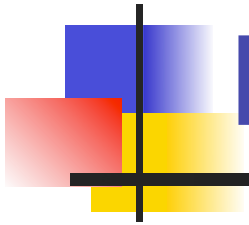
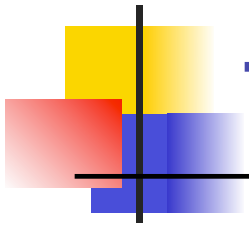


Choosing a JVM Web Framework

Matt Raible

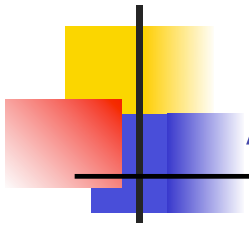
matt@raibledesigns.com

http://raibledesigns.com



Today's Agenda

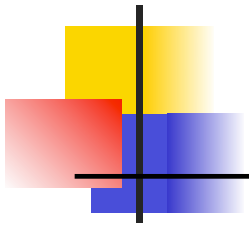
- Introductions
- The Problem
- Is there a solution to *The Problem*?
- How do you choose?
- 6 Important Factors
- Case Studies
- What do I think?
- Q and A



Audience Matters

- This talk is designed for teams trying to navigate the confusion in the JVM Web Framework space
- It is designed to be a discussion, not a presentation
- These are my opinions...
- ... and options are like ...
- Your opinions *will* influence mine
- So **please** share them!

Colorado Software Summit

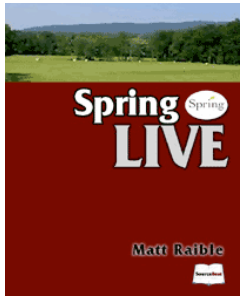


Introductions

- Your experience with web applications?
- 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 2, Stripes, JSF, Tapestry, Wicket, Rails, Grails, Flex

Who is Matt Raible?

- Power user of Java Open Source Frameworks
- Author of *Spring Live* and *Pro JSP 2.0*
- Founder of AppFuse and AppFuse Light
- Member of Java EE 5, JSF 1.2 and Bean Validation Expert Groups
- Committer on Apache Projects: Roller and Struts
- Java Blogger since 2002



Colorado Software Summit

The Problem

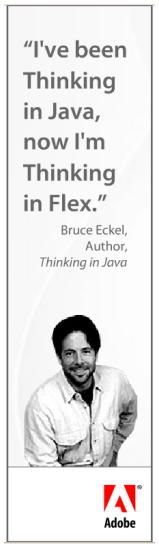
webwork Struts

Struts 2

tapestry

RAILS

Spring 2.0



My Faces

GRAILS

Stripes

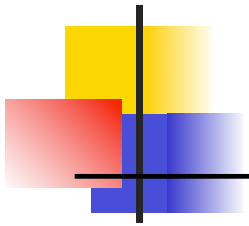
ZK Simply Rich

WICKET

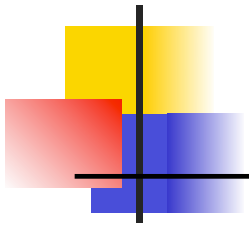
Google Code

Google Web Toolkit

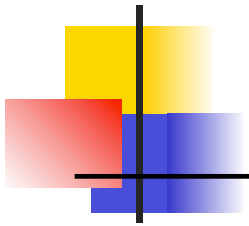
RIFE



How do you choose?

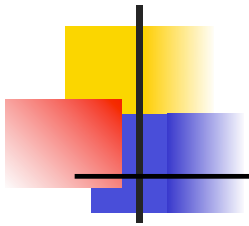


Eliminate, don’t Include



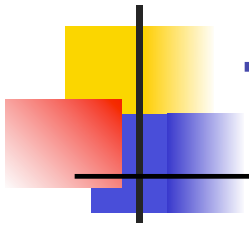
6 Important Factors

- Request, Component or RIA Framework
- Ease of Development
- Project Community
- Project Future and Roadmap
- Maintenance
- Technical Features



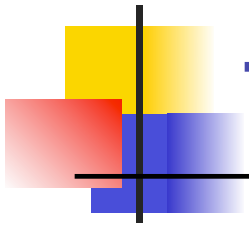
Request, Component or RIA

- Are you developing...
  - ... a consumer facing site?
  - ... a heavy forms, desktop-like, application?
  - ... a media-rich site?
- If the framework is built for state-fulness, should you use it in a stateless architecture?



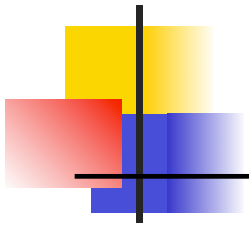
Types of Applications

- High-traffic, internet facing, infinite scalability
- Intranet-based, behind the firewall, few users
- Products, to be maintained for 5-10 years
- Legacy Backend
- Others?



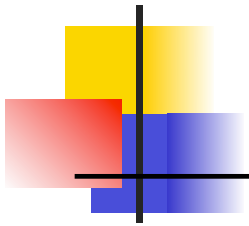
Types of Frameworks

- Request Based Frameworks
  - Struts 1/2, Spring MVC, Rails, Stripes
- Component Based Frameworks
  - JSF, Tapestry, GWT
- Rich Internet Applications
  - Flex, OpenLaszlo
- One Size Fits All
  - Grails



Do they matchup?

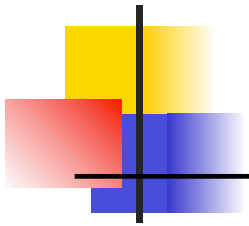
- High-traffic, internet facing, infinite scalability
  - ➢ **Request-based frameworks**
- Intranet-based, behind the firewall, few users
  - ➢ **Component-based frameworks**
- Products, to be maintained for 5-10 years
  - ➢ **Largest Community, Most Vendor Support**
- Legacy Backend
  - ➢ **Same Language as backend**
- Others?



Ease of Development

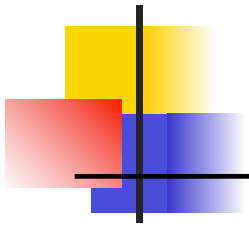
- Can new developers learn the basic concepts on their first day?
- Does the framework follow the principle of least surprise?
- Are you migrating from an existing web framework?

Colorado Software Summit



Project Community

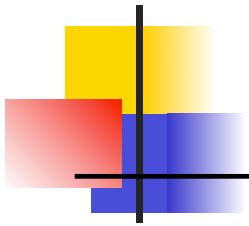
- Is there a company backing the project?
- Are vendors jumping on board? What about consultants?
- Does it have healthy mailing list traffic?
  - Users and developers are both important
- Are there frequent releases?
- What's the level of real-world usage?
- Are there books written about it?



Project Future & Roadmap

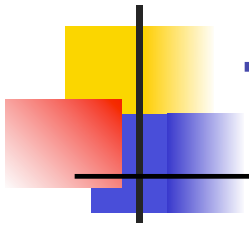
- Does the project have ambitious goals for its future?
  - JSF 2 and Tapestry 5
  - Both will sacrifice backwards compatibility
- Are you willing to be an early adopter?

Colorado Software Summit



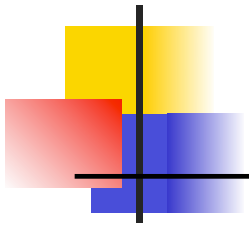
Maintenance

- Is the framework easy to test?
- Is backwards compatibility a concern for framework developers?
- Are releases backwards compatible?
- If not, is there an upgrade guide or adapter?
- Can 5 years go by?



Technical Features

- Are you migrating from an existing web framework or creating a new application?
- If creating new, are you doing "greenfield" development?
- If migrating, does the new framework have all the features your current framework has?
- If migrating, can you migrate one module at a time?
- Is the framework's "full-stack" a feature or a barrier?

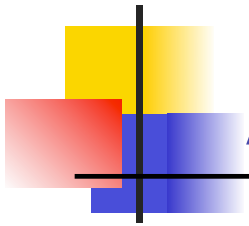


Don’t believe the Hype

- Don’t believe blogs and articles
- Try it yourself
- Believe developers, not evangelists
- Believe developers that are experienced with the framework and have used it in production
- Beware of corporate interests - they can twist marketing
- Are books a good sign?

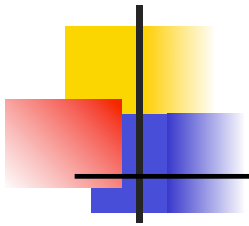
Best Tool for the Job

- Do you believe in Application Categorization and Framework Specialization?
- If so, pick the top players in that category...
- ... and prototype!
- If prototyping is painful, switch
- If prototyping isn't painful, is it work trying others?



After you've decided

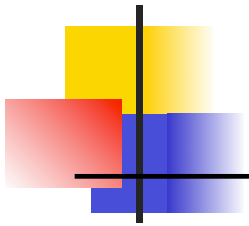
- Document the reasons for your decision
- Allow developers to challenge it
- Allow your prototype to be written with other frameworks
- Don't be afraid to try new frameworks
- Don't be afraid to use old frameworks
- Don't be afraid to keep your existing framework



Case Studies

http://raibledesigns.com/rd/entry/choosing_a_jvm_web_framework1

Colorado Software Summit



Wicket over JSF

I had a project I started with JSF and spent about 2 weeks coding the dao and service layer when finally starting the work on the UI. I ran into some nasty problems I couldn't came around (displaying buttons in a datatable that work even when the backing bean is not stored in the session). Then I found Facelets which solved some of my JSF problems, but not all - although it was nicer than plain old jsp-jsf soup.

...

I played around with wicket 1.0 some months before this project and thought I would give it a try. I spent about 6 days on the UI with JSF when doing the switch to wicket and in about 6-7 hours all of my UI pages were converted from JSF to wicket.

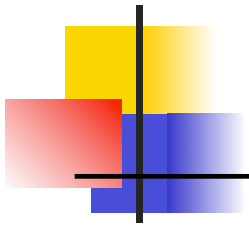
...

We like the technology very much, it's really easy and straightforward *once you get it*.

Colorado Software Summit

Janos Cserep

http://www.metaprime.hu

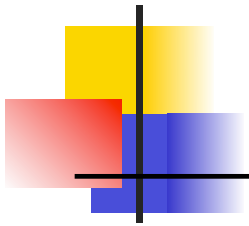


Stripes over Struts

My favourite option is Stripes. I have tried Struts (1.x) and some home-grown frameworks before but I don't have any reason to switch to other framework since I've met Stripes. It fully satisfy my needs. Easy integration with AJAX, easy configuration and yet the framework allows for easy extension as well. Easy integration with Spring and Hibernate. Nice validation. I have used Stripes for more then 5 projects now and the fact that I haven't find any show-stoppers yet is self-explanatory I guess. For me Stripes is the real option when the project does not require other framework to be used.

Lukas Vlcek
http://blog.lukas-vlcek.com

Grails over Struts & Spring MVC



I have used Struts 1.x with hibernate and a home made service layer. I have also used Spring + Hibernate + SpringMVC (AppFuse 1.9.x). Nowadays I am using Grails 0.5.x (but have used it since Grails 0.3).

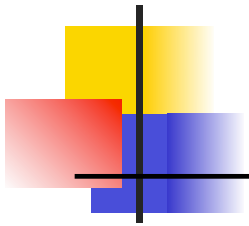
...

...my elimination process was the balance between productivity and production environment stability.

...

In terms of productivity, I haven't seen anything like Grails in JVM platform. It is so productive to develop with. And fun! A Grails app is very smooth to maintain. If you like Code By Convention over configuration, Grails is great. But if you need to do something different from the convention, you can configure Hibernate, or Spring with their xml config files.

Felipe Nascimento
http://felipenasc.blogspot.com



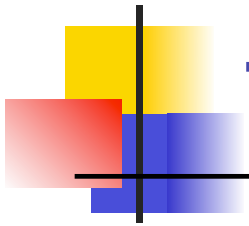
RIFE over Wicket

I've been using RIFE for a few years now and it has treated me extremely well. It is a single solution that includes most things you'd need for building J2EE web applications. I have tried a few other solutions, which most recently was Wicket, and found that the time/code it took to get something done was much greater than with RIFE.

...

I feel like RIFE's templating design and syntax spoils me. When I tried to convert a RIFE basic page to Wicket, I did it with much less Java and HTML than I did with RIFE. I feel spoiled by RIFE's very simple syntax and markup support. No need to use special tags in my HTML or anything.

Jeremy Whitlock
http://www.thoughtspark.org



Tapestry *vs.* JSF and Wicket

We spent some time (almost a week) in the jungle of Java Web Framework, just reading comparison, looking at the "ecosystem" of the framework (community, project linked to it, press buzz, etc). Quite quickly, the choice list drop to JSF, Wicket and Tapestry.

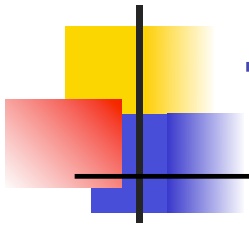
...

Wicket as a really good press, and was a kind of buzz work (not as bussing as RoR, but intrigating). Tapestry was the old one component framework, the one that open the way, and I had several friends that really love it, from a long time. JSF was the "obvious" choice, the norm, it has to be part of the choice.

...

Quickly, JSF was dropped : too complex, too huge, too much XML, it reminded us the worst day of Struts or EJBs.

continued on next page ...



Tapestry *vs.* JSF and Wicket

Wicket was impressive. The mailing list is really responsive, the approach of the framework is innovative (at least, for me who doesn't have any Swing background), the separation between code and style is really strict, etc.

...

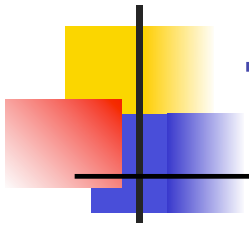
Drawbacks: Wicket is hard, at least in the beginning. Stateful by default.

...

Tapestry 5 was... Wow. It was the effect it did to me and the other tester. Its simplicity is amazing : its a joy to create new components (simple POJOs with some conventions and annotation), test them and reuse them along your application. The separation between HTML and logic is rather good, the reload facility a must have (Wicket also have this), the injection framework is amazing.

Francois Armand

Colorado Software Summit



Tapestry *vs.* JSF

I did an extensive study for a dissertation, lasting a year on web frameworks - but particularly Tapestry 4.

...

I also wrote a complex web application spanning over 1000 personal man hours. My findings are related to productivity and interoperability as these were seen as perhaps the most important areas for businesses. Tapestry was a success for the project, but had two critical failings: 1. Backwards compatibility 2. Documentation and project resources.

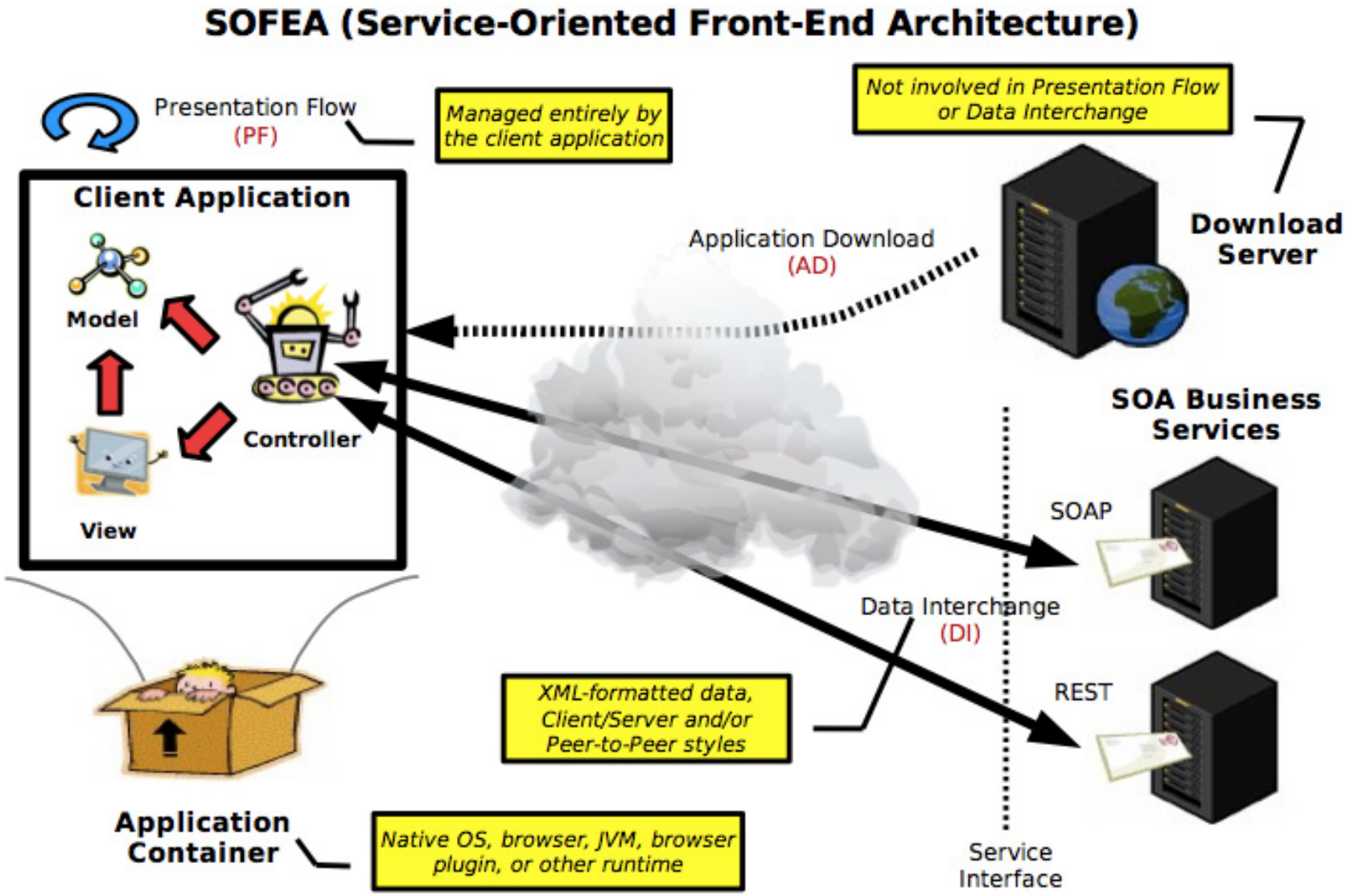
...

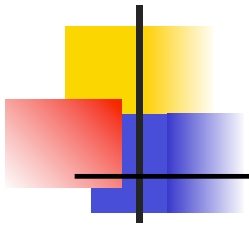
JSF is the most serious contender for longer term investment by a company in a web framework, yes it is immense, but they have heavy investment and rock solid documentation and training material. Once you are using it, it is really simple.

Peter Stavrinides

What about SOFEA?

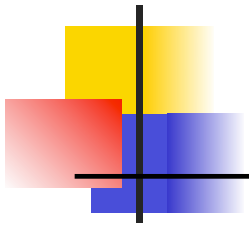
- Web Architecture for SOA Applications
- Client should be downloaded from a web server
- Presentation Flow should be on the client
- Data Interchange should happen via REST or SOAP
- http://wisdomofganesh.blogspot.com/2007/10/life-above-service-tier.html





SOFEA Frameworks

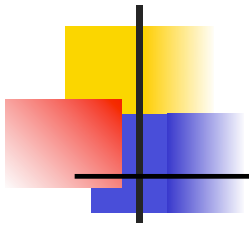
- DHTML/AJAX frameworks for Current Browsers
  - Hand-coded with third party JavaScript libraries
  - Google Web Toolkit (GWT, GWT-Ext)
  - TIBCO General Interface Builder
- XML Dialects for Advanced Browsers
  - XForms and XHTML 2.0
  - Mozilla XUL
  - Microsoft SilverLight/XAML



SOFEA Frameworks

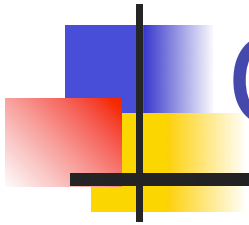
- Java frameworks
  - Java WebStart (with/without Spring Rich Client)
  - JavaFX
- Adobe Flash-based frameworks
  - Adobe Flex
  - OpenLaszlo

What do I think?



Conclusion

- The future is bright because of all the competition
- Developers should know more than one web framework
- You should try a framework before dissing it
- The plethora of web frameworks is a good thing!
- Doing proper research can save time and money
- Testing is the best path to future maintenance



Questions?

matt@raibledesigns.com

http://raibledesigns.com

Download presentation from:
http://raibledesigns.com/rd/page/publications