Extracted from:

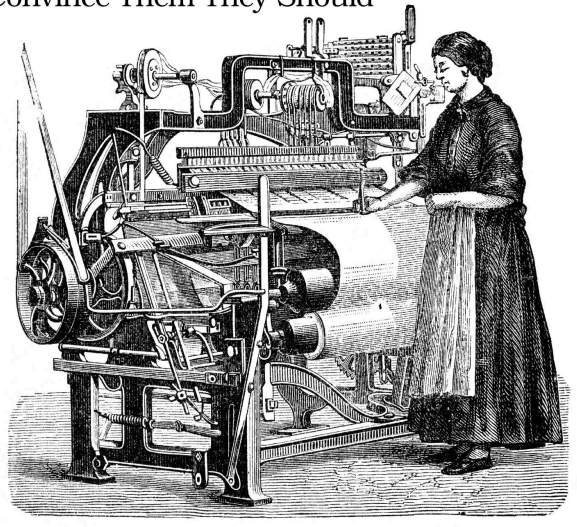
Driving Technical Change

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 17

Create Trust

Having the correct facts is a huge part of convincing people to come over to your way of thinking. However, a large, often overlooked part of the story is trustworthiness. If people don't trust you, you have an uphill battle justifying even the most obvious of tools and techniques.

Trust is a little harder than the other techniques in that there is no formula or checklist for it. You can take a class to Gain Expertise, but no class will get you to be trusted. In fact, gaining trust is often more about what negative things you don't do than what you do. But have no fear, although there are no shortcuts, that doesn't mean there aren't ways to get people to trust you.

FUD Factor

Shailaja's company was in the market to invest in a new database installation. Because of the high cost, she was looking at alternatives to their current technology choices. After careful consideration, she decided to change the entire company system over to MySQL. In her case, MySQL was cheaper, was more supportable, and had more public information available.

Shailaja came up against a lot of opposition. Some of it was irrational, people clinging on to old technology, but some of it was reasonable: change has cost; in this case, they may have been too high. Also, MySQL had just had just passed to a new owner, and their future wasn't written in stone, even if all signs were good at the moment.

The discussion got heated, and at one point someone blurted out, "I've heard that the company that owns our old system is going to discontinue it next year."

FUD

FUD stands for Fear, Uncertainty, and Doubt. Though the phrase was coined in the mid-1970s, the concept has been around since the first caveman traded a rock to another one "in case the mastodons come back." More recently, it's been marketers, public relations flacks, and sales guys who use this on you. Basically, the idea is to tell you something that will make you afraid of a rival's tool, enough so that you invest with the FUDer.

At a smaller level, this happens in the workplace a lot. Developers with experience with proprietary tools spread rumors about crazy license implications of open source tools. Open source adherents spread horror stories of hidden code in proprietary toolkits.

It's ultimately self-defeating. At best it can win people some sort of short-term gains, but in the long term, it is a road to nowhere. Eventually people wise up to be bullied repeatedly, and some people speak out. This spread of information inoculates the rest, and the technique becomes ineffective.

People were really swayed by this. There was one problem with it. Shailaja knew it wasn't true. Flamebaiters had made that up as FUD in a forum somewhere, and the company had been fighting it ever since.

She was in a quandary. She liked the effect that the FUD had on her co-workers, but it was factually wrong. She knew that if people found out the real story behind it, they would be pissed at trying to be bullied. But the odds of them finding out were slim—her co-workers weren't forum people.

In the end, Shailaja corrected the FUD. MySQL was adopted anyway. Her co-workers went with her recommendation partially because she had always been aboveboard with them.

17.1 Why Does It Work?

Create Trust works because people do not like to be manipulated. Tougher than that is that people don't want to even feel like they're being manipulated. This is why the used-car salesman has such a bad common stereotype in our culture. They're seen as deceptive and manipulative (even when they often aren't).

Oftentimes when changing tools and technologies, you reach a point in the decision process where all of the empirical stuff has been revealed, choices have been weighed, and you still don't have a complete victory for one tool or another. When an organization reaches that point, often they need a tiebreaker to make that decision. That tiebreaker is going to come from something not quite so rational, not quite so empirical. It is going to come from something personal like trust. Making sure that you are trusted clearly has a lot of value when you get to that stage of the decision-making process.

17.2 Developing Trust

As I said earlier, there are no magic bullets for creating trust. There are things you can do to develop trust over the long term. I'd make the argument that you should be doing these things anyway, but to each their own.

Don't Lie by Commission

Lies by commission are pretty obvious because they're what we think of normally when we think about lies. Saying something like "MYSQL isn't a relational database because it doesn't handle foreign keys" is a lie of commission. This is pretty straightforward. It's possible that if caught, you can claim it was a mistake. But go to that play too often, and people will catch on to you.

I know our kindergarten teachers told us not to lie when we were young, but I'm telling you not to do this now because we can still find excuses to do it, especially when there is a kernel of truth to it like in the MYSQL statement I just made. By default MySQL tables don't do foreign keys, but if you set the table type to InnoDB, it can. That's where people get into trouble. Subtle issues like this make it easy to justify lying. Just don't do it.

Don't Lie by Omission

Lies of omission are a little bit trickier. These are cases where it's not necessarily that you didn't say something that was untrue. It's where you failed to say something that was true and necessary to the conversation. So, saying something like "MYSQL table names are case sensitive" is a lie of omission because it's leaving out the rejoinder "on certain operating systems." If you were talking about issues such as

converting old databases to MYSQL, this becomes very significant to the discussion.

This isn't as destructive as a lie of commission because people give the benefit of the doubt and assume it is possible that this was a mistake or oversight. This is what makes it so attractive. There's high benefit and little risk. But the more you do it, the more likely people will come to one of two conclusions about you:

- You're leaving out things intentionally and therefore untrustworthy.
- You're mistaken a lot and therefore untrustworthy.

Neither outcome is good for your credibility.

Never, Ever, Ever Resort to FUD

FUD is by its very nature destructive in that you're motivating people through fear. Sooner or later people will get this. When they do, there are personal repercussions. Getting caught spreading FUD, especially if it is untrue, will ruin your reputation for months or years. Some people will simply never trust you again.

Even if they don't necessarily get that you are manipulating them, they might think that you are "the boy who cries wolf." If you predict doom and gloom and massive catastrophe enough and it doesn't happen, people start to realize that they shouldn't listen to things you say. That is the opposite of being trustworthy.

This does beg a question, though. "How can I speak about competing technologies if I can't spread FUD? At some point, don't I have to say that the other tool or technique is worse?"

I would say, always speak from your own tool first. The other tool isn't worse; yours is better. When asked to address a competing tool's shortcomings, you do have to say something, though. So, make sure that you speak plainly, unemotionally, and unexaggerated. Also, if you can, have industry sources for that criticism at the ready—that would help.

Admit Mistakes

Unlikely as it may seem, you will make mistakes from time to time. You may even be completely wrong. You may be tempted to ignore it and take attention away from it. This temptation will be even greater when you are in the middle of trying to get something adopted.

Believe it or not, being wrong might be the best thing to happen to you. It gives you the opportunity to inform people that you are wrong. Yep, you read that right. See, most people aren't used to people owning up to their errors. Even if they are, most people have to be confronted to own their mistakes. But the fact that you willingly admit mistakes allows people to trust you. Because you tell it like it is, even when it is bad for you.

All that being said, if your mistake is about the tool or technique, you need to reevaluate. Admitting you're wrong about something but continuing to insist that people do it is pure foolishness.

17.3 Skeptics That It Counters

This technique tends to be very effective against these skeptics.

The Burned

The Burned (Chapter 8, *The Burned*, on page 40) by nature have both experience with what you are talking about and trouble believing what you are saying by virtue of the fact that it disagrees with their experience. Any perception of dishonestly will immediately cause them to clam up. Don't give them any excuses.

The Cynic

The Cynic (Chapter 7, *The Cynic*, on page 36) doubts everything people say anyway. Giving them a reason to doubt you will only redouble their efforts to expose you for the fraud or fool they already think you to be. Don't compound this problem by actually being a fraud.

The Irrational

The Irrational (Chapter 11, *The Irrational*, on page 49) are looking for any excuse to shut down the change you are selling. You being a liar is a pretty good one. They will ride any mistakes in honesty you make for the rest of your working relationship. That's not grief you want.

17.4 Pitfalls

Telling the truth has few pitfalls. But there is one big one: in your effort to be seen as trustworthy, you point out and emphasize the flaws in your solution. You want to acknowledge that your tools have flaws,

downsides, or compromises, but advertising them is another matter altogether.

Sadly, there is little to be done about this. Because nothing is perfect, you will occasionally be put into the position of saying something bad about the tool or technique you are advocating. Own up to it, acknowledge the weakness, and spell out why you don't think that the weakness outweighs all your solution's strengths.

17.5 Wrapping Up

It seems pretty silly to have to encourage honesty. However, the temptations to lie are pretty big, especially in the short term. In the long term, though, the costs are too high. Tell the truth, ride out short-term issues, and play the long game.

Putting It Into Practice

Here are a few suggestions to try to improve your trustworthiness:

- Create in your mind a scenario where a competing technology makes more sense than yours.
- Investigate the weaknesses of your own solution. Make sure you know where it falls down.

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)