

Extracted from:

# Driving Technical Change

---

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

This PDF file contains pages extracted from Driving Technical Change, published by the Pragmatic Bookshelf. For more information or to purchase a paperback or PDF copy, please visit <http://www.pragprog.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 © 2010 The Pragmatic Programmers, LLC.

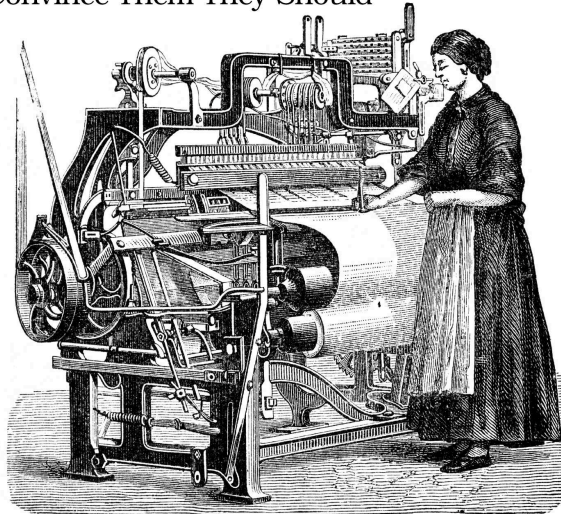
All rights reserved.

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

# 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 7

# The Cynic

---

You've just spent twenty minutes doing a presentation for your teammates on adopting source control. Yeah, they don't do source control at all. Yep, not at all—it's as if the last twenty years of computer science never happened. But better late than never, and frankly any source control is better than none, because disaster is one errant delete away. You did your homework, and you've tested for yourself. You tried and retried a number of solutions, but the combination of this particular group, solution maturity, availability of tools, and industry acceptance have led you to choose Subversion. Anyway, you've rocked the presentation, and now you're taking questions.

"I've heard that Subversion is yesterday's news. With Git on the rise as the next big thing in source control, do we want to do Subversion just to change in two years?" asks Cindy.

You were ready for that, "Great question, Cindy. I reviewed a bunch of solutions including Git. The long and short of it is that in our group we need tools that integrate into Eclipse. It's my opinion that the Git tools for Eclipse are just not there yet. They might be there in a year or two, but we need a solution yesterday. So, Subversion is more correct for us today. If we do need a feature set that Git provides, there are tools for migrating from Subversion to Git, so I'm not worried."

You assume that such a tome of an answer will be enough, but still Cindy fires the questions at you.

"I've heard that Subversion adds all of this extra metadata to a project, and the bigger the project, the more extra metadata gets created. I hear the metadata is easily corruptible."

“It’s an issue, but I think most of our projects are small enough that it shouldn’t be a huge problem. There are best practices for dealing with our larger projects,” you fire back.

It goes back and forth like this for awhile. Sometimes you nail the answer, sometimes you don’t. Sometimes you wonder if Cindy didn’t just call up the Wikipedia article for Subversion and just scrolled down to the “Current limitations and problems” section. How else does someone who has never touched Subversion know about the downsides of it?

At the end Cindy announces with assumed authority, “With all of these issues and considering we’ve been doing well without it, I don’t see why we need to add extra complexity to our environment.”

Other people in the meeting may not completely agree, but they’ve been sitting watching your verbal tennis match for a while. Much coffee was consumed. So, now you’re in a fight with Cindy, with people’s attention spans, and with their bladders. What should have been a slam dunk is now a political battle.

It’s a common story when you’re dealing with a Cynic.

## 7.1 Underlying Causes

There are a lot of causes to this behavior. Some people simply like to argue. Others like to prove that they are smarter than someone else. Others have worked in industry for a while and have been repeatedly disappointed and therefore never see the upside of anything.

But there is one really important reason that you will encounter this: *in our industry, this behavior is rewarded.*

Most of our currency in this industry is based on what we can produce with our minds. Smarts are important, but more important than *being smart* is *looking smart*.

Being smart can happen anywhere, doesn’t require an audience, and therefore often goes unnoticed. In fact, the distractions of being in front of people often sabotage smarts. That’s why you usually prepare well before you talk to a group about an issue. You can make reasonably sure you’ll deliver your full brain power to an issue in the privacy of your own cubicle, but in front of a crowd, you’ll have unknown variables, nervousness, and often bad luck keeping you from your full potential.

### **Criticism Is Good. Cynicism Is Bad**

Semantics are a tough thing. One person's cynicism is another's due diligence. I don't want you to think you need to handle any criticism as an attack of a Cynic. You should justify your tool or technique. People shouldn't blindly accept that you're steering them the right way.

However, the criticism I describe here as cynicism isn't trying to defend people from poor choices. It's designed to block progress for blocking progress's sake or to score cheap points. Some of what I say here should help you differentiate. But at the end of the day, it's like Potter Stewart said about obscenity: "I know it when I see it."\* It's hard to define but easy to feel; sometimes people are criticizing to achieve the best outcome for all, and sometimes people are being cynical to keep themselves from having to grow or to make themselves look good at someone else's expense.

\*. Potter Stewart was a Supreme Court Justice who when deciding a case about obscenity famously said, "I shall not today attempt further to define the kinds of material I understand to be embraced within that shorthand description ("hard-core pornography"); and perhaps I could never succeed in intelligibly doing so. But I know it when I see it."

Looking smart, on the other hand, requires an audience but delivers the impression that you can be smart without necessarily having to have the mental horsepower to actually pull it off.

What does all of this have to do with our healthy cynic? There are two ways to look smart:

- Be very smart in front of an audience (which we've established is tough).
- Be smarter than someone else in front of an audience.

The second one is pretty easy to do, and this is where our Cynic comes in. In challenging you every step of the way, they are keeping up with you, which makes them look smart. All it will take is one weak spot in your prep for them to look smarter than you.

## 7.2 Effective Countering Techniques

Countering the Cynic is about two things:

- Refusing them entry points to arguments
- Preparing enough so that you cannot be refuted

Refusing them entry points is about not allowing them to ask the questions in the first place. You can deliver your message smoothly. You can anticipate likely questions and answer them as part of your pitch. If you've done this correctly and they do ask a question, you can put them off until you answer it as part of your prepared material. This allows you to control the conversation, not them.

On the other hand, you can't prepare for everything. You can't anticipate every question. First answer their questions authoritatively; even if you don't know the answer, say "I don't know" with confidence. Going further, gain and use knowledge of the subject to prevent them from having opportunities to look smart at your expense.

To those ends, the following methods are especially helpful with these skeptics:

- Chapter 13, *Gain Expertise*, on page 55
- Chapter 14, *Deliver Your Message*, on page 64
- Chapter 3, *Solve the Right Problem*, on page 23
- Chapter 15, *Demonstrate Your Technique*, on page 70
- Chapter 21, *Create Something Compelling*, on page 103
- Chapter 18, *Get Publicity*, on page 87
- Chapter 20, *Build a Bridge*, on page 97

## 7.3 Prognosis

Assuming that you can prepare and refute as outlined here, you can usually do well by this group. With your delivery and expertise, you can make your solution *the smart solution*. By positioning your solution as the smart one, it follows that questioning it makes them less than smart. Their self-imposed pressure to look smart will keep them from sniping at you. Who knows, if you make your solution look smart enough, they might even become your biggest boosters.

# 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/](http://pragprog.com/titles/trevan/).

## Contact Us

---

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