Patterns of Agile Journeys

George Dinwiddie, Susan DiFabio, Oluf Nissen,
Rich Valde and Dan Neumann

Patterns of Agile Journeys

George Dinwiddie, Susan DiFabio, Oluf Nissen, Rich Valde and Dan Neumann

This book is for sale at http://leanpub.com/agilejourneys

This version was published on 2016-03-13

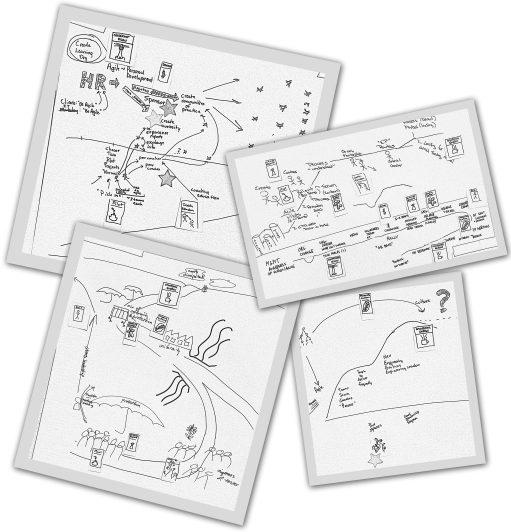
This is a Leanpub book. Leanpub empowers authors and publishers with the Lean Publishing process. Lean Publishing is the act of publishing an in-progress ebook using lightweight tools and many iterations to get reader feedback, pivot until you have the right book and build traction once you do.

Contents

Introduction

This book got its impetus from an Open Space session facilitated by Susan DiFabio at the Retrospective Facilitators Gathering 2015. We drew maps illustrating the journeys of various organizations we'd known as they underwent "going Agile".

Agile Journeys entail changes that are varied and extensive. Each journey was unique and was shaped by the people involved and the context in which they operate. At the same time, there were similarities that jumped out and grabbed our attention when we looked at more than one transition. As we recognized these, we gave them names and decorated our journey maps with sticky notes bearing hand-drawn icons marking where we recognized them. George Dinwiddie suggested we collect these as patterns and publish them for others.



This book contains some Agile journey patterns the authors have encountered. We share these to help you recognize situations you may find yourself in on your own journey. Remember, you are not alone. Use the tips in this book to reinforce or counteract the patterns you see. Notice how they mesh and intertwine into a complex whole. To make the patterns easier to reference, we gave each pattern a name. We hope that the names will help us, and you, talk more clearly about our Agile Journeys, and understand each other more easily.

This catalog of patterns is by no means complete. Undoubtedly we will discover more patterns as we continue to observe and discuss Agile transitions. You will, too, we hope. May the collection continue to grow for as long as these patterns are helpful.

How this book is organized

While writing this book, we noticed that some patterns seemed to relate to phases of a journey, so we grouped them into the following two categories: “In the Beginning” and “Down the Road.”

Some patterns seemed to fit better into a “cautionary tale” theme, so you’ll find these in the “Caution” section.

We highlighted the pattern names in **bold** and linked them for easy navigation and cross-reference.

Each pattern consists of a name, an “essence summary” sentence, a description, and example cases. Some patterns also have comments on reinforcing or counteracting the pattern and potential complications.

About this sample

This is a sample version of the book, and does not contain all of the content of the full version. This sample is intended to give you a small taste, so you can decide if you would like to purchase the full meal.

Find this book on Leanpub at https://leanpub.com/agilejourneys[1].

Acknowledgement

We thank Beth Stauter for editing the final manuscript before publishing. She helped us simplify language, clarify thoughts, and produce a consistent voice throughout the book.

Cover

Cover photo by Jay Mantri[2] released into the public domain under Creative Commons CC0 1.0 Universal (CC0 1.0)[3].

[1] https://leanpub.com/agilejourneys
[2] http://jaymantri.com/post/99567464559/download
[3] http://creativecommons.org/publicdomain/zero/1.0/

In the Beginning

In this section you'll find patterns that are most applicable at the beginning of an Agile journey. Read through these patterns to get a sense of what has helped other organizations start out well.

Leadership Vision

> Someone with influence and spending authority has a vision of Agile as a better way of working than their current approach. They articulate that vision up and down the hierarchy and act to implement that vision.

Description

I suppose it's possible for a major change in the way you work to happen spontaneously, with all of the participants contributing equally. That's not the way it usually happens, though. Usually someone has a vision of a better world than the one where they currently live. That person then articulates their vision to others in their world. It helps if that person already has influence in the organization. The sweet spot for this pattern is a manager with sufficient positional power to implement their choices, but also well regarded by peers and those lower in the power hierarchy such that others will willingly follow.

This well-regarded leader concludes that an Agile approach would be advantageous. Having influence with those in power, as well as discretionary spending authority, this leader is able to initiate a **Pilot**, usually with the aide of **Outside Experts** who can help make the pilot a success. This leader often excels at **Translating Into**

Traditional Terms to describe the work within the **Pilot** to higher managers and some peers. This ability to translate helps to create a **Protective Bubble** in which teams can experiment with new patterns of work and thrive without undue attention and meddling.

At least as important as managing upwards, this leader can articulate a vision for the future. The leader entice people to willingly suspend disbelief and give Agile development an **Honest Attempt**, in both action and mindset.

Agile journeys involve change, and change is often difficult for all involved.
Having a leader with vision shining a light on a clearly-articulated positive future can be instrumental in easing the difficulty and paving the way for success.

Example Case

In an organization of a little more than 100 people, a new software development director was hired after the previous director left for a new opportunity. The organization had looked into adopting Agile methods before, but a management re-organization had put a temporary stop to the effort. The new director had previous experience of good use of Agile development methods and knew that they could be useful in this environment as well. The new director worked with the people in the organization for a while to learn what was not going well. At one point the director published an email memo, outlining the highlights of the vision they had for the organization. The director influenced the organization to set aside money for training and bringing in **Outside Experts** to get the effort started. They also participated in regular transformation team meetings to stay involved with the efforts that were in progress.

The vision outlined in the email memo helped energize people, and the sponsorship for training and bringing in outside experts resulted in a good start for the transformation.

Honest Attempt

> Individuals, teams, and leadership need to be present, participative, open, honest, and embracing of failure; leadership needs to make it safe to do the above.

Description

A successful adoption of Agile needs to work at many levels. Individuals need to commit to their new roles within the process and try not to backslide into past patterns. Teams need to commit to delivering a working product and striving to continuously improve. The organization needs to commit to giving teams the support and autonomy they need to be successful.

In Agile, we want people to make an **Honest Attempt** to be present and to participate in new processes. Initially, some people may fight the process or look for other opportunities. They may think that moving to Agile is a passing fad or think that they do not need to change. Instead, they need to be open, honest and committed to the their team.

The individuals in the team need to trust each other. It's alright for people to fail - in fact we should embrace failure as a way to **Make Time To Learn**. People need to make themselves vulnerable to their teammates - admit their failures and admit when they don't know

the answers. Through their failures, individuals and teams can learn and grow. In a healthy organization, leaders and coaches model this behavior by openly admitting their failures and sharing how they learn from these failures.

Teams need to make an **Honest Attempt** at continuously delivering a working product and at continuously improving. Both the product and the change will happen incrementally. Teams need to try something new, evaluate and discuss the results, and choose the next thing to try. When teams don't make an **Honest Attempt** at delivering and improving incrementally, they will stagnate. Leaders and coaches can model this behavior by openly seeking honest feedback from their reports and openly sharing how they will work together to improve their behavior based on that feedback.

The organization also needs to make an **Honest Attempt** to commit to the process. The **Leadership Vision** needs to include a plan to **Shape the Path** - change the environment around the teams. Otherwise, it will be difficult, if not impossible, for teams to change. Teams can easily backslide into old habits. The organization needs to change in order to enable teams to maintain their successes.

When individuals, teams, and the leadership make an **Honest Attempt** to make Agile work, then the organization has the potential for delivering high value and meaningful work. Don't try a half-assed attempt at Agile, incorrectly label it Agile, and pronounce that it isn't working. Without making an **Honest Attempt** the organization will likely stagnate - individuals and leadership will become frustrated with their inability to improve.

Reinforcing this Pattern

Leaders and managers can help by making it safe for people to fail. Do not punish people's failures or make an arduous process to discover the reason for failures. Leaders should **Make Time To**

Learn and bring in **Outside Experts** as part of their commitment to making the process work.

Counter-Patterns

The obvious counter-pattern is one of resistance and rejection. We've heard many people in many organizations say "that won't work here," or even "that's impossible." These naysayers are generally people that you don't want in your **Pilot** program, lest it fail for lack of trying. There should certainly be other places in the organization where these people can productively contribute, and they may come around in time.

Perhaps worse are the people who say they agree and put on a face of compliance, but passive-aggressively undermine the effort. Perhaps they only half-heartedly attempt a new way of working, or attempt it only when they think someone is watching. Perhaps they spread rumors through the rest of the organization, bad-mouthing the Agile journey the organization has undertaken. These people can be a bit harder to detect and a bit harder to convince than the ones who reject the notion outright.

Example Case

The leader of a large organization decided to move to Agile and hired two directors to lead the divisions within her organization. The first director knew the Agile concepts well and openly spoke about his devotion to the process. However, his actions told a different story. He hired traditional project managers to manage the process. He had his functional managers and project managers plan future work without input from the teams, yet he made the teams responsible to outside commitments. Over time, management became frustrated with the continued late delivery. Team members

became frustrated with the lack of autonomy and looked for new opportunities.

The second director openly questioned the value of Agile but remained open-minded. He made an **Honest Attempt** at making the process work. He gave teams true autonomy to decide what they could deliver. He changed the reporting structure for scrum masters, so they could freely push back on product desires. An energy began to grow within the organization and teams swarmed around project deliveries to ensure the customers were delighted.

Down the Road

In this section you'll find patterns and examples of things people have found helpful as they continued their initial journeys and started approaching sustained change. Read these patterns to prepare yourself for what you may need to do in the long term to help your Agile journey succeed.

Shape the Path

> Make changes to physical and structural elements of the environment in such a way as to remove barriers and tacitly enable Agile ways of working.

Description

The **Shape the Path** pattern is the act of making changes to physical and structural elements of the environment in such a way as to remove barriers and tacitly enable Agile ways of working. This pattern takes its name from the work of Chip and Dan Heath in their book entitled *Switch: How to Change Things When Change Is Hard.* [Heath2010]

When the environment looks the same and the organization feels the same, it is easy for employees to view the announcement of a corporate Agile journey as yet another methodology fad that will come and go.

They may use the new terminology but view that terminology as simply new words for the same old concepts. E.g. they may use "backlog" as the new term for requirements, "scrum meeting" as the new term for status meeting, "sprint plan" as the new term for project plan, and "team member" as the new term for resource allocation.

They may go through the motions of Scrum or Kanban ceremonies, but how they interact with teammates and how they accomplish work may not actually change.

So… what can break these old habits of thought and action? **Shaping the Path** – changing the physical environment and/or organizational structure.

Shaping the Path may take the form of environmental changes such as:

- replacing individual cubicle spaces with team spaces
- providing pairing stations

and/or organizational changes such as:

- realigning reporting structures to reduce competition across functions and encourage shared, cross-functional goals
- revising HR policies to place greater focus on employee collaboration and less focus on employee competition

Shaping the Path works on two levels.

First, it sends a powerful message something has fundamentally changed and the organization is serious about the change. The environment no longer looks the same when one walks into work in the morning. The location of one's desk may have changed. One may report to a different person. It is *not* business as usual!

Second, **Shaping the Path** enables the new way of working. Gathering for a daily scrum meeting is much less troublesome when teammates are already physically together. Learning to pair program has one less obstacle when one has a shiny new ergonomic pairing station. Shifting one's sense of identity from a functional specialist to a T-shaped team member is easier when one's reporting manager is focused on team outcomes instead of specialty skills. A

T-shaped person is one who is still specialized in one area but has helpful familiarity in others.

Shaping the Path can be instrumental in moving a change initiative from a state of fear, uncertainty, and doubt to a state where the new normal can be seen as a beacon of light on the horizon.

Example Case

A large traditional company embarked on an Agile journey. They planned it out like a project. They had executive **Sponsors**. They brought in **Outside Experts**. They scheduled the training, the launch of **Pilot** teams, the feedback gathering, and the launch of all remaining teams. The project was going according to plan. Yet many employees remained unconvinced that it was a good journey or that it would last. Agile ceremonies still felt like unnecessary overhead as people were seen traipsing from their cubicles to daily stand-up meetings in cramped "scrum rooms" away from their desks. There was still talk in the hallways that this was the "methodology of the year", that it "had its problems, but no need to worry because there would be another methodology next year."

Then the organization decided to replace cubicles with team spaces a.k.a. Agile pods. They created pilot pods for two teams and used feedback from those teams to finalize the design of a full build-out of Agile pods for all software development teams.

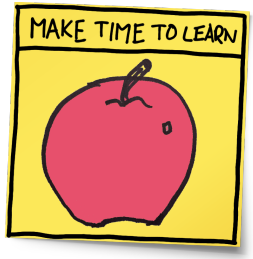
The language in the hallways changed. There was no more talk that this was just a fad. Many more employees began to see promise in team-based work. Stand-up meetings were visible in the pod spaces as you walked by throughout the day, but participants no longer looked like they were being dragged off to some horrendous ritual. They looked physically energized and engaged. An office that had previously been full of heads-down individuals in cubes was now abuzz with people collaborating around the whiteboards and extra-large monitors of the new pod spaces.

It wasn't perfect, of course. There were complaints of losing private space and increased noise levels. Although pods facilitated intra-team communication, they, in some ways, isolated teams from each other. But regardless of the pros and cons, there was no denying that things had changed. Agile was there to stay! Team members began figuring out how to make the best use of their new spaces. Pods facilitated the process of thinking and behaving differently and helped further the Agile journey.

Make Time to Learn

> An Agile transition requires learning, and learning takes time and practice.

Description



The word "transition" makes it sound like you can change the way your organization works in a smooth and rapid manner. In truth, change can be quite chaotic. It's not that everyone knows how to work in this new fashion and has just chosen not to do so up to now. Learning new ways of working takes time. Agile transitions are not accomplished by merely providing some training classes. Training is often a good start, but real learning requires absorbing the information and putting it into operation. It requires employing that information to guide new behavior. And it requires practice. People must practice the new behaviors until they are natural and used in times of crisis or time pressure. Until then, they need to have time to focus on learning to do them well.

An effective adoption of Agile requires learning new skills and observing how well those skills are working. This effort, of course, distracts people a bit from the work at hand. The distraction caused by learning subtracts a bit of capacity from the primary work. If

the learning is successful, then this distraction is temporary. If the learned skills are beneficial, then it's expected that the primary work will benefit enough to make up for the temporary loss of capacity. The goal is to increase our effectiveness and capacity for the future.

This temporary loss of productive capacity while learning doesn't have to be a big drain on current productivity. Be careful to take on smaller and/or fewer learning objectives at a time. Smaller changes require less learning and less shifted focus. By ordering the changes wisely, the early return from initial lessons helps make room for later changes. Select changes that are easier to accommodate and provide noticeable benefit to achieve progress quickly, minimizing disruption. Early success promotes confidence in the team's ability to learn as a group. Tackle more difficult topics based on the breathing space provided by earlier learning.

There are many practices to encourage learning.

- Start a book club or study group to investigate ideas and practices from others.
- Hold periodic "Coding Dojos" or "Programming Katas" to gain shared experience in new software development practices.
- Build Communities of Practice to share ideas and knowledge between people with similar interests.
- Organize "Lunch and Learn" programs for informal sharing.
- Explicitly make time for intentional practice, such as taking Friday afternoons to explore new areas.
- Hold an annual conference, perhaps using Open Space Technology, within the organization to share information and encourage learning.
- Perhaps the easiest of all, ask people regularly what they have learned recently. Reward people for taking a risk, even if they fail. Demonstrate by your actions that learning is valued.

On a successful Agile journey, the investment in learning never ends. Much of Agile is about taking action, noticing the effects, and then adjusting toward the desired results. Being Agile means being a learning organization. This is a high standard and aspiring to it is also a form of learning.

Example Case

A team that was about to embark on re-creating a piece of software for a new platform decided that they needed to build this new version on a better foundation than the previous version. To do that, they decided to use a particular design pattern that matched the platform's preferred way of writing applications. They decided that learning this new pattern would have the most impact if they all got together for a few hours each afternoon to attend an online video class on the pattern. Everyone watched the video together, and people were able to discuss questions and concerns as they came up during each session. This experience was so positive for the team that they continued getting together most afternoons to create the new application foundation together, in the style of *Mob Programming*. As a result, they were able to produce a solid, high quality foundation for the new software in just a few months.

Caution

In this section you'll find some patterns that have the potential to derail your journey into an Agile way of thinking and behaving. Read through these patterns to be prepared for roadblocks you may encounter and to see how you might prepare yourself to overcome them.

Cookie Cutter

> Cookie Cutter approaches (i.e. all teams do it the same) are well meaning attempts at efficiency, but they undermine motivation and long-term effectiveness.

Description

Rich recalls: "During each of my childhood holiday seasons, my mother pulled her grandmother's sugar cookie recipe from her recipe box and mixed a batch of cookie dough. A couple of days later, my sisters and I gathered around the kitchen table to decorate the cookies my mother cut from the dough using our favorite holiday cookie cutters. The cookie cutters enabled her to efficiently and effectively mass-produce many cookies of the same shape, which we could turn into childhood masterpieces that could be shared with family and friends."

In the Agile world, a cookie cutter approach is a standardized practice or approach that is applied repeatedly to many teams with the hope of mass-producing results across a large organization. For instance, a company **Pilots** a new approach or process with a handful of teams. Much time is spent with these pilot teams to enable them to reach a level of success using the new process. After achieving success, the organization attempts to replicate the results

by rolling out the identical process to many teams. One way to revise this concept is by the company identifying best practices that successful teams inside or outside the company use, then attempting to get other teams to adopt these processes. This is done with the hope of efficiently and effectively improving the performance of the teams.

In Agile, the goal is to build self-organized teams comprised of motivated people who will work together to build working software that meets ever-changing customer requirements. Once the team has been challenged with *what* they need to develop, they need to be given the freedom to adjust the process in ways that enables them to work effectively. Since the individuals on these teams are different and their work domains are different, the personality and needs of each team will be different - a process or practice used by one team may not be effective for another team. By efficiently applying the same practice to all teams, the organization is not treating them as self-organized entities. Instead, the organization is treating them as interchangeable entities that can be used to mass-produce results. If the organization wants truly autonomous teams that can creatively solve problems, it needs to recognize them as individual collaborative units. Teams should share information and learn from each other. Teams can also learn from others in the industry. But this should be a process of discovery for each team. Look for ways to make the successes of teams visible and enable teams to recognize, experiment and adapt practices for themselves.

In his book *Drive* [Pink2010], Daniel Pink shows that people are motivated by autonomy, mastery and purpose. This suggests that people need to be given room to choose aspects of the who, what, where and when they work. They need an environment that encourages and supports the desire to experiment and improve. Lastly, they need to know the "why" of their work - how their work will make a difference. Most engineers have a strong desire to learn and to apply their knowledge to improve the world around them. Cookie cutter approaches are often applied by well meaning

individuals because they believe those approaches will be efficient. But usually people will feel a lack of choice in the process, leaving them unmotivated and making the process ineffective. Long term efficiency and effectiveness can be gained through allowing slack in the immediate short-term efficiency.

Counteracting this Pattern

We need to focus on setting teams up for success by clearly stating *what* they need to develop and providing the context of why they need to develop it. It will then be the team's responsibility to determine *how* the work gets done within the boundaries of an Agile framework. Coaches and managers should develop an awareness of the bright spots where teams are working effectively and make those successes visible. They can also encourage teams to experiment and embrace short term failures.

Example Cases

Case 1

A large traditional development organization faced difficulty getting products into the market. They faced long cycle times of analysis, design, development and testing. Additionally, political boundaries existed between the phases that made it difficult to work cooperatively. The leader of the organization challenged the leadership team to reduce the amount of time that it took to deliver new solutions. Agile development was presented as a potential solution, and pilot teams were identified to experiment with the process. After the successful pilot phase, the leadership team decided to roll out the process to the entire development organization. A process was designed to train teams and move them into the process. The rollout was managed using a traditional management approach -

looking to make it as fast and efficient as possible by applying the same process to each team. Little consideration was given to the variance in the types of teams within the organization. Some teams were essentially "startups" that could release updates to the market more quickly than others. Some teams, which were working on systems that required a high degree of quality, had less room for experimentation with what was built. Some teams were component teams within very large initiatives and required a different level of coordination with other teams. Most teams adopted the Scrum ceremonies, but little time was given to enable teams discover the process that would be most effective for their situation. The result was that many teams struggled to reach true autonomy and to develop the process of continuous improvement that enables teams to thrive.

Case 2

An organization ordered rolling whiteboards for each team to track their progress through the Sprint. They had them pre-painted for the Sprint Kanban, with a predetermined set of columns. Ironically, the pilot teams had redefined the columns they used a number of times before settling down on a consistent set, and continued to make small short-term adjustments for special conditions. The mass-produced teams with the painted whiteboards found themselves unable to represent the work, as they saw it, on their Kanban board. Being unable to change it, they had to keep the real status of the work in their heads rather in the visual management tool. This resulted in confusion within the team, and slow or incorrect status being reported on the board.

Dissipating Energy

> Agile journeys often start with a lot of energy, but that energy often dissipates over time for a variety of reasons.

Description

Kicking off an Agile journey is an exciting endeavor! It may get started by a visionary leader (see **Leadership Vision**) or grow from a grassroots movement that reaches critical mass. The leader (or a small team of leaders) may inject momentum by becoming **Sponsors**. They may create or identify Agile teams to conduct a **Pilot**, helped along by **Outside Experts**. Perhaps the organization starts a program to **Create Internal Coaches** and feels that it is well on its way.

But then something odd happens. People become complacent. Sponsorship wanes, perhaps following the **Exit** of the visionary leader. People become lax on writing high quality code. Maybe the budget gets cut, so team members can't travel anymore to meet with people who are essential to their continued success. Or perhaps the **Pre-Existing Culture** is finally catching on to what the organization is trying to do and adding friction to the progress, slowing things down despite best efforts to keep things moving. Congratulations – this is the **Dissipating Energy** pattern.

We've seen this pattern multiple times. Most change efforts start off with lots of enthusiasm, good intentions, and plenty of initial momentum. Over time, as with a cup of hot tea or coffee, this initial energy will naturally dissipate. People will have taken some training on the mechanics of Agile and perhaps even on the necessary technical software development practices. Over time, though, the mechanics stay just that - mechanics. The development practices are hard and deadlines are looming every two or three weeks now. It becomes harder and harder to stay disciplined about planning the work and estimating things, so much of that is let go.

The people who were leading and coordinating things in the beginning don't feel they are making any difference, and so they slowly start withdrawing their efforts. Mindsets among developers and management ranks haven't really changed. People may say "we're Agile now" after they've taken the initial training, but they really aren't. They're still thinking in Waterfall frames. What was once a promise of a better future becomes just another failed change effort.

In some organizations, an Agile transition is seen as a "project" to be executed with a finite budget and a defined end point. Once the end point has been passed, the funding to keep the effort going is withdrawn and energy around it goes away.

Counteracting this pattern

Dissipating energy is natural and to some extent necessary. No organization can sustain the energized activity that often characterizes the initial phase of an Agile journey. Leaders need to anticipate this, allow for periods of rest and integration, and have measures in place to pull people up after the lull.

Decide to form a dedicated team of Agile leaders. Find people who have passion for the hard work of changing habits and minds and who are psychologically equipped to have the right kind of conversations, even the difficult ones. Realize that becoming

Agile doesn't end after finishing a few classes and running two handfuls of sprints. Commit money and time to ongoing training and travel. Build communities of practice and support within your organization.

Injecting a lot of energy at the start may set things off on a good footing, but re-injecting energy periodically down the road is critical for long term success. Inject energy from time to time by gathering everyone in the same room to reconnect and strengthen their personal bonds.

Make sure everyone works at a sustainable pace, especially the people who are most fully engaged in helping change along. This will conserve the energy for the long haul and direct it in a productive manner.

Example Case

An organization of about 120 people (including about 70 software developers) welcomed a new director-level manager to its development group. In conversations with people, the manager found that things were not going as well as he or she would like them to. The manager kicked off an effort to identify problem areas, and that effort eventually turned into the formation of an Agile transformation team. This team met regularly, using Scrum to manage their own work, including sprint planning and holding retrospectives. The team was co-located with the manager and worked through many activities that an Agile journey involves, such as curating training curriculum, finding outside experts to help, bringing in coaches for specific activities (especially at the team-coordination level), and producing a periodic newsletter. Money was invested in travel to compensate for geographical dispersion of people in the organization. People were given time for training.

After a while, the manager accepted a different opportunity in the larger organization and took a few of the team members along.

The team composition changed, with a new program manager stepping in. The remaining members were geographically separated, which made collaboration harder. Slowly, the money for travel was cut, and the geographic dispersion of the software development teams made collaboration harder and harder. The remaining managers started considering the Agile transformation completed and stopped meeting regularly to see what needed attention and improvement. The quality of the software that the teams produced began to deteriorate, a theme that was reflected in employee satisfaction surveys. The energy around the transformation had dissipated, exacerbated by concerns about business pressures, money, and lack of true buy-in by the managers still left in the organization.

Bibliography

Chip and Dan Heath, *Switch: How to Change Things When Change Is Hard*

[Heath2010]

Chip and Dan use compelling, real-world stories to illustrate three fundamental aspects of facilitating a change journey - the emotional, the logical, and the environment. The **Shape the Path** pattern name is drawn directly from their work. Many of the stories show how small changes can have dramatic positive impacts.

Daniel Pink, *Drive*

[Pink2010]

In *Drive* Daniel Pink describes recent findings about what motivates people to be engaged in a knowledge-worker setting. Exposure to this material can deepen your appreciation for the Agile Manifesto principles and values and help you understand in greater depth why they make sense for software development teams.

Simon Sinek, *Start with Why*

[Sinek2009]

Simon Sinek goes into detail about how to build strong cultures. His thesis is that companies with a clear sense of "Why" outperform others significantly over time. He also warns about what might happen if the "Why" isn't made clear and firmly embedded in a company's institutional knowledge and "folklore." Organizations can use the information in this book to help guard against the effects of the **Exit** pattern.

Tom DeMarco, *Slack - Getting Past Burnout, Busywork, and the Myth of Total Efficiency*

[DeMarco2001]

For years I thought *Slack* was merely about leaving space in your time schedule. I should have known better, having read other Tom DeMarco books. This book is chock full of ideas about being more effective rather than more efficient, about the limits of efficiency, and the typical problems that inhibit effectiveness.

Lyssa Adkins, *Coaching Agile Teams*

[Adkins2010]

Adkins provides a comprehensive guide to what it means to coach Agile software development teams. The book covers self-management, teaching, mentoring, coaching and many practical tools that coaches can use to help the teams they work with become high-performing.

About the Authors

Rich Valde

My Agile journey began at HERE - formerly Navteq. Before our Agile adoption, Navteq used a traditional gated development process. As a software developer, I experienced first-hand the pain of working in a siloed organization. Since the adoption, I have participated in a Coach mentorship program and have met others in the Agile community through conferences such as the Retrospective Facilitators Gathering. Before working in software, I worked as a singing waiter, musician and a voice coach for a small youth theater. My interest in technology and learning was fostered during my pursuit of a Master's degree in Instructional Technology.

Oluf Nissen

I heard about Agile software development while working at a small start-up company in Denmark, developing communication software for industrial process control and getting frustrated by my experiences with that. When I left the start-up to join another one, I did so purposely in order to experience Extreme Programming first-hand. After landing at a Fortune 50 company in Silicon Valley, I joined in the formation of an Agile Special Interest Group there in 2004. I've been a Scrum Master with training from Ken Schwaber and Jeff McKenna since 2007. I've worked on a handful of different

teams in various geographic configurations, and in 2013-2014 I was involved in an implementation of SAFe as an internal consultant and part-time Agile coach.

George Dinwiddie

Back in 1999 and 2000, I was researching Design Patterns on Ward Cunningham's Wiki, the Portland Pattern Repository. I noticed a lot of discussion, some of it heated, about something called Extreme Programming, but I ignored that as noise. Then one day I used "Extreme Programming" as the punchline for an impromptu joke. "Extreme Programming?" my co-worker asked. "What's that?" This stopped me in my tracks. "I don't know," I replied. "I guess I should find out."

That was the beginning of a never-ending journey for me. I explored Extreme Programming on Ward's Wiki, joined the Extreme Programming Yahoo group, and started experimenting. Mostly it was an approach that I naturally followed when left to my own devices, but there were a few things that seemed totally foreign. Through discussions with others and exploration on my own, I made sense out of it and started encouraging others where I worked to understand it, also. Through that job, and the next, I made some progress in getting bits of it adopted, but never fully. Then I went off on my own as a consultant and have helped numerous organizations, tiny to large, with their own journeys.

Susan DiFabio

I spent many years as a software programmer/analyst/generalist/-consultant and I watched the industry "evolve" from one where it

was expected that I sit and talk (sans methodology) directly with my customer and write code to meet their needs, to one characterized by document sign-offs and Gantt charts and code freezes and blame. Then one day in 2001 I was browsing a book-store and found the book "Planning Extreme Programming" by Kent Beck and Martin Fowler. I bought it, I read it, and I exclaimed out loud, "Oh my gosh, there are other people in the world who believe what I believe about the way things really work on software projects!"

Today I am an independent Agile coach focused on helping teams and organizations find success applying Agile principles. I witnessed first-hand the success of iterative, feedback-driven development as compared with waterfall processes. I also experienced the importance of valuing people in the pursuit of creating work places that are not only productive but also happy, humane, and creative. I am passionate about sharing what I've learned and helping my clients discover the Agile journey that is right for their unique situation.

Dan Neumann

I was not Agile before there was Agile. I was a Computer Science major that hired into a CPA and consulting firm right out of college. We followed a waterfall process. Plans, contracts, and process were paramount. Working software came late, and when push came to shove, manual testing got cut short. That was how I worked for a decade.

I got introduced to Agile as part of a career transition. I served as Scrum Master for both collocated and distributed teams. Collaboration with other Scrum Masters provided the first taste of "coaching." When the corporation announced the closure of our office location, I sought opportunities in other organizations that were also transitioning to, or practicing, Agile methods. That set

me on a course that led to more Agile coaching, filling that role as both a consultant and employee.

Continuous learning and engagement with others in the Agile community has been key to continuing to develop and provide value as a professional.