



From Developer to Coach – Evolving an Agile Mindset

DAVID FRINK, Fidelity Investments

People become agile coaches from a variety of different backgrounds and for a variety of different reasons. While we all have a shared belief in the power of an agile mindset, we have different experiences that inform our beliefs about what agile truly means. This report details how I grew into my version of an agile mindset and walked the path from developer to coach.

1. INTRODUCTION

“There has to be a better way. I can’t do this to my family again.” I was at the end of a 9-month long project where I’d known from the start that I’d be working 60+ hours a week for the final 2 months and not focusing enough on my new family. And I felt powerless to do anything about it. After I finished the project, I wrote a message to my manager and their manager. It was somewhat long and very professional, but the core message was, “I won’t keep working this way.”

They had already been thinking of breaking from our plan-driven, waterfall process and experimenting with an agile approach. My message was both catalyst and opportunity. We formed a small agile team and got to work. We were immediately successful and very quickly the rest of the development organization started working in an agile way.

That was my first formal introduction to agile. But, like many of us, the more I learned about what it is to ‘be’ rather than to ‘do’ agile, the more I realized that agile simply put words to what I’d felt and done for years. I didn’t find agile, agile found me.

In the same way, I didn’t know what coaching was until I’d already been doing it for a while. And when I found out what it was to be an agile coach I thought, “this is what I want to do, but I don’t know how to get there.”

In retrospect it is obvious that I’ve always been on the path to being an agile coach and a leader who serves. But all paths look that way if you start from the end and look backwards. If you view the journey from the beginning without knowing where each turn will take you, the outcome isn’t predetermined. As Teddy Roosevelt once said, “In any moment of decision, the best thing you can do is the right thing, the next best thing is the wrong thing, and the worst thing you can do is nothing.” Not all of my decisions were the right thing, but the act of deciding kept me moving forward.

I will share with you how it felt to walk the path from an introverted developer to an agile coach. Instead of spending all day trying to cut out distractions by wearing headphones, I now spend all day talking with teams and individuals. I went from being able to measure my progress with working software to only being able to evaluate my progress based on the change in others...something that is not always easy to see or measure. On this journey I moved from being the star of my own movie to an uncredited extra in other people’s. And the irony is I create far more value in the background than I ever did center-stage.

2. BACKGROUND

I am a lead agile coach at Fidelity Investments (disclaimer—my opinions are my own and do not necessarily reflect the opinions of my employer). I lead a team of coaches who support ~100 Scrum teams as well as the managers and leaders who support the success of those teams. In my career I’ve been a solo developer, agile team member, manager, leader, release train engineer, and coach. I’ve formally used agile practices since 2007 and I’ve supported companies small and large, and early, middle and late in their transformation.

The common thread of my career has been the need to learn and grow. Any time I start to feel like I know how to do all parts of my job well, I look for a new challenge or learning opportunity. Earlier in my career, my wife worried that I would never be satisfied because as soon as things started to feel easy, I wanted to make a

change. Instead of needing to change jobs or companies, I found a role in coaching that presents a never-ending series of new challenges and opportunities to learn and grow.

3. THE JOURNEY

On my journey from developer to coach, I made several transitions. My skills development started out focused on technical skills and moved toward people skills. I went from improving my skills for the job I had to proactively investing in the skills I needed for the job I wanted. I uncovered my core values and let that drive my direction, and I began to evaluate success in a new way.

3.1 Developer—The tech isn't the hard part; What agile feels like

I was a developer for 7 years before I had the opportunity to work on an agile team. Some of it was solo project work, some was partnering with other developers both in the US and in India. I was successful, but not completely satisfied by simply implementing other people's solutions.

As I grew from writing overnight batch loading programs to developing web applications, I decided I needed to improve my skills in application design and usability. I discovered the book *Getting Real* [Fried]. This book became the bedrock of what it means to me to work in an agile way. Key lessons like "Fix Time and Budget, Flex Scope", "Embrace Constraints", "Ignore Details Early On", "Epicenter Design", and "Half, Not Half-Assed" seemed to be the most logical way to create a web application. The philosophy of this book became my agile mindset, long before I'd ever heard the term 'agile.'

After writing that, "I can't do this anymore," message to my managers and receiving their fantastic, supportive response, we built a small agile team and started sprinting. Our team was two full-stack developers, one QA/analyst, and a product owner. We were fully co-located into a pod of 4 cubes. We shared all team responsibilities equally. If we needed documentation more than code, we all wrote documentation. If the PO was out when we needed to demo the release to our customers, we all pitched in to do the demo.

We released 3 times a year. We set the release dates a year in advance and never moved them. At the start of a release, we had a set of problems we had to solve plus *some* hard requirements. We'd do a high-level estimate to make sure we weren't about to promise something we couldn't deliver and then we'd get to work. Because the date was fixed and the budget was fixed, we got really good at managing scope. And when I say managing scope, I mean we got ruthless at prioritization.

We did good work, and because we were given the problems and charged with creating the solution, we were able to innovate and create new designs and interfaces. As we innovated and prioritized some features over others, I found myself needing to explain, justify or defend the decisions we'd made. I recognized I needed to improve my ability to communicate and influence for change. The books, *Made to Stick* [Heath2007] and *Switch* [Heath2010] taught me how to communicate complex concepts in a clear and compelling way.

We worked in this cycle of fixed-date releases for 3 years. Three years with the exact same team of people. Those years are the most consistent time in my career that I've been excited to come to work in the morning. When you are on a high-performing agile team, you deliver great value to customers, but you also have fun doing it. Of course, there were bumps along the way, but our shared commitment to our goals and our respect for each other ensured we always moved forward and became stronger as a team.

All coaches can tell you what a high-performing team looks like. Not every coach can tell you what a high-performing team *feels* like. I learned that once you've had that experience, you want to have it as often as you can. And you want to help as many other people have that feeling as possible. That feeling is what motivated me to become a manager.

3.2 Manager—How to know if you've had a good day; Management as a discipline; Create space for the team to thrive

I didn't become a manager because I wanted to manage. I became a manager because I wanted to protect the agility we'd built over the past 3 years. One of the managers who had supported the agile transformation was considering moving to a different part of the company. I was concerned that without their support and advocacy, the teams would lose some of the autonomy that had helped us be so successful. I felt that being a manager would let me protect the agility of the teams.

I was not a good manager at first. I tried to ensure that people were making decisions the same way I would have made them. I was micromanaging and annoying the team. I leaned on my technical expertise because I didn't know how else to help the team. I was uncomfortable not being able to measure my success by the code I wrote.

I didn't receive much training as I became a manager, and I didn't want it to take 10 years for me to get good at the skill of management. I decided to treat management as a discipline distinctly different from what I'd done before. I couldn't think of myself as a developer who was also a manager. Instead, I needed to think of myself as a manager first and learn the necessary skills to be good at my job.

I received some great mentoring and that helped. Plus, I found a podcast that allowed me to focus in on the skills I needed to be effective [Auz]. The hosts assumed you knew nothing about good management and distilled it down to basic practices. I learned how to delegate effectively and the importance of regular check-ins with your team members. As someone who is introverted, I was terrified at having to give redirecting feedback, but I knew it was part of my new role. The podcast helped me break feedback down into situation-behavior-impact and encouraged me to give only positive feedback for several weeks before trying any redirecting feedback. Their structured approach helped me get more comfortable spending most of my day interacting with others rather than coding with my headphones on. I was a little rigid and robotic as I practiced these skills, but because I was at the 'shu' level of experience, I just needed to put in the time and repetition to increase my skill and comfort.

These skills as a manager were foundational to my journey. By the time I became a coach, I was comfortable observing others and seeing what they were doing well and where there were opportunities for improvement. I was comfortable giving both reinforcing and redirecting feedback whenever needed.

I discovered that my favorite part of being a manager was having one-on-one conversations with the people on my team. We would talk a little about the work, but the more impactful discussions were about how they were doing as a team member and as a person. We discussed what they found particularly challenging or rewarding. Those coaching and mentoring discussions became what I looked forward to most each week. Nobody else in the company held 30-minute one-on-ones every week with each member of their team. I got pushback on this approach but decided to do it anyway because it helped the team be more effective.

I began to let go of the details of the work. It was assumed that managers were involved in the details and providing tactical direction. I found that the less I was involved in the details and the more time I spent on the people, the better the work went. I still helped the team arrive at good solutions, but it took the form of asking powerful questions to pressure-test the designs rather than making suggestions on how to do things. I paid attention to my language and ensured that most of the things I contributed were questions, rather than opinions.

I knew if I let go of the details and something went wrong, I'd get in trouble. But I had faith in people and the effectiveness of the teams. It was my job as a leader to create space for the team to have enough autonomy to be creative and own both the problem and the solution.

Although I didn't realize it, I was developing my coaching stance even though I had no idea what coaching was. The skills I developed helped me define what effective leadership looks like in an agile environment. And yet, it felt odd to do things differently from other managers around me. But, I learned that if you're getting the right outcomes, people are less likely to inspect your inputs and outputs. That's been a key approach for me since: deliver such great outcomes that if people come to look at what you're doing, working differently is seen as innovation rather than non-conformity.

3.3 Production Support –Lean in practice; Success at a cost

After managing agile teams doing new feature development for a few years, I moved over to lead a production support team. The production support team was responsible for fixing production bugs, biweekly maintenance releases and keeping the production applications running well. They had a 24/7 on-call rotation and when I joined the team, there were multiple stability issues that made it necessary to monitor and restart servers, day and night.

Initially, I wasn't excited to take the role. I'd been asked to take it multiple times but I was hesitant. This was a group with an extremely hard job. They were often overworked and tired from having to give up their nights and weekends to deal with problems they didn't cause.

I finally decided to take the role for the good of the company. At least, that's what I told everyone else. The real reason I took the role was because the more I thought about the people on the team and their challenging jobs, the more empathy I had for their situation. I wanted to help them in any way I could. If you asked anyone in the company what the production support manager's job was, they would have said it was to ensure the production system never went down. What I've never said in public before is that I defined my primary responsibility as making life better for the people on the team. I believed that a happy and healthy team would create a happy and healthy production system.

The team had been without a manager for a time and they were keeping up with new issues for the most part, but system stability wasn't improving and the team was starting to burn out. After talking with the team, we found we needed to do two things. First, we needed to act less like firefighters and more like fire marshals. Firefighters put fires out after they start. Fire marshals work to prevent fires from starting at all. Second, we needed to figure out how to create enough space to start working in this new way.

To create space to work differently, we started applying lean principles. We defined our value and looked at our value stream. As we examined our value stream, we found several things that didn't align to our primary mission. We asked ourselves, "What are the things *only* we can do?" We were the only team that could keep production healthy, so we decided to prioritize that and deprioritize all other types of work.

At the time, the production support team was also responsible for client feature requests and creating internal tools. These two things didn't align with the mission of the team, and there were other teams within the company who were more capable of delivering on these needs. It took some convincing of leaders to move these responsibilities off the team, but they ultimately agreed. This plus other value stream optimizations began to create the space to get ahead of problems before they started.

Being on the production support team was simultaneously the most proud and most out of balance I've been in my career. After 18 months, I asked to be moved back to manage teams that didn't have on-call responsibilities. I was burned out. I was having trouble sleeping. Every time my phone would vibrate, I would involuntarily tense up and check if it was a production issue that would cause me to drop everything to stay up all night supporting the team as they fixed the problem.

The irony was, we'd been wildly successful. At the beginning, there were many opportunities for improvement. We had servers misbehaving and being restarted multiple times each week. There was a long backlog of bugs to fix. And the testing suite took 2 hours to run each maintenance promotion. I worked with the team to set ambitious goals we didn't know how to achieve. One goal was to go a month without an unplanned server restart. I still remember seeing one of the team members, who had been restarting servers all week, stifle a laugh when I proposed that goal because they didn't think it was possible. We needed to shrink our bug backlog so we could hit our guideline dates for all the applications we supported. For the testing suite, I challenged the team to reduce the smoke test run from 2 hours to 10 minutes so we could have smoother maintenance promotions and take up less of the team's time off-hours.

The team rose to the challenge. We addressed the bug backlog by rearranging the on-call process to allow more focused time for bug fixes. System stability improved as we caught up on overdue upgrades and as we got to the root causes of recurring issues. We achieved our goal of going over a month without any unplanned restarts. For the test suite, I'd assumed they'd figure out how to run the suite in parallel to hit the goal. Instead, they re-evaluated the test suite using a risk-based mindset. They decided that what we'd been calling our 'smoke test' was really a suite of in-depth functional tests that were incredibly unlikely to have broken in lower environments without us noticing. They pared the tests down to the true 'smoke test' of verifying that the system was live and the services were running. 'Maximizing work not done' caused no increased risk or drop in quality.

People had good work-life balance, and we'd figured out how to get several more people into the on-call rotation, so the burden wasn't being shouldered by the few on behalf of the many. We initially worried about who had enough experience to get woken up at 3 a.m. and make a snap judgment that would either fix the problem or make it worse. By using blameless post-mortems [Frink2018], we were able to train new people for overnight support. But, the biggest thing that made it easier to get more people into the on-call rotation was making the system so stable that being woken up at 3 a.m. was the exception, not the norm.

We'd managed to make working on production support a calm, predictable, and fulfilling experience. And yet, I was burned out. I wasn't able to be who I needed to be for my family. Even though there was an on-call rotation, if somebody didn't answer the alert within 3 minutes, it went to my phone. And if I didn't answer it within 5 minutes it went to the head of the department. Even though I wasn't the one who would log in and fix things, I felt like I was 3 minutes away from being interrupted for 18 months. We had fixed most of the recurring problems, and even though getting an alert was a rare event, I felt I had to be ready to react to an alert at all times. I'd managed to help the team fix their work-life balance, but at the cost of my own.

I asked to change roles and my manager supported moving me back to a position that wasn't on-call. I'm appreciative to this day for their willingness to help me find a role with better balance.

Walking away from that team was one of the hardest things I've done in my career. I cared for the people and respected the tough jobs they did every day. I felt like I was letting them down. However, I knew that if I'd been successful as a leader, then the success of the team wasn't about me. If I stepped back and the team

couldn't perform, then I hadn't done my job in growing a high performing team. A high performing team can't depend on one person. That isn't resilient. That isn't scalable. That isn't agile.

Success is about results. But it isn't only about results. True success is the right results, created in the right way with an acceptable cost. I'd helped the team achieve success, but I wasn't working at an infinitely sustainable pace. I wasn't working in a way that worked for me and my family.

3.4 You can't get there from here—What do I value; Practice out loud; Agile job hunting

When I moved out of production support, I was contemplating leaving the company. I was burned out, but that was only part of it. I'd been at that company for 14 years and I wasn't growing or learning rapidly enough anymore. However, moving back to be a development manager was exactly what I needed. I needed time to think about what to do next.

At various points in my career, I've sought the guidance of a career counselor. Whether I was frustrated as a developer or unsure if I wanted to take on a new role, talking with a career counselor helped me clarify what to do next. One of the exercises they had me do was to prioritize what I valued. On the top of my list were family, helping others, honesty, integrity, practicality, moral fulfillment, knowledge, creative expression, and work-life balance. On the bottom were competition, power, authority, structure, predictability, working under pressure, and supervising others.

This sorted list of values was independent of skill or aptitude. I had demonstrated that I could successfully lead a team in a high-pressure environment. But this values exercise helped me understand why I was so drained, even when we were successful. It showed me that what I'm good at and what I find fulfilling can be very different. Prior to that values discussion, my main question in considering a new role was, "Will I be successful?" After getting explicit about my values, it became "Will I be successful, and will it align with what I value and find fulfilling?"

I still didn't know what type of work I wanted to do, but I was starting to get a sense of what skills I needed to do work that aligned with my values. I value knowledge, helping others and doing work that makes a positive change in the world. If I was going to live those values, I realized I would have to get more comfortable speaking in front of groups. As an introvert, speaking up in groups didn't come naturally to me. When I was a developer, I was sometimes quiet and my manager encouraged me to share my perspective in group meetings because they knew I had great insights. I worked on that and got better at it, but I still needed to improve.

Around that time in 2014, a new agile conference (TriAgile) was being organized in Raleigh, NC. My company sponsored the event as a recruitment opportunity. Because the conference was new, the organizers offered sponsors a speaking slot if they wanted it. A few people inside the company suggested that I tell the story of our agile transformation from 8 to 80 people. I was nervous, but I agreed.

I worked on the presentation in the months leading up to the conference with input and support from the other people who had been there from the beginning. The presentation was coming together well, but I had a strong feeling of impostor syndrome when I saw all the agile experts who would also be presenting. There would even be one of the authors of the agile manifesto speaking! I knew I wasn't in the same league with these people.

But as I worked on telling the story of our transformation, I found a new perspective that helped my confidence. I wasn't speaking because I was the expert in some aspect of agile. I was speaking to share an experience and tell a story that would inform and inspire others. As nervous as I felt, I knew none of the other great speakers were capable of telling the story of *our* transformation. I was the only speaker who was an expert in my own lived experience. With that shift, I became more comfortable just focusing on telling our story. I was still nervous about speaking in front of a room of strangers, but I now felt confident I was the right person to tell our story.

On the day of the conference, I had a full room of people. And although I was nervous and I know I used lots of 'ums' and filler words, I had great engagement and feedback. Most importantly, I proved that although I had a lot to learn, I could speak effectively in front of a group.

Although I was working on the skills I needed to live my values, I felt I wouldn't make enough progress unless I moved to a different company. If I stayed where I was, I'd likely stick to my old habits and patterns of thinking. I needed the challenge of learning to work in a new place to give me energy to continue my journey of self-discovery. I needed to see how a different company approached agile.

Because I wasn't confident in my job-hunting skills, I decided to take an MVP approach. I was worried that I'd apply for a job I wanted, but that my resume wouldn't be good enough or my interview skills would be weak. I began applying for jobs I *didn't* want. I was gathering data on whether or not my resume would get me

invited to a phone interview. I tweaked my resume based on the feedback I received and began applying for jobs I *did* want.

The MVP phase of my job hunt also allowed me to improve my interviewing skills. Being introverted, I sometimes have trouble coming up with a quick and fluid response when I'm asked a question. I've got lots of thoughts and ideas, but it is sometimes hard to arrange the words in a way that I can get them out of my mouth without taking a long pause first. I decided I needed to work on the skill of having good, quick responses to interview questions. To practice this skill, I would use my daily commute to record myself answering sample interview questions out loud. I would review the recordings and make adjustments to my answers based on how I sounded. After a while, I stopped recording myself because I was learning to listen to myself as I was speaking and adjust in the moment. But I still kept the practice of running through a sample interview day after day. The goal wasn't to memorize my answers; it was to become comfortable as I communicated. To this day, this is the same technique I use if I have a presentation to give or an upcoming crucial conversation. I'll practice my presentations standing up, out loud, over and over. For crucial conversations, I'll anticipate the questions or topics and practice answering them out loud. Because I've taken the time to pre-wire both my thoughts on a subject and my ability to articulate them verbally, I'm able to be more present in the session and better able to react to the unexpected.

My preparation paid off and I was hired as a development manager at a new company. Exactly 15 years after the day I started, I left my old company for my next adventure. My experience of what an agile team *feels* like, learning the discipline of management, clarifying my values and intentionally working on the skills I needed to move forward laid a strong foundation for what was to come next. The pace of change increased, and a lot of important things happened in the next 18 months.

3.5 “Agile Coach in Dev Manager’s Clothing”

I took a few weeks off before starting the new job and cleaned out my garage. I also built shelves above my garage doors. I had many mis-steps and mistakes along the way. I began to realize that designing and building those shelves caused me to re-learn some agile lessons I'd forgotten. I thought it might be an interesting story to tell [Fink2014]. I'd wanted to share my stories and opinions in writing for several years, but impostor syndrome and fear of judgment held me back. But, I remembered the lesson I learned presenting at TriAgile: “You are the only expert in your own lived experience.” I didn't need to be an authority on a topic before I wrote about it. I just needed to tell my own stories in my own way.

Telling that story was fun and easy. Publishing it online forced me to create a ‘walking skeleton’ version of a website, and that lowered the barrier to future writing and publishing. After that point, all I had to do was write down what I wanted to share and click, ‘publish’. I discovered the biggest risk in writing isn't that someone will judge you; it is that nobody will read what you publish. To this day, the #1 reason someone comes to my site is because they're trying to build overhead garage shelves using slotted angle iron. Whether they stay to read about agile, I don't know.

When I arrived at the new job, things were different from what I'd experienced before. The main office was in a different city. And although the company had a decent-sized presence where I lived, I was the only leader from my business unit in that office. I was simultaneously surrounded by people and alone as a leader at the same time. This was an intentional growth choice. I'd spent the first 15 years of my career co-located with my peers, teams and managers. This meant that it was easy for me to ask my manager's opinion on what to do about something, because they'd been able to witness the situation first-hand. I decided to test whether or not I was good at my job by putting myself into an environment where I had less of a safety net. It isn't that I didn't have the support of peers or my manager. But without them in the office with me, I knew that I would work on issues by myself for longer before asking for help. I also had a team in the UK. This was my first opportunity to manage people who were remote from me. Those were new challenges, plus I was remote from the rest of the leadership team. Every leadership meeting I attended there was a conference room full of people, with me as the only person dialed in on video. I learned how difficult it is to be the only person not in the room.

Even though my title was manager, my stance was as a coach. I had no desire to guide work or be directive. I wanted to help the teams learn to hold *themselves* accountable. Much as David Marquet stopped giving orders [Marquet], I stopped giving explicit direction. I would partner with the teams to discuss and analyze a tough problem, but all of the ideas came from them. My goal wasn't to create the best solution for today's problems. My goal was to create the best *team* who could solve the problems of today and the problems of the future.

This group had been working in an agile way for a while, but some things weren't working smoothly. There were conflicts between architecture and senior developers on how to design the new system. Maintenance and

production support were hard to handle. QA never felt they had enough time or a large enough voice. Releases only happened once or twice a year. And 6 teams spread across 3 offices and 2 continents had coordination challenges.

They didn't need another development manager. They needed a coach. And without even knowing to describe it that way, I became a coach. I worked with the architects to design a process and working agreements on how architectural designs would be created, proposed, debated and selected. We analyzed the flow of production support work, and optimized it to remove waste and decrease cycle time. QA shifted left and provided more input earlier in the story creation and refinement phases. We had a scrum-of-scrums several times per week to raise and address blockers. And through an intense period of continuous improvement, we began producing a production release every sprint.

I helped the group through various challenges while using a coaching stance. While I did take ownership of some parts of the improvement activities, most of the work was done by the teams and not by me. Without even knowing it, I was developing my coaching muscles.

Over this same period, I felt I needed to spend more time with people who 'got' agile. I joined a few meetup groups for agilists. Emotional intelligence was the topic of the first meetup I attended. I enjoyed the speaker and had some additional thoughts I wanted to discuss, so I reached out over email and we decided to meet for coffee. That was my first opportunity to spend time with an agile coach and talk shop. They were very interested in my journey and wanted to help me along the way, so we entered into an informal coaching/mentoring relationship.

Over the next few months, I learned more about what it is to be a coach and the different stances you use as you coach. This was years before I learned the IC Agile coaching competency model [ICP], but we discussed aspects of teaching, mentoring, coaching and facilitating. I was learning and growing and attending every meetup I could to learn more and meet new people.

A turning point on my journey to being a coach came in the form of a tragedy later that year. A friend and former co-worker died in an automobile accident on the way home from work. Their death was tragic for their family, friends and co-workers. As I processed my own grief, a thought came to my mind. They died on the way home from a job they loved. How much greater would the tragedy be if they'd died on the way home from a job they hated? This tragedy motivated me to speed up my journey of becoming a coach.

As my conversations with my mentor took on greater urgency, the discussions shifted from exploratory and theoretical to focused and tactical. The question wasn't 'if' I'd become a coach, it was 'when.' We discussed specific actions I could take to move me farther down the path.

I became more active in the meetup groups. I began hosting meetups at my company's offices. I gave a five-minute lightning talk at the end-of-year meetup. It was the first time I'd ever given a talk on a topic that wasn't directly related to my lived experience. I was nervous but the talk was well received and it gave me confidence that I could share ideas and concepts that weren't specifically related to things I'd done myself.

I came to appreciate that the work I was doing was coaching. But, because 'coach' wasn't in my job title, I had difficulty thinking of myself as a coach. Nobody had given me permission to be a coach. Nobody I was working with thought of me as a coach.

With the support of my mentor, I realized that I would have to self-recognize myself as a coach before anyone else would be able to see it. My self-perception as a coach was more important than what other people thought. I was nervous to describe myself as a coach, but I knew that if I wanted my next job to be 'coach', I'd have to celebrate my successes as due to coaching, not management. I changed my 'headline' on LinkedIn to, "Agile Coach in Dev Manager's Clothing." I felt it highlighted what I really did without trying to represent myself as something I wasn't.

I connected with people I knew within the agile space. Those people connected me with other people, and I began having lunches with experienced coaches. I also recognized the need for external credibility as a coach. I'd been working in an agile way since 2007 and leading with a coaching stance for several years at that point but I had no formal credentials. I took the test and got my PSM I. I got certified as ICP-ATF and ICP-ACC. I applied to and was accepted to speak at TriAgile in 2016 on the topic of engaging developers in an agile transformation. While doing all of that, I never told anyone at work what I was doing. As far as they were concerned, I was just a development manager who happened to be helping them get better working in an agile way. I knew what I had to do for my own growth and development, and I didn't need permission from anyone to do it.

Nine years after I started working on an agile team, and 6 years since I started walking the path toward coaching without even knowing it, I stopped being a development manager. I took new job and finally had my first formal role with 'coach' in the title.

4. WHAT'S NEXT

I've learned many lessons since becoming an agile coach and had many supportive teachers, coaches, mentors, managers, peers and friends. My understanding of what it is to be a good coach evolves and matures every year. I've continued to develop my skills as a speaker by speaking at regional and national conferences. In recent years I've invested in becoming a better facilitator and that has resulted in the opportunity to facilitate crucial conversations for executives. I recently became a certified executive coach and I'm on a path to get my International Coach Federation – Associate Certified Coach credential.

I began my agile coach journey focusing on the 'agile' part. What I've learned over time is that I'm able to make more impact the more I focus on and invest in the 'coach' part. I've discovered the power of simply being fully present for another person and committed to their success with no agenda of my own. My recipe for success as a coach is to help the person in front of me uncover their biggest challenge or opportunity and then to help them move forward.

On this journey, I redefined what success means to me. When I initially defined success as comfort and stability, choosing to challenge myself every few years didn't seem to make sense. But when I redefined success as lifelong growth and learning, I realized you can't just keep doing the things you already know how to do well. You must build on what you're good at to get you to the next place where you can challenge yourself to grow in new ways.

The act of walking is nothing more than putting yourself out of balance to create forward momentum, taking a step to regain your balance, and then repeating until you arrive at your destination. After 20 years I've accepted that the journey of learning itself is my destination. As long as I'm on that path, I'm exactly where I need to be.

5. ACKNOWLEDGEMENTS

Through this journey I've been supported and influenced by many wonderful people. Thank you to my wife Heather Dye Frink and daughters Isabel Frink and Eleanor Frink, who have supported me as I've evolved, especially when that evolution means I'm always trying something to challenge myself... including trying to cover a 20-year journey in 8 pages. I've been fortunate to work with many amazing people who have helped me on this journey. Though there are too many people to thank here, I'd like to recognize: Daryl Broddle, Mary Thorn, Tia DeMaria, Kristen Somers, Sheri Hester, James Elmore, Mark Greene and Carey Rogers. Thank you to Woody Catoe, who helped me find direction when I needed it. And a special thanks to Ronnie Angerer, Art Pittman and Frank Forte. Without your mentorship, guidance and encouragement I would not be the coach I am today. Thank you for nudging me forward and seeing something in me I wasn't yet able to see for myself. Finally, thank you to Nate Ashford for your thoughtful questions, suggestions and edits. I couldn't have created this experience report without you!

REFERENCES

- [Fried] Jason Fried, David Heinemeier Hansson, 2006. *Getting Real: The Smarter, Faster, Easier Way to Build a Successful Web Application* [ONLINE] <https://basecamp.com/books/getting-real>
- [Heath2007] Chip & Dan Heath, 2007. *Made to Stick: Why Some Ideas Survive and Others Die*. Random House
- [Heath2010] Chip & Dan Heath, 2010. *Switch: How to Change Things When Change Is Hard*. Crown Business
- [Auz] Michael Auzenne, Mark Horstman. [ONLINE] <https://www.manager-tools.com/>
- [Frink2018] David Frink. 2018. Blameless Postmortem [ONLINE] <https://dfrink.com/blameless-postmortem/>
- [Frink2014] David Frink. 2014. Agile Home Improvement [ONLINE] <https://dfrink.com/agile-home-improvement/>
- [Marquet] David Marquet. 2014. What is leadership? – With David Marquet [ONLINE] https://www.youtube.com/watch?v=pYKH2uSax8U&ab_channel=LeadershipNudgeswithDavidMarquet
- [ICP] Agile Coaching Institute [ONLINE] <https://agilecoachinginstitute.com/agile-coaching-resources/>