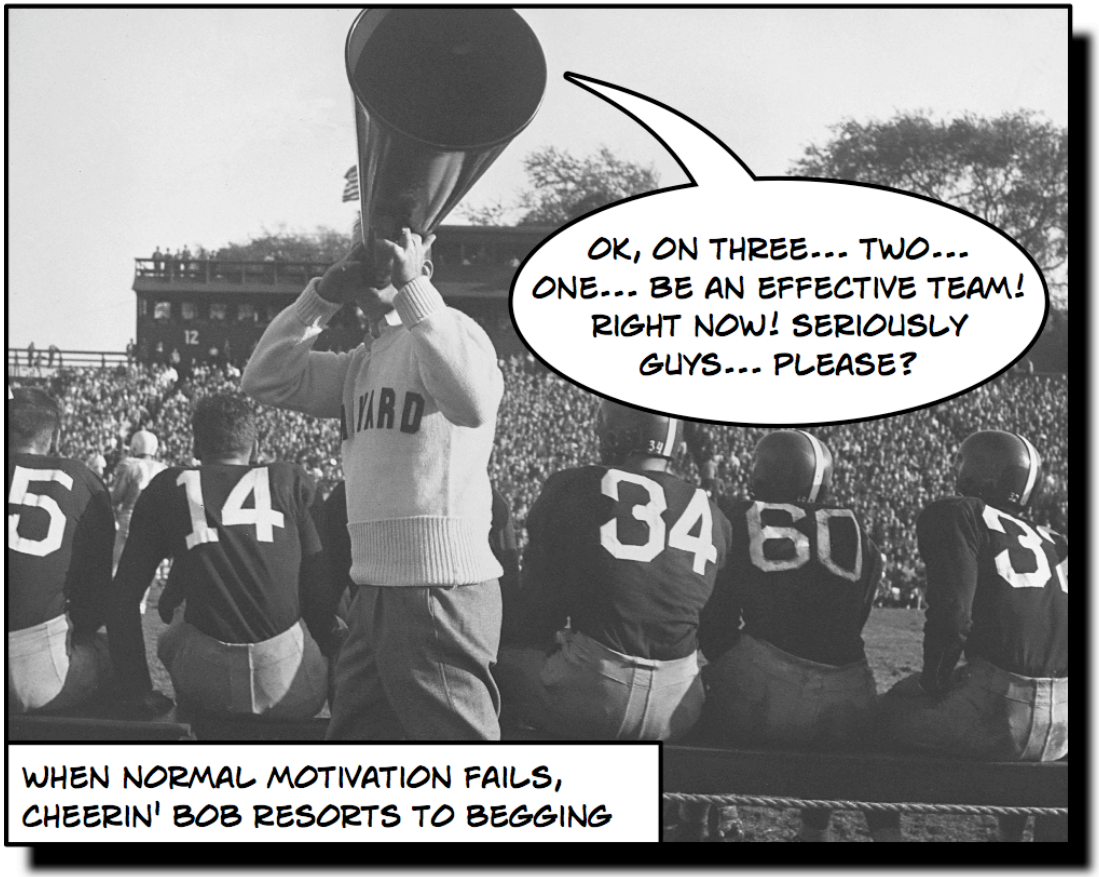
Beautiful Teams



Lessons we learned from veteran team leaders

A presentation by Jennifer Greene and Andrew Stellman

for Boston SPIN

April 21, 2009

Who we are...

Jenny's spent the last 15 years or so managing development and test teams

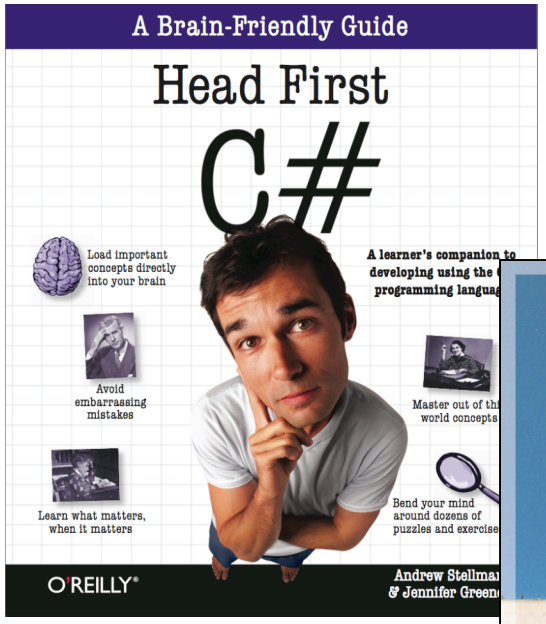
Andrew started programming in the 80s, and lost count of how many languages he's worked with.

She's currently doing consulting and training for a group with a large (500 person) IT team

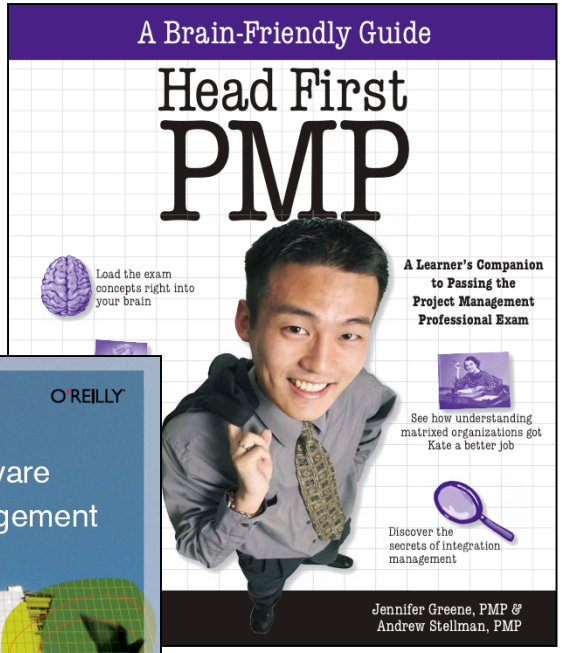
He's led teams of programmers, requirements analysts and process engineers.

Jenny and Andrew truly believe that with better development practices and good programming habits, we can all build better software.

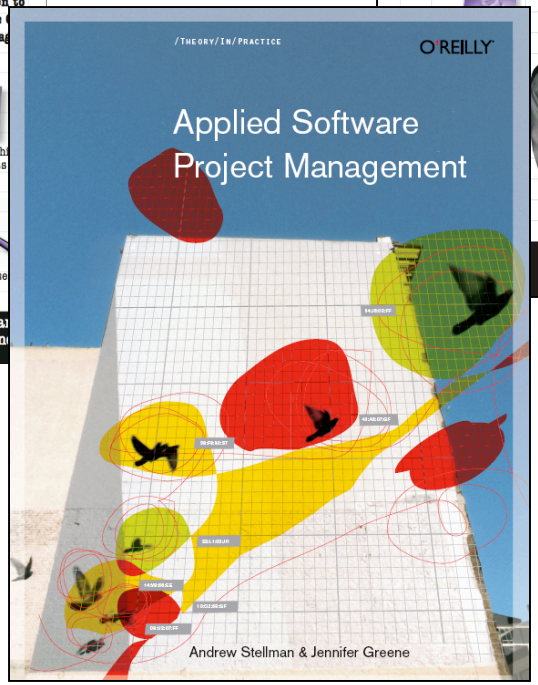
Our first three books...



2008



2007



2005

Our latest book: Beautiful Teams

We put this talk together to bring you the lessons we learned from putting together *Beautiful Teams*. It's a collection of stories from and interviews with some of the most innovative thinkers in the software industry. This talk is our way of sharing some of the ideas that helped us learn more about what makes teams work.

WARNING: This is NOT an academic presentation

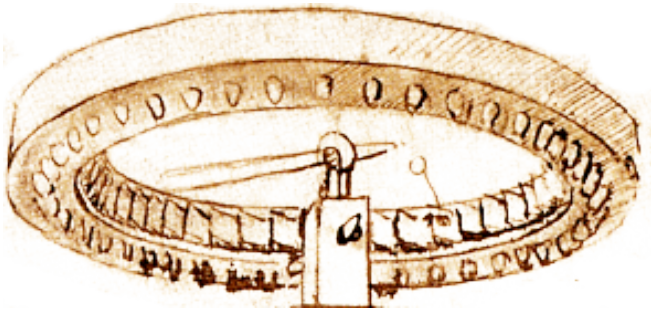
The topics we are about to cover may be deadly serious, but we won't be

If you want academic slides, we've got 'em:
http://www.stellman-greene.com/slides

Practices aren't everything...

When we started working on *Beautiful Teams*, we thought we'd be talking to people about the practices they used in different kinds of companies across many industries. We were naïvely looking for a recipe for a great team. But what we discovered is that there is no one "best" practice.

Just to be 100% clear, we don't think this is true!



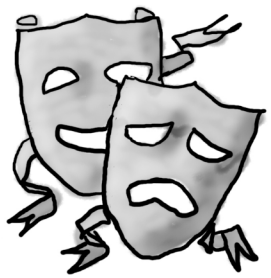
"If you've got the right practices, the people are interchangeable cogs."

Different practices work for different people at different times. A practice that works great for one team can cause problems for another.

...but you need more than just smart people!

In fact, some of the smartest developers seem to be the most susceptible to serious and enormously costly problems:

- Diving right into a project without planning
- Ignoring users when they try to explain their needs
- Antagonizing team members by acting like prima donnas
- And gold plating the software they do build



It's not just about training people

People are as important as the process… and even the best practices won't resolve the drama of a bad team situation.

You can't just tell a team to work together

Software people are great BS detectors. They want to feel like they're being treated respectfully and intelligently. That's why they respect people who are genuine and smart.

It takes more than good coffee in the pantry to make a good team... but bad coffee can turn the team against you!

This may SEEM like common sense...

When you're thinking about building a team in the abstract, it seems so simple. Just go on a retreat, or do a team-building exercise, or maybe get a foosball table for the break room. But it's rarely that easy in real life.

So how do you make a team work?

That was the question we set out to answer when we started *Beautiful Teams*. We talked to some of the brightest people in the software industry, and their advice came to us in four categories:

- **People**
  Who's on the team has a huge impact on how they work
- **Goals**
  You need a common goal to bring the team together
- **Practices**
  How you do the work makes all the difference
- **Obstacles**
  When they get in your way, you need to be smart

People

Teams are messy. They're full of emotional connections, often between people who are at their wits' end trying to solve problems that may not necessarily be solvable.

When you have people who, in the midst of a situation like that, are willing to be themselves, put themselves on the line, listen to the people around them, and help everyone get through the late nights and the frustration, that's what makes a team great.

Up Close: What makes developers tick

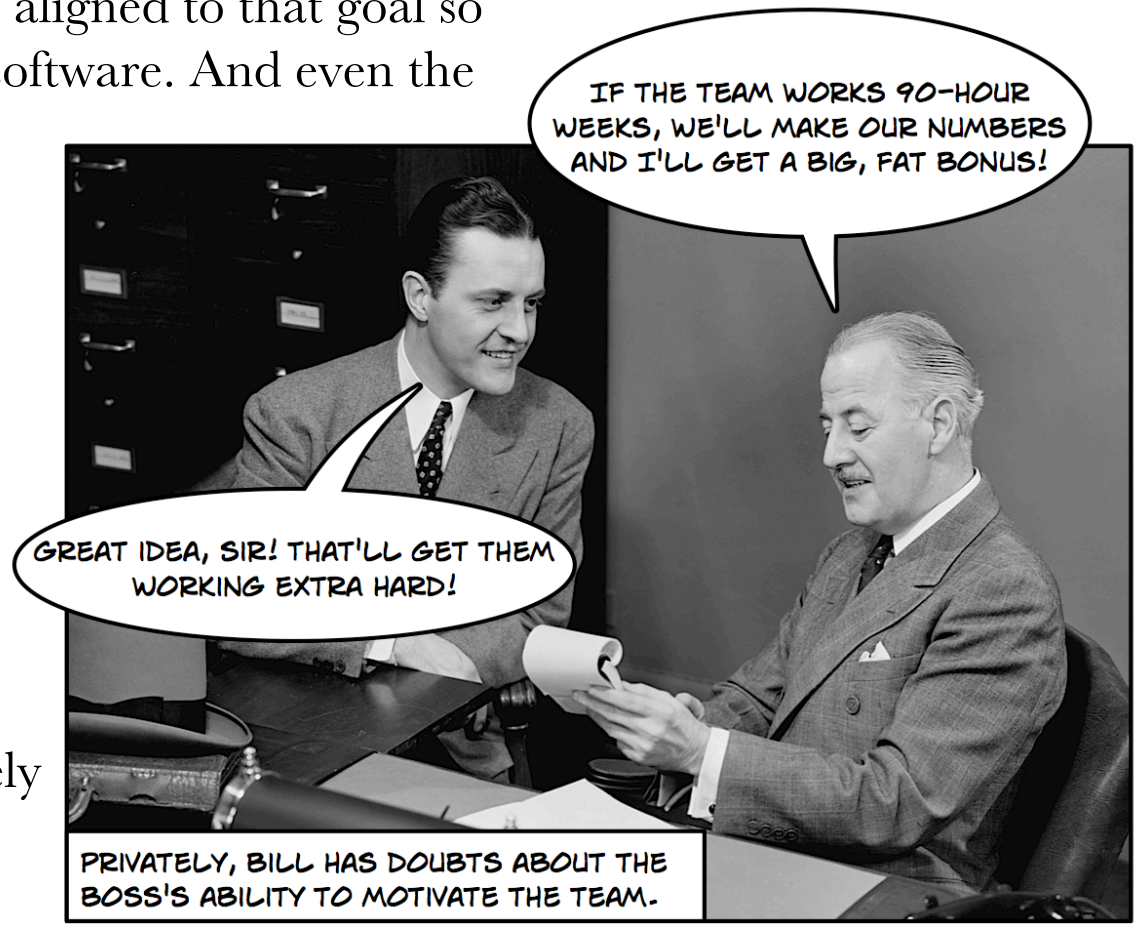
Beautiful Teams contributor Andy Lester spent his career figuring out what motivates developers, and how to help them integrate into their teams.

> *I was on a team once where I said, "At the very least, can we just have minimal respect for everyone here?" And I was asked quite seriously by someone else, "Well, what if not everybody on this team is worthy of respect?" And that's baffling to me as a human, but it's also not uncommon. And that minimal amount of respect is something that many just don't get.*
>
> *Andy Lester (chapter 5)*

Goals

One of the biggest challenges of working with a team is keeping everyone aligned to that goal so that they build the right software. And even the best teams can have conflicts around those goals, conflicts that can tear a team apart.

But if you align people to those goals from the beginning, and keep everyone in the loop as they change—and they always change—the project is much more likely to be a success.



Up close: Establishing an "elevating goal"

Beautiful Teams contributor Steve McConnell talks about how a team leader needs to establish what he calls an "elevating vision or goal," and he's not the only contributor who mentioned that.

> *If you're out digging ditches, that's not very elevating or inspiring. But if you're digging ditches to protect your town that's about to be attacked by an enemy, well, that's more inspiring, even though it's the same activity. And so the leader's job, really, is to try to determine or frame the activity in such a way that people can understand what the value is.*
>
> *Steve McConnell (chapter 16)*

Practices

It takes a visionary to see the value in a new practice. It takes a salesperson to convince management to pay for it, and to convince the team to do it. And in a lot of cases, it takes an above-average team to be open-minded enough to change the way they work.

Up close: Better practices at NASA

Beautiful Teams contributor Peter Glück explained how the practices they use at NASA's Jet Propulsion Laboratory keep spaceships working.

Our missions to Mars cost $400 million, and your typical business application only costs $1 million to develop. You're looking at a difference in scale and in cost of failure. The cost of failure if a business application doesn't work quite right is a few irate customers. It can be bad for the business if they really get it wrong, but chances are in the nominal cases it's going to work fine, and in the off-nominal cases they'll get some complaints and they can issue a patch. If we get it wrong in the off-nominal cases, we may not get a second chance.

Peter Glück (chapter 18)

Obstacles


DICK DISCOVERS WHY IT'S NOT ALWAYS EASY BEING A BUSINESS ANALYST
SO I UNDERSTAND YOU NEED A HORSE.
THAT'S RIGHT, A HORSE. RIGHT AWAY.
BUT YOU DON'T NEED TO RIDE IT.
OF COURSE NOT. I SAID I NEED A HORSE.
AND IT SHOULD HAVE HORNS, AND YOU'RE GOING TO NEED TO MILK IT.
WHAT ARE YOU, A HALFWIT? WHAT ELSE WOULD I DO WITH A HORSE?
I'LL SEE WHAT I CAN DO.

Up close: Getting past obstacles

Beautiful Teams contributor Scott Ambler spent his career focused on making enterprise-wide changes, and doing that successfully means having to overcome obstacle after obstacle.

> *You can't always tell when a team is dysfunctional. If there's negative shouting and screaming, that's a problem. But sometimes, you can be on a team where everyone's trying, but nobody's communicating, and nobody's reaching their goals.*
>
> *The easy analogy is that during the day you've got people digging a hole, and at night other people are filling it up. Everybody's working really hard digging and filling holes, but in the end nothing of value is actually occurring.*
>
> *Scott Ambler (chapter 26)*

Bringing it all together

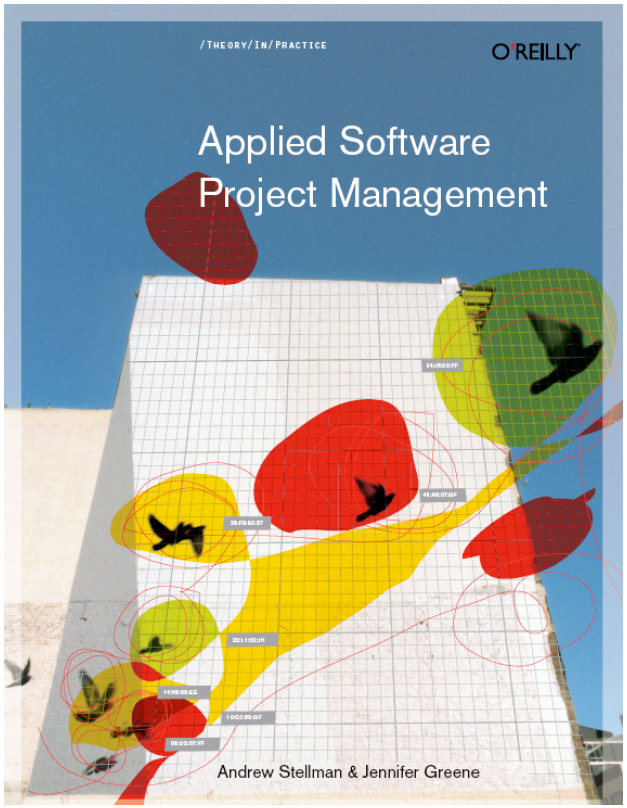
For me, the essence of getting people to work together is to have an aesthetic vision that you can get them to sign up for. Where you build a shared vision of the truth that you're building, where you've expressed an ideal. Because then you set people free to pursue that ideal on their own.

Tim O'Reilly (chapter 1)

One of the signs that I have for the health of an organization is that they're reticent to fail. Organizations that are totally anal and refuse to fail at all tend to be the least innovative organizations, and they're hardly any fun, because these people are so fearful of failing they take the most conservative actions. On the other hand, organizations that are freer in failing, not in a way that will destroy the business, but are given some freedom to fail, are the ones that are more productive, because they're not in fear for their life with every line of code.

Grady Booch (chapter 9)

One last quick note from the marketing department...



Buy these books!

And check out our blog, "Building Better Software"

http://www.stellman-greene.com/

We'll post these slides there in the next few days.