

Extracted from:

# My Job Went to India

## And All I Got Was This Lousy Book

This PDF file contains pages extracted from My Job Went to India, published by the Pragmatic Bookshelf. For more information or to purchase a paperback or PDF copy, please visit <http://www.pragmaticprogrammer.com>.

**Note:** This extract contains some colored text (particularly in code listing). This is available only in online versions of the books. The printed versions are black and white. Pagination might vary between the online and printer versions; the content is otherwise identical.

Copyright © 2005 The Pragmatic Programmers, LLC.

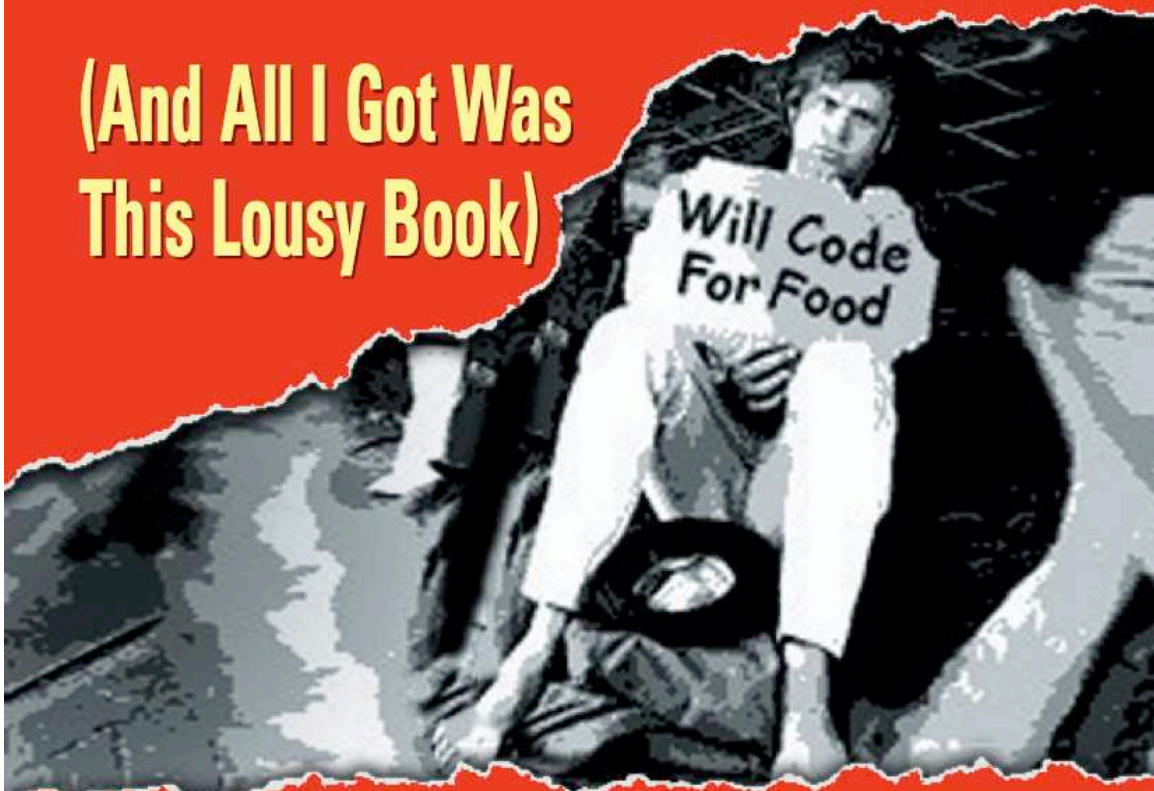
All rights reserved.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form, or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior consent of the publisher.

The  
Pragmatic  
Programmers

# MY JOB WENT TO INDIA

(And All I Got Was  
This Lousy Book)



## 52 Ways To Save Your Job

Chad Fowler

## 7 Don't Put All Your Eggs in Someone Else's Basket

While managing an application development group, I once asked one of my employees, "What do you want to do with your career? What do you want to be?" I was terribly disappointed by his answer: "I want to be a J2EE architect." I asked why not a "Microsoft Word designer" or a "RealPlayer installer?"

This guy wanted to build his *career* around a specific technology created by a specific company of which he *was not an employee*. What if the company goes out of business? What if it let its now-sexy technology become obsolete? Why would you want to trust a technology company with your career?

Somehow, as an industry, we fool ourselves into thinking *market leader* is the same thing as *standard*. So, to some people, it seems rational to make another company's product a part of their identities. Even worse, some base their careers around non-market-leading products—at least until their careers fail so miserably that they have no choice but to rethink this losing strategy.

Let's take a moment again to remember that we should think of our career as a business. Though it's possible to build a business that exists as a parasite of another (such as companies who build spyware removal products to make up for inadequacies in Microsoft's browser security model), as an individual, it's an incredibly risky thing to do. A company, such as the spyware example I just mentioned, can usually react to changing forces in the market such as an unexpected improvement in Microsoft's browser security (or Microsoft deciding to enter the spyware removal market), whereas an individual doesn't have the bandwidth or the surplus cash to suddenly change career direction or focus.

---

Vendor-centric views are typically myopic.

---

The sad thing about a vendor-centric view of the world is that, usually, the details of a vendor's software implementation are a secret.

You can really learn only so much about a piece of proprietary software until you reach the *professional services barrier*. The professional services barrier is the artificial barrier that a company erects between you and the solution to a problem you may have, so that it can

profit from selling you support services. Sometimes this barrier is intentionally erected, and sometimes it's erected as a side effect of the attempt the company makes to protect its intellectual property (by not sharing its source code).

So, while a single-minded investment in one particular technology is almost always a *bad idea*, if you *must* do so, consider focusing on an open-source option, as opposed to a commercial one. Even if you can't or don't want to make the case for using the open-source solution in your workplace, use the open-source option as the platform from which you can take a deep dive into a technology. For example, you may want to become an expert in how J2EE application servers work. Instead of focusing your efforts on the details of how to configure and deploy a commercial application server (after all, *anybody* can figure out how to tweak settings in a config file, right?), download the open-source JBoss or Geronimo servers and set aside time for yourself to not only learn how to operate the servers but to study their internals.

Before long, you'll realize you're naturally changing your view. This J2EE thing (or whatever you chose to get into) really isn't all that special. Now that you see the details of the implementation, you see that there are high-level conceptual patterns at work. And, you start to realize that, whether with Java or some other language or platform, distributed enterprise architecture is distributed enterprise architecture. Your view changes from narrow to wide, and your mind starts to open. You start to realize that these concepts and patterns that your brain is sorting through and making sense of are much more scalable and universal than any specific vendor's technology. "Let the vendors come and go—I know how to design a system!"

### Act on it!

1. Try a small project, twice. Try it once in your home base technology and once, as idiomatically as possible, in a competing technology.

# The Pragmatic Bookshelf

---

*The Pragmatic Starter Kit* series: Three great titles, one objective. To get you up to speed with the essentials for successful project development. Keep your source under control, your bugs in check, and your process repeatable with these three concise, readable books.

*Facets of Ruby* series: Learn all about developing applications using the Ruby programming language, from the famous Pickaxe book to the latest books featuring Ruby On Rails.

The Pragmatic Bookshelf features books written by developers for developers. The titles continue the well-known Pragmatic Programmer style, and continue to garner awards and rave reviews. As development gets more and more difficult, the Pragmatic Programmers will be there with more titles and products to help programmers stay on top of their game.

## Visit Us Online

---

### **My Job Went to India**

[pragmaticprogrammer.com/titles/mjwti](http://pragmaticprogrammer.com/titles/mjwti)

This book's home page, including errata and other resources.

### **Register for Updates**

[pragmaticprogrammer.com/updates](http://pragmaticprogrammer.com/updates)

Be notified when updates and new books become available.

### **Join the Community**

[pragmaticprogrammer.com/community](http://pragmaticprogrammer.com/community)

Read our weblogs, join our online discussions, participate in our mailing list, interact with our wiki, and benefit from the experience of other Pragmatic Programmers.

### **New and Noteworthy**

[pragmaticprogrammer.com/news](http://pragmaticprogrammer.com/news)

Check out the latest pragmatic developments in the news.

## Buy the Book

---

If you liked this PDF, perhaps you'd like to have a paper copy of the book. It's available for purchase at our store: [pragmaticprogrammer.com/titles/mjwti](http://pragmaticprogrammer.com/titles/mjwti).

## Contact Us

---

Phone Orders:	1-800-699-PROG (+1 919 847 3884)
Online Orders:	<a href="http://www.pragmaticprogrammer.com/catalog">www.pragmaticprogrammer.com/catalog</a>
Customer Service:	<a href="mailto:orders@pragmaticprogrammer.com">orders@pragmaticprogrammer.com</a>
Non-English Versions:	<a href="mailto:translations@pragmaticprogrammer.com">translations@pragmaticprogrammer.com</a>
Pragmatic Teaching:	<a href="mailto:academic@pragmaticprogrammer.com">academic@pragmaticprogrammer.com</a>
Author Proposals:	<a href="mailto:proposals@pragmaticprogrammer.com">proposals@pragmaticprogrammer.com</a>