

# Choosing a Web Framework



By : Zvika Markfeld & Adi Baron

# Agenda

---

- Introduction
- The Problem
- How do you choose?
- Important Factors
- Q & A



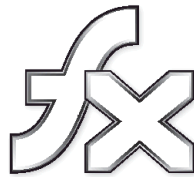
# Introduction

---

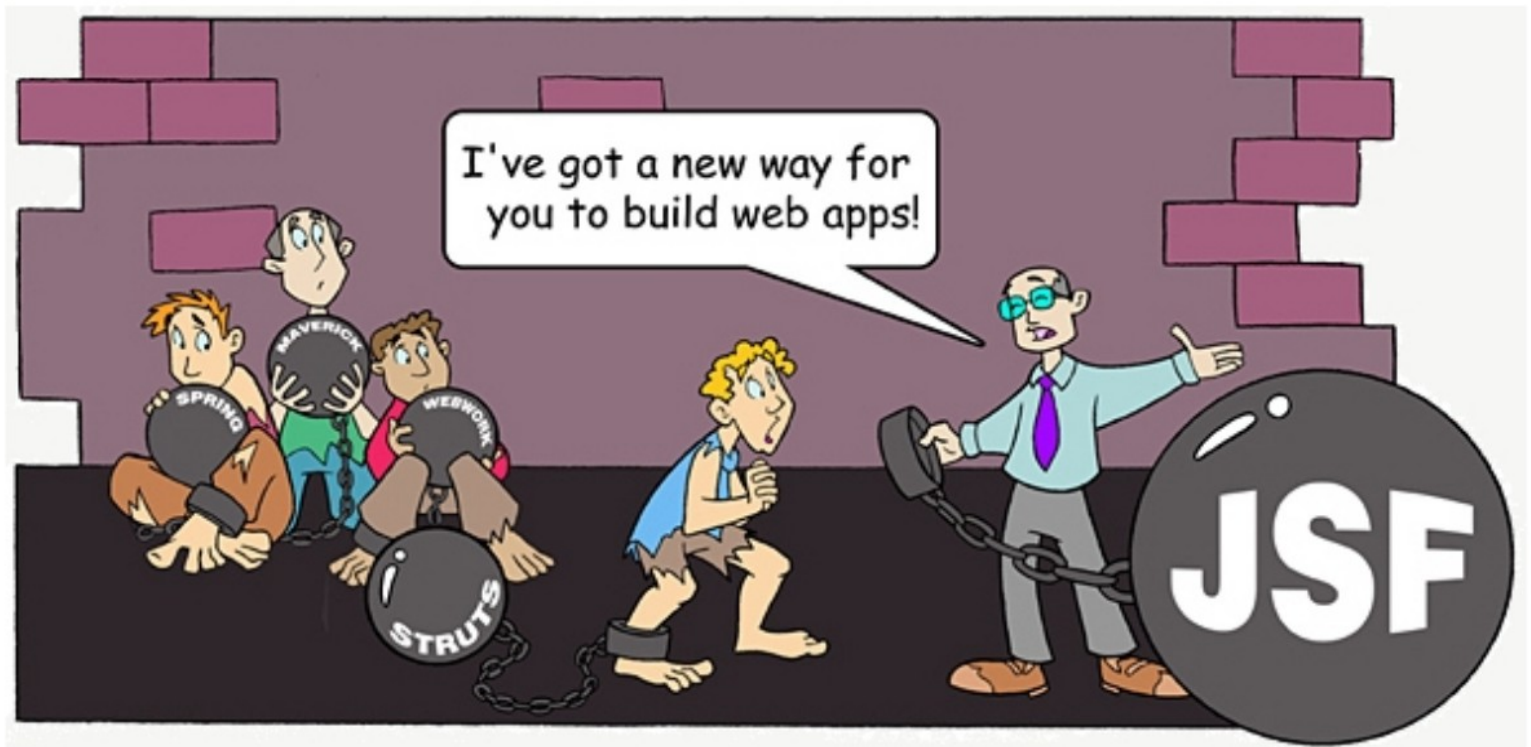
**So, what's this presentation about?**



# How Do You Choose?



# Pros and Cons



# Wicket



- Pros:
  - Great for Java developers
  - Tight binding between pages and views
  - Active community
- Cons:
  - HTML templates live next to Java code
  - OO is a requirement
  - Everything is done in Java only



# Flex

---

- Pros:
  - Produces Flash UI
  - Founded by Adobe
  - Great looks
- Cons:
  - Not search engine friendly
  - Doesn't render HTML content well
  - Printing issues



# JSF

- Pros:
  - Java EE Standard - lots of demand and job
  - Proven on production
  - Lots of component libraries
- Cons:
  - Tag soup for JSPs
  - Doesn't play well with REST/Security
  - No single source for implementation





# GWT

- Pros:
  - Write Java - Get JavaScript
  - Tools / Testing
  - Easy to learn and develop
- Cons:
  - Hard to attach on top of existing code
  - Not enough advanced widgets
  - No clear future
  - Performance Issues



# SpringMVC

- Pros:
  - Integrates with many view options seamlessly: JSP/JSTL, JSF, Tiles, Velocity, FreeMarker, Excel, PDF
  - Inversion of Control makes it easy to test
- Cons:
  - Almost too flexible - no common parent Controller
  - No built-in Ajax support



# Seam

- Pros:
  - A full stack framework
  - Conversations with jBPM
  - Heavily funded by JBoss/Red Hat
- Cons:
  - JSF
  - Works best on JBoss AS
  - Designed for EJB 3



# Struts2

Struts<sup>2</sup>

- Pros:
  - Simple architecture - easy to extend
  - Tag Library is easy to customize
  - Controller-based or page-based navigation
- Cons:
  - Missing documentation
  - Missing properties, invalid OGNL expressions

# Grails



- Pros:
  - Less LOC => awesome productivity
  - Easy to learn for Java Developers
  - Spring/Hibernate based
- Cons:
  - Not as performant as using the raw frameworks
  - Hard to sell to stakeholders
  - Virtually unknown

# Other Frameworks

---

- Tapestry
- Stripes
- WebWork
- Zk
- Dojo
- Extjs





# So, How to Choose?



# Eliminate, Don't Include

---

**It's not about including choices,  
it's about eliminating them**



# Don't believe the Hype

---

- Don't believe blogs and articles
- Try it yourself
- Believe ***developers***, not ***evangelists***
- Appreciate ***real*** developer experience
- Beware of corporate / marketing interests
- Books are a good sign



# Trial And Error

---

- Pick 2-3 frameworks for your type of application...
- ... and prototype!
- If prototyping is painful, switch
- Make sure you prototype more than one
- Do a presentation comparing the pros and cons of each
- Collect feedbacks
- 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



# 6 Important Factors

---

- What type of Application are you building?
- Ease of Development / Is full-stack an option?
- Project Community
- Project Future and Roadmap
- Testability
- Technical Features



# Types of Applications

---

## Request, Component or RIA?

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



# Types of Frameworks

---

- Request Based Frameworks
  - Struts 1/2, Spring MVC, Stripes
- Component Based Frameworks
  - JSF, Wicket, Tapestry, GWT
- Rich Internet Applications
  - Flex, OpenLaszlo
- One Size Fits All?
  - Seam, (G)Rails



# 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?



# Project Community

---

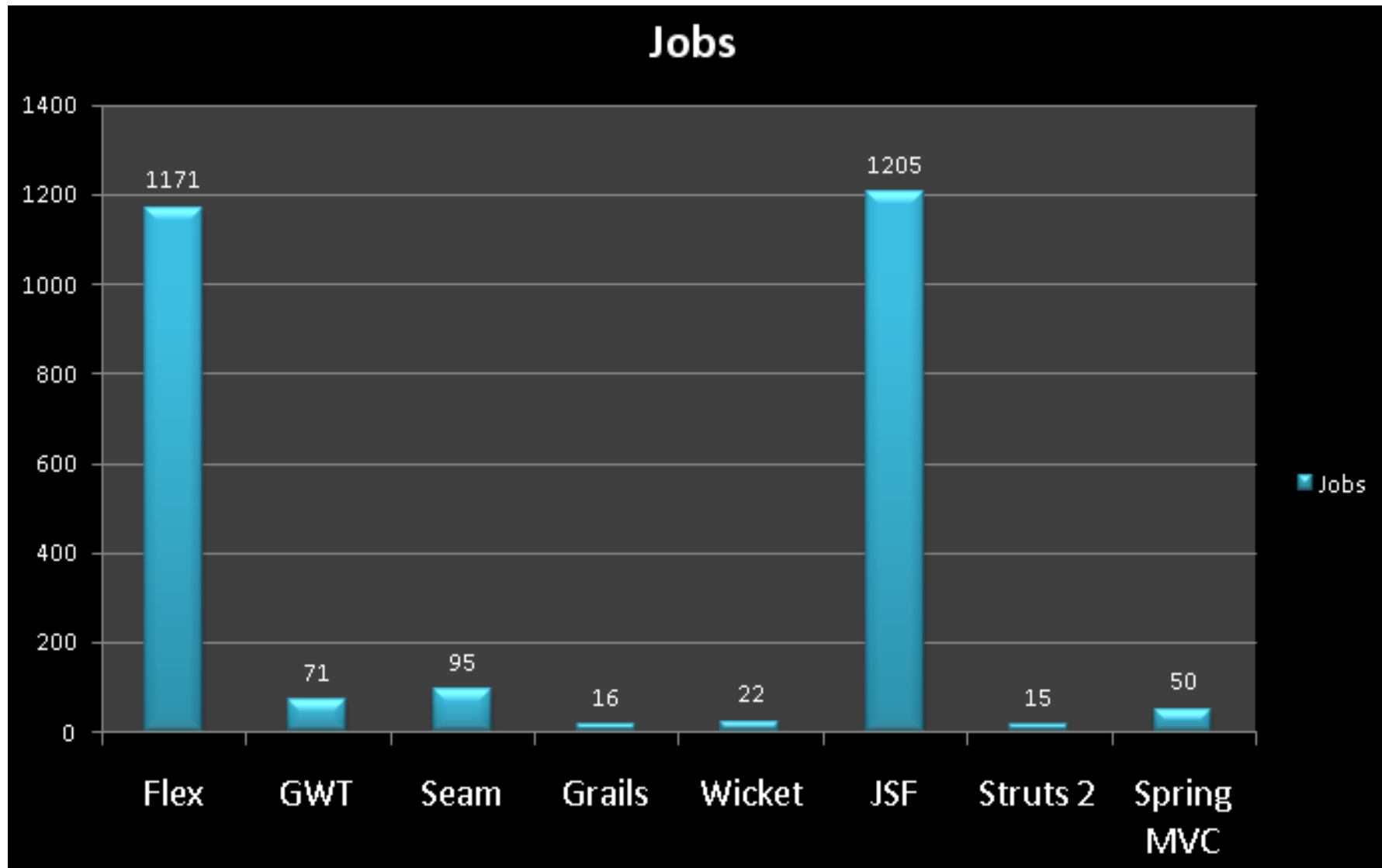
- Is there a company backing the project?
- Are vendors jumping on board? Consultants?
- Healthy mailing list traffic?
  - Users and developers are both important
- Are there frequent releases?
- What's the level of real-world usage?
- Are there books written about it?



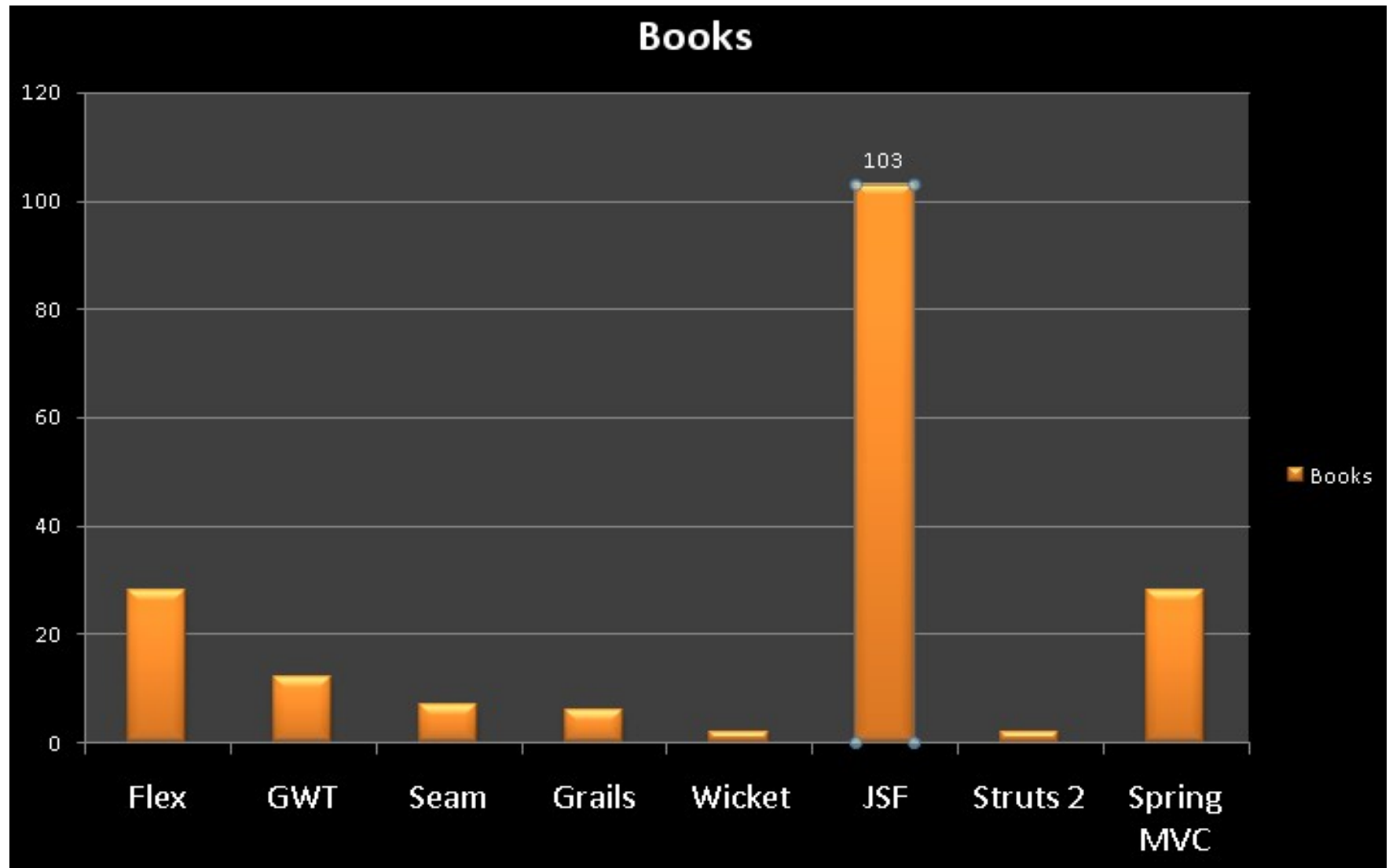
# Pretty Graphs



# Jobs on Dice.com

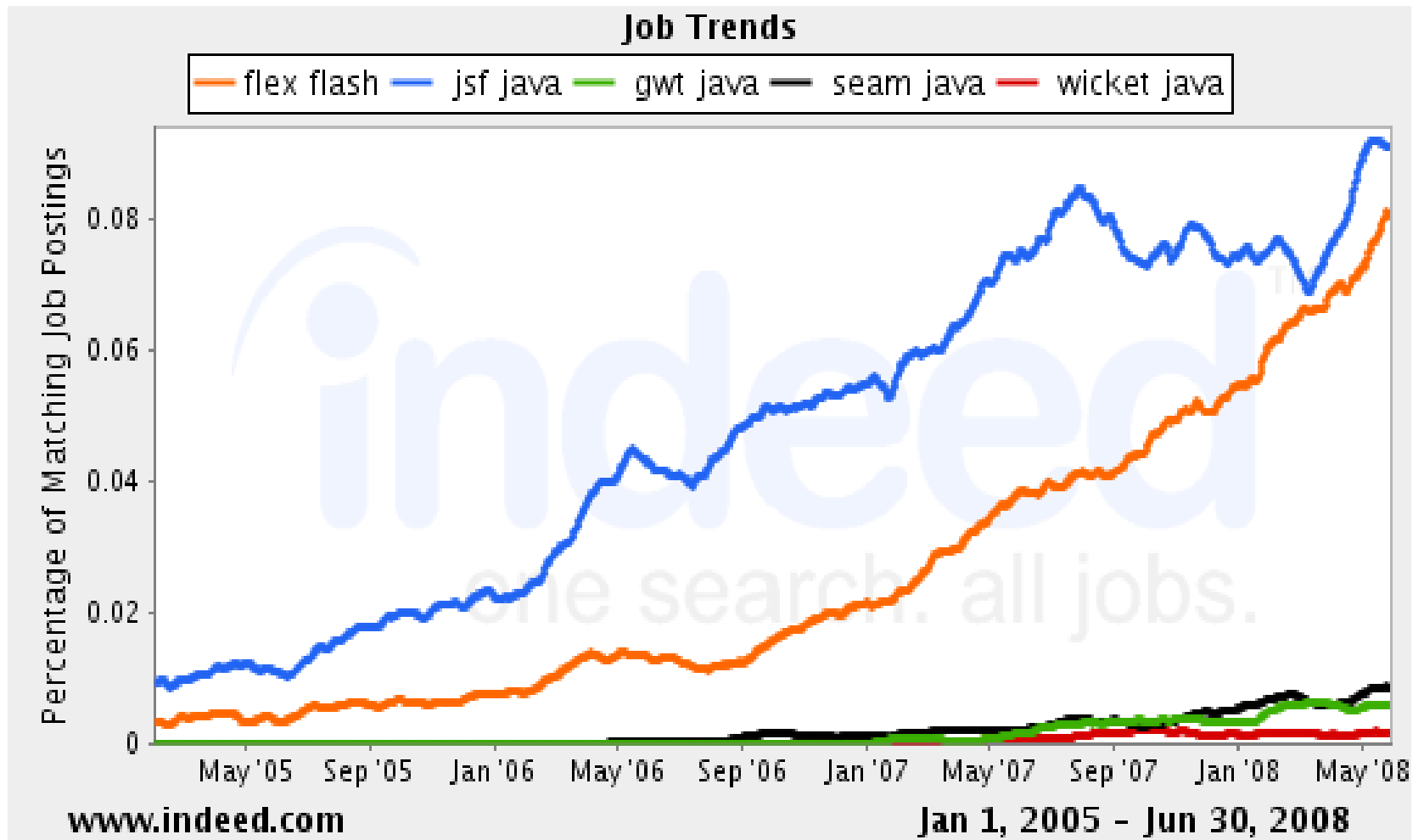


# Books on Amazon





# Job Trends





---

***Thank You!***