11.5	14

<http://bit.ly/jvm-frameworks-matrix>

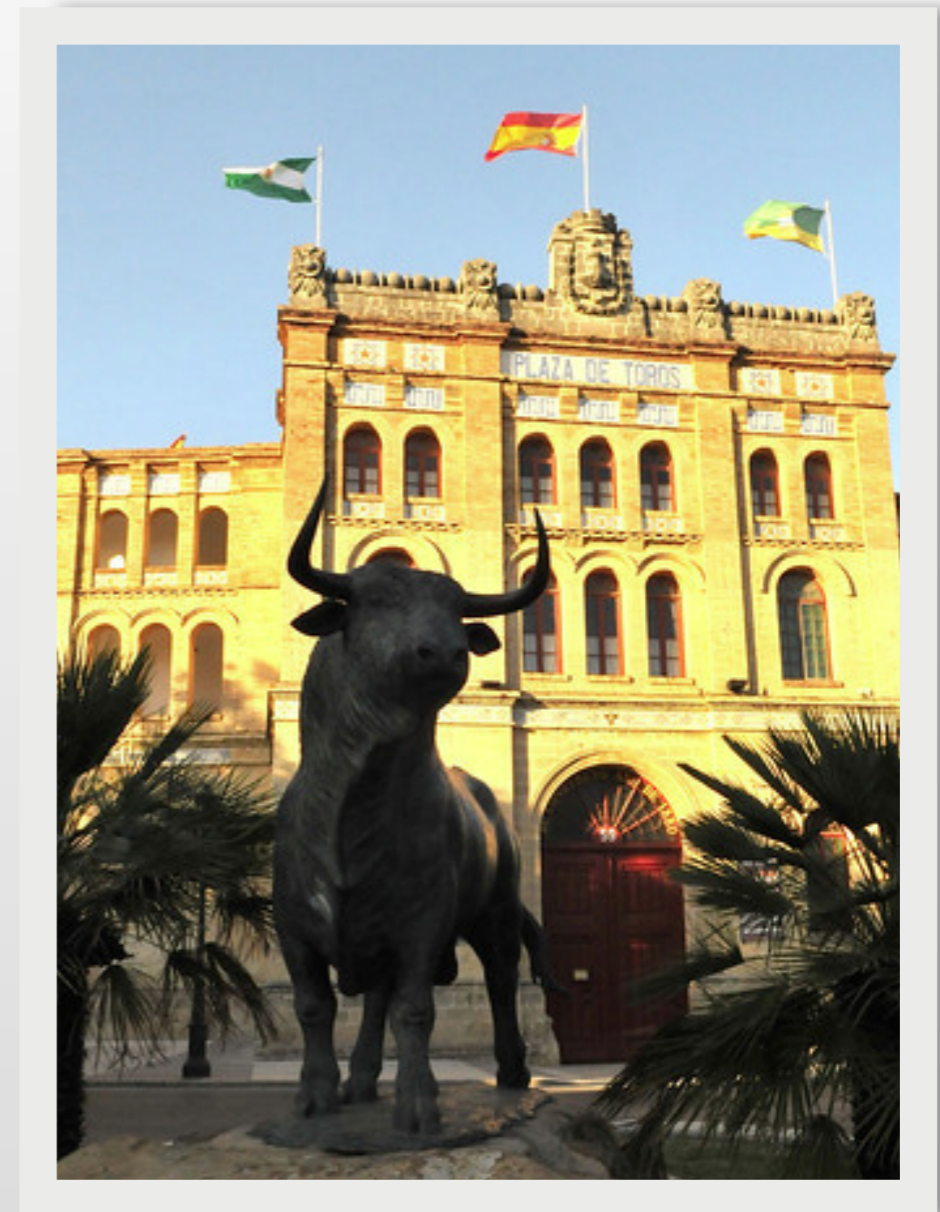
Matrix Results



Matrix Results



- ▶ Grails (17.5)
- ▶ GWT (17)
- ▶ Ruby on Rails (17)
- ▶ Spring MVC (17)
- ▶ Vaadin (15.5)
- ▶ Tapestry and Wicket (15)



Weighted Matrix



Weight	Criteria	Struts 2	Spring MVC	Wicket	JSF	Tapestry	Stripes	GWT	Grails	Rails	Flex	Vaadin	Lift	Play
10	Developer Productivity	5.00	5.00	5.00	5.00	10.00	5.00	10.00	10.00	10.00	0.00	10.00	5.00	10.00
0	Developer Perception	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
5	Learning Curve	5.00	5.00	2.50	2.50	2.50	5.00	5.00	5.00	5.00	5.00	5.00	2.50	5.00
5	Project Health	2.50	5.00	5.00	5.00	2.50	2.50	5.00	5.00	5.00	2.50	5.00	5.00	5.00
5	Developer Availability	2.50	5.00	2.50	5.00	5.00	2.50	5.00	2.50	5.00	5.00	2.50	0.00	2.50
0	Job Trends	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0	Templating	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0	Components	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
5	Ajax	2.50	5.00	2.50	2.50	2.50	2.50	5.00	2.50	2.50	2.50	5.00	5.00	2.50
5	Plugins or Add-Ons	2.50	0.00	5.00	5.00	2.50	0.00	5.00	5.00	5.00	5.00	5.00	2.50	5.00
10	Scalability	10.00	10.00	5.00	5.00	5.00	10.00	10.00	5.00	5.00	5.00	5.00	10.00	10.00
10	Testing	10.00	10.00	5.00	5.00	10.00	10.00	5.00	10.00	10.00	0.00	5.00	5.00	10.00
0	i18n and l10n	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
5	Validation	5.00	5.00	5.00	2.50	5.00	5.00	5.00	5.00	5.00	5.00	5.00	2.50	2.50
10	Multi-language Support (Groovy / Scala)	5.00	5.00	10.00	10.00	10.00	10.00	0.00	10.00	0.00	0.00	10.00	0.00	5.00
10	Quality of Documentation/Tutorials	5.00	10.00	5.00	5.00	5.00	10.00	10.00	10.00	10.00	10.00	10.00	10.00	10.00
0	Books Published	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
10	REST Support (client and server)	5.00	10.00	5.00	0.00	5.00	5.00	5.00	10.00	10.00	5.00	5.00	5.00	5.00
10	Mobile / iPhone Support	10.00	10.00	10.00	10.00	10.00	10.00	10.00	10.00	10.00	5.00	10.00	10.00	10.00
0	Degree of Risk	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
100	Weighted Totals	70	85	67.5	62.5	75	77.5	80	90	82.5	50	82.5	62.5	82.5

Weighted Results



- ▶ Grails (90)
- ▶ Spring MVC (85)
- ▶ Ruby on Rails (82.5)
- ▶ Vaadin (82.5)
- ▶ Play (82.5)
- ▶ GWT (80)



Fighting for 5th



► Top at Devovx 2010

- GWT
- Rails
- Spring MVC
- Grails
- Wicket / Struts 2



Fighting for 5th



► Top at Rich Web Experience 2010

- Grails
- GWT
- Rails
- Spring MVC
- Tapestry / Vaadin



Ratings Logic



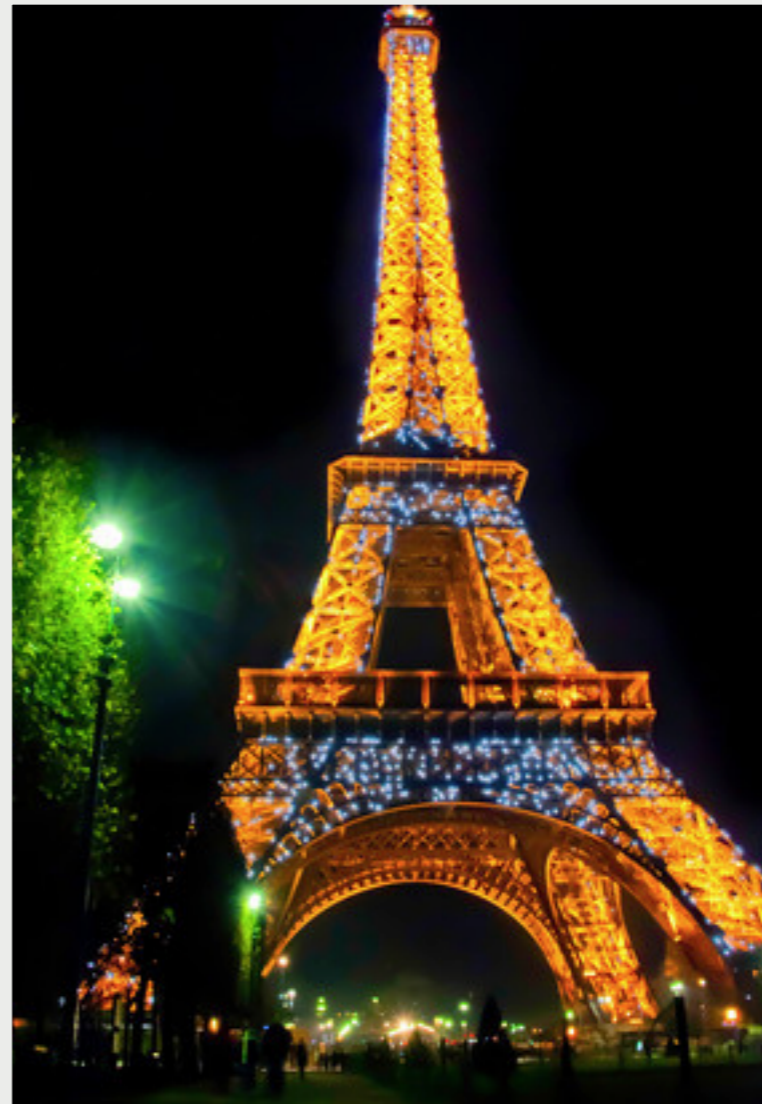
- ▶ Developer Productivity
- ▶ Developer Perception
- ▶ Learning Curve
- ▶ Project Health
- ▶ Developer Availability
- ▶ Job Trends



Ratings Logic



- ▶ Templating
- ▶ Components
- ▶ Ajax
- ▶ Plugins or Add-Ons
- ▶ Scalability
- ▶ Testing Support



Ratings Logic



- ▶ i18n and l10n
- ▶ Validation
- ▶ Multi-language Support (Groovy / Scala)
- ▶ Quality of Documentation/Tutorials
- ▶ Books Published
- ▶ REST Support (client and server)

Ratings Logic



- ▶ Mobile / iPhone Support
- ▶ Degree of Risk

[http://raibledesigns.com/rd/entry/
how_i_calculated_ratings_for](http://raibledesigns.com/rd/entry/how_i_calculated_ratings_for)



David Pollack's Lift Ratings



- ▶ Developer Productivity: Lift gets a 11, Rails gets a 5, most Java-based frameworks get a 1 or less.
- ▶ Developer Perception: Every web framework gets a 1.
- ▶ Learning Curve: Lift gets a 2.
- ▶ Job Trends, yep, it's zero.

* Matt's scale is 0-1 and my ratings are on Matt's scale, except mine goes to 11.

<http://lift.la/my-take-on-matt-raibles-spreadsheet>

Peter Thomas's Perfbench



- ▶ Seam / JSF vs. Wicket Performance Comparison
 - January 2009: Seam 2.1.1 and Wicket 1.3.5
 - Average page response time in milliseconds

action	Seam / JSF					Wicket				
	users-1	users-5	users-10	users-15	users-20	users-1	users-5	users-10	users-15	users-20
get login	569	521	406	385	747	154	40	37	45	63
post login	392	626	979	973	1189	303	166	191	268	296
ajax post search	35	117	158	288	336	15	40	95	152	224
get view hotel	19	106	154	250	294	8	43	49	88	82
post book hotel	28	71	118	181	207	28	43	82	77	96
ajax post cc number	23	59	84	148	152	4	11	19	28	23
ajax post cc name	18	61	88	132	119	3	3	10	22	18
post booking details	16	52	111	149	105	9	20	37	53	42
post confirm booking	33	126	273	488	756	29	159	317	471	808
get logout	15	24	292	107	180	8	58	71	44	46

Peter Thomas's Perfbench



- ▶ On the Seam / JSF side, the 20 sessions each take up about 800 KB adding up to around 16 MB total. On the Wicket side the 20 sessions add up to around 1.5 MB.

	Seam / JSF	Wicket
Heap Size (bytes)	23,947,512	9,496,312
Classes	6,344	4,489
Objects	585,123	179,028
Class Loaders	370	234

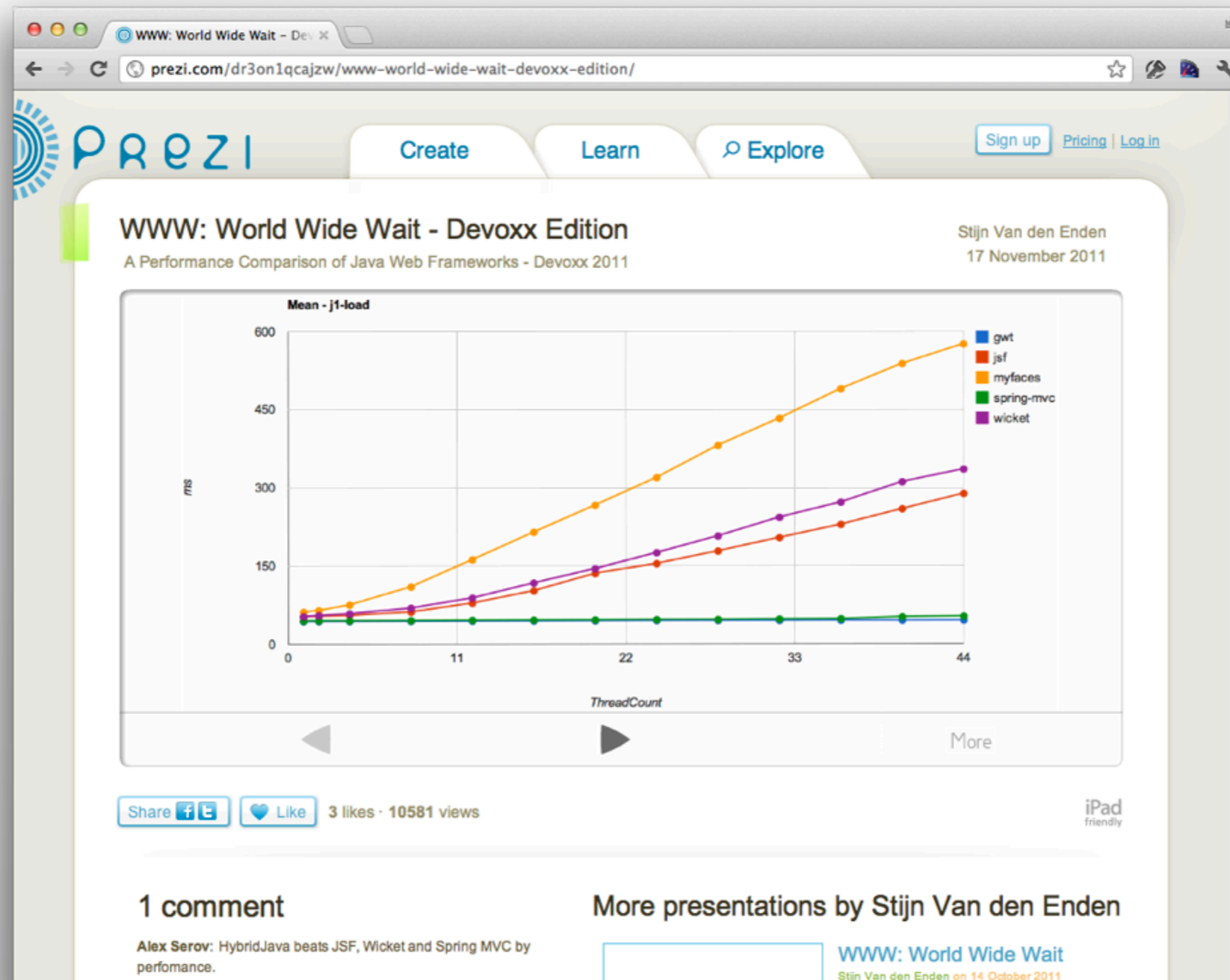
- ▶ + Lots of banter between Peter and Seam developers @ <http://bit.ly/3X50Gc>

Peter Thomas's Perfbench



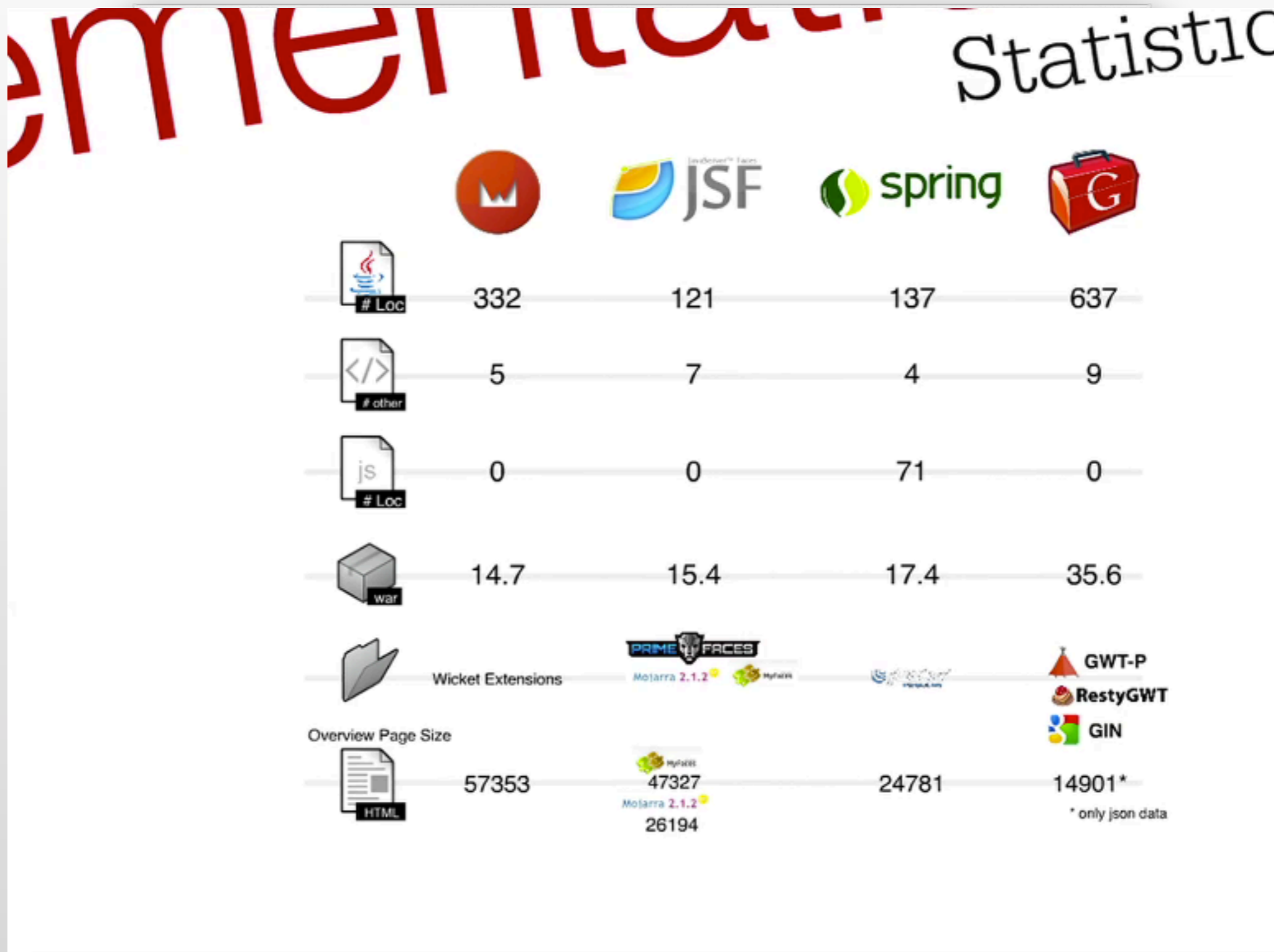
- ▶ Peter's Observations:
 - Grails was far more productive than Tapestry 5.
 - Grails still has some ways to go in terms of performance.
 - Overall, Wicket is fastest, with Tapestry coming a close second.
 - Wicket takes up the least amount of heap.
 - Session usage of the Seam + JSF combination is significantly higher compared to the rest.

World Wide Wait - Devovxx



<http://www.parleys.com/d/2942>

World Wide Wait - Devvoxx



World Wide Wait - Devovxx



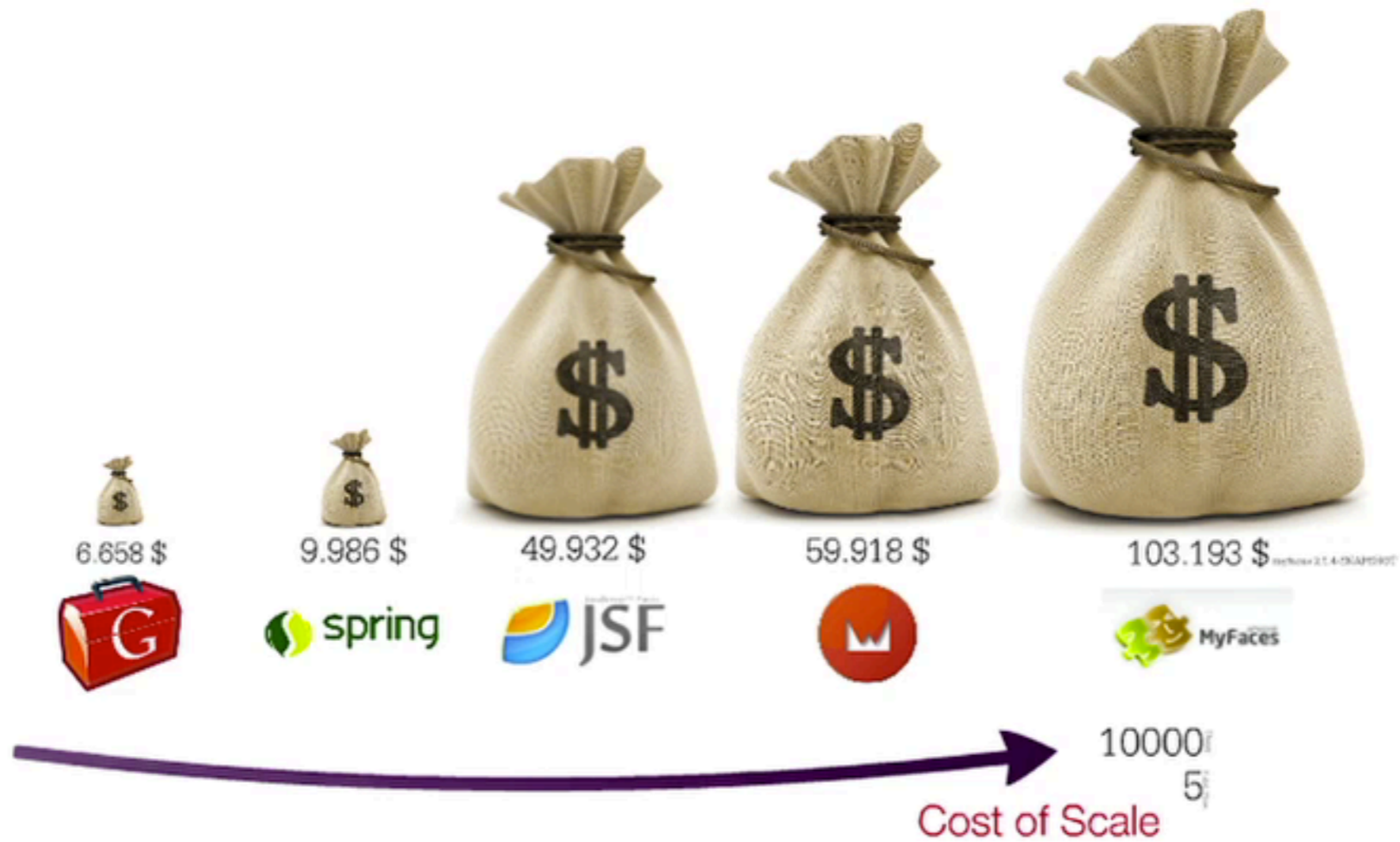
> 16 GB of data

World Wide Wait - Devovxx



>700 hours of test runs

World Wide Wait - Devovxx



World Wide Wait - Devovxx



WWW: World Wide Wait - Devovxx Edition
A Performance Comparison of Java Web Frameworks - Devovxx 2011
Stijn Van den Enden
17 November 2011

Framework	Performance (ms)
gwt	~300
jsf	~350
myfaces	~550
spring-mvc	~300
wicket	~350

HybridJava beats JSF, Wicket and Spring MVC by performance.

1 comment
Alex Serov: HybridJava beats JSF, Wicket and Spring MVC by performance.

More presentations by Stijn Van den Enden
WWW: World Wide Wait
Stijn Van den Enden on 14 October 2011

HybridJava - Really?



HybridJava is a [Java™](#)-based server-side *component-oriented* **MVC** technology for programming dynamic web content. Most of existing frameworks claim to support "components", but what they really mean is often one or several of the following:

- components of page [Model](#)
- components of page [Controller](#)
- components of page [View](#)

For instance, a popular book on Struts "Programming Jakarta Struts" has chapters 5-7 titled "Struts Controller Components", "Struts Model Components", and "Struts View Components". Those three types of components are notably uncorrelated, so making them work together takes quite an effort. This situation did not change much with introduction of Struts 2.

In truly *component-oriented* frameworks (Wicket, Tapestry, Click and **HybridJava**) **each** component follows the **MVC** paradigm independently, incapsulating its own Model, its own Controller, and its own View within a single entity.

Historically, *component-oriented* approach flourished in development of desktop applications UI. Wicket, Tapestry and Click openly admit to their inheritance of Visual Basic. The core of VB approach of constructing UI from components is having objects of a programming language organized in a data structure isomorphic to UI. Recursive visiting of this structure makes nodes add some output to the screen. Objects representing buttons are designed to fire events as traditional method calls up along the tree and so on.

With invention of markup languages it became possible to conveniently depict the structure of UI using tags without doing any programming. Wicket, Click and Tapestry, however, still use a programmatically built structure. In particular, Wicket is overly redundant at that. To add a Wicket subcomponent means to add a Java class node, depicted both in mark-up as well as in configuration. **HybridJava** technology finally moves the task of constructing pages and components completely to the mark-up area, eliminating the need for programming and configuring.

Pros and Cons





▶ Pros

- Easy dynamic language transition for Java Developers
- Groovy
- Plugins for all types of applications

▶ Cons

- Groovy learning targets Java Developers
- Stack traces are horrendous
- Knowledge of underlying frameworks not required, but helpful



▶ Pros

- Write Java => Produces Optimized JavaScript
- Easy to learn and develop with standard Java Tools
- Vibrant Community

▶ Cons

- You have to know Java
- Slow to compile, difficult to test
- More like a JSP Tag Library than a web framework

Ruby on Rails



▶ Pros

- Easy to learn and understand for Web Developers
- Lots and lots of documentation
- Passionate Community

▶ Cons

- Slightly less performant by default
- Dynamic language means more tests
- Development Tools and Debugging

Spring MVC



▶ Pros

- Easy Configuration with Annotations and Conventions
- Integrates with many view options seamlessly: JSP/JSTL, Tiles, FreeMarker, Excel, PDF, JSON
- Excellent REST Support

▶ Cons

- Instant reload not built-in, need JRebel or Spring Roo
- No open development process, need to be SpringSource
- Ajax requires 3rd-party library (can be a good thing!)



▶ Pros

- Uses GWT API for developing view
- Vibrant Community and company backing
- Excellent Themes and Layouts support

▶ Cons

- Large memory footprint, state stored in session
- Sketchy (?) because backed by a commercial organization
- Joonas keeps telling me my cons are wrong

Wicket



▶ Pros

- Great for Java Developers
- Tight binding between pages and views
- Active community - support from creators

▶ Cons

- No Jobs or Developers
- Stateful by default
- HTML Templates live next to Java code by default

Tapestry



▶ Pros

- Live Class Reloading
- Development emphasis on performance and scalability
- Excellent Exception Reporting

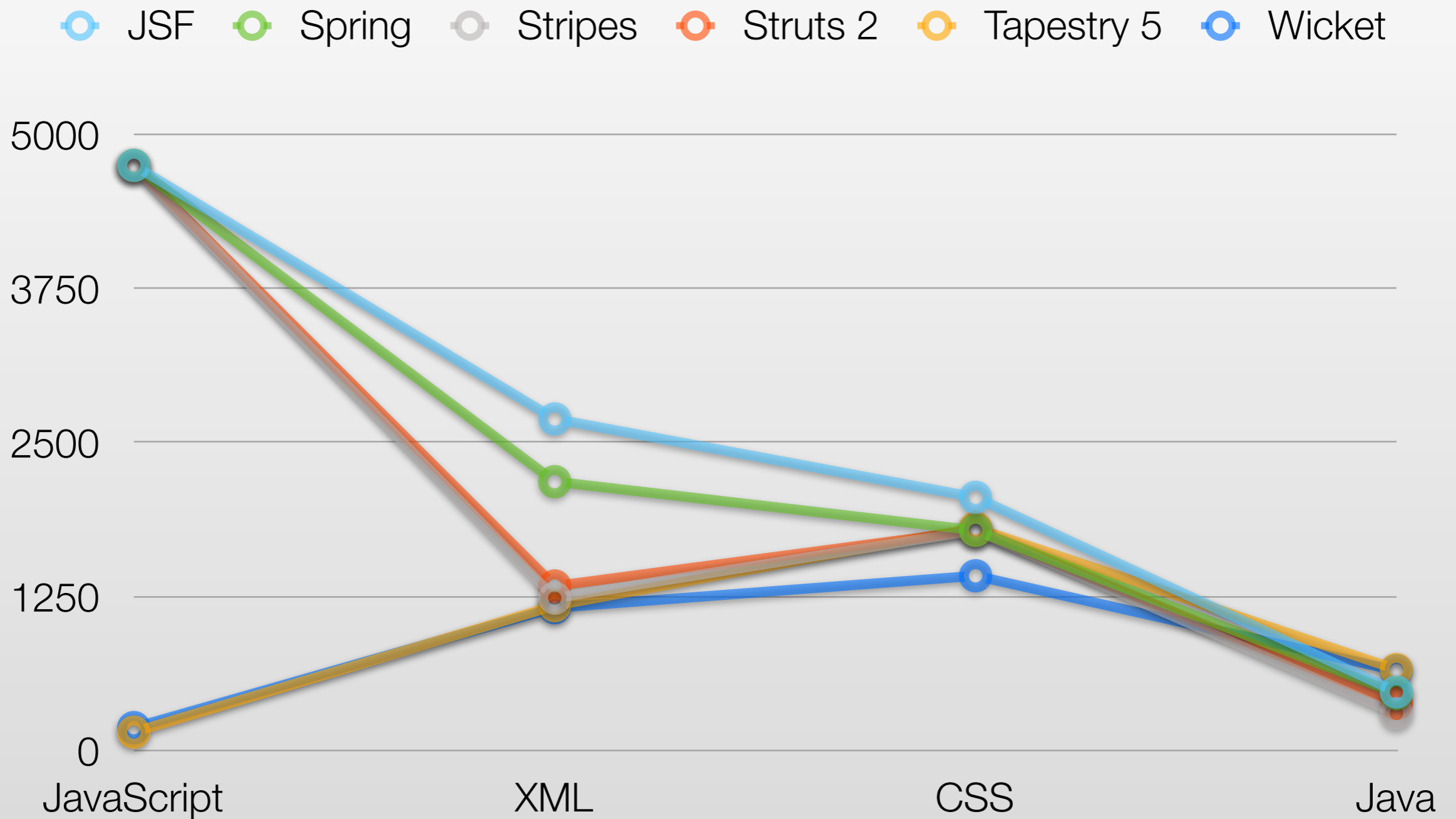
▶ Cons

- No Jobs
- Prototype baked in for JS Library
- Annotations vs. Conventions

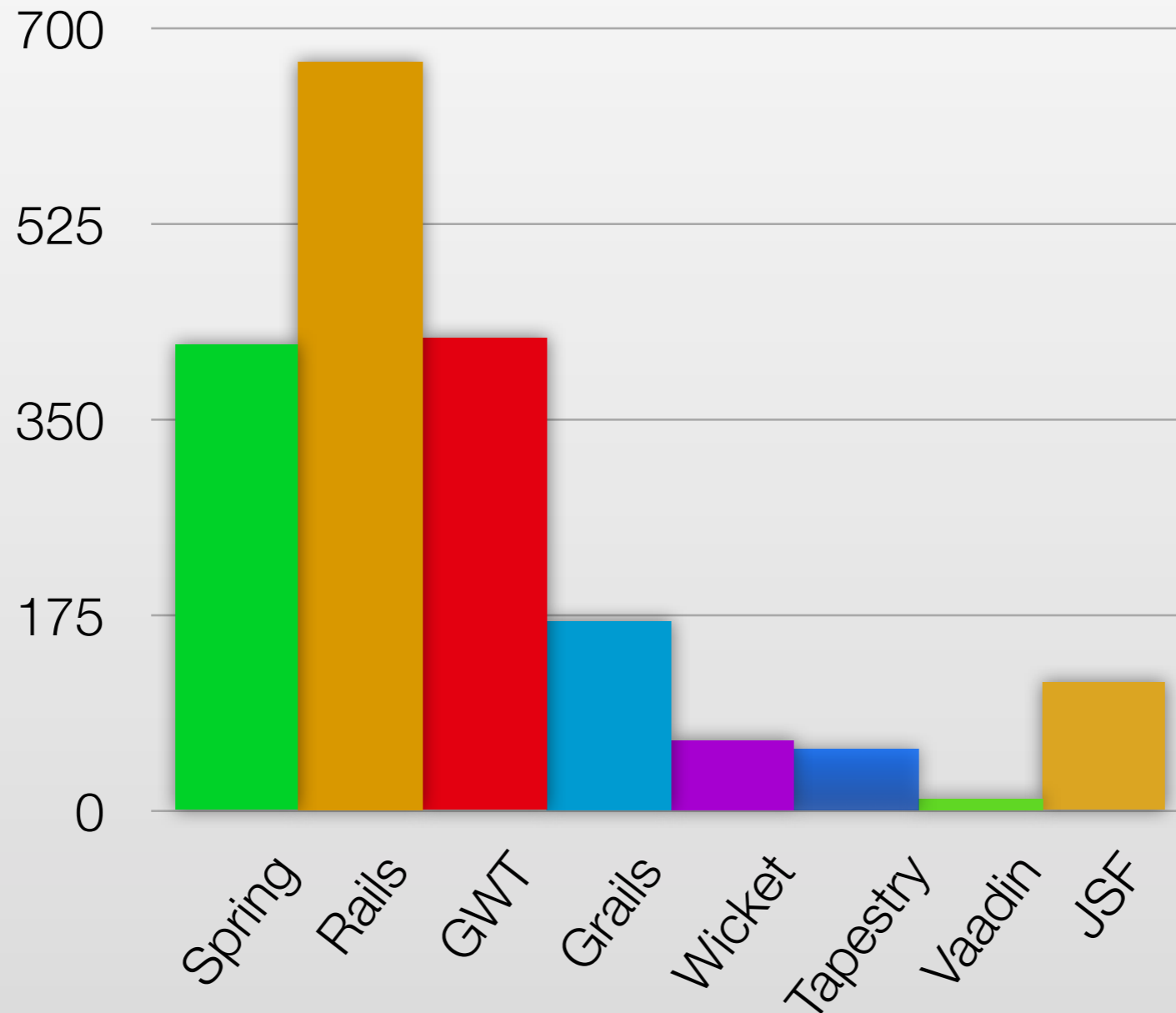
Pretty Graphs



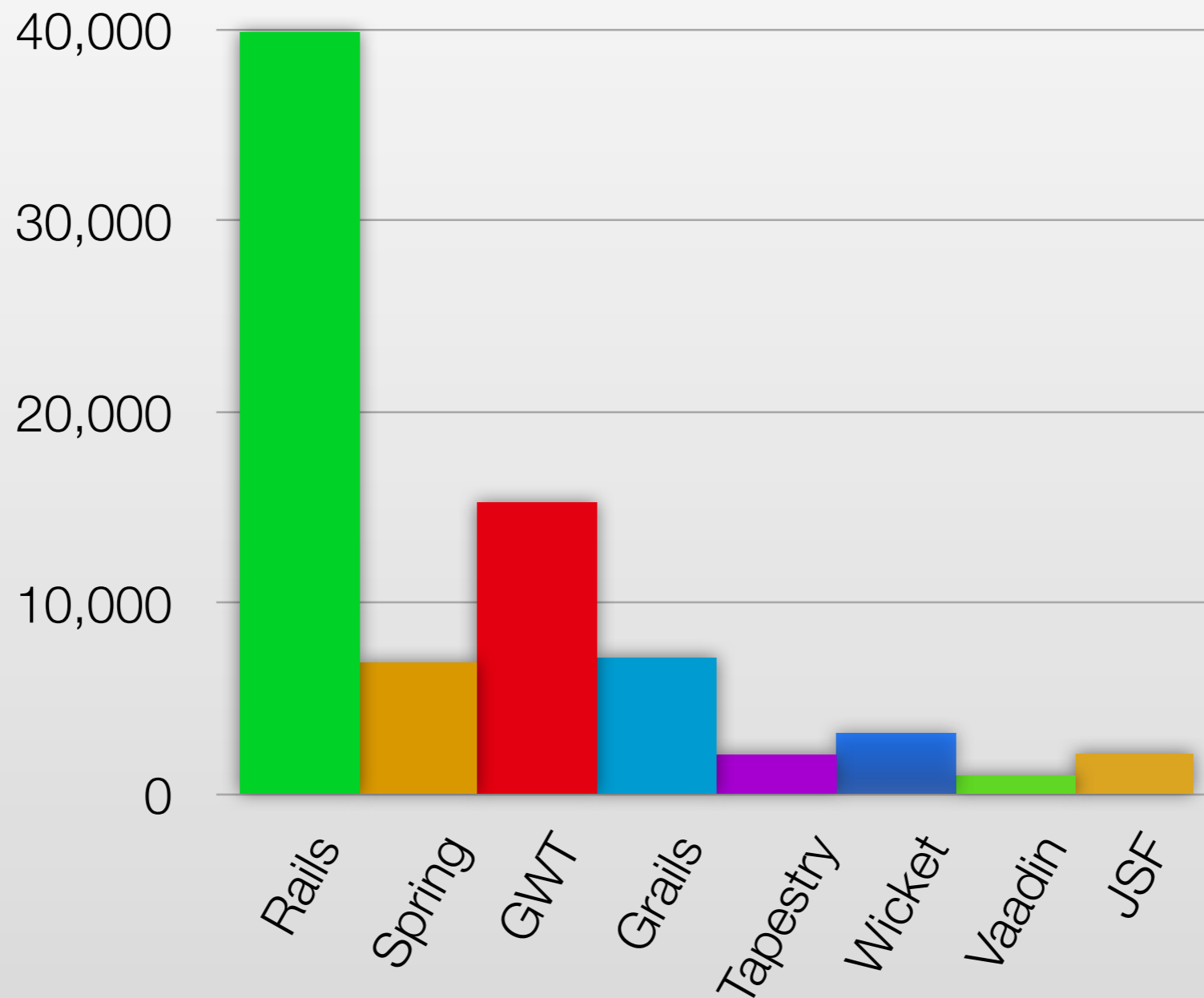
LOC in AppFuse Light



Jobs on Dice (World)



LinkedIn Skills (World)



Pretty Graphs



Google Trends

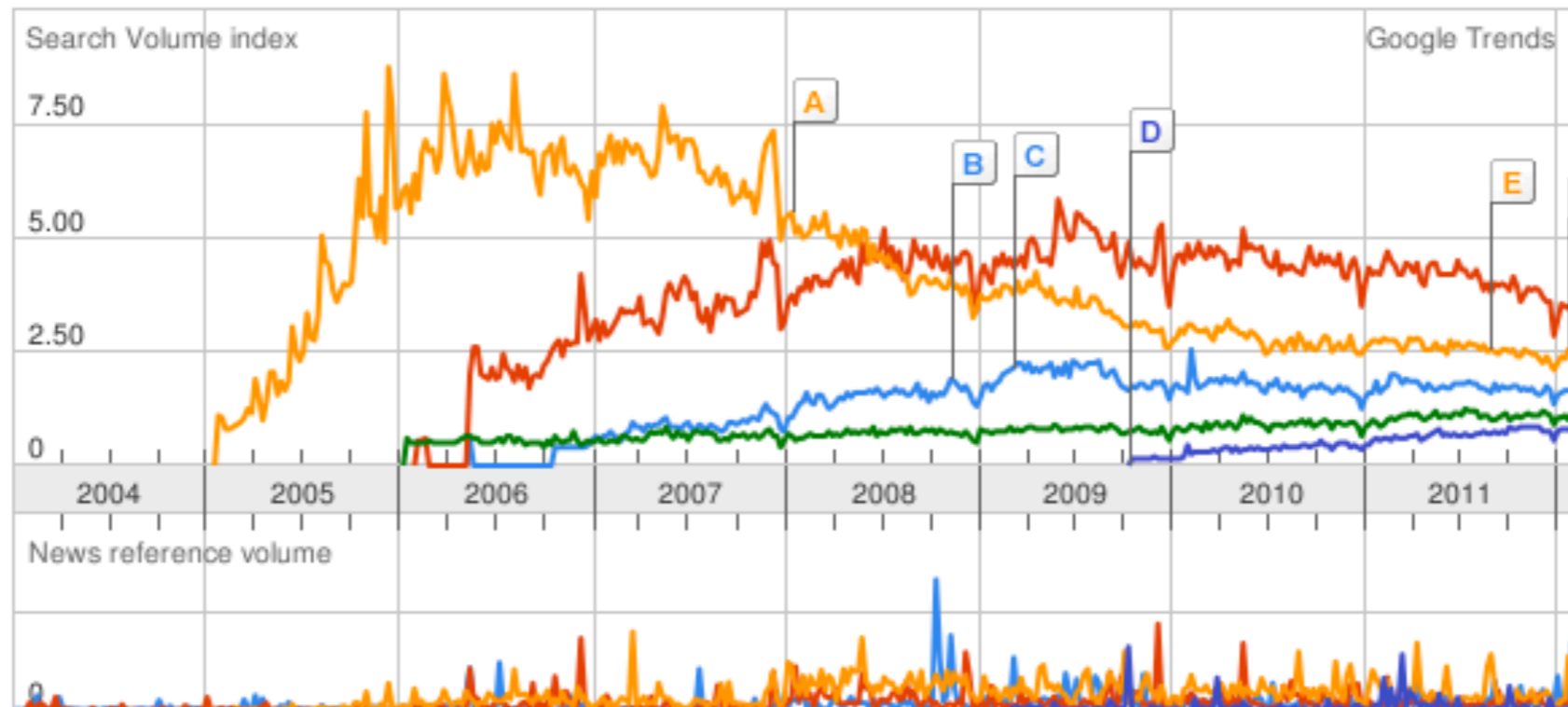
Tip: Use commas to compare multiple search terms.

Searches [Websites](#)

- Scale is based on the average worldwide traffic of **grails** in all years. [Learn more](#)
- An improvement to our geographical assignment was applied retroactively from 1/1/2011. [Learn more](#)

grails 1.00 **gwt** 2.85 **ruby on rails** 3.75 **spring mvc** 0.60
vaadin 0.15

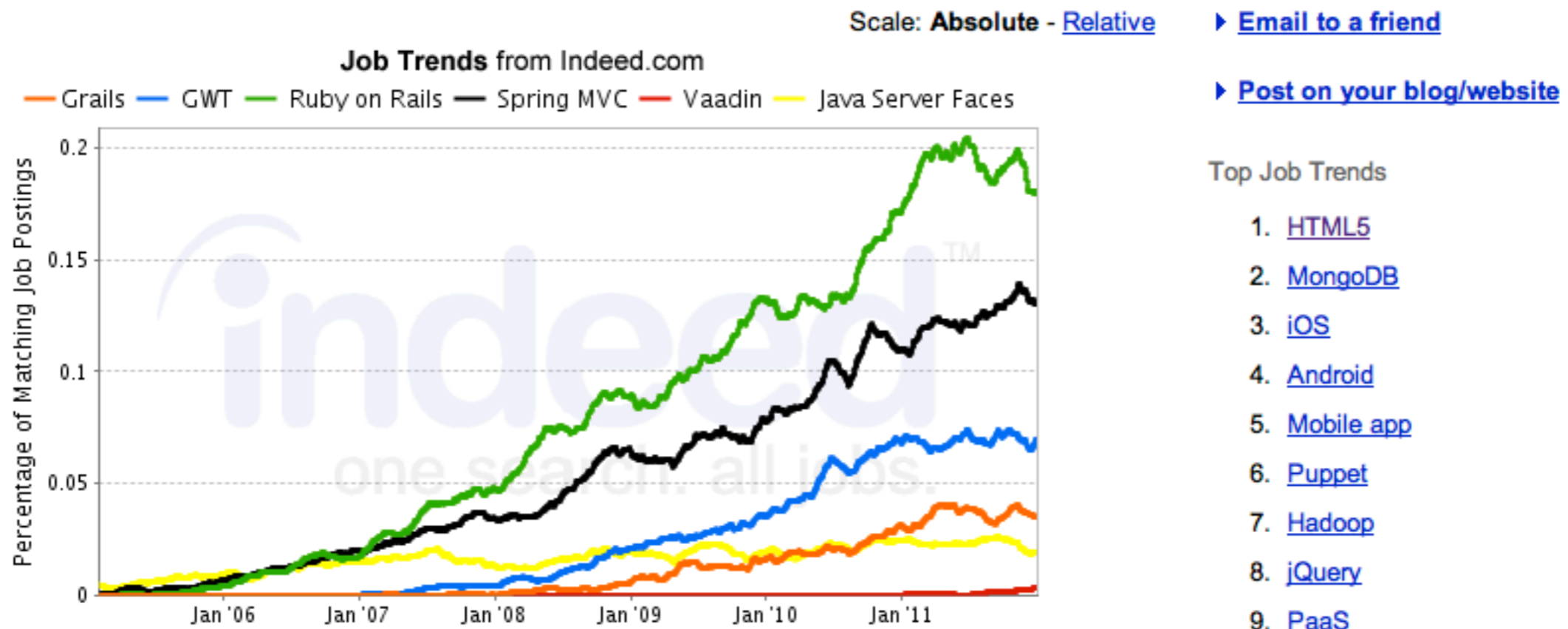
Showing only first 5 terms



Pretty Graphs



Grails, GWT, Ruby on Rails, Spring MVC, Vaadin, Java Server Faces Job Trends



Top Job Trends

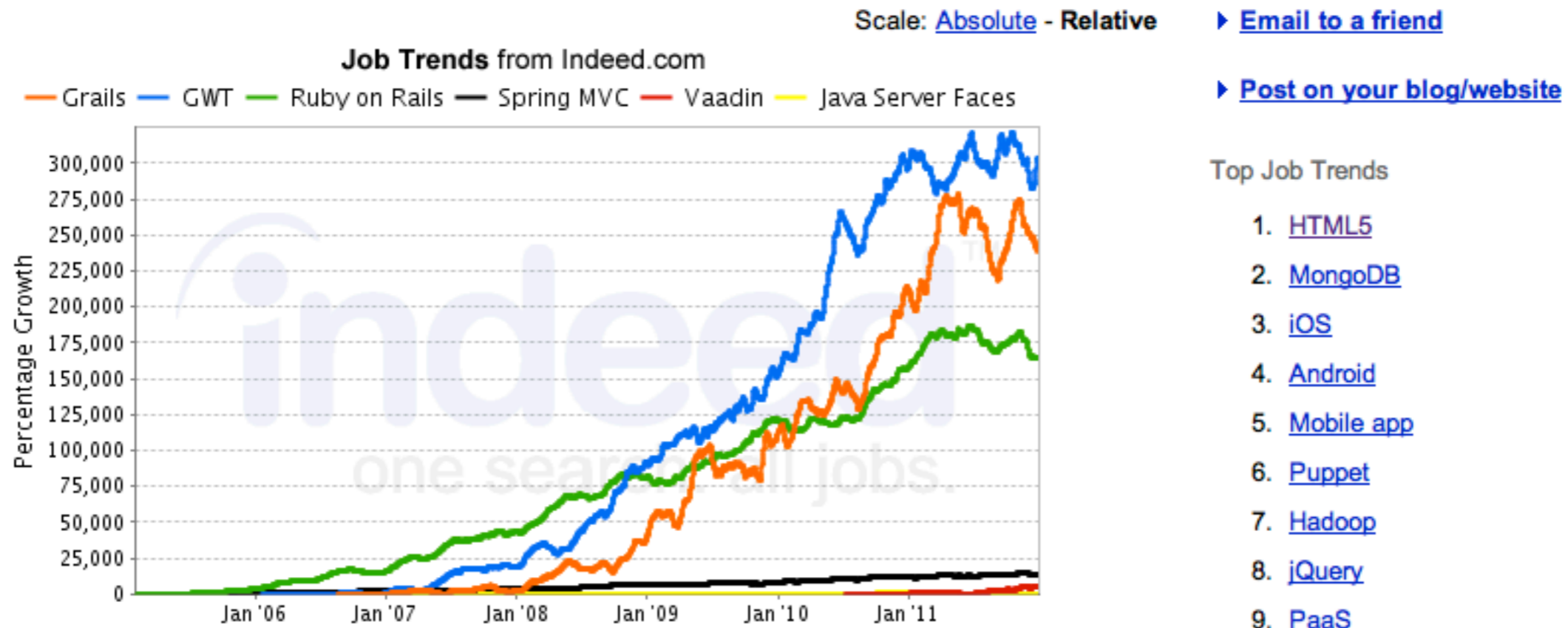
1. [HTML5](#)
2. [MongoDB](#)
3. [iOS](#)
4. [Android](#)
5. [Mobile app](#)
6. [Puppet](#)
7. [Hadoop](#)
8. [jQuery](#)
9. [PaaS](#)
10. [Social Media](#)

Indeed.com searches millions of jobs from thousands of job sites.
This job trends graph shows the percentage of jobs we find that contain your search terms.

Pretty Graphs



Grails, GWT, Ruby on Rails, Spring MVC, Vaadin, Java Server Faces Job Trends



Top Job Trends

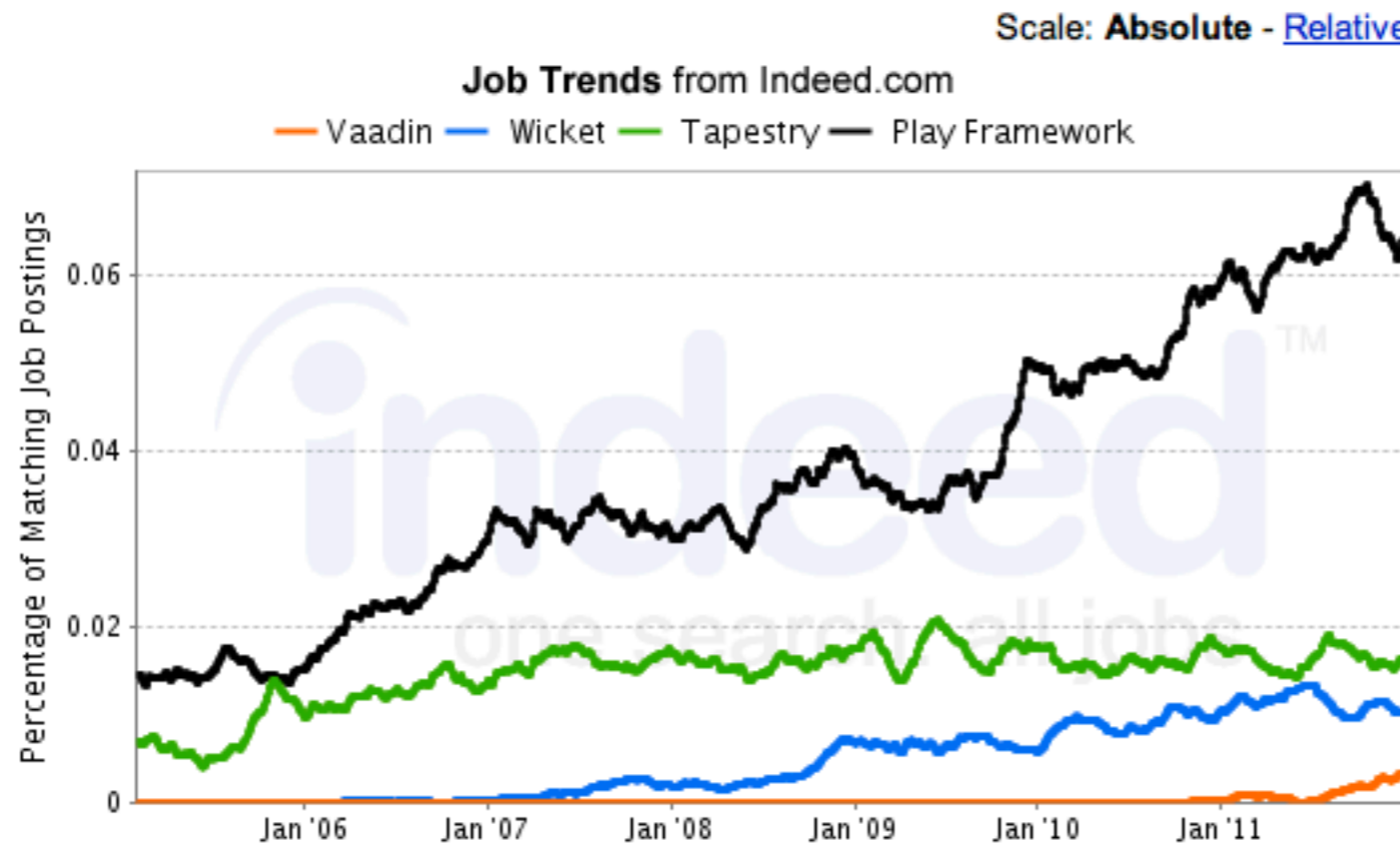
1. [HTML5](#)
2. [MongoDB](#)
3. [iOS](#)
4. [Android](#)
5. [Mobile app](#)
6. [Puppet](#)
7. [Hadoop](#)
8. [jQuery](#)
9. [PaaS](#)
10. [Social Media](#)

Indeed.com searches millions of jobs from thousands of job sites.
This job trends graph shows relative growth for jobs we find matching your search terms.

Pretty Graphs



Vaadin, Wicket, Tapestry, Play Framework Job Trends



▶ [Email to a friend](#)

▶ [Post on your blog/website](#)

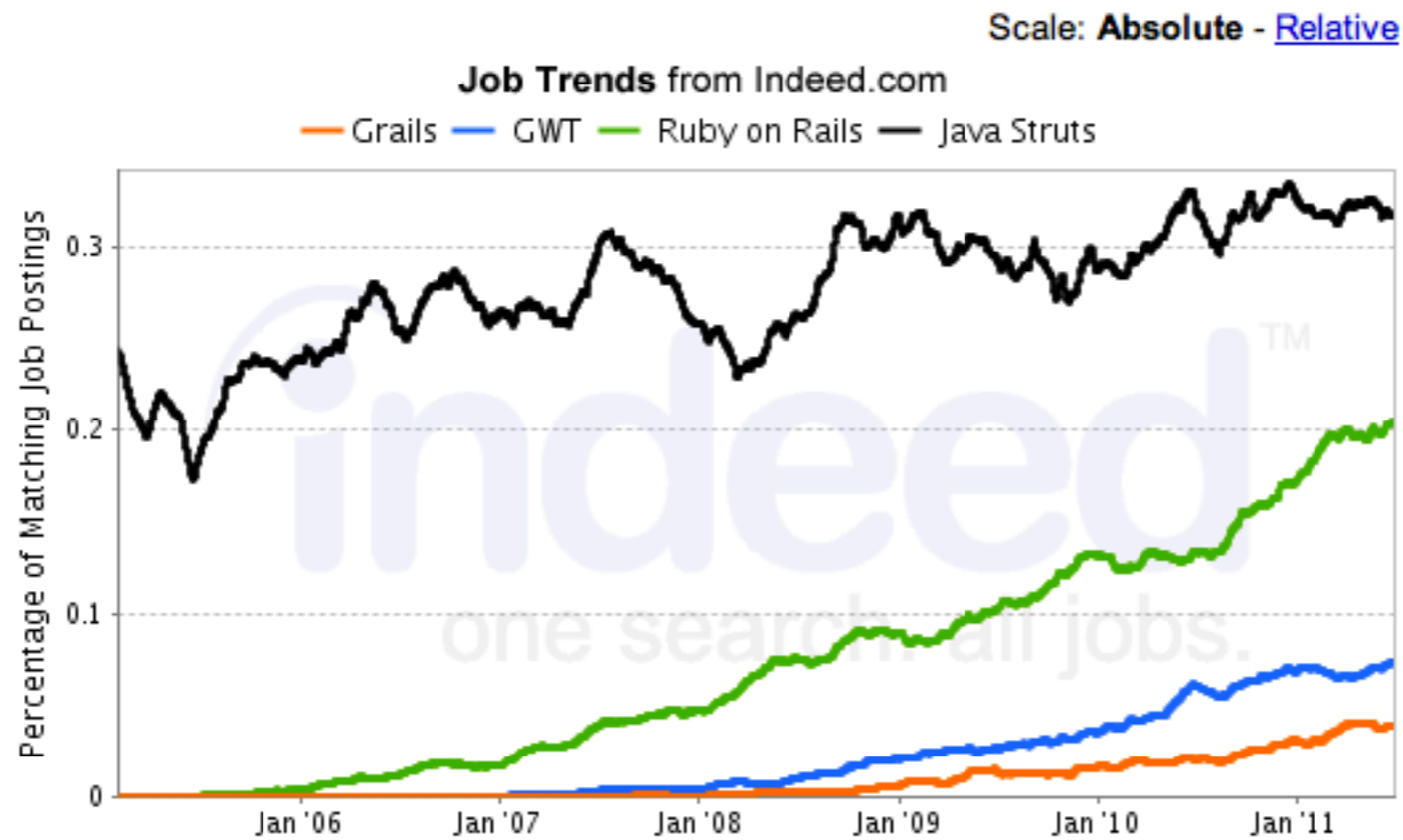
Top Job Trends

1. [HTML5](#)
2. [MongoDB](#)
3. [iOS](#)
4. [Android](#)
5. [Mobile app](#)
6. [Puppet](#)
7. [Hadoop](#)
8. [jQuery](#)
9. [PaaS](#)

Pretty Graphs



Grails, GWT, Ruby on Rails, Java Struts Job Trends



► [Email to a friend](#)

► [Post on your blog/website](#)

Top Job Trends

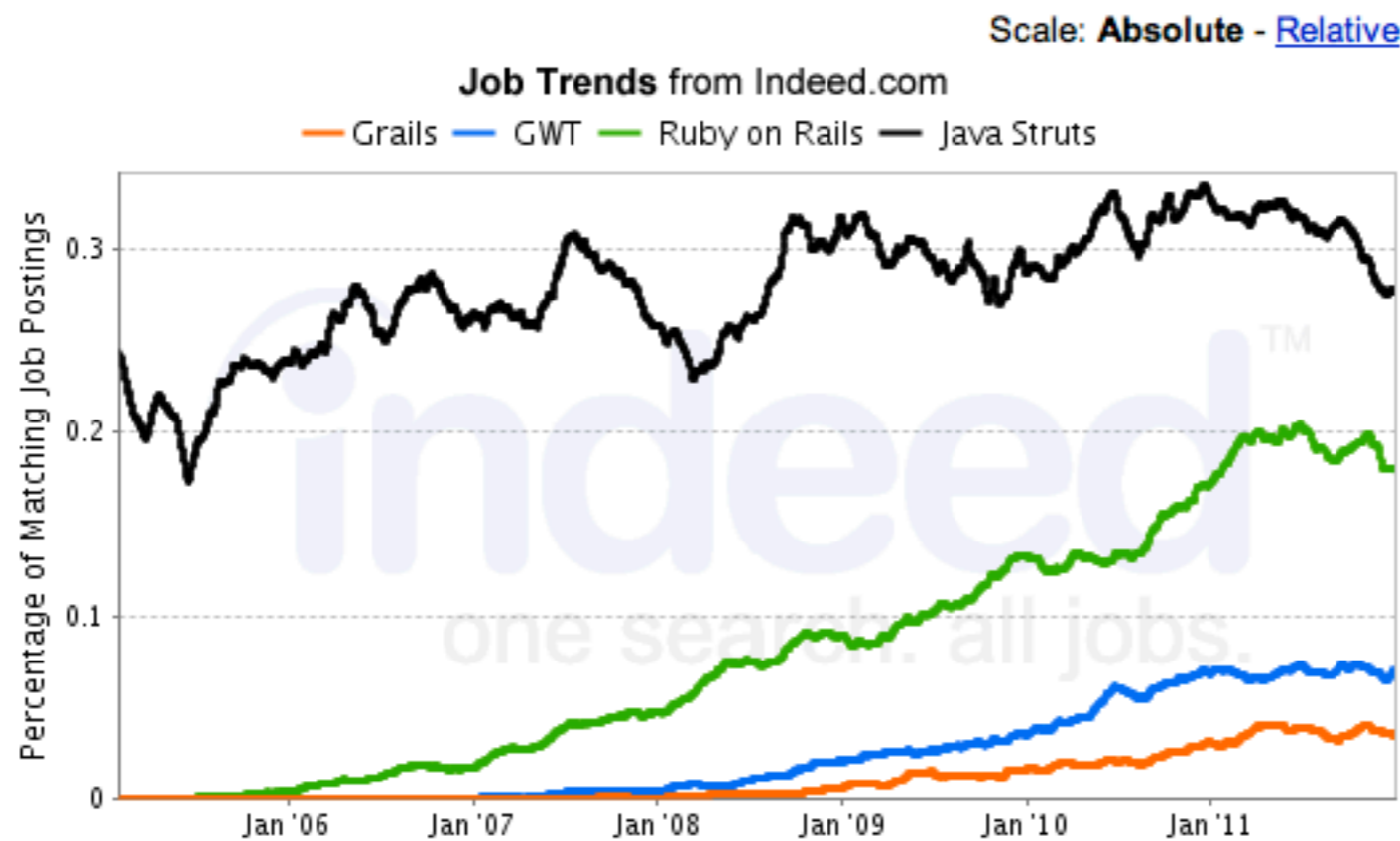
1. [HTML5](#)
2. [Mobile app](#)
3. [Android](#)
4. [Twitter](#)
5. [jQuery](#)
6. [Facebook](#)
7. [Social Media](#)
8. [iPhone](#)
9. [Cloud Computing](#)
10. [Virtualization](#)

Indeed.com searches millions of jobs from thousands of job sites.
This job trends graph shows the percentage of jobs we find that contain your search terms.

Pretty Graphs



Grails, GWT, Ruby on Rails, Java Struts Job Trends



▶ [Email to a friend](#)

▶ [Post on your blog/website](#)

Top Job Trends

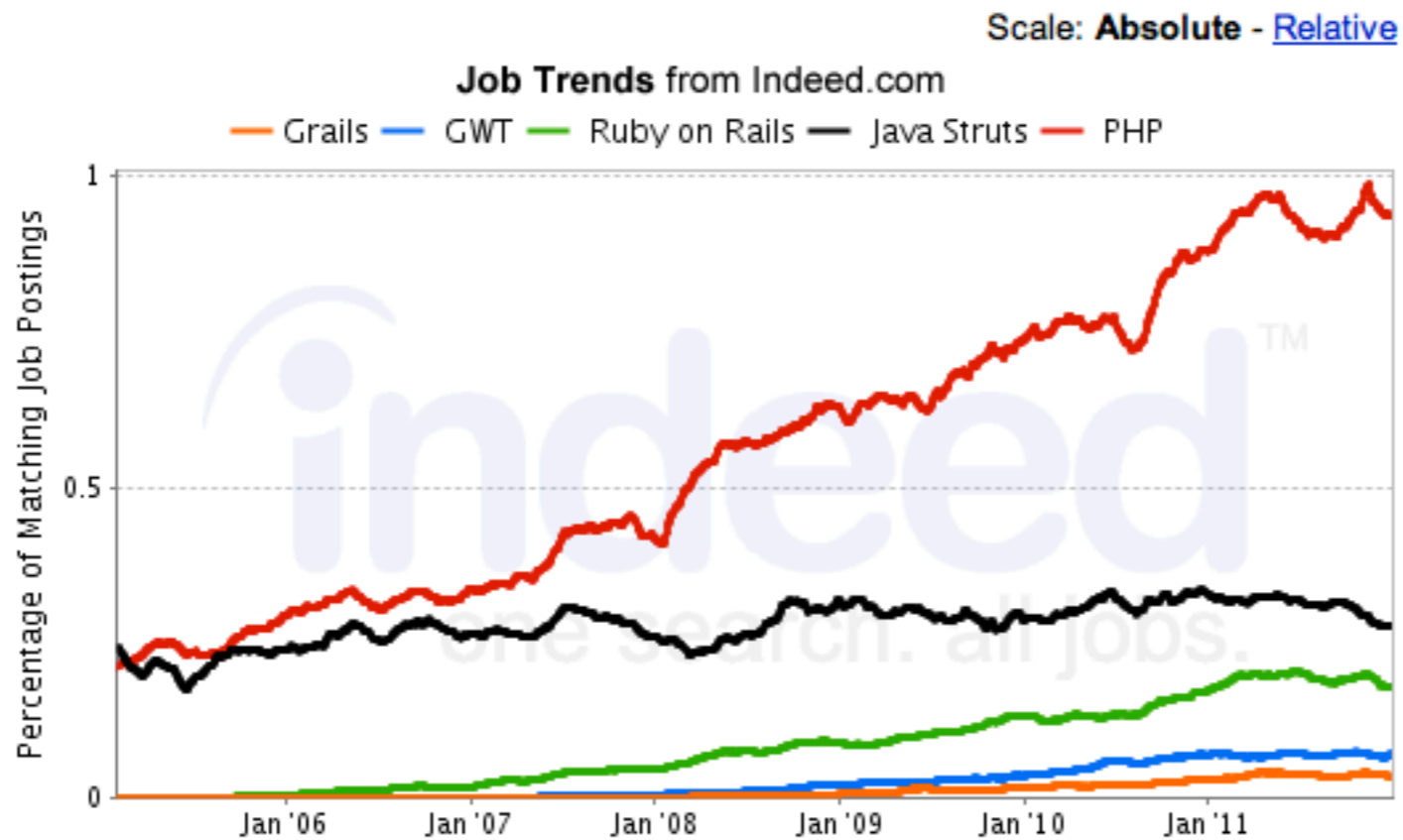
1. [HTML5](#)
2. [MongoDB](#)
3. [iOS](#)
4. [Android](#)
5. [Mobile app](#)
6. [Puppet](#)
7. [Hadoop](#)
8. [jQuery](#)
9. [PaaS](#)
10. [Social Media](#)

Indeed.com searches millions of jobs from thousands of job sites.
This job trends graph shows the percentage of jobs we find that contain your search terms.

Pretty Graphs



Grails, GWT, Ruby on Rails, Java Struts, PHP Job Trends



► [Email to a friend](#)

► [Post on your blog/website](#)

Top Job Trends

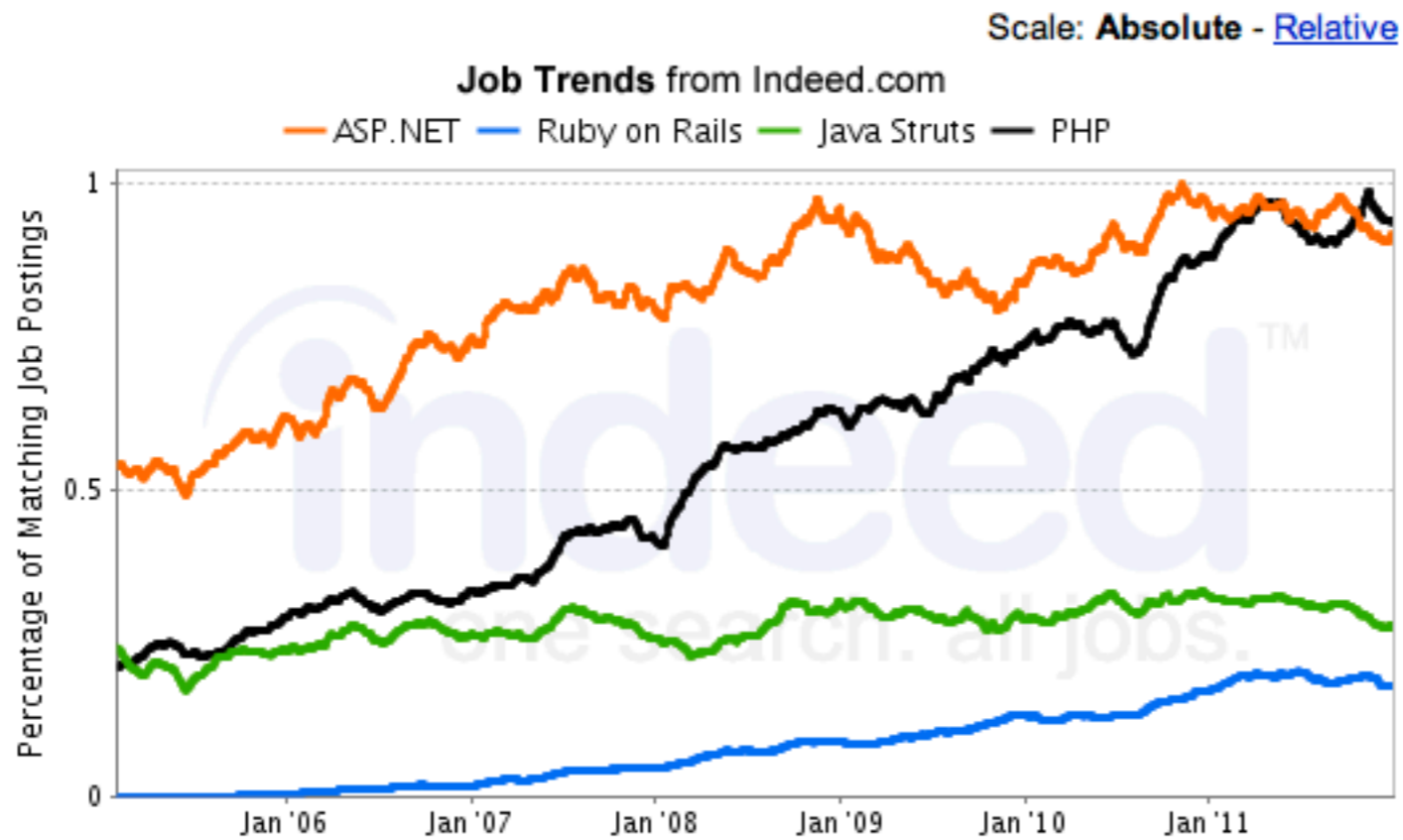
1. [HTML5](#)
2. [MongoDB](#)
3. [iOS](#)
4. [Android](#)
5. [Mobile app](#)
6. [Puppet](#)
7. [Hadoop](#)
8. [jQuery](#)
9. [PaaS](#)
10. [Social Media](#)

Indeed.com searches millions of jobs from thousands of job sites.
This job trends graph shows the percentage of jobs we find that contain your search terms.

Pretty Graphs



ASP.NET, Ruby on Rails, Java Struts, PHP Job Trends



▶ [Email to a friend](#)

▶ [Post on your blog/website](#)

Top Job Trends

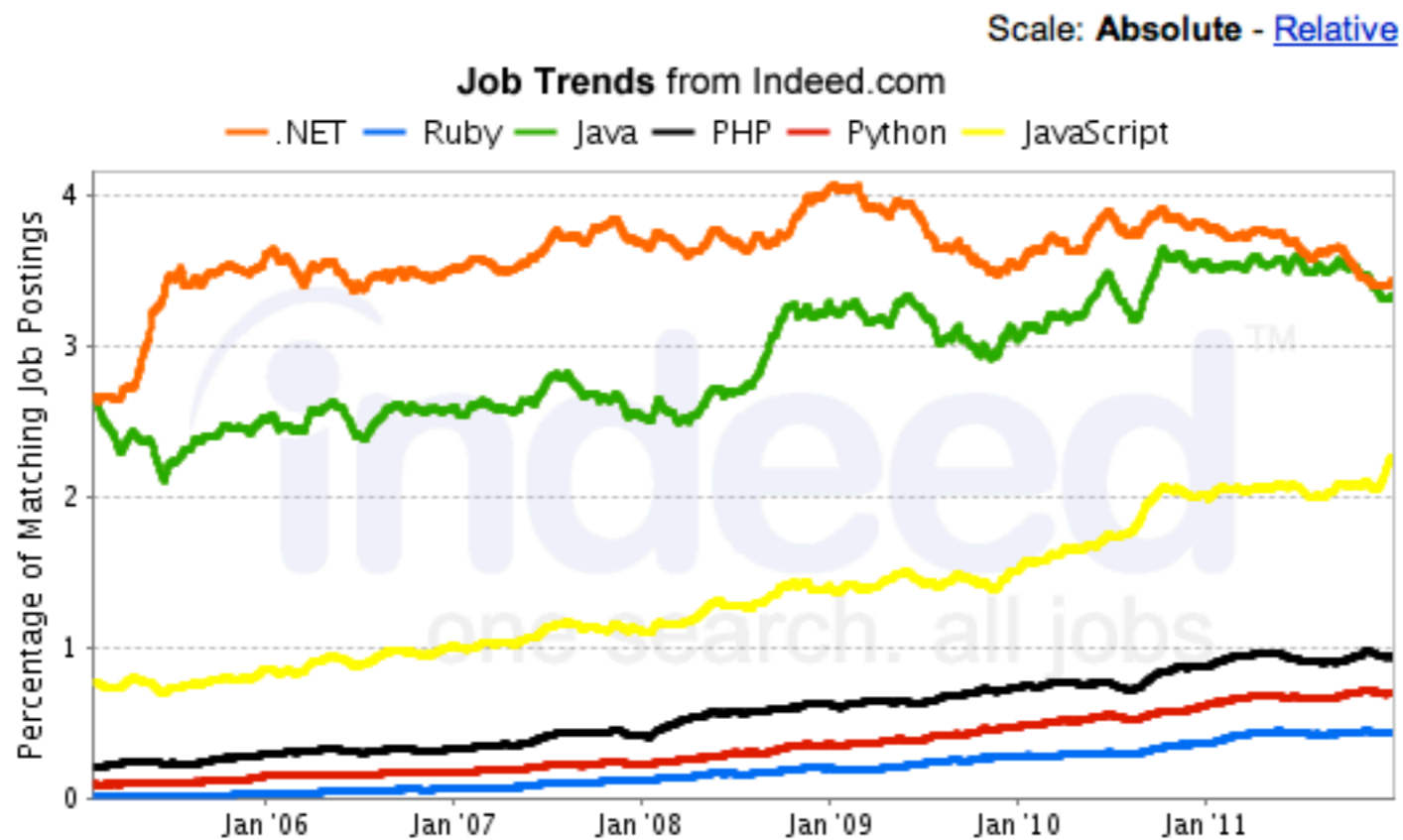
1. [HTML5](#)
2. [MongoDB](#)
3. [iOS](#)
4. [Android](#)
5. [Mobile app](#)
6. [Puppet](#)
7. [Hadoop](#)
8. [jQuery](#)
9. [PaaS](#)
10. [Social Media](#)

Indeed.com searches millions of jobs from thousands of job sites.
This job trends graph shows the percentage of jobs we find that contain your search terms.

Pretty Graphs



.NET, Ruby, Java, PHP, Python, JavaScript Job Trends



▶ [Email to a friend](#)

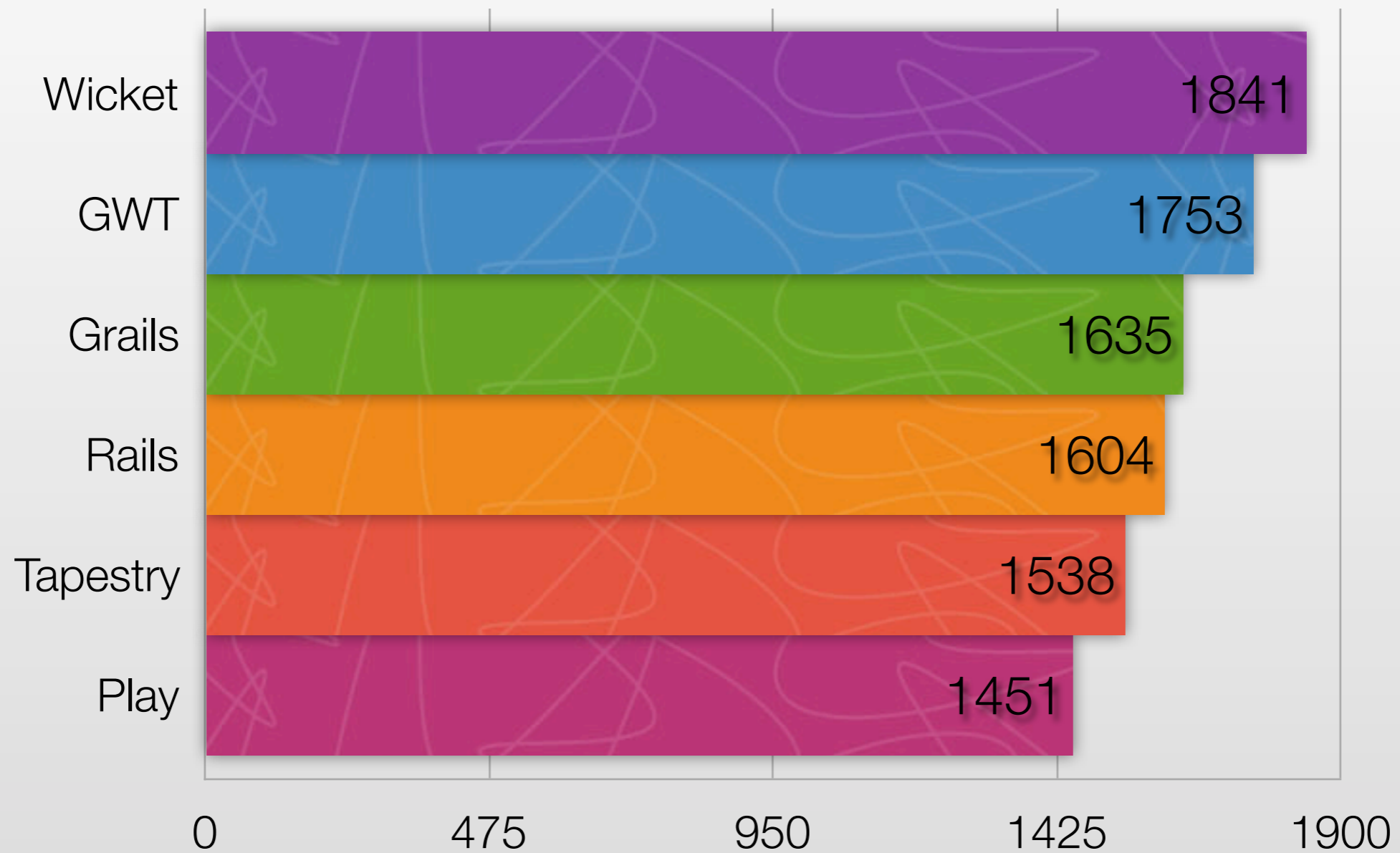
▶ [Post on your blog/website](#)

Top Job Trends

1. [HTML5](#)
2. [MongoDB](#)
3. [iOS](#)
4. [Android](#)
5. [Mobile app](#)
6. [Puppet](#)
7. [Hadoop](#)
8. [jQuery](#)
9. [PaaS](#)
10. [Social Media](#)

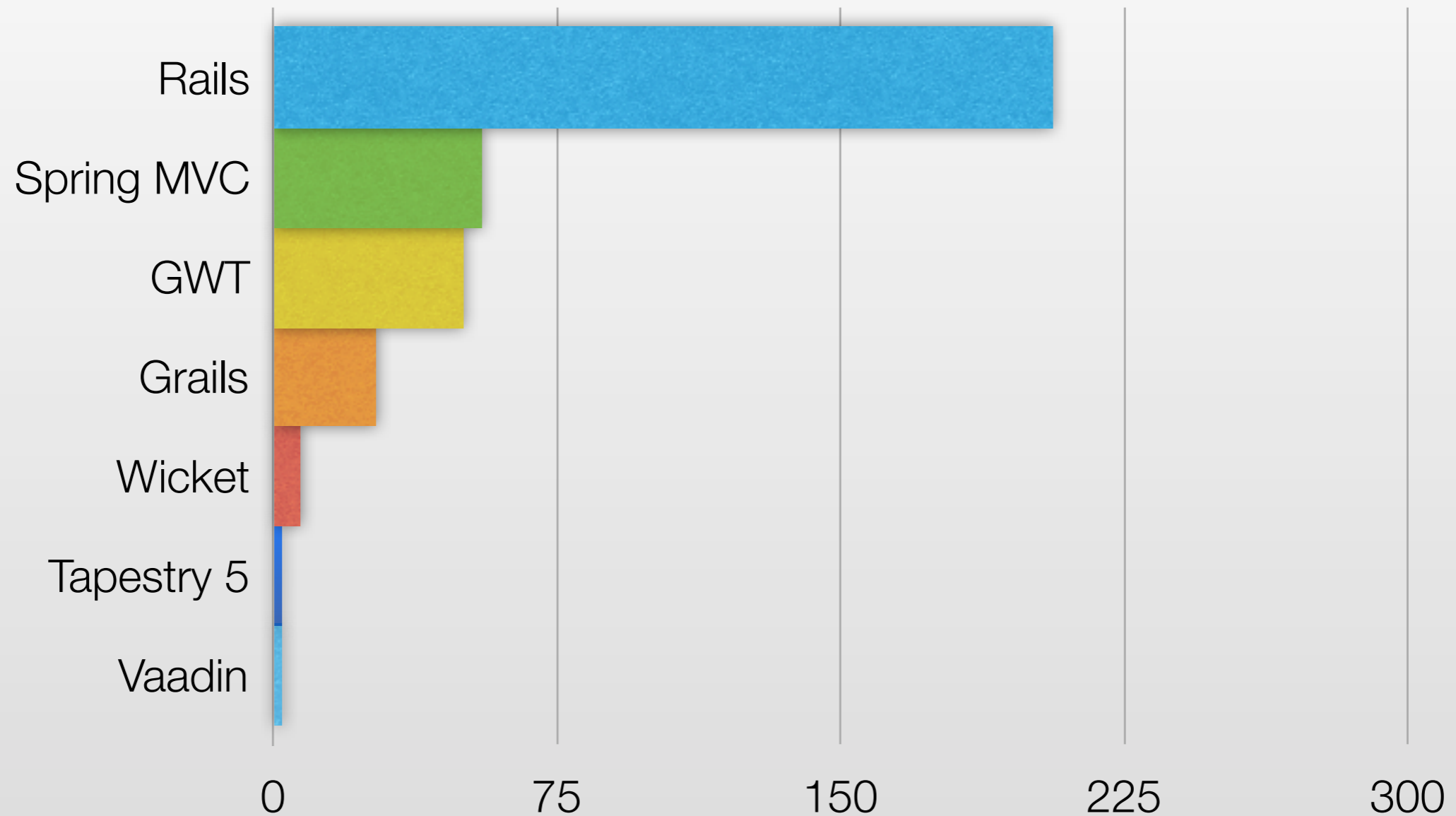
Indeed.com searches millions of jobs from thousands of job sites.
This job trends graph shows the percentage of jobs we find that contain your search terms.

Mailing List Traffic



* Spring MVC and Vaadin use Forums, which don't provide this data.

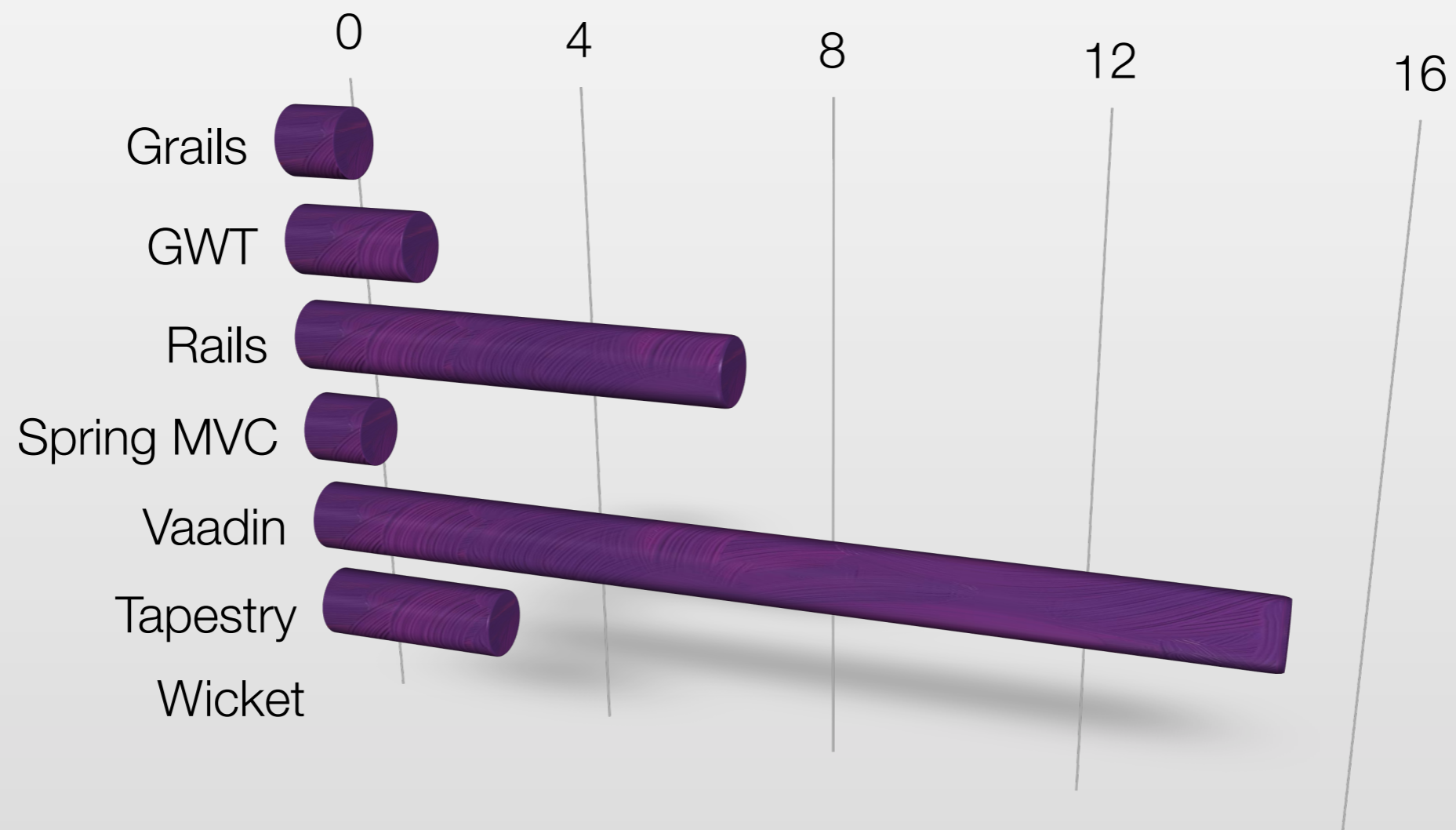
Books on Amazon



2011 Releases



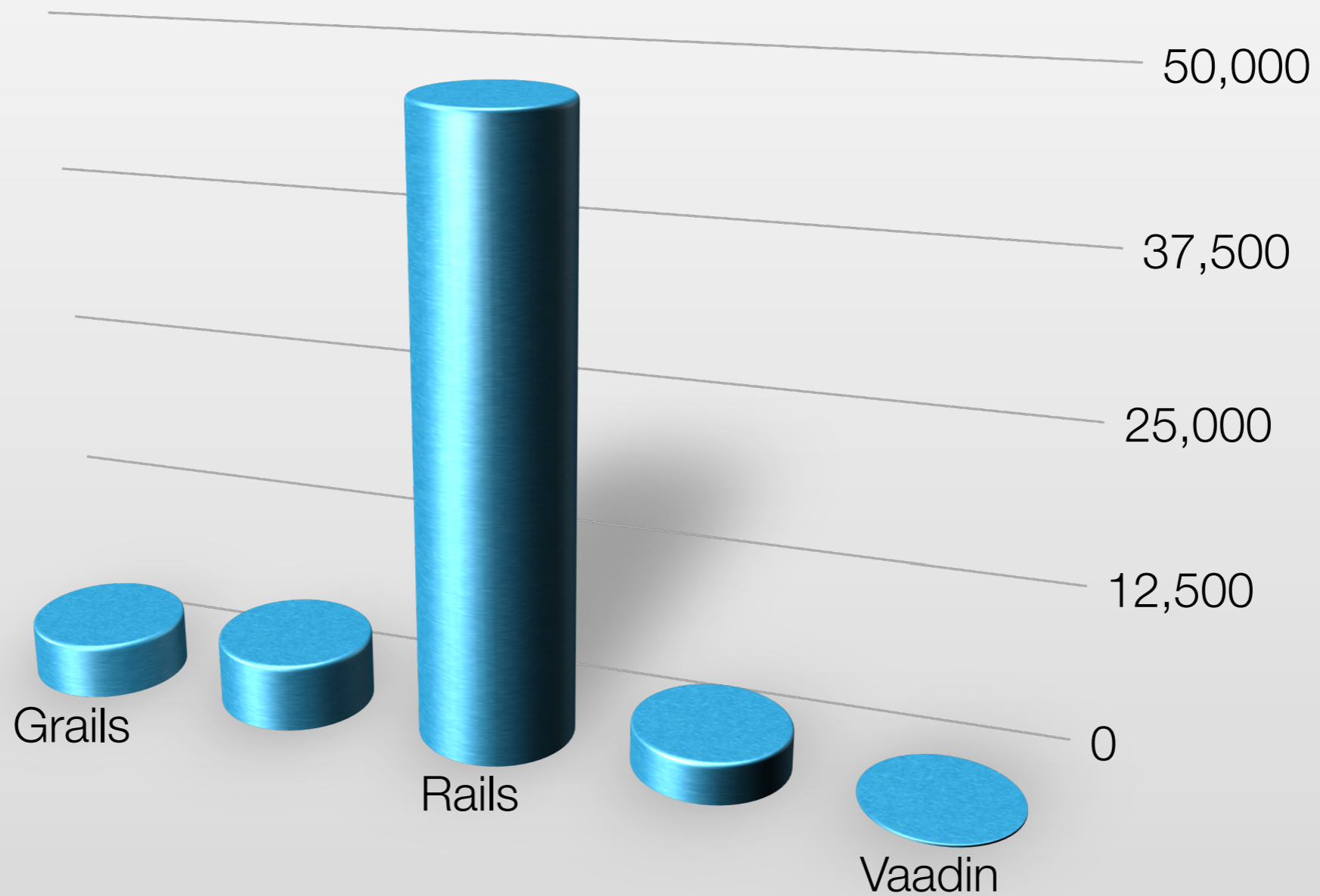
■ As of August 24, 2011



StackOverflow



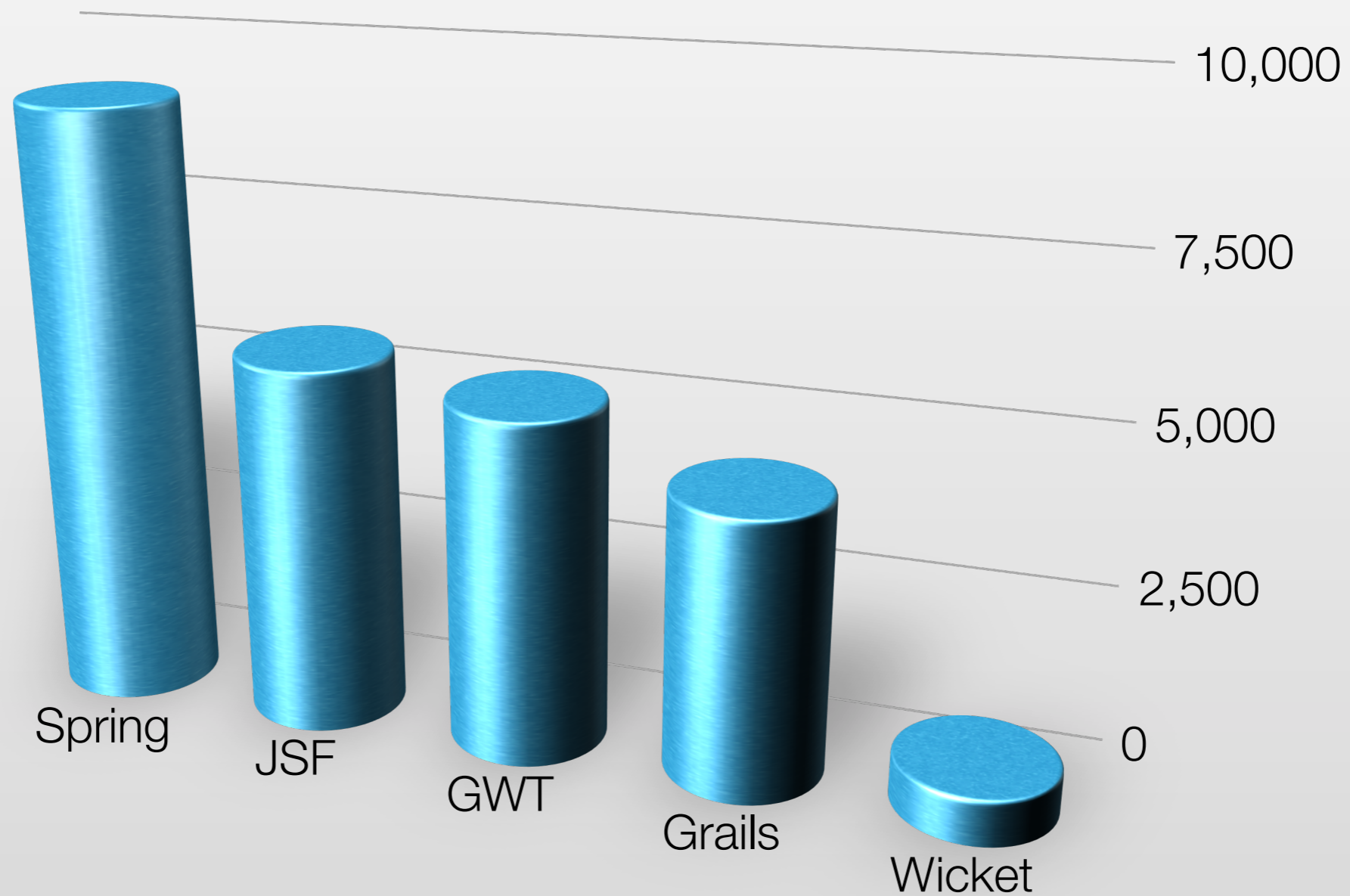
■ Tagged Questions (August 24, 2011)



StackOverflow



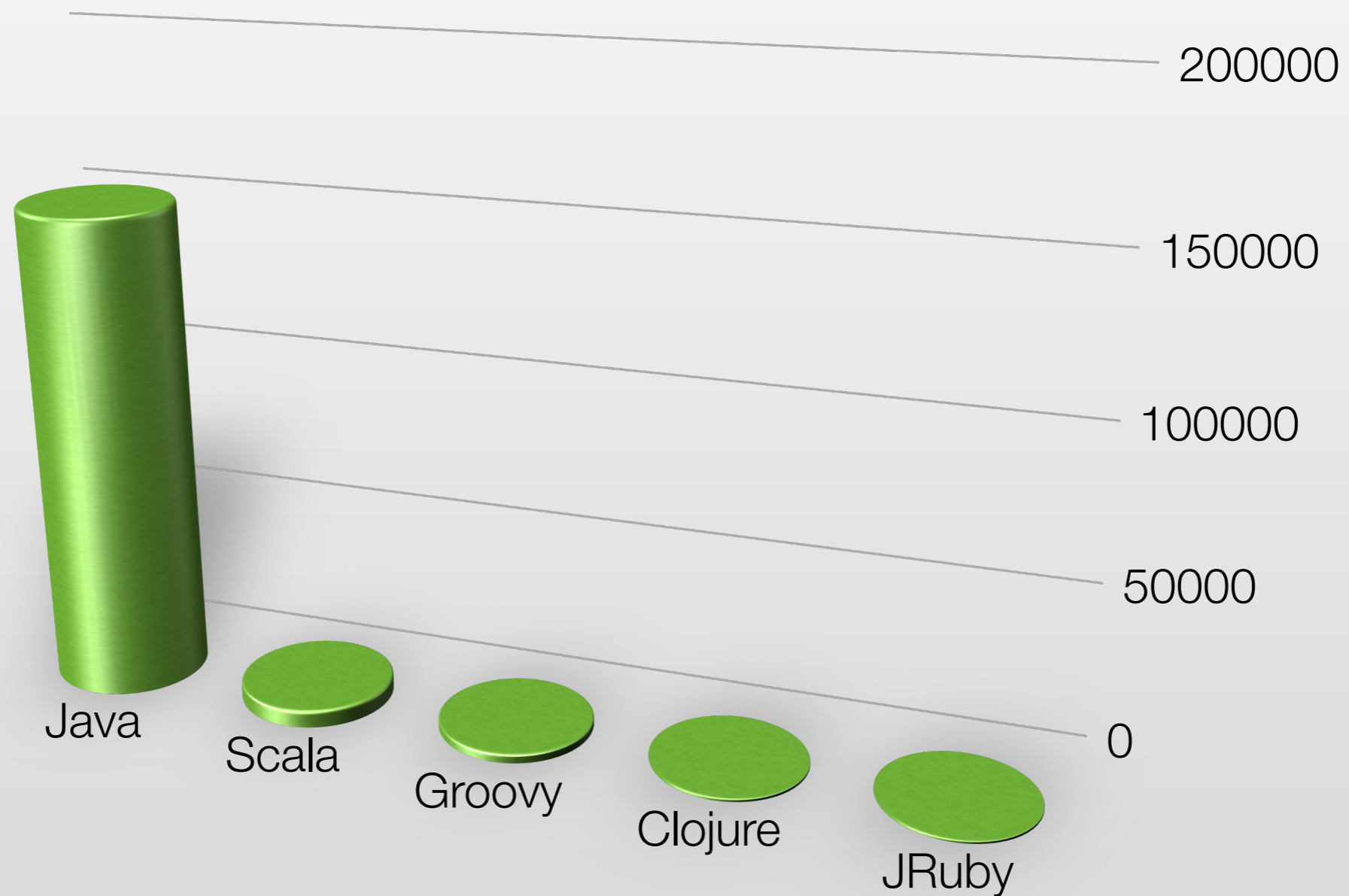
■ Tagged Questions (August 24, 2011)



StackOverflow



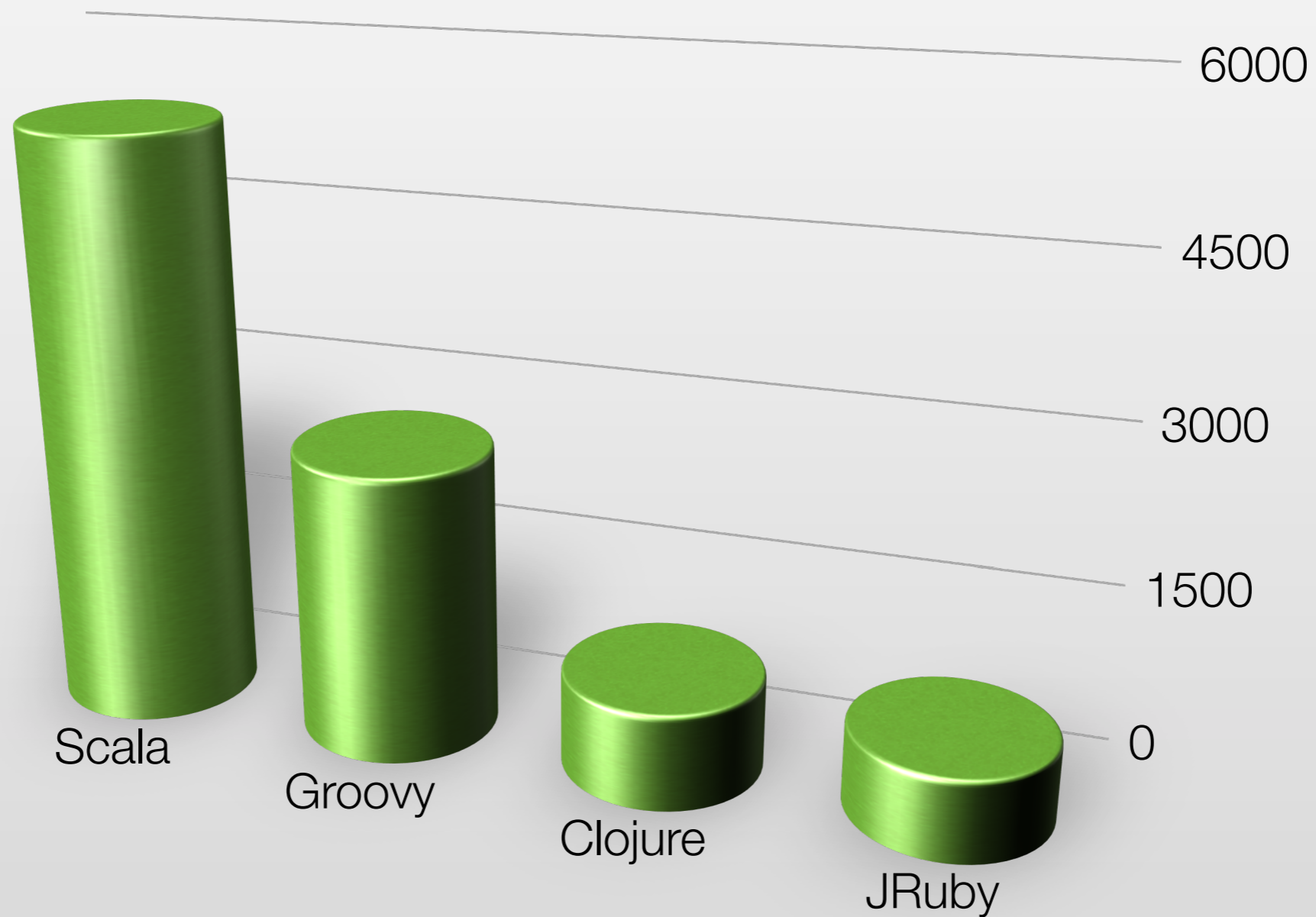
■ Tagged Questions (August 24, 2011)



StackOverflow

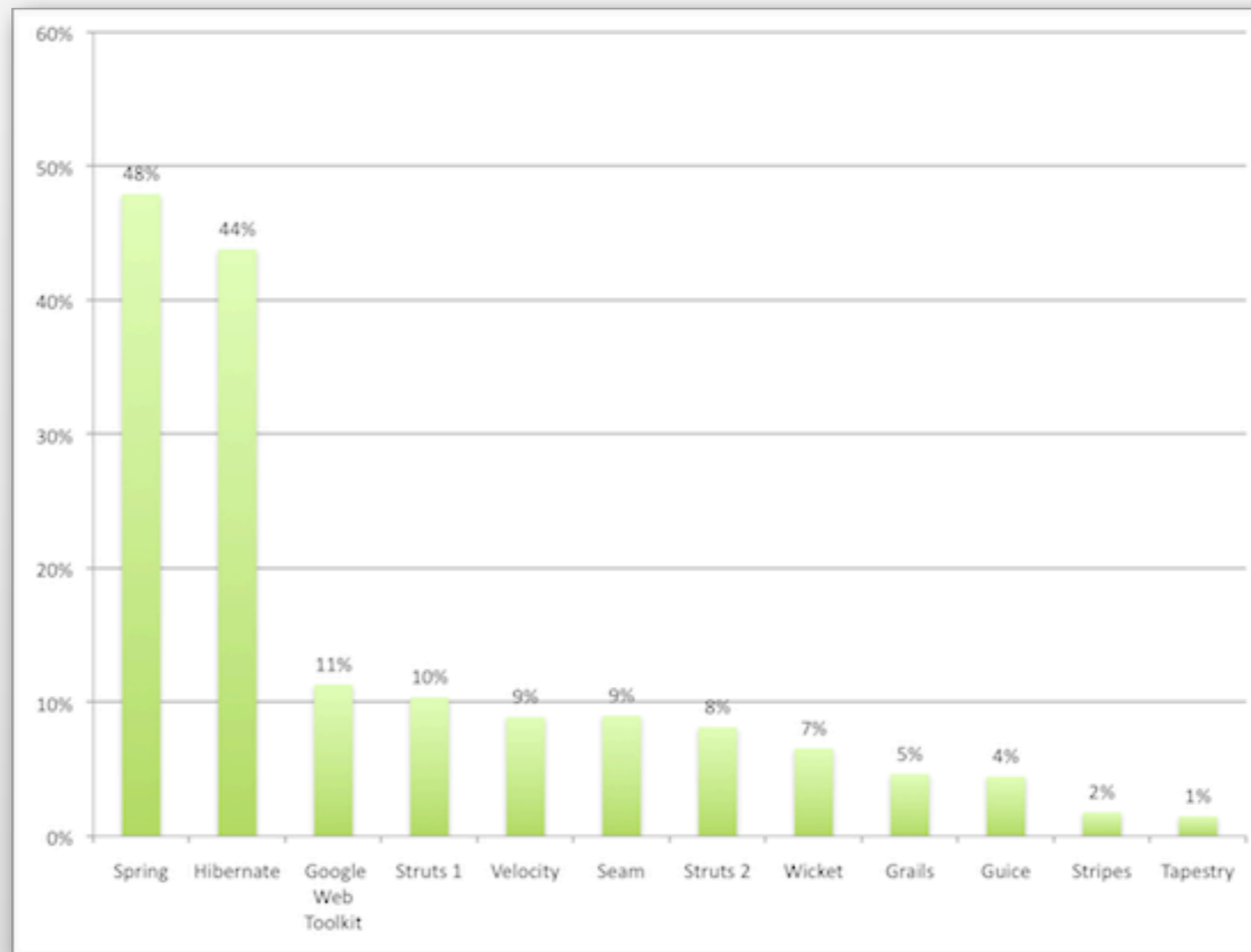


■ Tagged Questions (August 24, 2011)



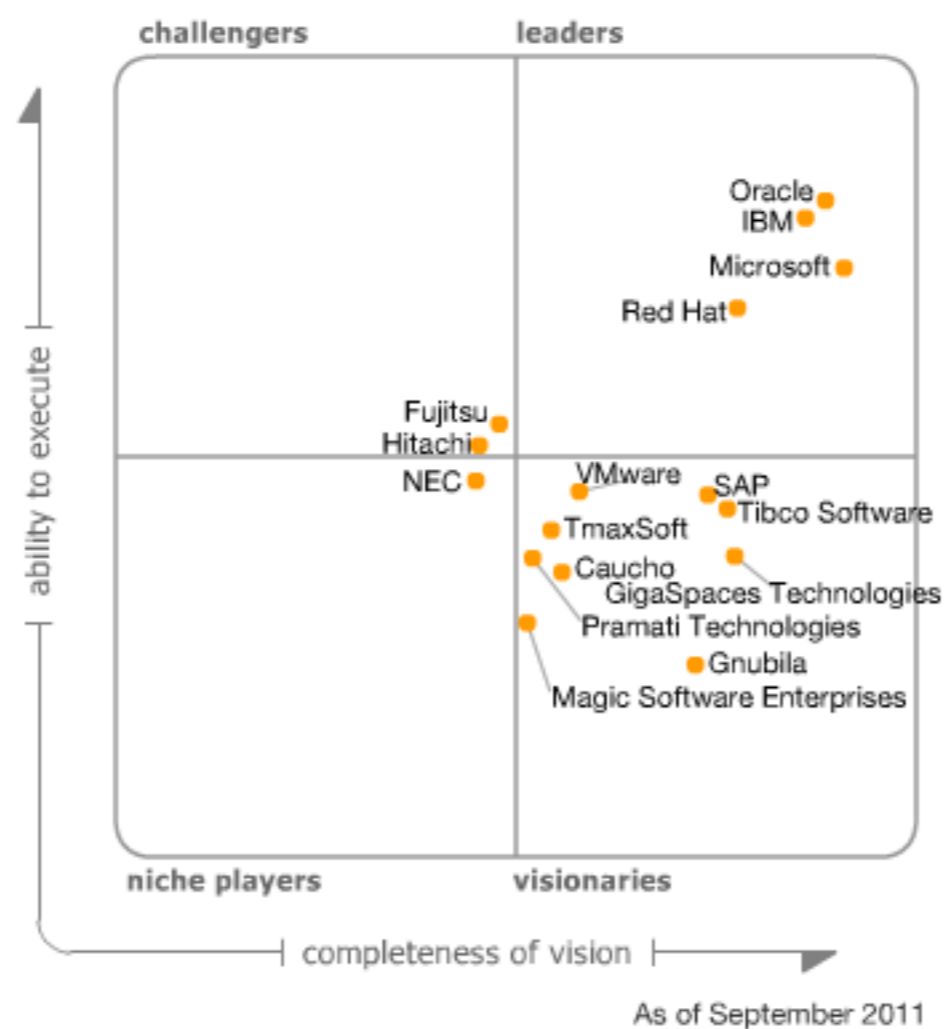


Framework Popularity



Source: ZeroTurnaround's Java EE Productivity Report 2010

What we need is...



Innovators



Struts™



Play! ▶

Modern Principles



The screenshot shows a web browser window with the following elements:

- Address bar: `blogs.atlassian.com/2012/01/modern-principles-in-web-development/`
- Navigation links: [Atlassian Home](#), [Get Help](#), [Plugin Exchange](#)
- Atlassian Blogs logo and search bar (Search Blogs)
- Article title:

Modern Principles in Web Development
- Author: By [Rich Manalang](#), Developer Advocate
About [Developer](#)
On January 18, 2012
- Social sharing: [+1](#) 11, [Tweet](#) 162, [Like](#) 8
- Text:

I've been kickstarting a bunch of small web apps lately. It seems like every time I start a new project, there's always something new that causes me to adjust my development principles. I thought it might be good to take a snapshot of what's "in" today. I like to think of web development phases starting from idea to delivery... all of it backed by strong principles of how to build great apps.
- Text:

The following are my core web development principles today:
- List of principles:
 - Designing for mobile first (even if you're not building a mobile app)
 - Build only single page apps
 - Create and use your own REST API
 - "Sex sells" applies to web apps

Web Developers



Matt Raible
@mraible

 Follow

If you call yourself a web developer, but don't know JavaScript or CSS, it's time to do some learning.

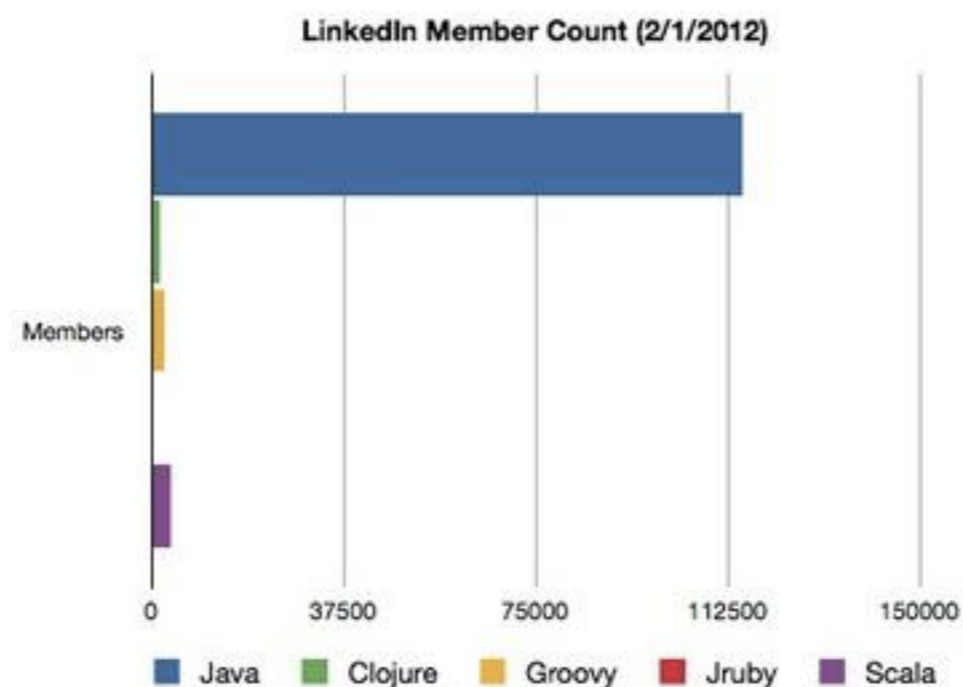
8:22 AM - 8 Feb 12 via Twitter for iPhone · Embed this Tweet

 Reply  Retweet  Favorite

Java Developers



“Java remains – in spite of the fragmented programming language landscape – a viable, growing language.”



<http://redmonk.com/sogrady/2012/02/08/language-rankings-2-2012/>

The Modern Web Developer



- ▶ ... embraces JavaScript
- ▶ ... is learning mobile frameworks
 - jQuery Mobile, Sencha Touch, PhoneGap or Native
- ▶ ... is using HTML5 and CSS3
- ▶ ... is developing REST APIs with the stateless framework that best supports their language
- ▶ **IE6 is dead**, IE7 isn't far behind...

You have to decide...



Are you a web developer?

I've seen the
FUTURE
It's in my
BROWSER



You have to decide...



Or are you a services developer?



Client-Side MVC



PH Client-Side MVC frameworks x

paulhammant.com/2012/02/13/client-side-mvc-frameworks-compared/

Inversionism Paul Hammant's blog

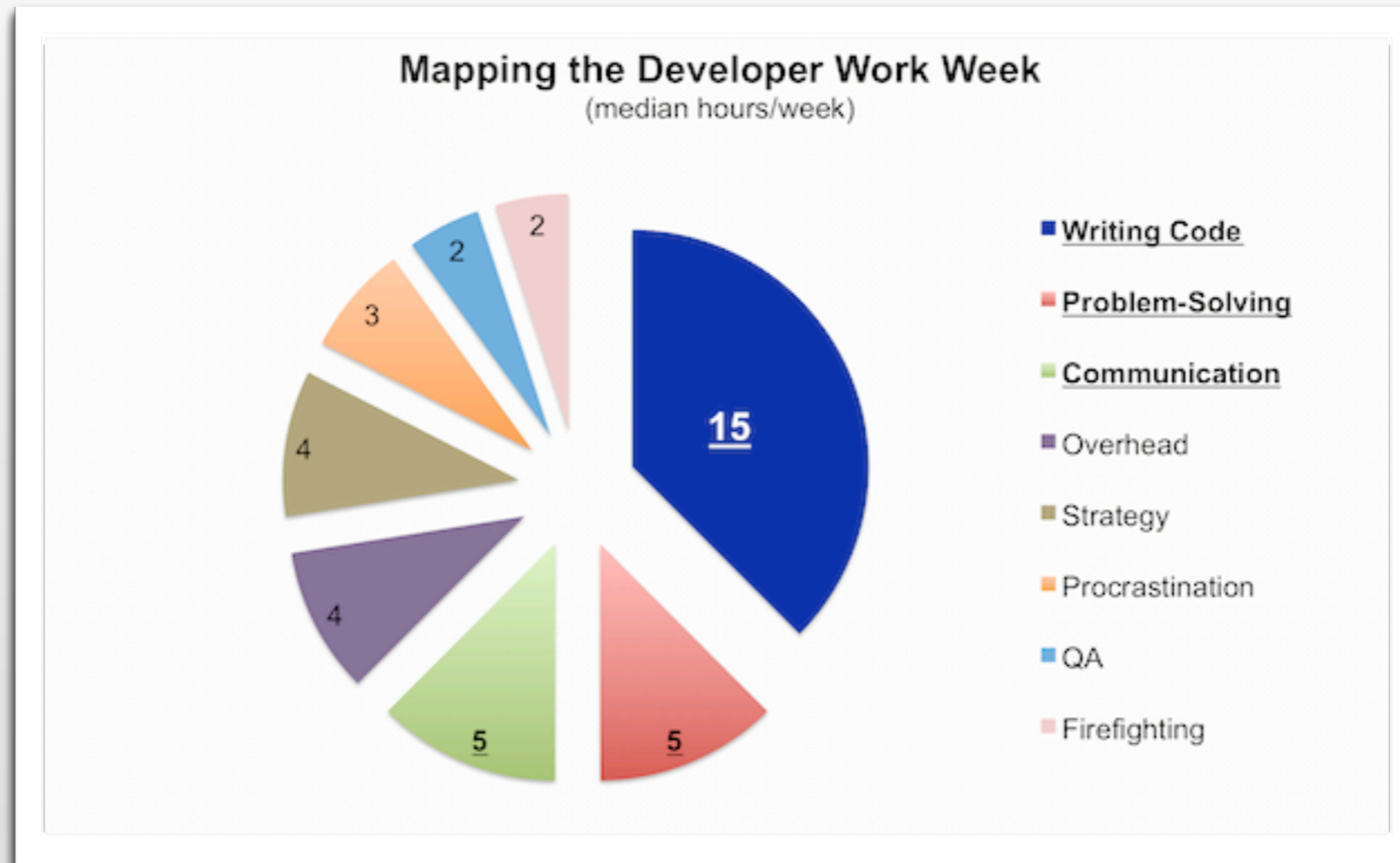
Client-Side MVC frameworks compared

Published: 13 Feb 2012

A month ago Gordon L. Hempton wrote about twelve JavaScript frameworks in the Client-Side MVC space. His rating criteria were different to mine. One that really sticks out is that I like the logic not forcing the template HTML to migrate to `<script>` tags. Depending on the sophistication of the app, I like to be able to see the app in a browser or DreamWeaver when the framework is **not running**. It gives me a way of gauging the composition of the app. It appeals to a WYSIWYG leaning that I have. I like my UI frameworks to be *built for designability* if you like.

Addy Osmani has a number of implementation of a TODO app on a github pages site. For composition purposes, this really is the definitive place presently. Using those, I'm going to scrutinize the HTML and how it the app looks without JavaScript. I checked out Addy's repo, then recursively deleted all javascript files, before loading the main page for each into a browser.

Developer Productivity

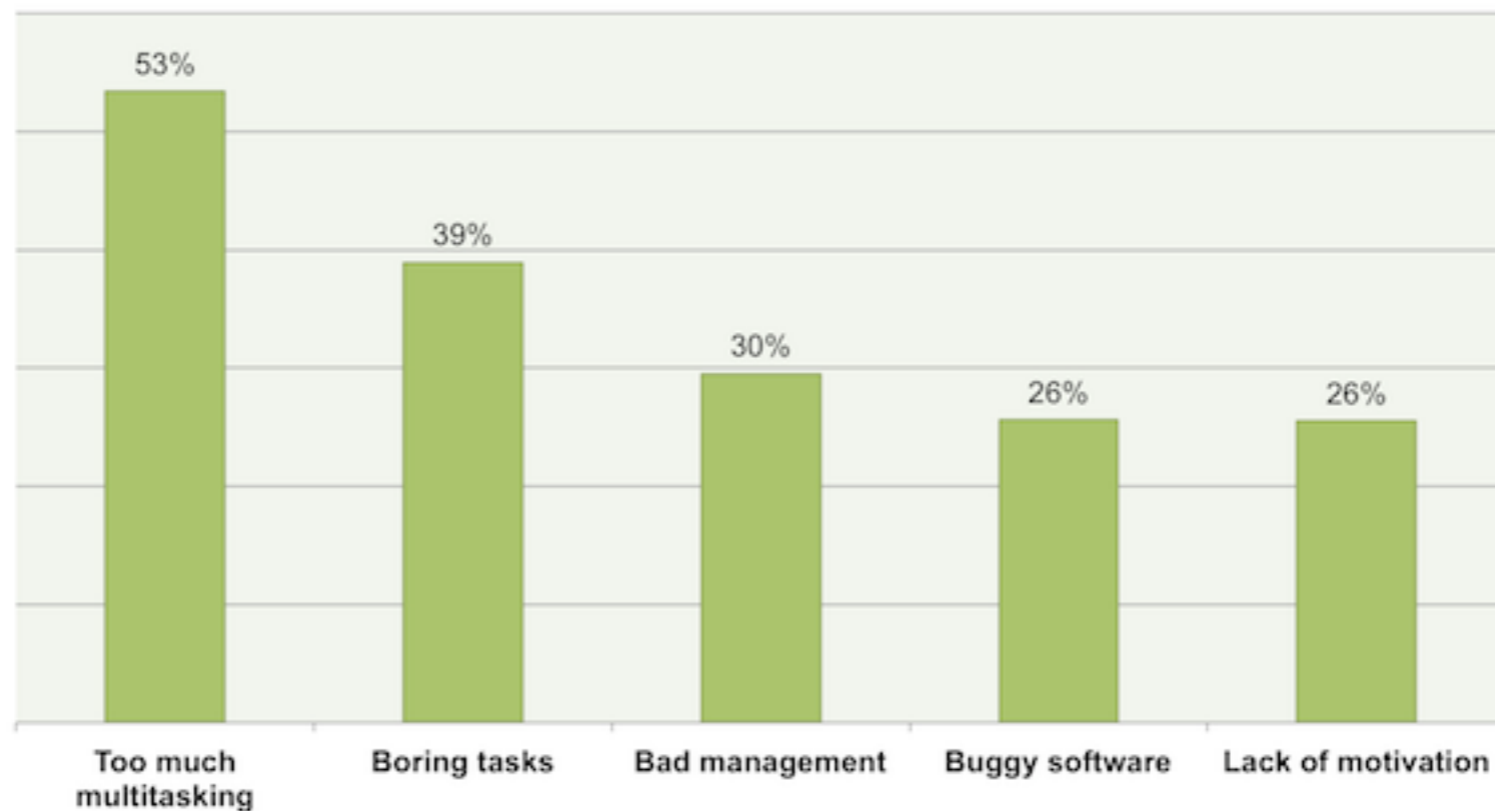


<http://zeroturnaround.com/blog/developer-productivity-report-part-1-developer-timesheet/>

Developer Productivity



What keeps you from doing your work?



<http://zeroturnaround.com/blog/developer-productivity-report-part-3-developer-efficiency/>

There is no “best” framework



Just lots of awesome choices...

Don't listen to me!



Choose your own!



- ▶ Prioritize a list of features that are important to your application.
- ▶ Pick 3-4 frameworks and do a 1-week spike with each, developing the same application.
- ▶ Document and rank each framework against your list of features.
- ▶ Calculate and choose!
- ▶ ... Or just pick one and get to work...

But don't forget...



The screenshot shows a web browser window with the address bar containing the URL `blogs.atlassian.com/2012/01/modern-principles-in-web-development/`. The page header features the Atlassian logo and the text "Atlassian Blogs" on the left, and navigation links "Atlassian Home", "Get Help", and "Plugin Exchange" on the right. A search bar labeled "Search Blogs" is also present. The main content area displays the title "Modern Principles in Web Development" in a large blue font. Below the title, there is a profile picture of Rich Manalang, followed by the text "By Rich Manalang, Developer Advocate", "About Developer", and "On January 18, 2012". To the right of the author information are social sharing buttons for "+1" (11), "Tweet" (162), and "Like" (8). The main text of the blog post begins with "I've been kickstarting a bunch of small web apps lately. It seems like every time I start a new project, there's always something new that causes me to adjust my development principles. I thought it might be good to take a snapshot of what's 'in' today. I like to think of web development phases starting from idea to delivery... all of it backed by strong principles of how to build great apps." This is followed by the sentence "The following are my core web development principles today:" and a bulleted list of four principles.

Modern Principles in Web Development

By Rich Manalang, Developer Advocate
About Developer
On January 18, 2012

+1 11 Tweet 162 Like 8

I've been kickstarting a bunch of small web apps lately. It seems like every time I start a new project, there's always something new that causes me to adjust my development principles. I thought it might be good to take a snapshot of what's "in" today. I like to think of web development phases starting from idea to delivery... all of it backed by strong principles of how to build great apps.

The following are my core web development principles today:

- Designing for mobile first (even if you're not building a mobile app)
- Build only single page apps
- Create and use your own REST API
- "Sex sells" applies to web apps

Questions?



▶ Contact

- <http://raibledesigns.com>
- @mraible

▶ Download

- <http://slideshare.net/mraible>

