

Instilling Agile Values for Creativity, Self-Improvement and Organizational Change - A Manager's Perspective

Overview

The scale and speed of an agile adoption are external measures that don't speak to the founding values of the practice. Collective ownership, continuous improvement and high trust are hard won, take time and discipline but lead to craftsmanship and joy. They are enabling conditions for innovation and beneficial change.

I will retrospect on my contributions both positive and negative towards cultivating these values in two teams. The first was a practice that matured over four years, led to a new mission for the team and direct collaboration with the founder and CEO. The second, is a new team finding its way at the end of its first year.

What will I do more off? What will I do less of? What impediments got in the way?

Agile Values

...the meteoric rise of interest in and sometimes tremendous criticism of Agile Methodologies is about the mushy stuff of values and culture.

– Jim Highsmith (<http://agilemanifesto.org/history.html>)

Agile Manifesto

We all know the agile manifesto:

- Individuals and interactions over processes and tools
- Working software over comprehensive documentation
- Customer collaboration over contract negotiation
- Responding to change over following a plan

As advocates we tend to focus on practices. To justify these practices we talk about delivering value faster and more frequently, realizing cost savings and increased quality.

But we define “value” too narrowly in terms of short term metrics or cede the definition entirely over to others. At the same time we minimize how hard it is to achieve an agile practice.

So, we set ourselves up for failure.

The astounding truth is that value - *narrowly and subjectively defined* - does not require a humane workplace or a maintainable code base. It does not even require that the software be useful.

Instilling Agile Values - A Manager's Perspective

That's why we need something more than case studies and a toolkit of practices. We need the manifesto, a set of principles for a way of doing which despite human fallibility and economic necessity inspires us to create good work .

Which is also why Bob Martin is lobbying to add a fifth principle:

"We value craftsmanship over execution" (or "craftsmanship over crap")¹

But beyond the manifesto, we need to lobby for value in the larger sense of the word. For both the creativity and happiness of the people who create software and the benefit to the distant end users who are confronted by the software they create.

To what end?

"... I've got to have more experience with junior [children] than a lot of the people who are telling me what I should be doing with them... I think I could help bring a lot to it and nobody ever asks... They just go ahead and proclaim and we have to follow."

– Anonymous Teacher²

As a manager, my goal is to deliver according to senior management's goals for the business as best as I interpret them through the efforts of other people, my team. People make their best contributions when they have a clear idea of what they are supposed to achieve and are empowered to work together as they best see fit to accomplish that end.

People deserve dignity. They deserve to feel pride in their work. They should have an opportunity accomplish something meaningful. They should have their measure of creativity, joy and personal fulfillment both at the office and at home. They should be called upon to temper their own self-interest for the benefit of the community in which they live and work.

Values explicitly addressed in the manifesto

- Client/Employer - delivering business value, visibility
- Product - adaptability to changing requirements
- Colleagues - collegiality and collaboration
- Self - Sustainable work/life balance, excellence, retrospection

Values not explicitly addressed in the manifesto

Other Stake holders

- society
- end users

¹ <http://blog.objectmentor.com/articles/2008/08/14/quintessence-the-fifth-element-for-the-agile-manifesto/>

² **What's Worth Fighting for In Your School**, Andy Hargreaves & Michael Fullan

Instilling Agile Values - A Manager's Perspective

- peers and reputation of field

We don't just build the bridge we're told to build...

- public good
- access to benefits to the disabled or economically disadvantaged
- temper all technical judgements with human value
- conflicts of interest
- learn applicable standards and laws
- avoid associations with unethical organizations

We should leave the campground better than we found it³

- volunteerism & good works
- report ethical violations
- encourage colleagues to behave ethically
- avoid prejudice

Instilling (a retro*)

Disclaimer

* Clearly I will respect my past and current employers, coworkers and colleagues.

Definition of the word⁴

in-still - To introduce by gradual, persistent efforts; implant: "Morality . . . may be instilled into their minds" (Thomas Jefferson).

People

Nothing that isn't in a ton of books. I won't spend too much time on this.

Goals

- Passionate
 - care about quality
 - care about improving their own skills
 - need to accomplish something
- Courageous
 - admit their own limitations
 - take criticism from peers
 - give constructive criticism to peers
- Smart
 - Sensitive to other people
 - Know their s---

What I try to do

- involve the team in hiring peers - group interview

³ Bob Martin Clean Code tip (borrowed from the Boy Scouts)

⁴ The American Heritage® Dictionary of the English Language, Fourth Edition copyright ©2000 by Houghton Mifflin Company. (<http://www.thefreedictionary.com/instill>)

Instilling Agile Values - A Manager's Perspective

- skills test made by the team
- encourage team to participate/contribute to the community
This happens over time if you get to know your team as people, learn about what they want.
- help them know what they want their next step to be
community projects and groups and even better if you assume a leadership or mentoring role. Much better to present at than attend a conference. Great recruiting tool.

What I need to do better

- keep in touch with the people
Informal or formal but regular one on one time
- keep touch with the work
Participate with the team, find time to mix in and code if possible but don't become a bottleneck. The little overhead you create will be made up for your improved ability to feel your team's pain and to root out their impediments.
- make changes as soon as necessary
You have to know when someone is a liability to the team and make the change. I am not satisfied that I have done this as soon as I could have.
- create context to give feedback
I have no problem giving honest feedback in the right context but unless I create that context for myself, i.e. one on one retrospection on a frequent basis, I'm not as effective.
- maneuver the recruiting world
All resume's lie. Coached resume's lie absolutely.

What I need to never do

- The guy who eats a sandwich during meetings
Almost every good leader I've worked for has at one time or another had a trusted lieutenant who didn't carry their wait. He's the guy who sits in the meeting reading his blackberry or eating his lunch. Who is always very busy but who doesn't DO anything. What they're good at is taking credit for other people's work and selling it to their leader. I neither want to be that person nor do I want to have one of those people.

Team

This is the heart of the managers work is to build a good team.

Goals

- colocation
- high trust
- 6-8 of them per team
- diversity
- matrix across disciplines
- project teams

What I try to do

- articulate a cause
- don't lie to them

Instilling Agile Values - A Manager's Perspective

- defend them
Team building is not a linear progression to happiness. There will be wrong turns. The team will also be held accountable to things they have no control over. You have to advocate for your team sometimes even to the point of becoming the dead sheepdog.
- let the team take risks and hold them accountable
My current team pushed for a major refactor of a less than one year old codebase visibly adding cost, risk and time to a major project. I defended that decision because I believed they had the hands on knowledge I did not, because I knew they could pull it off, and because I need a team that rallies to a cause and to each other, champions quality and owns the code.
- celebrate victories
Acknowledge those small victories along the way. Incremental changes that improve the team or the relationship of the team to the larger company. Definitely cheer the big victories, even if you find yourself the only one giving a standing ovation in the auditorium.
- identify an “other”
It helps to compare your team’s performance with another group. At best it can be a beneficial mutual competition. I’ve heard of that happening in large agile adoptions. But in my case, it’s usually us rallying to be better than “them”. Just make sure you aren’t dependent on them to deliver your projects because then you’re enflaming the politics.
- allow myself to be overruled (divided team)
- give them time to grind the pebble smooth (Storm)
- adopt their own practices make their own decisions - allow self-organization
*In one team, we spent the better part of three months teasing out the definition of roles for UX, product owner and developer. It took some fraught conversations teasing out what issues were tied to business value and which to subtleties of user experience. The product owner had to learn to not answer questions and let the UX director step into the vacuum.
On that same team, we were planning to divide the team in two to take on different streams of work. The team refused and opted to work on both backlogs within one sprint. We worked on a 60%-40% allocation. Because they owned the decision they took upon themselves the extra work to make it happen.*
- bring in outside expertise to challenge and validate
- bring in a “closer”
Someone in the team lead or scrum master or delivery manager role who closes out tasks, who can be trusted to trail blaze a new role and who gives me unvarnished feedback

What I need to do better

- functional roles as process boundaries (“handoff’s suck”)
- not one team - offshoring
- “certified agile”
- banishment to the admin tool
- giving too much information

Instilling Agile Values - A Manager's Perspective

I think I've actually learned this lesson for the most part. In my effort to be transparent to my teams, I used to let them know too much about the impediments I was working on for them. As one of my team once told me, "I don't need to know this because I can't do anything about it. All it does it depress me."

Ecosystem

This is what determines whether the team creates real value, whether you are able to manage according to your values and whether your team will survive.

Goals

- Where the people with vision collaborate directly with the people who have the means to build that vision.
- An organization that can consistently create original, valuable products/services
- An organization that people want to work for
- An organization that respects its employees and receives the most contribution from its employees.

What I try to do

- choose leaders that are worth working for
You're entrusting them with your labor and you're recruiting and building a team off of you're own reputation. Make sure you have something to learn from them and something to accomplish with them. Make sure they will protect you if you need and deserve protection.
- demonstrate loyalty - earn trust
People seem to be surprised that they have to earn trust. My experience is that you always have to earn trust particularly if you want permission to build the team you want and manage them the way you want. The best way I've found to earn trust is to deliver on an ambitious goal but it also involves small actions that let your boss and co-workers know you have their back. Genuine collegiality helps. False collegiality hides dysfunction and is a form of politics.
- walk in other people's shoes
A cliché, but don't judge a co-worker until you've tried to do what they do
- know when you're winning
Best advice I ever got from an entrepreneur is know when things are turning in your direction because there's a difference between a death threat and death throw and if you're not aware, you'll over react and hurt your own momentum.
- make enemies
When doing so creates necessary change. Sometimes you need to fire the client even if the client is another department. Better have earned some trust with your boss before you do this.
- not make enemies
When doing so does not effect change. This just enflames the politics. You all lose sight of why you're there.

What I need to do better

- encourage the leader to collaborate with the team

Instilling Agile Values - A Manager's Perspective

- help the team participate in brainstorming the thing
- place each project in context
- not becoming a black hole of information
- attack the what is value? question

We go into so many projects with subjective and volatile measures of success. How can a team rally to that. Fight to isolate a set of measurable benefits but realize that if they don't capture the real reasons why a project is being done delivering on them won't matter. Be willing to acknowledge intangible benefits: corporate reputation, demonstration of ability, learning for the company...

Things I can't control

- time - lifecycle of the business/lifecycle of the team (local optima)
There's a huge risk it will take longer to build a real team than a business can maintain the specific environment required to sustain that team. Sounds negative but it sucks to spend years building something to have the business change management and for that team to hollow out in weeks.
- top secret
Any business has secrets. Sometimes those secrets are material to decisions you would make about a product roadmap. Sometimes, the people shaping that roadmap don't know. Sometimes knowing that information would change the product enough to avoid failure.
- the budget cycle
If we don't know what we want until we see it, how can we know how much it will cost months before we start work. Annual budget cycles divert so much energy and create so many artificial constraints on projects. Because the problem is, sometimes you can't solve a time and budget constraint by trading scope. Sometimes you're just baking in failure.
- hierarchical organizations
command and control wastes so much initiative and hands on knowledge.

Ken H. Judy
<http://judykat.com/ken-judy>