Extracted from:

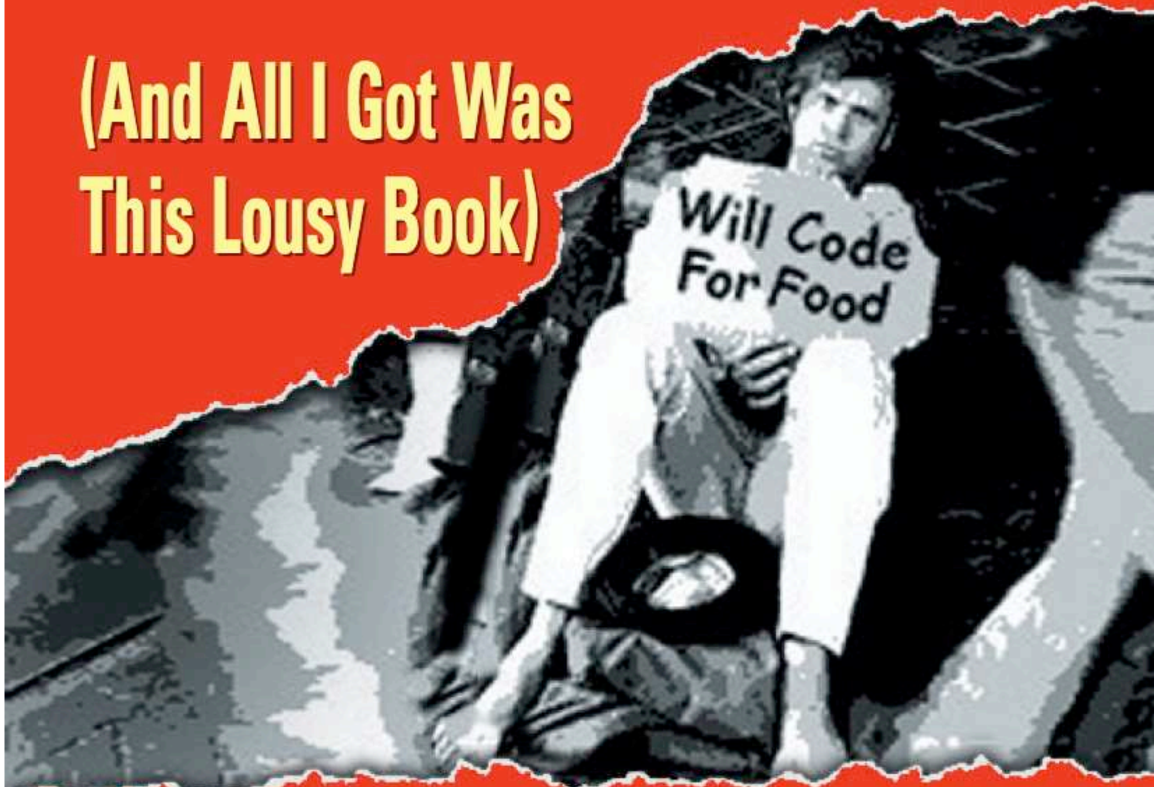
My Job Went to India

And All I Got Was This Lousy Book

The Pragmatic Programmers

MY JOB WENT TO INDIA

(And All I Got Was This Lousy Book)



52 Ways To Save Your Job

Chad Fowler

48 The South Indian Monkey Trap

In *Zen and the Art of Motorcycle Maintenance* [Piroo], Robert Pirsig tells an enlightening story about how people in South India used to catch monkeys. I don't know if it's true, but it teaches a useful lesson, so I'll paraphrase it.

The people of South India, having been pestered by monkeys over the years, developed an ingenious way of trapping them. They would dig a long, narrow hole in the ground and then use an equally long, slender object to widen the bottom of the hole. Then they would pour rice down into the wider portion at the bottom of the hole.

Monkeys like to eat. In fact, that's a large part of what makes them such pests. They'll jump onto cars or risk running through large groups of people to snatch food right out of your hand. People in South India are painfully aware of this. (Believe me, it's surprisingly unsettling to be standing serenely in a park and have a macaque come suddenly barreling through to snatch something from you.)

So, according to Pirsig, the monkeys would come along, discover the rice, and stretch their arms deep into the hole. Their hands would be at the bottom. They would greedily clutch as much of the rice as possible into their hands, making a fist in the process. Their fists would fit into the larger portion of the hole, but the rest of the narrow opening was too small for the monkeys to pull their fists through. They'd be stuck.

Of course, they could just let go of the food, and they'd be free.

But, monkeys place a high value on food. In fact, they place such a high value on food that they cannot force themselves to let go of it. They'll grip that rice until either it comes out of the ground or they die trying to pull it out. It was typically the latter that happened first.

Pirisig tells this story to illustrate a concept he calls *value rigidity*. Value rigidity is what happens when you believe in the value of something so strongly that you can no longer objectively question it. The monkeys valued the rice so highly that when forced to make the choice between the rice and captivity or death, they couldn't see that losing the rice was the right thing to do at the time. The story makes the monkeys seem really stupid, but most of us have our own equivalents to the rice.

If you were asked whether it was a good idea to help feed starving children in developing countries, you would probably say "yes" without even thinking about it. If someone tried to argue the point with you, you might think they were crazy. *This* is an example of value rigidity. You believe in this one thing so strongly that you can't imagine *not* believing it. Clearly, not all values that we hold rigidly are bad. For most people, religion (or lack thereof) is also a set of personal beliefs and values that are unfaltering.

But not all rigidly held values are good ones. Also, many times something that is good in one set of circumstances is not good in another.

Rigid values make you fragile.

For example, it's easy to get hung up on technology choices. This is especially true when our technology of choice is the underdog. We love the technology so much and place such a high value on defending it as a choice for adoption that we see every opportunity as a battle worth fighting—even when we're advocating what is clearly the wrong choice. An example I encounter (and have probably been guilty of myself) is the overzealous Linux fan base. Many Linux users would put Linux on the desktop of every receptionist, office assistant, and corporate vice president with no regard for the fact that, in terms of usability, the toolset just doesn't compare to much of the commercial software that's available for a commercial operating system. You look foolish and make your customers unhappy when you give the right software to the wrong people.

It's hard to tell you're losing weight because you see yourself every day. Value rigidity works the same way. Since we live every day in our careers, it's easy to develop value rigidity in our career choices. We know what has worked, and we keep doing it. Or, maybe you've always wanted to be promoted into management, so you keep striving toward that goal, regardless of how much you like *just programming*.

It's also possible for your technology of choice to become obsolete, leaving you suddenly without a foundation to stand on. Like a frog in a slowly heating pot of water, you can suddenly find yourself in a bad situation. Many of us in the mid-1990s swore by Novell's NetWare platform when it came to providing file and print services in the enterprise. Novell was way ahead of its time with its directory services product, and those of us "in the know" were almost cocky in our criticism of competing technologies. Novell's product was enjoying a healthy majority in market share, and it was hard to imagine the tide turning.

No single event made it obvious that Novell was losing to Microsoft. Microsoft never made that magic Active Directory release that made us all say, "Wow! Drop NetWare!" But, Netware has slowly gone from bleeding-edge innovator to legacy technology. For many NetWare administrators, the water was boiling before they ever even realized the pot was warm.

Whether it is the direction your career is taking or the technologies you advocate and invest in, beware of monkey traps. Those originally intententional choices may become the last handful of rice you find yourself gripping prior to your career being clubbed to death.

Act on it!

1. *Find your monkey traps*—What are *your* rigid assumptions? What are those values that guide your daily actions without you even conciously knowing it?

   Make a table with two columns, *Career* and *Technology*. Under each heading list the values that you hold unfalteringly true. For example, under Career, what have you *always* known to be one of your strengths? Or your weaknesses? What is your career *goal* ("I want to be a CEO!,")? What are the right ways to achieve your goal?

   In the Technology column, list what you most value about the technologies you choose to invest in. What are the most important attributes of a technology that should be considered when making a choice? How do you make a scalable system? What's the most productive environment in which to develop software? What are the best and worst platforms in general?

   When you've got your list down and you feel like it's fairly complete, go one at a time through the list and mentally reverse each statement. What if the opposite of each assertion were true? Allow yourself to honestly challenge each assertion.

   This is a list of your vulnerabilities.

2. *Know your enemy*—Pick the technology you hate most, and do a project in it. Developers tend to stratify themselves into competing camps. The .NET people hate J2EE, and the J2EE people hate .NET. The UNIX people hate Windows, and the Windows people hate UNIX.

   Pick an easy project, and try to do a *great* application in the technology you hate. If you're a Java person, show those .NET folks how a *real* developer uses their platform! Best case, you'll learn that the technology you hate isn't all that bad and that it is in fact possible to develop good code with it. You'll also have a (granted, undeveloped) new skill that you might need to take advantage of in the future. Worst case, the exercise will be a practice session for you, and you'll have better fodder for your arguments.