Comparing Web Frameworks

JSF, Spring MVC, Stripes, Struts 2, Tapestry and Wicket

Matt Raible
matt@raibledesigns.com

Raible Designs

Today's Agenda

- Introductions
- Pros and Cons
- Sweetspots
- Web Framework Comparison: What each does well
- Conclusion
- Q and A

Introductions

- Your experience with webapps?
- Your experience with Java EE?
- What do you want to get from this session?
- Experience with Maven, Tomcat, Hibernate, Spring?
- Web Framework Experience:
    - Spring MVC, Struts, Stripes, WebWork, JSF, Tapestry, Wicket

webwork

WEB APPLICATION FRAMEWORK

Struts

Struts2

tapestry

Spring 2.0

My Faces

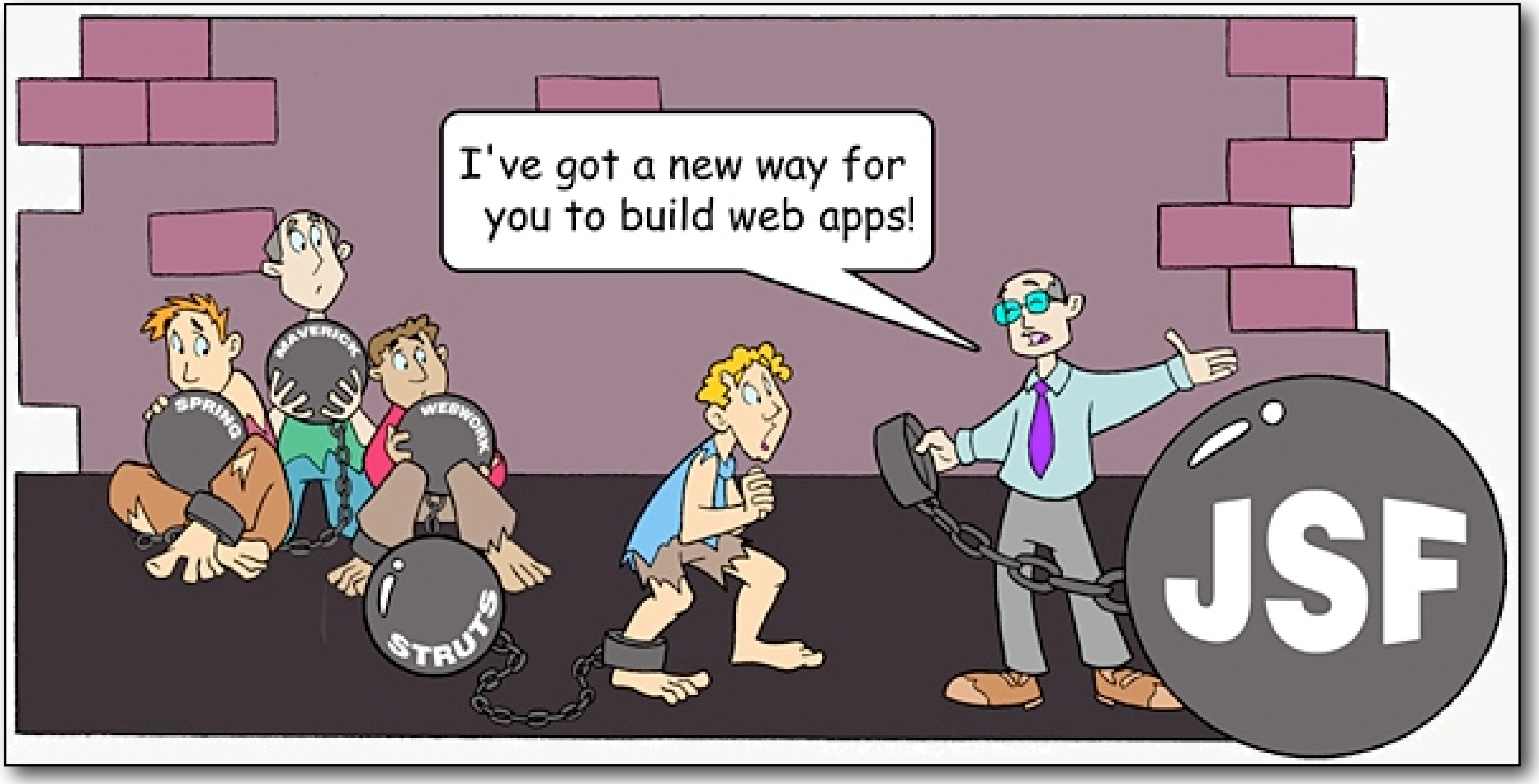
Stripes

WICKET

My Experience

- **Struts 1**: used since June 2001 - same time 1.0 was released.
- **Spring MVC**: used since January 2004 - before 1.0 was released.
- **Struts 2/WebWork**: used since July 2004.
- **Tapestry**: used since July 2004.
- **JSF**: used since July 2004 - both Sun's RI and MyFaces.
- **Stripes and Wicket**: learned them last week ;-)

Pros and Cons



JSF

- Pros:
  - Java EE Standard - lots of demand and jobs
  - Fast and easy to develop with initially
  - Lots of component libraries
- Cons:
  - Tag soup for JSPs
  - Doesn't play well with REST or Security
  - No single source for implementation

Spring MVC

- Pros:
  - Lifecyle for overriding binding, validation, etc.
  - Integrates with many view options seamlessly: JSP/JSTL, Tiles, Velocity, FreeMarker, Excel, XSL, PDF
  - Inversion of Control makes it easy to test
- Cons:
  - Configuration intensive - lots of XML
  - Almost too flexible - no common parent Controller
  - No built-in Ajax support

Stripes

- Pros:
  - No XML - Convention over Configuration
  - Good documentation (easy to learn)
  - Enthusiastic community
- Cons:
  - Small Community
  - Not as actively developed as other projects
  - Hard-coded URLs in ActionBeans

Struts 2

- Pros:
  - Simple architecture - easy to extend
  - Tag Library is easy to customize with FreeMarker or Velocity
  - Controller-based or page-based navigation
- Cons:
  - Documentation is poorly organized
  - Too much concentration on new features
  - Googling results in Struts 1.x documentation

Tapestry

- Pros:
  - Very productive once you learn it
  - Templates are HTML - great for designers
  - Lots of innovation between releases
- Cons:
  - Documentation very conceptual, rather than pragmatic
  - Steep learning curve
  - Long release cycles - major upgrades every year

Wicket

- Pros:
  - Great for Java developers, not web developers
  - Tight binding between pages and views
  - Active community - support from the creators
- Cons:
  - HTML templates live next to Java code
  - Need to have a good grasp of OO
  - The Wicket Way - everything done in Java

Sweetspots

Purpose of Experiment

- Discuss various open source Java web frameworks
- Highlight what each does well
- Debunk some myths
- Find out framework author's opinions of other frameworks
- Learn about the future direction of the framework
- Find out what the author's think of Ruby on Rails

Who Represented?

- JSF, Jacob Hookom
- RIFE, Geert Bevin
- Seam, Gavin King
- Spring MVC, Rob Harrop
- Spring Web Flow, Rob Harrop and Keith Donald
- Stripes, Tim Fennell
- Struts 1, Don Brown
- Tapestry, Howard Lewis Ship
- Trails, Chris Nelson
- WebWork, Patrick Lightbody
- Wicket, Eelco Hillenius

What's your "sweet spot"?

- **JSF:** For when you want to bring desktop-like functionality to the browser with the reliance of a standard specification and large amounts of third-party features.
- **Spring MVC:** Integrates a number of different technologies and as a result is applicable to a wide range of project types. It should be considered a strategic base platform for web application development.

What's your "sweet spot"?

- **Stripes:** Applications with lots of complex data interactions. Its type conversion, binding, and validation are very powerful and make it easy to manage large, complex forms and map them directly to domain objects, etc.
- **Tapestry:** Real strengths come through on medium- to large-sized projects (although you can have fun even on a single-page application). Those are the projects where you'll get leverage by being able to easily create new components.

What's your "sweet spot"?

- **WebWork:** Usually fits in best with small teams that are willing to get their hands dirty and learn a lot about the open source tools they use. WebWork is not meant for the “armchair programmers” who prefer drag-and-drop development.
- **Wicket:** Well suited for intranet/extranet applications, where the UI is relatively complex and where you want to make the best use of your developer resources.

What's your opinion?

The Smackdown

Evaluation Criteria

- **Ajax Support**: Is it built-in and easy to use?
- **Bookmark-ability**: Can users bookmark pages and return to them easily?
- **Validation**: How easy is it to use and does it support client-side (JavaScript) validation?
- **Testability**: How easy is it to test Controllers out of container?

Evaluation Criteria, cont.

- **Post and Redirect**: How does the framework handle the duplicate post problem?
- **Internationalization**: How is i18n supported and how easy is it to get messages in Controllers?
- **Page Decoration**: What sort of page decoration/ composition mechanisms does the framework support?
- **Community and Support**: Can you get questions answered quickly (and respectfully)?

Evaluation Criteria, cont.

- **Tools**: Is there good tool (particularly IDE) support for the framework?
- **Marketability of Skills**: If you learn the framework, will it help you get a job?
- **Job Count**: What is the demand for framework skills on dice.com and indeed.com?

Ajax Support

- Is Ajax support built-in and easy to use?
  - JSF: No Ajax support, use ICEfaces and Ajax4JSF
  - Stripes: No libraries, supports streaming results
  - Struts 2: Dojo built-in, plugins for GWT, JSON
  - Spring MVC: No libraries, use DWR & Spring MVC Extras
  - Tapestry: Dojo built-in in 4.1
  - Wicket: Dojo and Script.aculo.us (Wicket Stuff)

Bookmarking and URLs

- Using CMA allows users to bookmark pages. They can click the bookmark, login and go directly to the page.
  - JSF does a POST for everything - URLs not even considered
  - Stripes uses conventions, but you can override
  - Struts 2 has **namespaces** - makes it easy
  - Spring MVC allows **full URL control**
  - Tapestry still has somewhat ugly URLs
  - Wicket allows pages/URLs to be **mounted**

Validation

- Validation should be easy to configure, be robust on the client side and either provide good out of the box messages or allow you to easily customize them.
  - JSF has ugly default messages, but easiest to configure
  - Spring MVC allows you to use Commons Validator - a mature solution
  - Struts 2 uses OGNL for powerful expressions - client-side only works when specifying rules on Actions
  - Tapestry has very robust validation - good messages without need to customize
  - Stripes and Wicket do validation in Java - no client-side

Testability

- Spring and WebWork allow easy testing with mocks (e.g. EasyMock, jMock, Spring Mocks)
- Tapestry appears difficult to test because page classes are abstract, Creator class simplifies
- JSF page classes can be easily tested and actually look a lot like WebWork actions
- Wicket has WicketTester, a powerful solution
- Stripes has Servlet API Mocks and MockRoundtrip

Post and Redirect

- The duplicate-post problem, what is it?
- Easiest way to solve: redirect after POST
- Is there support for allowing success messages to live through a redirect?
    - Spring MVC allows you to add parameters to a redirect
    - Stripes, Tapestry and Wicket all have "flash" support
    - Struts 2 requires a custom solution
    - JSF requires a custom solution, i18n messages difficult to get in page beans

Internationalization

- JSTL's <fmt:message> tag makes it easy
- No standard for getting i18n messages in controller classes
- Stripes, Spring MVC and JSF use a single ResourceBundle per locale
- Struts 2, Tapestry and Wicket advocate separate files for each page/action
- JSF requires resource bundle to be declared on each page
- Tapestry's <span key="key.name"> is awesome

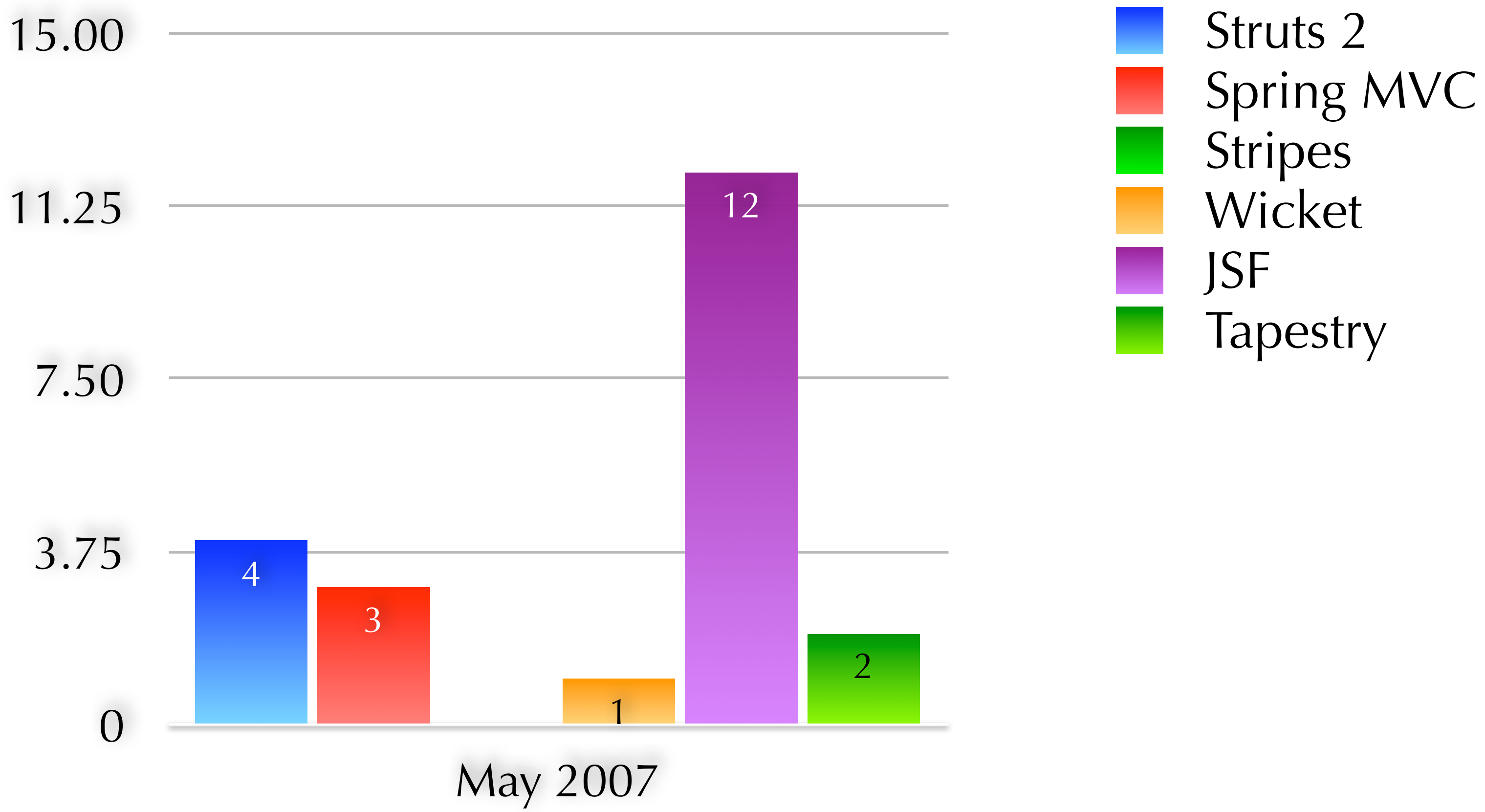
Page Decoration

- Tiles Experience: used since it first came out
- SiteMesh is much easier to setup and use
- Tiles can be used in Struts 2, Spring and JSF
  - Requires configuration for each page
- SiteMesh can be used with all frameworks
  - Requires very little maintenance after setup
- SiteMesh not supported or recommended for use with JSF, Tapestry or Wicket

Tools

- Spring has Spring IDE - only does XML validation, not a UI/web tool
- WebWork has EclipseWork
- Tapestry has Spindle - great for coders
- JSF has many, and they're getting better and better
- Stripes and Wicket don't have any official tools
- NetBeans has support for: Struts *, JSF (+Facelets), Tapestry and Wicket (no Stripes or Spring MVC)

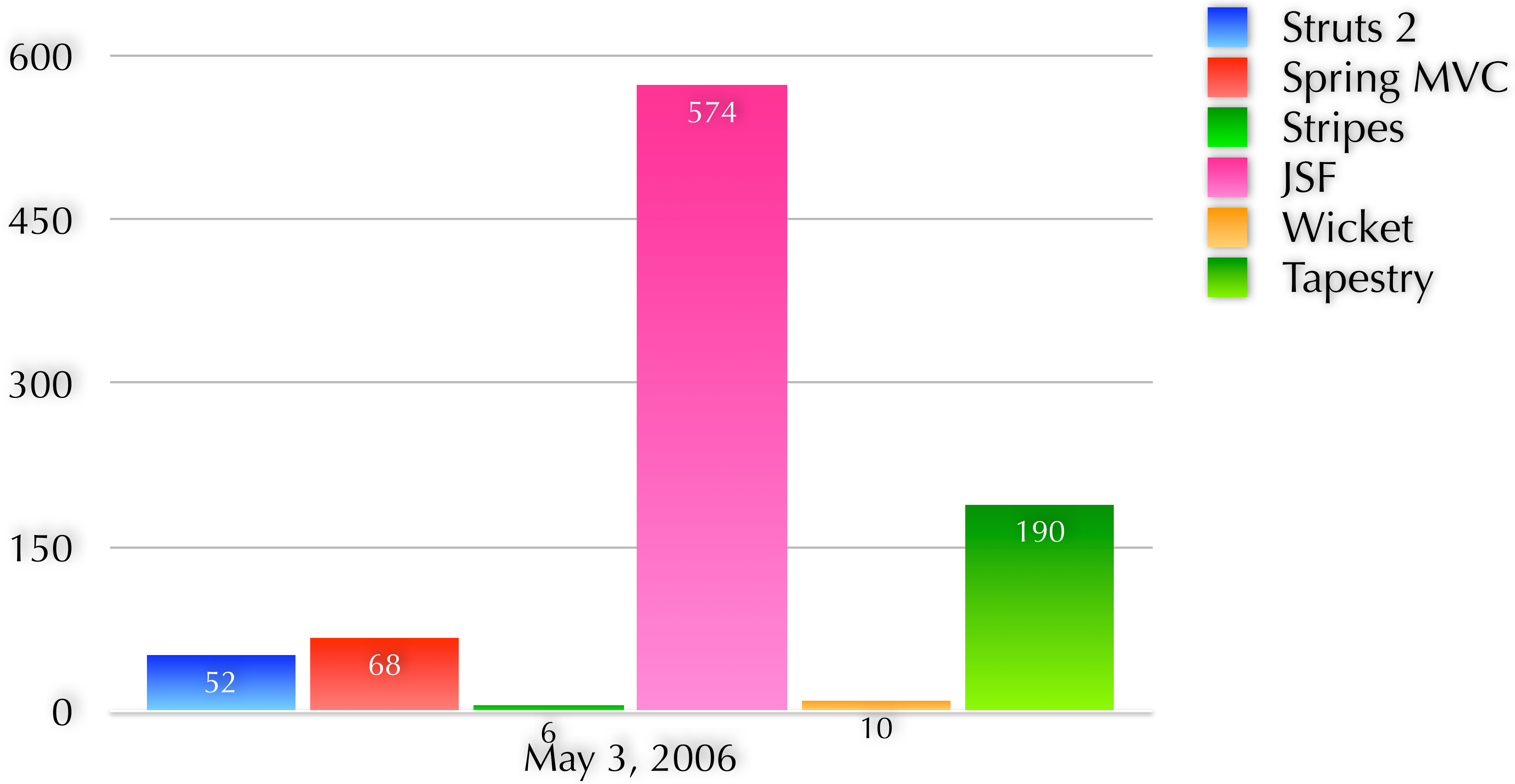
Tools Available



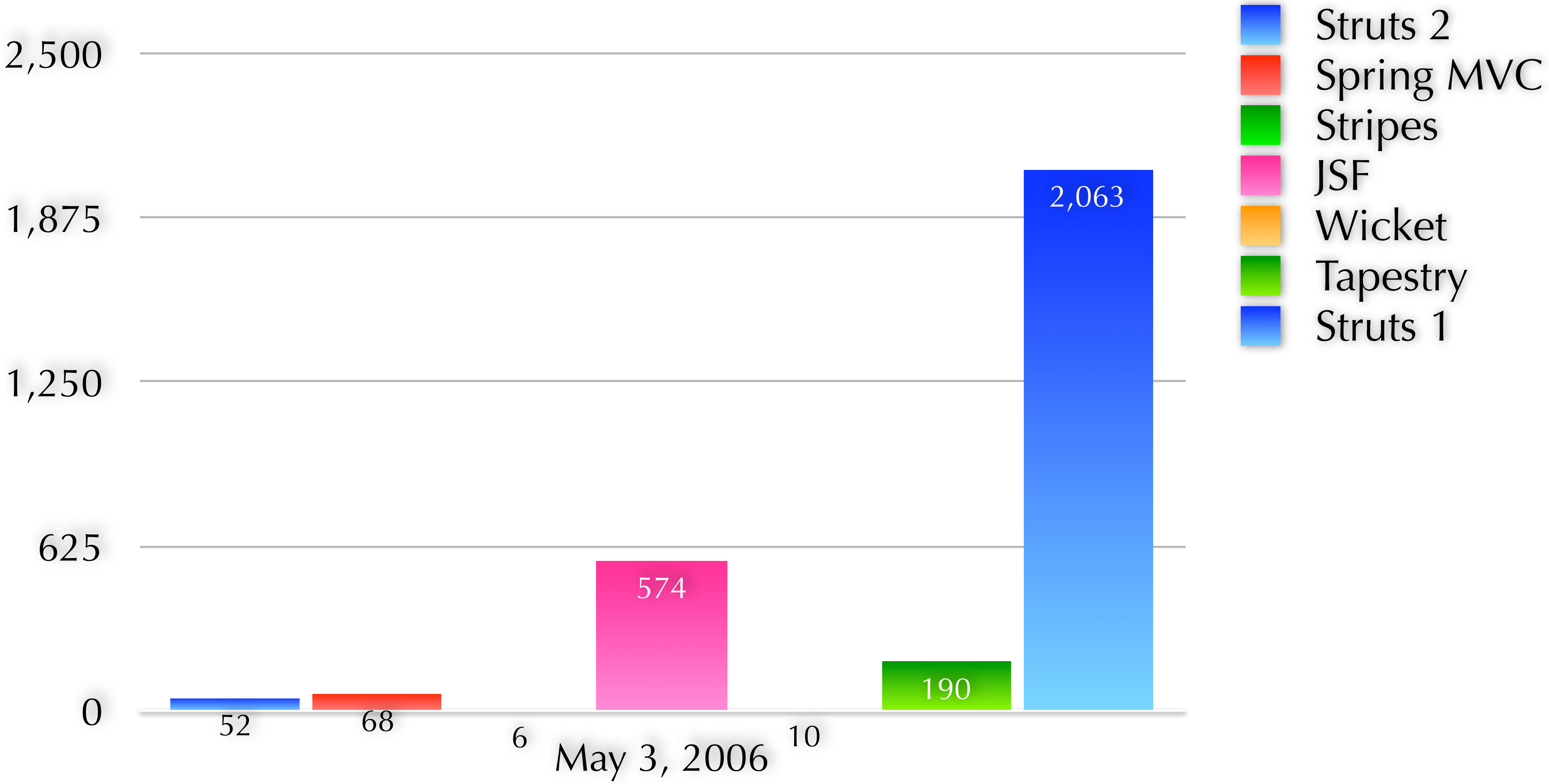
Marketability of Skills

- Struts 1 is still in high-demand and widely-used
- Spring is getting more press, but mostly due to the framework's other features
- JSF is quickly becoming popular
- Struts 2 is gaining ground, but very scarce on job boards
- Tapestry has increased in popularity in last couple years
- Wicket and Stripes are virtually unknown

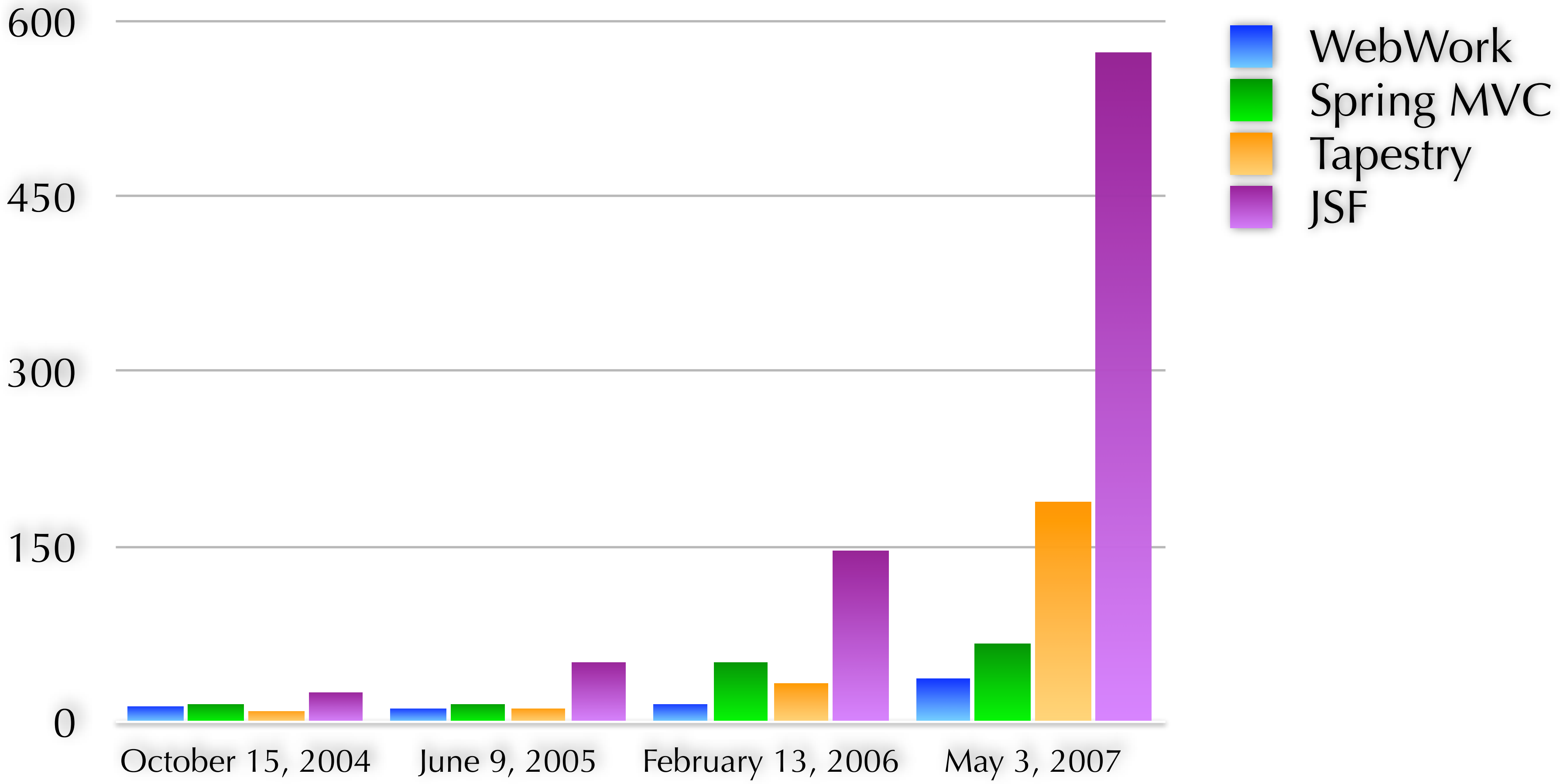
Dice Job Count



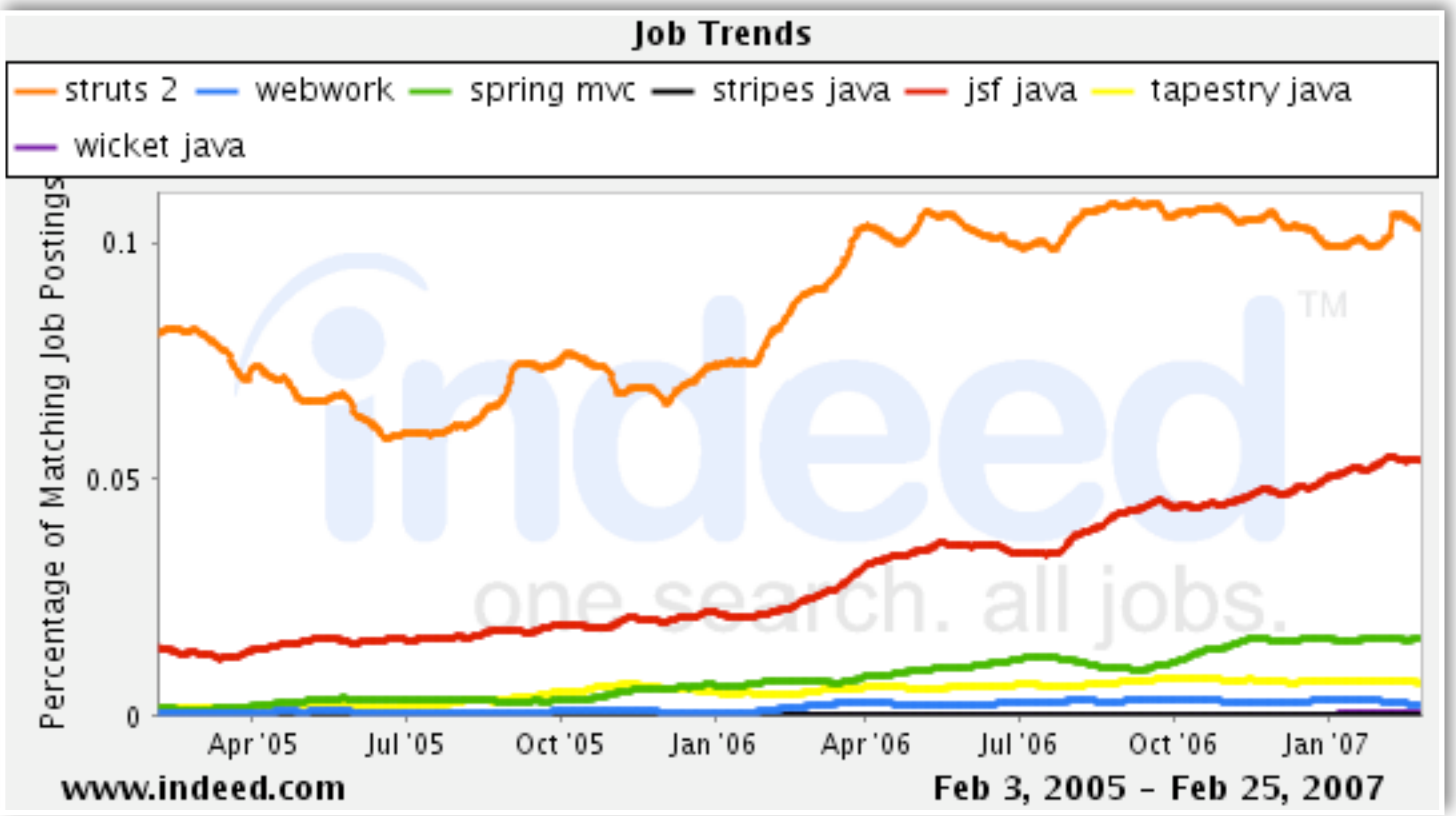
Dice Job Count w/ Struts



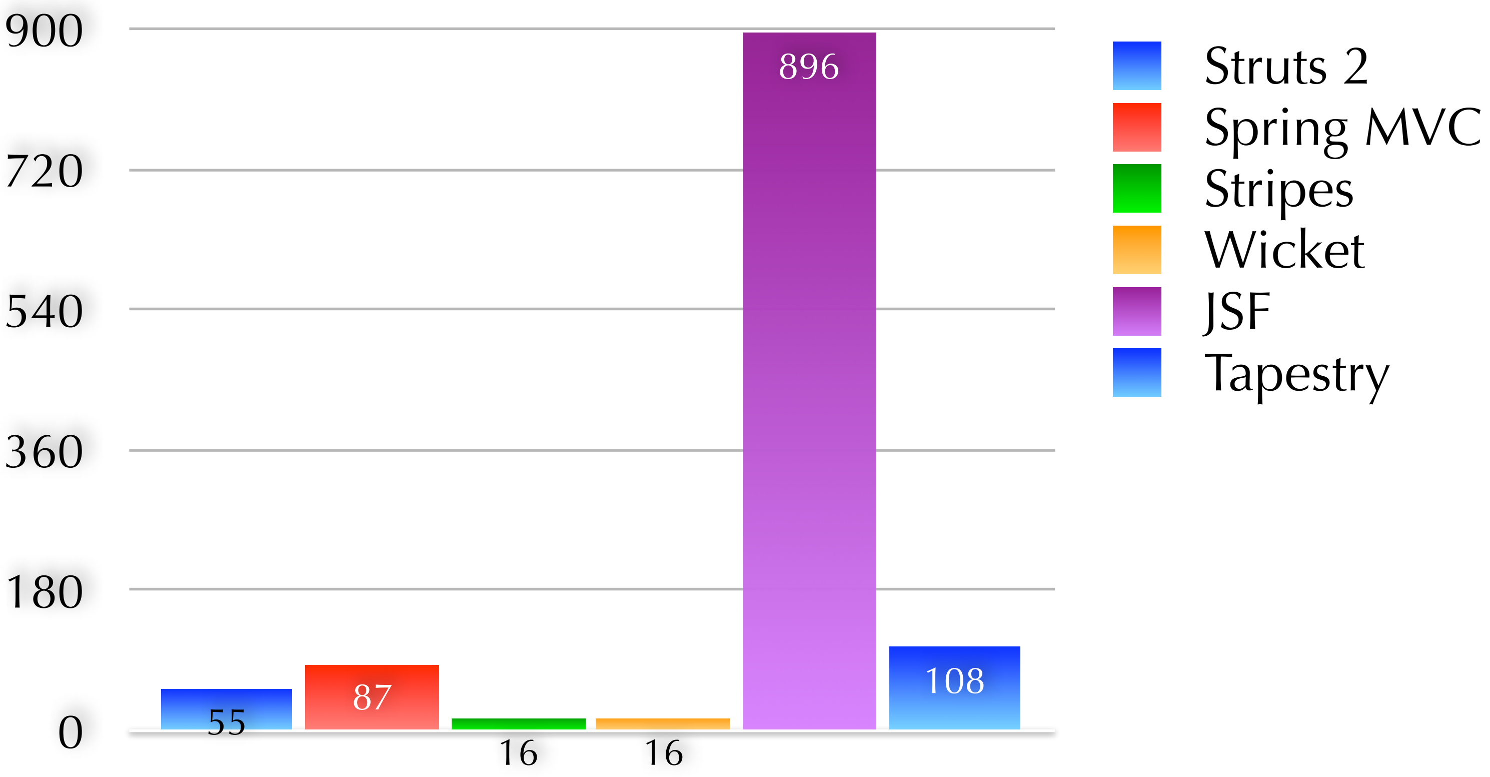
Job Trend sans Struts



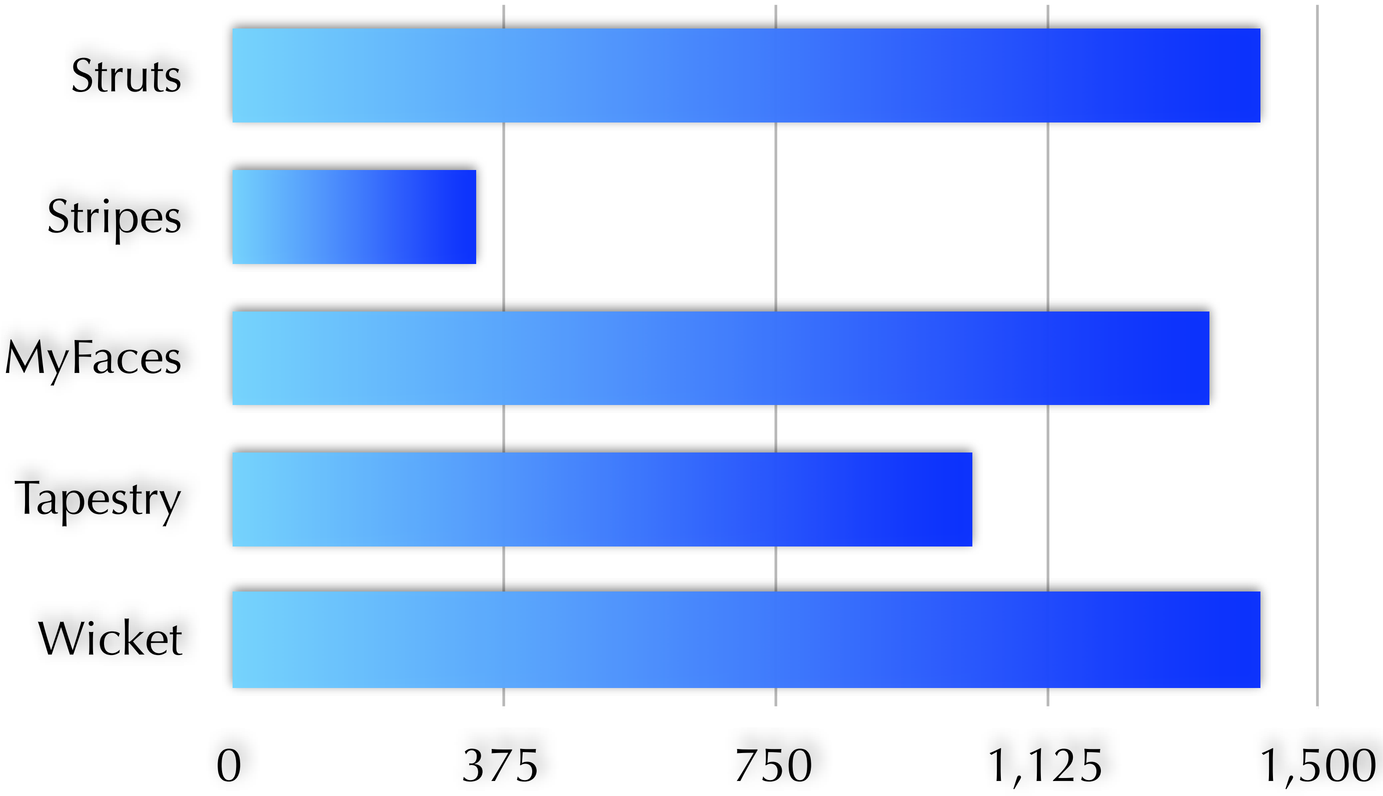
Job Trends


Job Trends
struts 2
webwork
spring mvc
stripes java
jsf java
tapestry java
wicket java
Percentage of Matching Job Postings
0.1
0.05
0
Apr '05
Jul '05
Oct '05
Jan '06
Apr '06
Jul '06
Oct '06
Jan '07
indeed
one search. all jobs.
www.indeed.com
Feb 3, 2005 – Feb 25, 2007

Employer Search on Monster.com
Resumes posted 4/3 ~ 5/3/2007

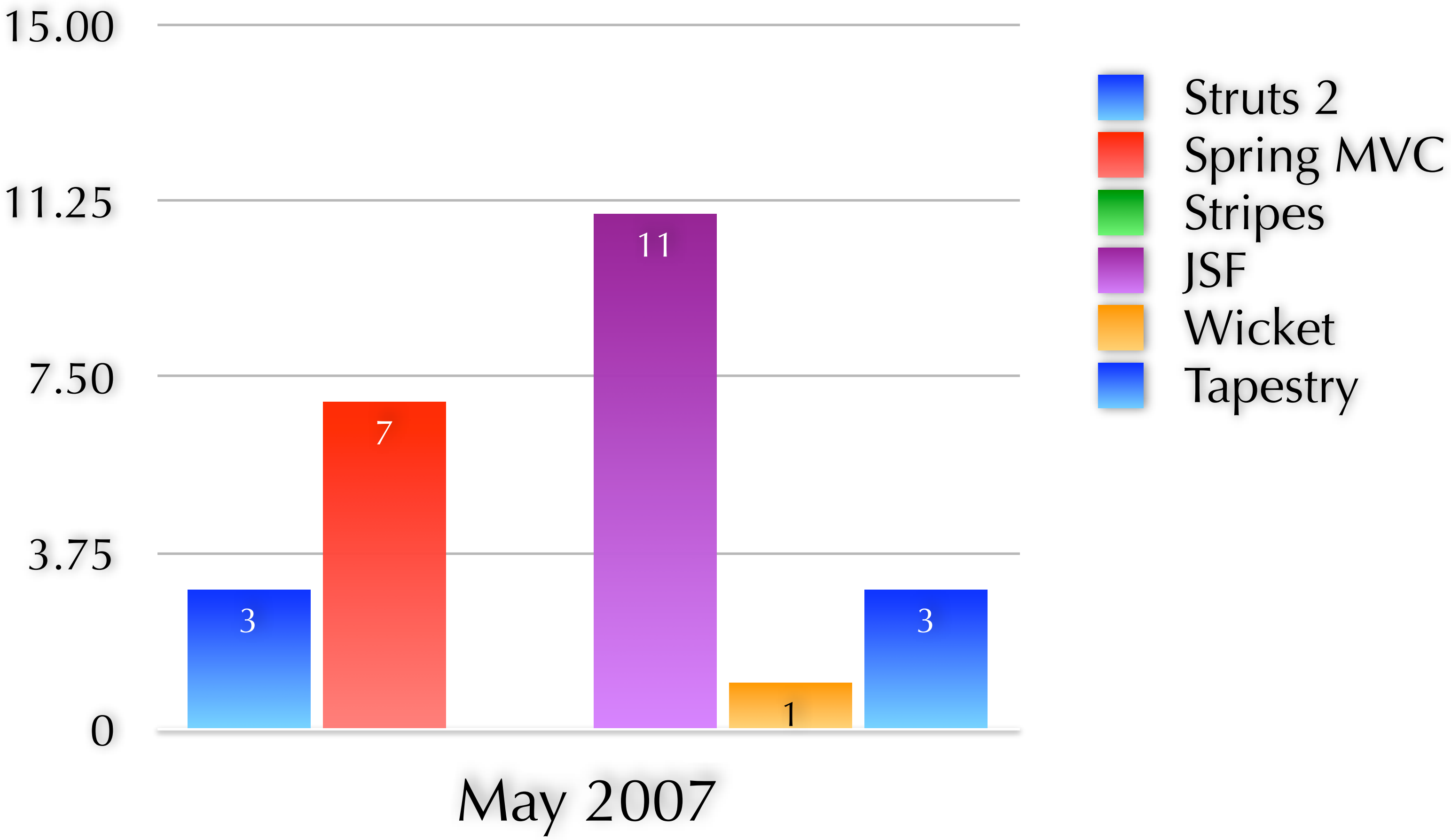


Mailing List Traffic



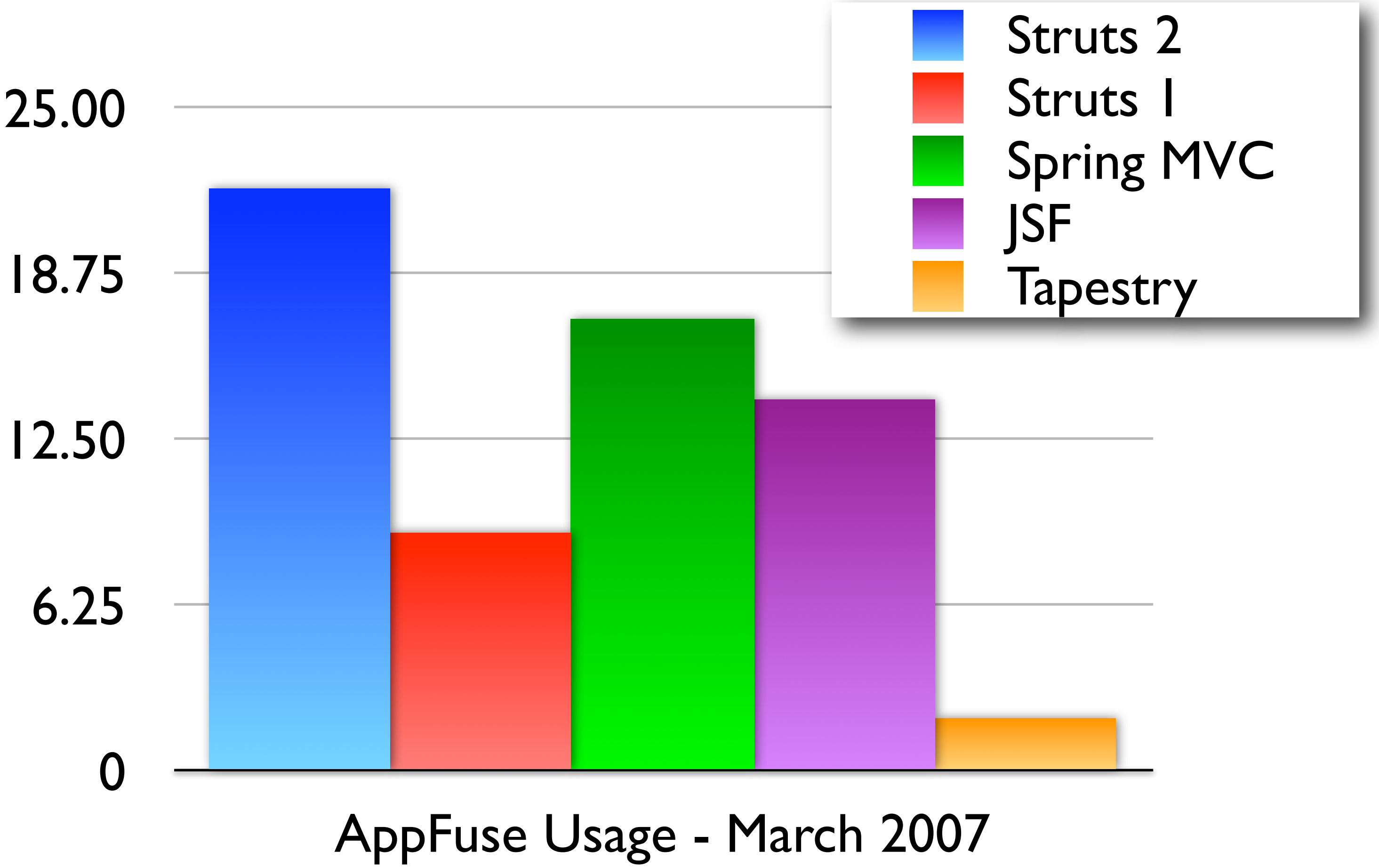
** Spring MVC is not listed here because they have a forum instead of a mailing list and I couldn't figure out a way to count the number of messages for each month.*

Books on Amazon



Which would I choose?

What do others think?



Resources

- Download this presentation
  - http://appfuse-light.dev.java.net/framework-comparison
- Struts - http://struts.apache.org
  - **StrutsTestCase**: http://strutstestcase.sf.net
- Spring MVC - http://www.springframework.org
  - **Spring IDE**: http://www.springide.org
  - **Gaijin Studio**: http://gaijin-studio.sf.net
- WebWork - http://opensymphony.org/webwork
  - **Eclipse Plugin**: http://sf.net/projects/eclipsework
  - **IDEA Plugin**: http://wiki.opensymphony.com/display/WW/IDEA+Plugin

Resources, cont.

- Tapestry - http://tapestry.apache.org
  - **Spindle**: http://spindle.sourceforge.net
- JSF - http://java.sun.com/j2ee/javaserverfaces and http://myfaces.apache.org
  - **Java Studio Creator**: http://sun.com/software/products/jscreator
  - **MyEclipse**: http://myeclipseide.com
- **IDEA**: http://www.jetbrains.com/idea
- **SiteMesh**: http://opensymphony.com/sitemesh

Resources, cont.

- Testing Frameworks
  - **JUnit**: http://junit.org
  - **EasyMock**: http://easymock.org
  - **jMock**: http://jmock.org
  - **jWebUnit**: http://jwebunit.sourceforge.net
  - **Canoo WebTest**: http://webtest.canoo.com
  - **Tapestry Test Assist**: http://howardlewisship.com/blog/2004/05/tapestry-test-assist.html
- **AppFuse** - http://appfuse.org

Books

- **Struts in Action**, Ted Husted and Team
- **Struts Live**, Rick Hightower and Jonathan Lehr
- **Spring Live**, Matt Raible
- **Pro Spring**, Rob Harrop and Jan Machacek
- **Spring in Action**, Craig Walls and Ryan Breidenbach
- **Professional Java Development with Spring**, Rod Johnson, Juergen Hoeller and Team

Books, cont.

- **WebWork in Action**, Patrick Lightbody and Team
- **Tapestry 101**, Warner Onstine
- **Tapestry in Action**, Howard Lewis Ship
- **Core JSF**, David Geary and Cay Horstmann
- **JSF in Action**, Kito Mann
- **Pro Wicket**, Karthik Gurumurthy

GWT

Grails

Seam

Flex

What’s Next?

Django

Ruby on Rails

Trails

OpenLaszlo

Who cares?

"If it works, use it!"

Questions?

matt@raibledesigns.com

Raible Designs