

# 8 Years Agile – From Startup ScrumMaster to Agile Coaching Group at a Company of 500

HEIDI HELFAND, AppFolio, Inc.

---

As a startup, we felt so strongly about XP and Scrum that we implemented them from a rather command and control perspective. Team dissatisfaction emerged. We dramatically changed our approach. What resulted was the development of a culture based on autonomy, team building and trust. We grew from having a dedicated ScrumMaster into a group of internal coaches supporting self-organizing teams that choose how they work. In this experience report, I will share the ScrumMaster's perspective in particular, and how we organize our Internal Agile Coaching Group.

---

## 1. THE SCRUMMASTER ROLE AT OUR STARTUP

8 years ago I became a ScrumMaster at a startup that had 10 people. There were mostly engineers, 3 of whom were remote. I had some social capital with this group since we worked together at a prior startup, and because of this they hired me into the new company strategically. I did not have training as a ScrumMaster, and so I went to Ken Schwaber's two-day course to become certified. I learned in other ways too, such as by talking with a local ScrumMaster who I was introduced to by one of our founders. He told me that "being a ScrumMaster is a process of making yourself obsolete." I did not know what that meant. It actually frightened me because I was, at the time, a Waterfall project manager converting into a ScrumMaster.

I also learned about my role from consultants engaged with us. Pivotal Labs was with our team for about a year and helped us really master XP principles and practices. Their engagement with us was very successful, and gave us an incredible foundation for years to come. What emerged, was a hybrid "Pivotal process" which involved pair programming 100% of the time, the use of Pivotal Tracker for managing our work items, continuous integration, test-driven development and some Scrum-like meetings such as standups and planning. We also wrote user stories and acceptance tests that we estimated with points. We planned our releases leveraging Tracker's planning capabilities, which at the time projected your release date based on story estimation. Here's what my role was like.

I facilitated Sprint planning. I worked with the product owner (who was at the time one of our founders) to understand and get the backlog ready for discussion prior to sprint planning. I'd get all of the stories into Tracker and ready for the team to discuss during the meeting. At the actual meeting, I connected in our remote engineers with GoToMeeting, and facilitated the discussion of each story. We added acceptance tests and played planning poker to get estimates. I also worked with the product owner to be sure that future work was groomed by pairs of engineers, and that it was in the Backlog in Tracker, ready to be picked up.

Coordinating the daily standup meetings was also my responsibility. We had three remote engineers at the time, and I was responsible for connecting them into our meetings. I held the space so that the team would focus on the three Scrum questions of the time - What did I do yesterday?, What am I going to do today?, and Are there any blocking issues? Beyond this, in each standup, engineers would determine who they would be pairing with that day.

During the sprint, I closely followed the Tracker stories that we worked on, and paid special attention to the state of each work item. If something was in a certain Tracker state (started, finished, accept/reject) for too long, I would ask about it. Any "clog" in the flow I would consider an impediment. In addition, I would view it as an impediment when someone did not have a pair.

Since we were a small startup, we were not fully staffed with all of the roles you might find in a more established company, and I helped to fill in the gaps. What this meant was that I wore other "hats" beyond ScrumMaster. For instance, I was the facilities department. I'd stock the snack cabinets, and help with anything related to office space and furniture. I'd order it, I'd assemble it (and get others to help) and I'd deliver it. I was also the Google Admin, a.k.a. the IT department. I would entitle people for email and I

would take misbehaving computers to the Apple store for servicing. I was also people operations. I would onboard new employees in and outside of our Engineering group. I was also a technical recruiter. I would organize recruiting events at the local university, post engineer job ads online, and I would schedule and host interviews for prospective team members. We grew steadily in the beginning, adding a few engineers to our group every few months. To add even more, I also planned and ran events. I'd even take out the trash if it needed to be done.

Those were a lot of different hats to wear. But it was interesting and different, and people pitched in to help when it wasn't distracting to them. As the years went on, I was able to "shed skin" as we gradually hired in people to fill those roles. In retrospect, I think it was a good fit to pair a ScrumMaster role with doing all of these other things at a small startup. No task is too big or small to take on if you have a servant leadership perspective.

Besides these other activities during the sprint, I also facilitated regular retrospectives with the team. I tried out different retrospective formats from the Agile Retrospectives book by Derby and Larsen. I watched our Pivotal consultants conduct retrospectives, and copied what they did. Retrospectives happened every 2 weeks to monthly. We have had consistent retrospectives like this for years. To this day, they help us to iterate how we work together as a team. One particular change that came out of the retrospective was the desire by the team to change to more "formal" Scrum as I'll explain next.

At one point, some team members read the book "Scrum and XP from the Trenches" by Kniberg, and were interested in doing sprint burndown charts, a practice that we had not tried yet. So the team decided in a retrospective that we should start creating a sprint backlog of developer-centric tasks that we would estimate with hours during each sprint planning. These would be then be expressed in burndown charts. This meant that on a daily basis, engineers needed to update their tasks with hours remaining so that we would have the data for the burndown charts.

Creating these burndown charts was a manual process due to lack of functionality at the time in Pivotal Tracker. It became my job to create them in Google Docs. But to do that, I needed the remaining hours from engineers at the end of each day. So my new daily ritual was to remind engineers about this. I'm sure I got quite annoying. After some time, I got feedback that people felt micromanaged by this activity. We also wound up abandoning this activity after some time since no one looked at the charts and they were essentially waste. We really didn't have a need for these charts.

## 2. COMMAND AND CONTROL AGILE: "WE FEEL MICROMANAGED"

I believe that the pressure of feeling micromanaged was building in our environment because of this burndown chart activity, and also because we were trying to enforce 100% pairing.

In standups, I would make sure that we knew who was pairing with whom. I even kept track of it on a magnetic board with photos of each engineer. I would put magnets with pictures of each engineer side by side demonstrating who the pairs were for the day.

After a while, if you would walk through our workspace, you could see that people were not pairing 100% of the time. Since it was a decision and value of the company to pair, somehow it became my responsibility to find out why it was breaking down, and to try to "enforce pairing". It became very uncomfortable and awkward for me to be the "pairing police." I looked for answers.

At Agile 2010 in Orlando, during a hallway conversation, I asked J.B. Rainsberger for advice on how to get 100% pairing. Instead of telling me some magical solution to my problem (which I wanted), he told me to read a book called "The Three Signs of a Miserable Job: A Fable for Managers (And their Employees)" by Patrick Lencioni.

The book is written as a fable, and describes a situation where a former CEO decided to buy and manage a restaurant in which the employees are very unhappy. The book talks of how he went about turning the situation around. I didn't need to get into the specific tactics that are discussed in the book to get my perspective shifted just enough to realize that I was contributing to a climate of unhappiness at my work. Looking back, I realized that we had a climate of "command and control Agile." People were unhappy. They felt micromanaged. People did not feel trusted. The "miserable job" book hit the nail on the head.

I was not the only one who realized this. Team members began to speak up. Shortly after that, we did the first of what has become an annual survey in engineering where people can share how they feel about our workplace individually and anonymously. It worked like this. Everyone got a survey via email from an outside consultant to answer these questions:

1. What is going well?

2. What could we do to improve the engineering team experience?
3. Of all of the things you thought of to improve the engineering experience, what is the one thing that would have the biggest positive impact?

Figure 1 shows some of the raw results that we compiled in 2010, which guided what I view as a “plate tectonic” shift in our engineering culture.

The four top areas going well:
--We have a talented team of smart, skilled people.
--The team is willing to help each other and collaborate.
--Test driven development works well and we have a good automated test system.
--We are agile and respond and make changes quickly. We do regular software updates. We continue to move at a rapid pace. We rarely miss a release.
The four top areas to improve:
--Reduce the amount of testing that engineers have to do. The zerg [manual testing cycle by the whole engineering team at the end of each sprint] is frustrating and takes too large a percentage of engineering time.
--Allow engineers to decide when and how much to pair. Pairing does not make sense 100% of the time.
--Have clear career goals and more coaching and management for engineers. Introduce architect roles. Have goals that lead to career objectives. Meet more frequently with engineers. It would be good to get more frequent feedback.
--Trusting the engineers. To manage their time and decisions. Be fair. [sic]
The top four areas that would make the biggest impact:
--Management and coaching for engineers. Clear career path. Career development. Team leads
--Relaxing the pairing policies
--More trust of the engineering team
--Hire a VP of engineering

Figure 1: Sample of our engineering survey results, 2010.

The action our Engineering management took when reading and digesting the results of this survey changed everything. Here are some of the specific things that resulted from this exercise.

- We had an organizational shift that promoted an engineer to be our first Engineering Director who built out career paths. People got more attention.
- We came up with a plan to add more test automation and had the vision in place to have “zero bugs” which influenced our teams going forward to distribute and fix bugs earlier in the cycles. Having “zero bugs” meant that we would not carry forward a queue of bugs that we did not fix in a timely fashion. After some time, we were able to abolish the “waterfall-esque” end-of-sprint testing cycle that we named the “Zerg”. Four years after, in 2014, we attained the goal of “zero bugs” and have maintained that since then.
- We also got the support from our management to relax the pairing. To address this, a group of engineers got together and figured out how they wanted to approach pair programming. They wrote a guideline and shared it with everyone in order to have buy in. In sum, the guideline was that developers could pair at their own discretion. If something made more sense to solo on, they would do that and get a code walkthrough. This act gave a lot of freedom, autonomy and trust to our engineers. Eight years later, if you walk through our space, you will see developers pairing, some soloing, and some mobbing, thanks to a visit from Woody Zuill last year. People have the freedom to work as they wish.

- As the ScrumMaster at this time, I took in all of the feedback and felt relief that I could change the way I was approaching the teams based on these newly identified needs.

In hindsight, I can say that we were starting to focus, as the Agile Manifesto emphasizes, more on the individuals and interactions and less on the perfect use of processes and tools (such as burndown charts and 100% pairing). In addition, doing qualitative research like this survey was the first indication to me that the role of the ScrumMaster (and later Agile Coach) needs to be guided by a needs analysis and not just passionate Agile or XP ideals.

### 3. ANZENEERING

Although we didn't know it at the time, we were starting to take more of what I interpret as an Anzeneering approach to our culture development (Kerievsky, 2014). "Anzen" is the Japanese word for safety. Joshua Kerievsky talks about safety in software engineering in a 2014 blog post. He talks about protecting people by establishing "anzen" in everything from relationships to workspaces, codebases to processes, products to services. He asserts that "protecting people is the most important thing we can do, because it frees people to take risks and unlocks their potential."

Our emphasis on policing perfect pair programming 100% of the time, as well as combining it with micro-Scrum burndown practices was not healthy, and it did not encourage any risk taking. Instead, it led to a lot of unhappiness as I have described. So we wanted to change, and needed to change into an environment of trust. In my view now looking back, and from my personal interpretation of the Anzeneering concept, we were starting to cultivate Anzen, and the development of a healthier workplace in particular. Listening to what our team members had to say and adjusting the environment to meet the needs in order to support them was the new focus. We shifted into a more bottoms-up mode.

### 4. THE TEAMBUILDING ERA

I did a lot of reflection on what I wanted my role to be like going forward based on the survey feedback. How I approached my role initially as the micromanaging pairing police and burndown nag was obviously not working. In particular, I thought a lot more about the statement, "Being a ScrumMaster is a process of making yourself obsolete", as I mentioned at the beginning of this report. At this juncture, I started consciously applying that philosophy. Here's how.

I set things up for the Scrum ceremonies - logistics etc, and just "held the space". I would make sure that the group convened, but I took a side role. By this time, either a product owner or developer would facilitate the sprint planning. I didn't need to be at the center of those meetings controlling the keyboard. Someone else closer to the information under discussion would be at the keyboard. I would sometimes skip standups and get people to be less reliant on me. I was also not religious about asking the three questions. Whatever the team wanted to do was fine. If two people went off on a tangent together for a while I would interrupt. Otherwise I let things go and just basically went into observer mode. Grooming over the years turned into a meeting led by the product owner. Many times I didn't attend. I did, however, own and keep the retrospective meeting, which helped us iterate on how we wanted to work together. I facilitated that continuous improvement. All in all, I feel like I was less of the "project manager ScrumMaster" focusing on the minutiae. I was actually becoming a coach on the side, but I didn't know it at the time.

Instead, what I focused on, was to try to elevate the happiness levels among the team. Lencioni's book talks about creating workplaces where people enjoy their jobs, and look forward to coming to work. I wanted to encourage that in our environment. So I was the catalyst who organized several extracurricular events. Our CTO and co-founder started to call this the Department of Fun. We had competitions monthly. For example, when we acquired new office space at one point, we had a huge, empty room next to ours. So, to make the most of it, we brought in a badminton net, and had a badminton competition.

We had other competitions too, such as a Guitar Hero video game competition, a juggling competition and we even made competitions out of boring things we needed to do such as putting together chairs and other office furniture.

Company-wide to this day we still have our guacamole making competition close to Cinco de Mayo in which members of every department are placed on teams together to battle for the best tasting guacamole, the best presentation, and most creative award. This is one example of how we've tried to help people get to know each other across the company as the years went on. I focused on making this event successful.

Another all-engineering and all-company event that emerged at this time was our Hack Day. Inspired by Atlassian's Ship It Days, we started the tradition of having a 24-hour period in which our Engineering and Product organizations as a whole form teams to build whatever they want from Thurs at 2pm until Friday at 2pm. We compete for handmade, goofy trophies and bragging rights for Best Technical Achievement, Most Creative, Most Company spirit, and Best Overall awards. Product features and enhancements have resulted from this innovation-driving event. We have replicated this event with the whole company multiple times, and have done more than 8 of these events over the years. Organizing and putting this on was a focus of mine.

We also took trips together as an engineering team. We've done that since our early days even before this culture shift I've been describing. The first trip that we had was white water rafting. Ever since then, rafting oars have become a symbol in our company to mark major milestones, such as the launch of a new product or the acquisition of another company. Annually we have done other trips together since then such as camping on an island off our California coast called Santa Cruz Island. We've even taken a trip together to Disneyland.

Besides taking trips together, we'd also do things locally in our beautiful Santa Barbara environment. We went to visit the Monarch butterfly preserve. We went hiking together. We were essentially tourists together going to local festivals like Old Spanish Days Fiesta or the Avocado Festival in Carpinteria, CA. These activities would be "opt in" for team members, and we have not had more than one per month. After all, we had software to build and ship.

So I became a "hands off" ScrumMaster and really emphasized these types of teambuilding activities and being a "support on the side" to teams. I trusted that people would do the right thing. Our management team trusted everyone. And teams delivered functionality that we released to customers on a regular basis, which showed our progress. We didn't need any special metrics to show the productivity we had. I really think that trusting the team as a result of our first engineering survey, doing continual retrospectives, and having regular team building events and outings repaired our culture.

What I've also learned is that retrospectives aren't enough to bring about the change that might be needed in a context. Collecting other types of qualitative feedback from team members such as via the individual, anonymous survey can help identify areas in the environment to improve so that people feel safe and comfortable to do their jobs without fear or obstacles. Doing research on the "now", or the current context can help guide and shape the support roles needed in an engineering environment. This research can help to illuminate the "starting conditions" that are present at the time, which help guide ScrumMasters and Agile Coaches in how to focus their roles so that they are "fit for purpose". (Anderson, 2015)

## 5. FROM SCRUMMASTER TO AGILE COACH

As I moved farther away from coordinating team logistics, I learned about Agile coaching by participating in an online coaching circle with Lyssa Adkins. I also went to a Coaching Agile Teams workshop. Through these experiences, I learned about the Agile Coaching Framework, which has been an influence in my own coaching, as well as in our Internal Agile Coaching Group.

During the teambuilding era that I have described thus far, I discovered and used this framework to guide my own growth. I felt like I had pretty deep knowledge of Scrum after all I'd been through, and also of teaching and mentoring due to my past training and work as a teacher, but what I really lacked was an understanding of what it meant to be a coach. I felt like I self-identified more as a coach rather than a ScrumMaster. I had stepped back a lot in the care that I gave to our nearly 7 teams. I was encouraging others on the teams to step up and keep our work flowing. And it was working. I decided to get coach training at the Agile Coaching Institute to learn more.

Soon after that, I wound up rebranding my role and title to be Agile Coach. I explained this rebranding to team members at the time simply by pointing out that in many teams I have been coaching others to step up and keep the work flowing through the teams by removing obstacles etc. And I also pointed out that we practiced Scrum on most teams, but that some were starting to experiment and practice Kanban. So it felt awkward to have a title tied to just Scrum.

As time passed, we hired another Agile Coach. He was influenced by the Agile Coaching Institute as well, and had deep knowledge of facilitation from an organization called Community at Work. He became the coach for two teams and I handled the rest. We both reported to our VP of Engineering. After 6 years I finally had a collaborator in the same role! Together we learned, and expanded our knowledge of coaching. What resulted from our collaboration was a different type of team building emphasis.

## 6. BUILDING RELATIONSHIPS IN THE TEAM

I have emphasized the idea thus far that taking trips together and going on outings and playing sports together helps people build and maintain relationships for successful team bonding. These shared experiences are like “relationship glue” or “team glue.” If you know, care about, and like who you are working with, then everything else is easier. John Whitmore in Coaching for Performance (2009) writes about this in the “Coaching Teams” chapter of his book.

We started growing and adding more team members at a rapid pace. We grew from having about 30 engineers in Jan 2013 to about 90 engineers in 2015. What we viewed as rapid growth became our new norm, continually creating, new starting conditions.

As coaches, we would handle this growth by helping team members get to know each other as people by doing various activities such as “team startups.” We again drew from our training by Lyssa Adkins and Michael Spayd and leveraged many of the activities that they taught us in their training such as Market of Skills and Journeylines. (Adkins, 2010).

This growth continues even today. We also incorporate more and more work from Organizational Relationship Systems Coaching (ORSC) such as helping teams “design alliances” together when they first form as a team. Creating a designed alliance is when you get with the team and together derive operational guidelines for how they want to work together as a team. We also further that by designing conflict protocols; i.e. identifying the ways that we want to behave with each other when we are in conflict. We view conflict as natural on teams. Conflict suggests that changes in the team need to emerge.

Besides when entirely new teams form, we also do these sorts of team identity and operational decision-making exercises when team compositions change. If there is a new person that joins a team, we typically review and adjust all of these things. We also have team kickoff social activities when we have new and changed teams.

Relationship building, and having coaches focus on helping the teams gel as quickly as possible has been a strategy that we have applied when faced with extreme growth as I have discussed.

## 7. BUILDING OUR AGILE COACHING GROUP

It became clear that we needed to hire a third coach. We decided we would try to have one coach for every two teams so that we could be present to coach team members at all of their meetings in order to understand what they would need to reflect on at their retrospective.

We decided to hire an apprentice, and “grow our own” Agile Coach since it was at the time extremely hard to find an experienced Agilist with a coaching mindset, wanting to relocate to Santa Barbara, CA. We guessed that it would probably take 6 months to get someone up to speed (and probably just as long to find a non-consultant, experienced Agile Coach willing to relocate to our coastal town). We knew we could mentor someone into this role.

When thinking about the qualities or characteristics in the person we wanted to hire, we set out to find someone who: 1) had a voracious appetite for learning - someone who reads a lot and loves to learn as they would need to learn on the job. And 2), we were looking for someone who was not a self-centered attention grabber, or needing to be the “hub” of information. Rather we wanted someone who was authentically interested in helping other people grow, learn and change.

We found a local math teacher who was into Non-Violent Communication and had facilitation expertise. We were confident that she could run retrospectives and bring a learning-focused emphasis to the role. She excelled at the role, and helped us improve our developer mentoring program. Then life happened and she moved out of town. Luckily we had hired two other experienced Agile Coaches to support teams.

I’ve come to learn that the coaches we hire each bring their own interests, strengths and influence on our teams. We have four coaches at the writing of this, and we each seem to have particular emphases - one brings a relationship systems coaching, facilitation and improv lens, the second a project management lens, the third a kanban lens and technical practices lens, and the fourth a teambuilding and one to one coaching lens inspired by the Coaches Training Institute. We are continually looking for other coaches to bring their special skills and talents to join our mission to create a happy, fun and productive environment.

Another role that we have in our coaching group is a Support and Events specialist role. The person in this role is in charge of administrative functions within the engineering team such as collaborating on purchasing supplies and special team swag, catering for team lunches and events, cross team event planning - such as our annual overnight tech retreat and also our bi-yearly 24 hour Hack Day events. She also is in charge of new hire onboarding logistics, and helps with our ever growing summer intern program.

This was an additional “shedding” of skin for me, as I had carried on these responsibilities for years. I list this specialist role here because this specialist could theoretically grow into an associate agile coach role based on their interest and aptitude. By sheer growth of the group, we became too numerous for our Engineering VP to manage, and as such I became the Director of this agile coaching group.

## 8. CONTINUING TO ANCHOR TO ENGINEERING TEAM NEEDS

Earlier in this report I talked about the qualitative feedback that we got from our Engineering team, which helped our organization shift towards a culture of trust and teambuilding. In early 2015 we did a qualitative survey on what the organization needs from Agile coaches. At the time I interviewed Engineering directors, VPs and other team members to derive what the organization feels that they need from us. Coaches ran with this information (as well as their professional opinions) and came up with a current mission statement and initiatives for the quarter, to reflect what the organization needs from us based on our new “starting conditions”. We got buy in for our initiatives and focus, and came up with a big visual board to use for our group in which we keep track of the things we are doing as a coaching group to meet each of our initiatives. We have our daily standups by that board, and use it as a tool to create more visibility into our endeavors. We will iterate on this activity in order to provide the best and most appropriate service to our engineering and product teams as our organization grows and changes.

## 9. CONCLUSION

In this report I have reflected on the past 8 years of the Agile I have experienced. I’ve discussed how we shifted our engineering culture of “command and control Agile” to a focus on teambuilding and interpersonal relationships based on qualitative research that pointed us to change from really “policing” processes and tools to focusing on individuals and interactions. I talked about how I viewed us as repairing our damaged culture by trusting that people could work as they chose - not by mandating and enforcing 100% pairing or any other Agile practice or tool religiously. I talked about the emergence of our culture based on events and teambuilding and the growth of our ScrumMaster role into an Agile Coaching Group. I reiterated that as our company grows and changes, we have different starting conditions and as coaches, we need to be flexible to adapt our approach so that it is continually “fit for purpose.” Furthermore, beyond retrospectives, I have also illustrated the need to do regular, qualitative research or needs analyses to guide the focus of our Agile coaching group so that we remain grounded in reality and not off in the clouds emphasizing passionate agile ideals.

### REFERENCES

- Derby, Esther and Larsen, Diana. "Agile Retrospectives: Making Good Teams Great". Pragmatic Bookshelf, 2006.
- Kniberg, Henrik. "Scrum and XP from the Trenches." Lulu.com, 2007.
- Lencioni, Patrick. "The Three Signs of a Miserable Job: A Fable for Managers (And Their Employees). Jossey-Bass, 2007.
- Kent Beck et al., Manifesto for Agile Software Development. agilemanifesto.org, 2001.
- Kerievsky, Joshua. Anzeneering, <http://www.industriallogic.com/blog/anzeneering/> 2014.
- Adkins, Lyssa. "Coaching Agile Teams: A Companion for ScrumMasters, Agile Coaches, and Project Managers in Transition." Addison-Wesley Signature Series (Cohn), 2010.
- Agile Coaching Institute.  
Community at Work.
- Whitmore, John. Coaching for Performance: GROWing Human Potential and Purpose - The Principles and Practice of Coaching and Leadership, 4th Edition. Nicholas Brealey Publishing, 2009.
- Anderson, David J. Kanban Coaching Masterclass materials. <http://dja.com/> 2015.
- Organizational Relationship Systems Coaching (ORSC) <http://crrglobal.com>  
Coaches Training Institute (CTI) <http://thecoaches.com>