Extracted from:

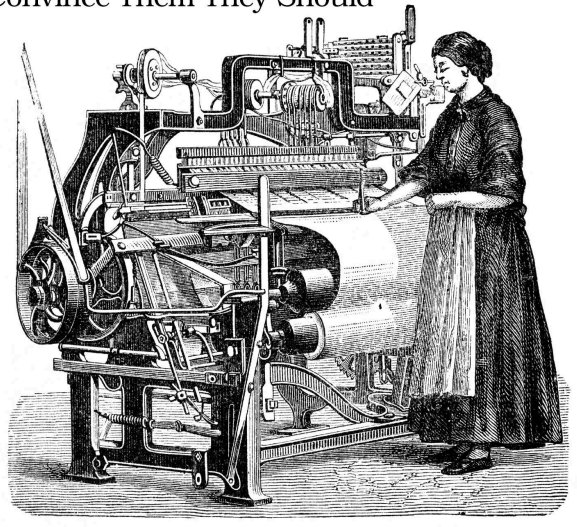
Driving Technical Change

Why People on Your Team Don't Act on Good Ideas, and How to Convince Them They Should

The Pragmatic Programmers

Driving Technical Change

Why People On Your Team Don't Act on Good Ideas, and How To Convince Them They Should



Terrence Ryan

Edited by Jacquelyn Carter

Chapter 1

Why This Book?

I was in the middle of a very typical meeting, with a very typical group, in my very typical company. I was in charge of our web application servers. My responsibilities included maintaining software, maintaining hardware, enforcing best practices, and getting people to upgrade. I was always trying to get people to upgrade.

In fact, we were talking about upgrades.

"I need you guys to set up some sort of schedule for moving your applications from ColdFusion 6 to ColdFusion 8," I said, for the fifth time in as many meetings.

The expected response was delivered with a sigh, "We can't move to the new servers. Every time we move our applications to a new server we have problems and incompatibilities. We just can't have that with our users."

I fired back, "That's a common problem in general. Have you considered using unit tests to be able to have more confidence when you move from version to version? That's not just a problem with an application server but also web servers, database servers, and your code base. You have to move from version to version. Unit tests help with this."

The excuses then flew, "It would take too much time. Why should we have to do this? We don't know how to do unit tests."

I could tell you that I argued with them. I could detail the rest of the argument. I don't really have to do that. You know how it went. I didn't get them on board. I couldn't convince them.

Over my time in that position, I had to try to sell several different advancements and techniques. I had the argument I just described and others like it many, many times. I lost a lot, I won a few, and some just ended up being wars of attrition. I did start to notice certain patterns:

- The same people tend to make the same arguments.
- Some people always went along with new things.
- Other people jumped on to an advancement once others had already converted.
- Some people can never be convinced.
- Certain arguments I made worked on some people but not on other people.
- Sometimes getting management involved was the only way of getting people on board.

I took those patterns, wrote down what I observed about them, and figured out that certain tactics worked better on some than others. I started reusing the same tactics on the same skeptics for different issues. My batting average went up. It became easier to sell advancements.

That's what this book is about—those advancements, those patterns, those arguments. My hope is that what I have to say can allow you to skip all of the go-nowhere arguments, avoid the frustration, and actually drive your organization forward technologically.

1.1 How Is This Book Organized

This book is a patterns book. That means the subject matter is based on a set of repeating forms, or *patterns*. Two main parts of the book are broken into collections of patterns. One part is about skeptic patterns focused on the reason some people resist your efforts. The other is about techniques that can be used to counter skeptics. Ultimately, this means the chapters in these sections are going to be highly structured.

While the two patterns parts are about who your co-workers are and how you should approach them, the final part of this book breaks away from the patterns and talks about strategy. It will help you sort out who to approach first, who to avoid, and how to turn your efforts into real change.

1.2 Why You Should Read This Book

The goal of this book is to enable you to convince co-workers to adopt new tools and techniques. You should be able to do this without having to become some sort of cutthroat office politician. This doesn't mean that you won't need to use politics, just that you don't have to use them evilly.

I will outline a cast of characters; some of them will remind you of people you work with. Once you identify those people, you should be able to match them up with countering techniques. You apply those techniques on your skeptics in a strategy I will lay out. Then change magically happens.

Well, not magically. It does take some work and effort. But it is that straightforward. You should be able to reap some benefits immediately after reading this book. The rest will come as you gain experience doing what this book outlines.

1.3 Who I Think You Are

I think you are a technical person. Perhaps you're a developer or programmer. You could be a server administrator, network engineer, or hardware engineer. Maybe you're a database administrator or even a designer who works with technical people.

It doesn't really matter what type of technical person you are, as long as you do some sort of technical work with other people.

My anecdotes and scenarios are going to be about developer topics. Sorry, that's who I am. However, it doesn't matter what language or tool set you are using. This is going to apply evenly, whether you are a .NET or a Java programmer, an open source fan, or someone in love with some company. This is for anyone who has tried to get co-workers to change the way they work, regardless of how you wanted them to work.

The Pragmatic Bookshelf

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 you stay on top of your game.

Visit Us Online

Home page for Driving Technical Change
http://pragprog.com//titles/trevan
Source code from this book, errata, and other resources. Come give us feedback, too!

Register for Updates
http://pragprog.com/updates
Be notified when updates and new books become available.

Join the Community
http://pragprog.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
http://pragprog.com/news
Check out the latest pragmatic developments, new titles and other offerings.

Buy the Book

If you liked this eBook, perhaps you'd like to have a paper copy of the book. It's available for purchase at our store: pragprog.com//titles/trevan.

Contact Us

Online Orders:	www.pragprog.com/catalog
Customer Service:	support@pragprog.com
Non-English Versions:	translations@pragprog.com
Pragmatic Teaching:	academic@pragprog.com
Author Proposals:	proposals@pragprog.com
Contact us:	1-800-699-PROG (+1 919 847 3884)