CHAPTER 4

Common Success Factors

CONTENTS

INTRODUCTION

In analyzing the different stories of how each organization implements Agile methods, it is clear that Agile UX teams tend to fall along a spectrum of success. On one end are the teams that have only "the scent of Agile" on them. These are the people who use the term *Agile* to describe what they do and claim to follow the process, although a close look at their day-to-day efforts might paint a different picture. In examining how these teams operate, it is clear that they do not really engage in the core values of the methodology and often skip many of the recommended steps prescribed by their chosen method. Because these teams are not actually applying Agile methods in concert with adopting the values and principles, it would be hard to see them as successful adopters of Agile. Since they are not necessarily following any particular development process, despite what they might think, they often embody the worst of all the styles that influence their work. It is typical to see on these teams a lack of

Agile User Experience Design. http://dx.doi.org/10.1016/B978-0-12-415953-2.00004-2
Copyright © 2013 Elsevier Inc. All rights reserved.

training to educate individuals about their Agile method and therefore a lack of real understanding of what it means to follow a particular process. It is also common for these teams to have many problems that are symptomatic of poor communication. The teams tend to struggle; and not only are they not effectively using Agile, they have trouble hitting their release dates with a good level of quality and containing the features that they committed to delivering. They are failing not just because they are not practicing Agile but because they are not following any defined process.

At the opposite end are the teams that achieved a kind of organic rhythm with Agile to the point where the process just disappears and they are no longer even aware of the process—they just do it. These teams have not only realized an expertise in practicing Agile, they are also putting out great products quickly. Being able to do this is the purpose of becoming Agile, so it is no surprise when an expert realization of the process results in robust and timely releases. The process has become so transparent to them that, when talking to members of teams that have been able to achieve such a seamless Agile process, it can be hard to pin down the details of the day-to-day efforts that the rest of us are so focused on navigating properly.

When I asked Christina York if her team used a Sprint 0 for planning, she told me that it did so at first but was evolving out of the process, because it did not really allow time for much research. That does not mean that the team is not doing research or planning but that it is *always* doing research and planning and is in a constant state of preparing for the next sprint or release cycle. It does not need a dedicated period of time to do this activity, because it found a way to plan constantly. I like to describe this of state as being effortless, but that is just how it looks to an outsider, as it probably takes a considerable amount of time and effort to get to that place. Since supporting ongoing research efforts is one of the questions that Agile UX teams have the most difficulty figuring out, the fact that her team could do this and make it a natural part of its rhythm is an indication of a very mature practice.

This kind of proficiency is a really great sign that a teams has been tending to the process: identifying what works and what does not and evolving its functioning at an optimal level. Jeff Gothelf describes this maturation process as progressing to the point where there is no process. He often compares it to the big fight in the *Karate Kid*, where our hero is no longer waxing, he is just doing karate without thought or effort. Once the values of Agile and the techniques have become ingrained in the way that the team works, it requires no conscious effort or hours of discussion to know how best to follow the process. It just becomes the way that the team works. Even the process of reflection becomes less of a ceremony and more of a natural conversation that occurs as a matter of course during the sprint cycle. This is an embodiment of the Agile Manifesto's values of giving priority to "individuals and interactions over process and tools." It can be easy to get caught up in all the trappings of Agile methods and lose sight of the real core values. There are so many events to execute: daily scrums, planning poker, retrospectives, and demonstrations. Checking off these activities can provide a sense of accomplishment, not to mention their ability to keep a whole team of people very busy and occupied. The real value of Agile is not in the ability to attend as many meetings as possible or aggressively keep them limited to the recommended time. The intention of these events is to provide the team a way to become highly efficient and collaborative, if it is not in that state already. A team can evolve beyond these methods or at least mature to the point where the process can be done without much conscious thought. When this happens, it is possible to give a seasoned team the ability to operate at the speed appropriate for the members and allow them the freedom to focus on completing the work and not on hitting prescribed deadlines.

A team just beginning its Agile journey should realize that the teams that have achieved this kind of seamlessness did not get there overnight, nor did they get there without some pain and suffering. The ability to transcend process comes from trial and error and, in some cases, years of effort. Achieving a seemingly effortless workflow first needs an exertion of effort to build the team's skills. No one starts out achieving a perfect adoption of Agile in the first sprint, and certainly no one immediately realizes an ideal integration of UX into the release cycle on Day 1. The ability to overcome obstacles and redirect the team's efforts is what ultimately determines the ability to move toward the more successful end of the Agile spectrum. The team's skill in navigating rough patches is almost more important than the success or failure of a single sprint. Learning from failure helps prevent additional issues down the road, if the team sorts out what went wrong; one bad sprint will not compromise the release but repeating the same mistakes will.

Also, many teams do not fit these extremes and fall somewhere along the middle of the spectrum between struggle and enlightenment and land solidly on competency. While I consider the teams that achieved a picture perfect process to be living at the optimal end of the spectrum, it is certainly possible to be a successful team without achieving this advanced expertise. Many teams live happily and indefinitely in the broad space in the middle, where things go well, the work gets done, and there is lots of process, hours of discussion, and plenty of negotiation around that effort. At the end of the day, these teams still release great high-quality software in a reasonably short period of time and with a wonderful user experience. It just might take them a little more thought and effort to get there. You need not feel less successful because your team has not achieved a totally organic practice of Agile. It might not be realistic for your situation, and it may not even be necessary to get to that state. However, there is comfort in knowing that it is possible and informative to see how it is done.

We can learn from all the use cases on both ends of the spectrum to identify what contributes to the success, or lack thereof, for any given team.

PROJECT OVER PROCESS

Not to be confused with the oft quoted "People over Process" from the Agile Manifesto, this speaks to the balance that successful teams strike between process activities, which keep them on track and moving forward, and the production of the software. The process is helpful only as long as it supports the release; and when it is not, a process refinement needs to occur. Agile methodologies provide a framework for a team to function as empowered, self-organizing individuals in support of a common goal. Before you can move beyond the process itself, you have to experience it and learn what things helpi the team be productive and efficient and what things inhibit the delivery of a high-quality product. When you first start looking at Scrum, it might be easy to think that a daily scrum meeting is a waste of time, only to learn, once you have gone through the process, that is actually a great team-building tool and very effective at increasing the communication and transparency of the team. The different methodologies should be seen as a starting point in the effort to create a more highly functioning work environment.

Unfortunately, a myopic focus on details can distract from the real issues and sometimes interfere with the team's ability to work as efficiently as possible. When I asked Archie Miller how long his team's sprint cycles are, he said they

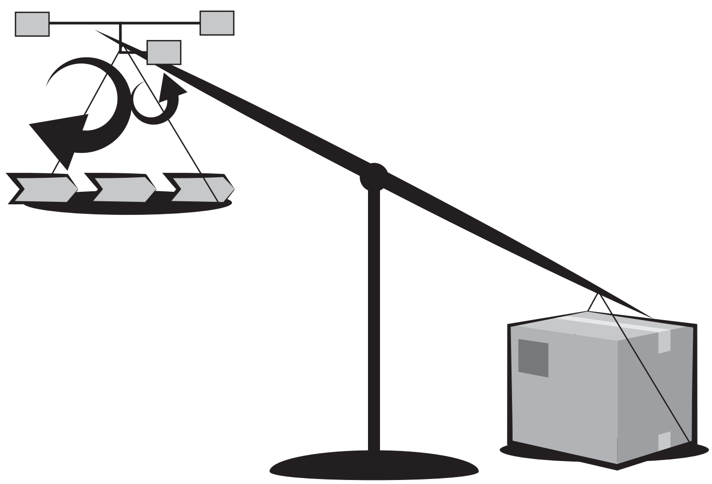

FIGURE 4.1
Project over process.

tend to be a week at minimum once they are out of the discovery phase. He did not have a definite number to give, because the team uses Kanban and choose to allow the work to move at its own natural pace. Because he is a part of a highly functioning Agile environment, he views the timing of the cycle a little bit differently from people who are still working within prescribed sprint cycles, saying, “Releases are like breathing for the team and nobody likes to hold their breath.” If his team were working in an environment where it was on a strict sprint schedule, that could easily interfere with the cadence and become a disruption or even a source of frustration. This is a company that took to Agile like a duck to water, and the corporate culture is such a good fit for Agile values that the people do not require as much scaffolding to support them in their efforts to focus on the project.

It might make more sense to see the more well-defined, prescriptive Agile methods as training wheels, providing a more solid structure and guidance around how to work, as everyone learns to think differently about the business of creating software. If a team has never worked with any type of Agile method before and is not certain of where the challenges will be, it is probably more helpful to use an Agile method like Scrum, which provides a very specific approach with which to work. However, there is no need to force a perceived evolution to a less-defined process, just because there is a certain amount of expertise with Agile. The process is just there to help to produce the work, if it is not getting in the way, then there is no need to “fix” it. If the team looks at a different technique, like Kanban and Lean UX, and sees the potential to help it work more efficiently, then it could be worth making the change. But if the process is not broken, fixing it may not be the right answer. The purpose of the process is to support the production of the software; if that is going well, then what problem would be solved by imposing a change to the process? In the end, the process is not the most important thing; the degree to which the team can function at an efficient and productive manner is.

The different elements of the techniques should be seen as tools and used as such. Since the tools might be new and different, they have to be tried out before they are refined or discarded altogether. If playing planning poker does not yield dividends over the course of several sprints, then the team should assess whether to continue investing its time in doing that. Most teams utilize the standup in some way, but if someone who is straddling multiple projects is already well integrated into the team, it is not uncommon for that person to attend standups only when his presence is necessary and skip the rest. And if the team agrees to this and it works, then this is an appropriate adaption of the methodology. The methods are not sacred, they are just guidelines known to be effective, and the team should not be afraid to question them once it has seen how they work for a given project. If, however, if the infrequent attendance erodes trust or inhibits communication, then the team must find another way.

Perhaps, the split resource needs to take on less work each sprint to allow enough time to attend the meetings and build relationships. It might be worth mentioning that it would be very unusual to completely eliminate the daily standup altogether. Even for teams across multiple time zones, some form of daily communication summarizing progress and identifying problems happens in one form or another. Having a meeting that occurs every day serves the purpose of good hygiene for keeping the team in the habit of interacting with each other and providing transparency into their actions. Many elements of Scrum get tweaked by different teams, but some version of this event should remain a part of the routine.

To look at what it means to value the project over process, we take the example of two imaginary teams at the same company. One team is committed to its daily standups, planning meetings, retrospectives, and daily review of burndown charts. However, the standups routinely run 30 to 45 minutes rather than the recommended 15 minutes. Planning meetings similarly run long, and the team is overcommitted for each sprint. Some team members are often scheduled at 200 percent of their capacity; and while the team reviews these numbers, adjustment to the workload are not made. Retrospectives are awkward and do not contain candid communication or resolution of issues. Not surprisingly, the team routinely fails to complete the scope of work it took on for the sprint.

At the same imaginary company is a second team, which does not conduct planning meetings but allows the individuals to estimate its tasks at a high level only to the degree they need to in order to identify how much work they can fit into the sprint. The team does not track velocity, but it does deliver everything on time and within sprint. Planning and commitment meetings require little time, as the team has a very clear understanding of the work. Daily standups are quick and do not serve as the single point of communication. The team has a sense of where it is in the cycle and what work remains.

The first team feels that it is Agile: It can check the box next to all the steps it has been told are necessary. And, since team members do not feel they are empowered to change, resigned acceptance seems the only answer. But they are not delivering working code within the time to which they commit, and many tasks go uncompleted during each sprint. The fact that so many meetings are running long is an indication of insufficient communication among team members and that they are not seeking out each other outside the scheduled events. A quiet retrospective indicates a level of discomfort among the team members that is inhibiting problem solving. The second team agreed to skip or minimize many of the steps prescribed by Scrum but the level of communication is very high and the members are executing successfully everything to which they commit and deliver the product on time, with high quality, containing the features they originally scoped, and with a great user experience.

If process is the success metric, then it might be tempting to describe the first team as successful. However, it is pretty unlikely that the team members feel that way, even if they cannot actually put into words what exactly is going wrong. If we instead look at the project work, it becomes very clear that the team is struggling. The second team may be skipping a few meetings and techniques, but the work is getting done quickly and with a healthy level of iteration and collaboration. The steps described by any Agile methodology are helpful only if they help the team make progress on the project and supports it to do the work more efficiently. If that is not happening, then the team needs to take stock and decide if its application of a given technique needs to be refined, if the particular tool needs to re-evaluated as to whether it should be in the repertoire, or if the team simply needs to try harder to make it work.

While it is important to make sure that the process is working and the team is not engaging in time-wasting activities or simply failing to accomplish its purpose, equally important is to go through the prescribed process before making modifications. Whether a team has chosen Scrum, Kanban, or something else, follow it as prescribed before discarding elements of it. This is not to say that the team should have a fear of modifying the process, but skipping over pieces of the methodology without actually trying them then having the team discuss its efficacy might mean missing out on applying techniques that may have sounded odd or time consuming but could be very helpful or easy to do. Teams often grumble about the daily standups and the planning meetings, but in the end, both events add tremendous value and they rarely get discarded. The other reason to go through the process before tweaking it is so that the changes can be precise and impactful. It may be that you do not need to eliminate the daily standup, but streamlining them so that some functional areas can have a single representative at the meetings to make it more efficient for teams that straddle multiple projects. To fix a problem, you have to actually know what it is and why you are fixing it, if you want to get to the right solution.

TEAM DYNAMICS

Ideally, the values of the larger organization would be such that moving to Agile would be a process that dovetails seamlessly with Agile principles. Archie Miller describes a carefully curated culture at Snagajob.com that, by design, works to balance the individual with the higher-level goals. Having such an environment makes the transition much less painful, because the core values expressed in the Agile Manifesto are already embedded in the corporate culture and the everyday behavior of the staff. It is even better if the company has made such a commitment to Agile that they invest in facilities to support a cooperative and collaborative environment. Christina York describes the kind of facilities support that many of us can only dream of; where the UX team members actually have two seats. One seat is with their project team, so that they can spend some of their time being colocated, and the other seat is with their fellow UX teammates, so that they can share product knowledge across the UX team as well as function as a cohesive group. This setup also balances the need for privacy by providing phone booths but still recognizes the need for collaboration with its flexible seating assignments for "chickens."

There is no question that being in such a nurturing environment makes it easier to make the move to Agile methods. But, many of us are not as lucky as Archie and Christina, and this does not mean that we cannot achieve a certain amount success with these development processes, despite our organizational cultures. Multiple office spaces are great, a really neat idea, but they are not required. Many teams have managed to engender a solid team environment across continents, so sitting on a different floor than your team is not the end of the world. An Agile friendly organization smooths the way for an Agile adoption, but it is not an absolute necessity, since most of the activities occur at the local project level. The only culture that is critical to the success of a given project is that of the project team. A dysfunctional team in an amazing company can still fail. But, if the team's dynamic is healthy and communicative, it can survive and thrive in the most hostile of corporate atmospheres. And, if the team's dynamic is not where it needs to be, then the team and not the organization can correct this.

Just as it is important to value the outcome of the project over the details of the process, it is crucial to make the healthy functioning of the team a priority. This means breaking down silos and not writing off issues because a problem affects another functional area. If the QE team is floundering because it has a hard time keeping track of the progress of the features and which items are done, this is a sign of a larger communication problem. As a UX person, when you hear an issue like that raised, it might be tempting to write it off as something that the development team and quality team will sort out, as it has no impact on your work. However, the only way that an Agile team can be effective is if it operates at a high level of efficiency, and that requires a cohesive team. If something is not working for one member of the team, then it is not working for the entire team.

This is something about which UX teams need to be especially conscious. Since we are so used to shrinking to fit the time allowed or just making it work, there is a tendency to try to get the work done and suffer in silence if the schedule is too tight or not enough time is planned to incorporate customer feedback into the design process. However, as an Agile team member, it is everyone's responsibility to flag obstacles and work with the scrum master to remove them. If the UX person cannot complete tasks on time, perhaps he or she is overcommitted to the sprint and needs to adjust, take a sprint to do some high-level design, or discuss the requirements more thoroughly with other team members or stakeholders. The problem belongs to the team and can be solved by the team.

This goes beyond just being a good team member, a typical Agile schedule contains little slack; and if one team member is having a challenge, the downstream effects are immediate and often snowball quickly to turn into bigger problems. An issue does not affect just the UX person but the developer who needs to implement the design, the QE person who needs to write a test case, and the documentation person who needs to write the help for it. It is important to recognize any symptoms that might indicate an underlying problem, identify the root cause, and come up with a targeted solution. Delaying the work of one, or all, of these team members can have implications for the overall timeline of the project. Making life better for one team member increases the quality of life for the entire team.

Another challenge that some UX teams struggle with is that this process does not lend itself to what Christina York refers to as "design divas." For the manager of a UX team making this transition, that is an area where team members might require some coaching and support. Some people are used to working on a design and handing off the resulting work of art to the development team for implementation, which may or may not match the design intention. For some UX professionals, this is the most comfortable way to work. It allows them some quiet time to really think through the design problem and work at their own pace to hash out a solution. Also, a certain amount of glory accrues to the person who solves a tough problem with a fabulously elegant solution that is presented in highly polished, pixel-perfect screenshots. You can spend time looking for around for inspiration and craft some gorgeous images that impress everyone. It can be tough to let go of being the hero who saves the day and always comes up with the coolest solution. When a UX team engages in an Agile process, these contributors need to get comfortable with the idea of working collaboratively and not necessarily owning the design. Some rock stars can learn to be part of a band, while others will be more comfortable remaining solo acts; for them, Agile will not be a very good fit.

COMMUNICATION

How do you improve the health of the project and the team? Communication is the grease that makes those wheels turn. The process provides a support for communication, creating opportunities and reasons to communicate and identifying the places where interaction is most important. However, the teams

FIGURE 4.2
Communication.

must engage in these activities with a genuine interest in sharing information and learning about what other teammates are doing. Otherwise, the structure only highlights existing dysfunctions. The process provides only the framework; the team needs to bring the right attitude to bear on the work. The good news is that good communication skills can be taught and nurtured if they do not exist already.

Events around grooming the backlog afford the chance to contribute to and discuss the user stories, which is where the team can exert the most influence over the process. Since these items are constructed to communicate the user viewpoint, any team discussion around them furthers a customer-centric perspective. The team can define acceptance criteria, specify the definition of *done*, and break the user story into specific, actionable tasks. Not only are the user stories structured in such a way that the team becomes habituated to speaking from the perspective of the user, but all the discussion around the user stories create a shared understanding of the customer needs. The work done around stories to turn them into tasks also serves to create a sense of ownership over the work. That is quite an accomplishment for something essentially the equivalent of functional requirements.

Sprint planning activities allow each person to commit to only what he or she can execute and control his or her workload. Not only does it give the team members the authority to choose what they can realistically achieve, but since they made the commitment explicitly, they are responsible for honoring it. Often, accountability can be diffused in traditional processes, and when no one has clear responsibility for something, there is also no real ownership and things tend to falter. The planning sessions also provide an opportunity to discuss a given task with other team members, if clarity is required.

Daily standups are designed to flag obstacles that need to be addressed and let the rest of the team know what each person is working on. Not only does this help catch and resolve problems as they occur, it creates the habit of frequent communication and teaches the team that talking and sharing information does not need to be a time-intensive activity. With its specific format and short time limit, it models the example that a meeting should be productive and efficient and not be held just for the sake of having a meeting. This meeting ensures that everyone knows what each other is doing, and it accomplishes that goal every time. This is a definite improvement over the more traditional meeting, where everyone's eyes glaze over as they listen to endless status reports or repeatedly glance at clock as someone rambles on in an attempt to fill the time. Even though people new to Agile often chafe over this meeting, as they consider it disruptive to the rhythm of their day, most teams end up keeping it in some form or another. It is a cheap investment and serves as good communication hygiene for the team, not unlike brushing your teeth in the morning.

The retrospectives at the end of the sprint are meant be a forum to discuss what went well, identify the opportunities for improvement, and suggest potential solutions. This meeting requires a degree of candor that is not quite as necessary in other discussions; this is the designated time to talk openly about problems that need resolution and work together to solve them. The Agile values and the intention of the meeting give everyone the permission and freedom to raise concerns without being branded as a complainer, and it channels the discussion in a productive direction. To avoid having a gripe session, there should be a habit of presenting a concern and immediately moving the conversation to identifying the root cause and possible actions to correct it. The ability to have a candid retrospective may not happen during the first sprint, but as the team builds trust and confidence in the process, the dialog will become easier to engage in.

These checkpoints should not be the only moments where mindless conversation occurs, they really are meant to set the tone and create habits, so that the team members get accustomed to working with each other. Even for teams experienced in Agile, if the team itself is made up of people who have not worked together previously, then this might still require effort until the team gels. Even when the individuals are used to the practice of Agile, they may not be used to working with each other, and the meetings give them the chance to orient to one another. If team members do not interact with each other outside these meetings, that will show up pretty quickly, as the scheduled meetings will routinely run long and not be very productive. The team members need to be in constant communication and comfortable reaching out to each other when they need something. In-person chats, emails, instant messages (IM), and the occasional formal meeting are great mechanisms for between-meeting interactions. The key is frequency and effectiveness. If you have a teammate who is more comfortable chatting over IM than face to face, then accommodate that and make it the preferred way to work with the person, if it results in more conversation and a better working relationship. Face to face is ideal, preferred, and should be encouraged, but individual styles should be respected.

Also take a look at what team building activities might be helpful. Collaborative design sessions can be a fun activity that brings the team closer together, especially if food is involved. While the team has many chances to connect and bond over work-based events, do not overlook the value of simply having lunch together. Spending a little time to get to know each other outside the context of meetings and talking about what you did over the weekend can go a long way to helping new team members feel like part of the group.

If the team is not entirely colocated, then there is a greater reliance on electronic methods. Austin Govella talks about having a virtual workroom that is a safe place for the UX people to interact and that allows him to bridge the gap with his colleague in another country. It is like having a virtual conference room, where he can put sketches up on the wall and have meetings. Having a persistent and dedicated place where design ideas and dialog can be shared, in privacy, can be the electronic equivalent of a recurring design meeting. Wiki pages can also help facilitate the communication between remote reams, but as they tend to be more public, they may not necessarily be a tool for engendering candid discussion. In addition to the day-to-day communication and dialog, some amount of design documentation may be helpful to facilitate a common understanding of the design, when teams are working at a distance. While Agile emphasizes not producing documentation for the sake of recordkeeping, nothing prohibits the production of an artifact if it benefits the team. Of course, it is best if that document is one of the items already naturally produced as a part of the design process—sketches, wireframes, html mockups.

Share these electronically, adding comments where they are needed. It is fairly important to share these items when they are created along the way, as they can help to provide insight into what the UX team is doing. Put your work up on the walls for on-site team members to see, post it virtually for remote colleagues. The purpose of doing this is not just to share information, some of the items on the wall might have no annotationsl, but to build a strong design culture. Designs that are tangible and somewhat omnipresent help immerse the whole team in the work of creating a great user experience. The UI stops being something on which the UX team works and specifies and becomes a living thing they look at every time they head to the kitchen. This may spark conversation and ideas that might not otherwise occur and builds a much greater awareness of design.

Regardless of where people sit, it is important that the ideas in one team member's head somehow make their way to the rest of the team. It is equally important that the information does not just go out but is processed by the others and that a healthy dialog ensues. Just like the events and ceremonies, the goal is not to have conversation for the sake of having a conversation but to engage in a real exchange of thoughts and ideas. Otherwise, everyone will drive each other crazy with nonstop chatter without actually adding any real value to the process or the product. Creating a genuine discourse is what allows the team to produce the best work possible.

DEFINE THE BIG PICTURE

Most of the people I interviewed for the case studies mentioned the need for an overall vision to help guide the work done during the course of a release. Agile teams can be especially susceptible to getting lost in the details of the work and lose track of how their tasks are meant to fit together. While many UX people mention the need for providing a framework for the design work, everyone seems to have a favorite way of providing that.

Personas can be a very effective technique to give the context to the design efforts. They are easy to produce and work into everything that the UX person touches. Because they are so easy to remember, it is easy for the entire team to use them as a touchstone to inform the decisions made throughout the process. Some UX people mention having a high-level design in place at the beginning to serve as a roadmap for the rest of the detailed design, which is developed during the release. This may not always be possible, but something visual that people can keep in the back of their minds as they move through the release can be a powerful tool to contribute to a coherent-looking release. This high-level design can be as simple as a workflow diagram or a quick sketch of one or two main screens. It cannot be too detailed, since it is created at the beginning of the cycle, when so much has yet to be defined. And it needs to be sufficiently vague so as not to stifle creative solutions that might be uncovered during the detailed design. Having such a design document in place from the beginning can serve as a veritable roadmap for the UI and a reference point for the entire team. Another tool often mentioned is a reference library. This tool is best used to increase consistency across the product rather than to indicate a specific design direction. However, these can be very useful when pulling several products together, extending the reach of a scarce UX resource, or minimizing the amount of unnecessary variation within a product.

TRAINING

Every successful team engaged in some form of Agile training, with many of them receiving training and coaching from Jeff Patton himself. Since many of the teams that are struggling had little or no training and are working a bit blind, there seems to be a pretty clear connection between training and a better implementation of Agile. On the extreme end, Suzanne O'Kelley and her teammates were able to participate in the LUXr residency program ("10-Week Residency," n.d.), a 10=week program where they were able to work on their real project problems in an environment where they received coaching. This level of training can be a factor in success, not just because it means the team receives a good, consistent education, but also because it indicates an organizational commitment to the process and a supportive management team. Other organizations routinely bring in coaches, so that the training is not a single event when they first begin to engage in Agile activities, but is a continuous effort that allows the teams to refine their process and get the guidance that they need to evolve. This also benefits employees who are new to the organization and were not onboard for earlier rounds of training. Not only have they the chance to be trained, they do so in the context of the environment in which they will work. This achieves consistency in the application of the process and a reinforcement of the best way to work at that company.

Unfortunately, not all of us can have Jeff Patton to provide coaching or attend a residency program. This does not mean that no other options can provide similar, although certainly not equal, value. Christina York started out with one user interface engineering session Jeff Patton presented. While no replacement for in-person coaching, it can be a very inexpensive way to get a lot of high-quality information into the hands of the team. It is even better if the entire team attends the training and then debriefs afterward over what the members learned. With one fairly inexpensive event, you can achieve some of the same education and team building results that come from a more formal training. You miss out on the tailored content and discussion, but it is a great replacement when the other options are not on the table. Another thing to bear in mind is that, after Christina's team achieved some success and the process was more ingrained at the company, there was more support for having team members attend conferences to stay current on Agile techniques. The organization saw that the process was working, and as time went on, it chose to invest in honing the skills that contributed to that productivity. Her tactics shows how to get your foot in the door and prove that investing in training for the team can contribute to good results and encourage continued investment.

However, even if you are in a situation with virtually no budget or organizational support for training, you should still engage in some kind of learning to equip yourself and get a good understanding of the road ahead of you. Many resources are available that require little or no money. When Jeff Gothelf started out, he simply reached out to practitioners who had experience and arranged to talk about their stories. It is amazing how generous perfect strangers are willing to be with their time, and a 45-minute conversation with someone can yield tremendous benefit and inspiration. Books, newsgroups, blogs, and local professional groups provide many opportunities to learn what your colleagues are doing in your neighborhood or around the world. Some of these resources might be broadly focused on Agile in general, while others are more specific to Agile UX. Explore the options available in both categories.

As much as possible, these resources should be leveraged for team-building events. Either the UX team should engage in training activities together or the entire project team can participate in the activities. Read the blog articles together and discuss them, have a book club, and share links to cool websites. Do not be afraid to share articles on Agile UX with other functional areas, after all they need to feel comfortable with how the UX team supports the project. Given how much training can contribute to the establishment of a high-functioning team, do as much of it as you can, and continue to do it even after you feel comfortable with Agile. It is an evolving methodology, and new information and ideas are coming out all the time that can inspire and help you solve problems. Your process and your team is evolving all the time as well, so a technique that held no interest for you a year ago might be incredibly relevant to you now.

Resources for learning about agile

When first starting out, it can often be difficult to even know where to look to learn more about Agile and Agile UX. I compiled a list of resources that I found to be helpful, but new forums are always cropping up, so do not be afraid to put your favorite search engine to use to find more.

Book

There are many books about Agile as a development process and quite a few of them get good reviews. There are fewer books about Agile UX, but this is likely to change. One book is a good starting point, no matter what your area of interest is or what you are looking to learn about, *Agile Software Development* by Alistair Cockburn (2002). This is the primer and the definitive book about Agile software development. The focus is on development, this is not a book that is going to speak about user experience or product design. Since this is *the* book about Agile and Alistair Cockburn is one of the original authors of the Agile Manifesto, reading this book provides a great insight into the values that started it all.

Online resources

Many resources are available online—blogs, discussion forums, and training. I like online options because they have so much immediacy, you can find what you need when you need it. This allows you to stay current with the newest ideas and techniques. You can access these items from anywhere in the world and supplement your locally available offerings. The list contains the sites I found most helpful and informative, presented in no particular order. Bear in mind that this is just the tip of the iceberg of what it out there.

- **LinkedIn**, **Agile Experience Design group.** For those of you who have LinkedIn accounts, joining this group, moderated by Anders Ramsay, is a great way to participate in or listen to conversations about Agile with a specific focus on design. It also lets you see who is interested in the same issues ast you and puts a face to the broader community. Much diversity is found in the type of questions that arise in this forum. It is an easy way to hear what people are talking about and learn from the ensuing discussion.
- **www.agileproductdesign.com**. This is Jeff Patton's website, and it is a goldmine of information about Agile, with a focus on incorporating user-centered design into the process. On this site is his blog, an archive of some of his presentations, links to articles he wrote, descriptions of the training he provides, and a calendar of the events in which he is participating. It is incredibly comprehensive and one of the single best resources for designers and researchers interested in learning more about Agile. It is not too strong a statement to say that, if you only go to one site to learn about Agile, you should go to this site.
- **Jeffgothelf.com.** Jeff Gothelf and his experience at TheLadders.com constitute one of the case studies in this book. Jeff is also a thought leader in the Lean UX movement and has quite a bit of insight to share on the topic. In fact, he wrote a book, *Lean UX: Applying Lean Principles to Improve User Experience* (2013), on the topic. His blog, links to published articles, and many of his presentations can be found on this site. There are articles and presentations about his experience of bringing Agile to TheLadders.com and you can also find his presentation "Lean UX: Getting out of the Deliverables Business." So much valuable content and up-to-the-moment discussion on the most current issues are on the site that this should be required reading for UX person finding his or her way with Agile.

- **www.balancedteam.org**. The organization's description says, "Balanced Team is a group of people who are interested in furthering processes and methodologies to create great things." The site has a cross-discipline focus, but many of the discussion center around UX. The blog on this site is a great read, slides from the Balanced Team Conference are also posted, complete with audio. If you cannot attend conferences or training events, spend some time with these materials.
- **www.agilemanifesto.org.** This is where the Agile Manifesto, and its principles, live. There are no blogs or discussion groups, and a little history. This is the source of it all. You might see the Agile Manifesto reprinted or referenced, including in this book, but it is worth a visit just to know that you have gone to the original source and seen the original text as it was first written.
- **www.scrumalliance.org**. This site is focused on Scrum, specifically from a development viewpoint. Some interesting articles are to be found under the Resource Center, as well as some information about training. For a nondeveloper, the most valuable part of the site might be its "Scrum 101" page, under "Scrum Basics." In a few paragraphs, it tells you everything you need to know, at a high level, about Scrum.
- **www.extremeprogramming.org**. The first thing you see when you get to this site are the words "Extreme Programming: A Gentle Introduction" and that it exactly what you get on the homepage. The rest of the site is more geared toward developers, providing links to white papers, articles, and books. If you are looking for some quick information about Extreme Programming, the first page of the site is the right place to go.
- **Andersramsay.com.** This site is broadly focused on user experience, but Anders is also an Agile coach and has many postings that speak to Agile and UX. On this site, you can find his presentation "Learning to Play UX Rugby," which is not only entertaining but describes some interesting techniques for UX practitioners. "The UX of User Stories," Parts 1 and 2, are great blog posts about what user stories mean to the UX practitioners. It contains some practical advice, and these posts are very helpful for a UX person looking to influence his or her Agile environment.
- **UIE.com.** This is the site for Jared Spool's User Interface Engineering consultancy; it provides a variety of content on Agile, including training and events that the company offers on the topic. There are also several blog posts and articles on Agile. In Jared's article "Essential UX Layers for Agile and Lean Design Teams," he describes a mental model for thinking about UX and how to fit it into Agile that is quite interesting and can help a UX team define, or possibly redefine, its approach to integrating with Agile methods.
- **Mountaingoatsoftware.com.** I admit that part of why I like this site as much as I do is because the company makes the coolest set of planning

poker cards, with goats on them. The site, which is done by Agile coach, consultant, and author Mike Cohn, has a wealth of information about Agile and Scrum. It is development focused and aimed toward educating people about the practice of Scrum specifically and Agile in general. Many training offerings are available, including some very reasonably priced "eLearning" sessions, which would be a very helpful resource for organizations where training is needed but there is not much budget to provide any.

- **Agile-UX.com.** This site belongs to Jean Claude Grosjean, an Agile coach and UX consultant. He has some great presentations and blog posts, but my favorite by far is "Collaborative Workshop and the Letter to Santa Claus" series of blog posts. This series is probably one of the most entertaining, adorable, and accessible examples of collaborative design.
- **UXMatters.com.** If you have not been to this site before, it is essentially an online magazine of really good articles about UX. As a whole, it is not Agile specific, but the subject comes up in a handful of articles, authored by experts. "Clash of the Titans: Agile and UCD" by Richard Cecil definitely wins the award for best title but is also a very thoughtful look at what Agile means for user-centered design. "Developing UX Agility: Letting Go of Perfection" is an interesting case study from the UX team at ProQuest about their adoption of Agile. I also found that the panel Q&A in "Integrating UX into an Agile Environment" to provide a nice summary of common concerns with great perspectives from a variety of contributors.
- **Johnnyholland.com,** As the site says, "Johnny Holland is an open collective exploring interaction design. Oh. and Johnny loves you." How can you not love a site that loves you back? That aside, it is a really cool site with a broad focus on interaction design and frequent contributors who bring the discussion around to Agile and Lean UX. Greg Laugero's article, "Lean UX Is Dead. Long Live Lean UX," is a thought provoking take on Lean UX as a strategic, rather than tactical, tool. A lot of content on this site will make you think differently about Agile specifically and design in general, so explore it at your leisure.
- **Nomad8.com,** Sandy Mamoli, whose title is "director of all things Agile" at Nomad8, has an Agile blog on this site that contains some of the most practical advice on Agile I have seen. Sometimes, the common discourse about Agile can get a little abstract or too philosophical for everyday use, but her posts on things, like "Should We Choose Agile or Iterative Development?" are accessible without being simplistic and her checklists for seemingly everything are a great resource for teams in the early stages of Agile adoption. Her "10 Ways to Fail with Agile" presentation is well worth viewing, and not just because it is a presentation that does not use PowerPoint.
- **leansoftwareengineering.com.** This site is full of essays from Corey Ladas and Bernie Thompson and their thoughts on a variety of Agile topics,

including Scrumban, on which Corey actually wrote the book. While the content is a few years old, it is still relevant and interesting and worth a read for their practical take on Agile processes, especially if your team is interested in exploring the use of Kanban (or Scrumban).

- **Slideshare.com.** This slide deck repository has everything, if you are willing to do a little digging or have time to get lost in the related links. Not every deck stands on its own very well, but so many presentations are there that you can easily move along to one that does. It also turns up more specific content than your average search engine, so it can make a great jumping off point for your Agile research.

ADAPT AND EVOLVE

The most powerful moment in Jeff Gothelf's story is when he has a UX team-only retrospective, and the team shows the flowchart it created, where every path leads to the conclusion that Agile "creates a negative environment that fosters failure and generates low morale." This very compelling that this moment is not the end of the story, but just one of the turning points in the team's Agile evolution. The team examined the problems, sorted out a few key issues, and kept going, eventually realizing a very successful process. The key was to have the kind of environment where it was safe for the team to express its dissatisfaction and acknowledge that a problem needed to be addressed. But, it is not enough for issues to be brought up—that is the easy part—commitment is needed to work to find a solution and a rhythm that is a better fit for the team. Most if not all teams will find that their transition is a rocky road, and rarely does team get it right from Day 1. But, with an environment committed to making things work for the team, it is possible to get over the initial hurdles.

As a UX person, it is highly likely that you will work with more than one project team. This means that even though you have managed to achieve an organic rhythm with one team, you might find yourself starting all over again with a new team. And perhaps, this time, the team members are scattered all over the world; the team has less Agile experience than your previous team; the team already began the project without you; any number of less than ideal factors can affect your preferred approach to Agile. This is where you adapt. Every Agile project is different because it comprises unique individuals. Feeling skilled at the process is the first step, the next is to have mastery at working as efficiently as possible with every group you support. For less-experienced practitioners, adaptation can be challenging, since it is hard to know whether you are making a reasonable and appropriate change to the process or abandoning it. Training can go a long way in giving a higher level of confidence with this, and returning to the Agile core values and principles helps keep you on track. More-experienced practitioners need to keep an open mind and look

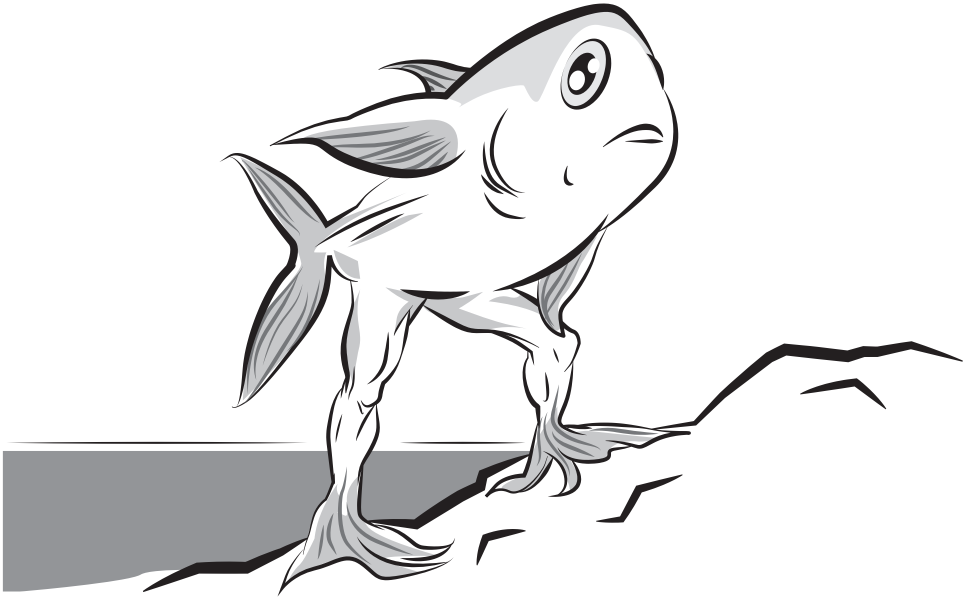

FIGURE 4.3
Adapt and evolve.

for opportunities to increase communication and enhance teamwork while accommodating whatever logistical challenges might be present. So much evolution occurs with Agile methods, especially with respect to UX, that it is almost impossible not to learn new tricks that help your own growth as an Agile UX practitioner.

CASE STUDY—SARAH KAHN, ADZERK

While there are many interesting stories, it is rare to find one that embodies so many of the success criteria. Sarah Kahn and Adzerk are that special situation. They are an example of what can happen when the team focuses on how the least amount of process can be used to support their work, working together and getting along well, healthy communication and interactions, and adaptation of the methods to fit the group's needs.

Sarah is a UX engineer at Adzerk, a small startup whose software helps content publishers optimize their advertising inventory. As the fourth employee, and currently the only UX person on the staff, she is in the unique position of helping define the process as it evolved at the company. Adzerk has a small team and a very strong culture of collaboration and communication, which it consciously created. The decision to adopt an Agile process was born out of the need to have a more-defined process in place as the team grew, married with a strong dislike for doing anything that takes the team away from writing code or introduces bureaucracy into the members lives. Sarah's story tells of an organic process that works very well to support the needs of the team and its customers.

When Sarah joined the company, the entire team could sit around a table and talk, so there was no real need for any kind of a process. They were able to simply decide what needed to be done and do it. As the company grew, it recognized the need for a more structured framework to support the development efforts and keep things moving forward in an organized way. However, many of the developers had negative experiences with heavy-handed traditional processes and did not want to spend their time doing paperwork or estimation exercises. They wanted to use the barest method possible, which naturally led them to Agile and its most lightweight method, Kanban. Sarah describes the company's version of Kanban as being more "Kanban inspired" than a by-the-book implementation of the process. The company took what it needed from Kanban and adapted the process to make it their own. As a UX person, Sarah was accustomed to fitting her work in wherever she could and in a lightweight manner. Since formal tools are expensive, she is used to making do with what she had. She always relied on pencil and paper and whiteboard sketching to get her work done. This spirit of working quickly and lightly fits well into her small startup and makes her transition to working with Kanban a little more seamless.

Adzerk originally thought it would have three columns for its Kanban board—"UX." "Working," and "Acceptance." Once ity started examining the workflow, it realized that the UX work really fell into two phases. The first phase would be the time when Sarah is doing initial design work and testing with customers. In the second phase, she would be debugging the designs and refining them. The real difference in moving an item from the first phase to the second is that it indicates that the UI design is ready for the backend to be developed. The second phase still holds the potential for rework, but the design is solid and stable enough for some coding to begin. Sarah also does the UI front-end development, so this piece of work is being done by her within the UX phases. Having adjusted so that the UX work is in two phases, the resulting Kanban board has four columns "UX Working," "UX Acceptance," "Working," and "Acceptance.. One of the firm rules is that a card moves only forward, never backward, on the board. Customer testing can occur either before the UX working phase or during it, as needed. Sarah also talks to customers as a part of rolling out new features, to validate them before turning feature flags on for a broader audience. The only drawback in talking to customers so frequently is a certain amount of feedback burnout because of the high frequency of customer contact and feedback solicitation, so Sarah is always looking for new ways to recruit users and get information to help the designs. When she is looking to explore an issue like language or buttons, she uses social media to gather feedback.

A task can be created and added to the team's backlog in four ways. An item can be a bug fix from a previous version. Sarah also works with customer support to identify rough spots and feature requests that need to be included in the backlog. They also include competitive features that are identified internally, often by their CEO. The final category is contract-driven feature work. When an item is created and put into the backlog, generally, some initial discussion takes place around it, maybe even a quick whiteboard session to do some design thinking. When the backlog is populated, each item is given an initial T-shirt size representing a rough level of estimate. XS might represent a half day worth of work, and L might map to a week's effort. If it is larger than that, then it requires the team to break it down further. Each item has two sizes, a UX size and a development size. Sarah assigns the UX T-shirt size and the developer that pulls the task assigns the development size. The team does no formal planning, since it has no traditional release cycle. There are no branches in the code base, once you commit code it goes in to production. This does not mean that the team does no planning at all; it is just more loosely structured. For more information on how Adzerk uses Kanban, see their blog (http://team.adzerk.com/post/12289837931/agile-ux-and-kanban-at-adzerk?goback¼%2Egna_3315113.)

The company started using AgileZen to help it to manage its backlog. It has no formal roadmap as much as a big wish list of items. It found that constantly staring at huge list of items could be a bit overwhelming, so it is useful to have a tool to manage the list. Each week, team members pick a few items to focus on for the week or two duration of the sprint. Sprints themselves can be of flexible length, depending on the situation. If a customer has a specific need and date that needs to be met for contractual reasons, then the timing and duration of the sprints are geared toward supporting that effort. One of the benefits to using AgileZen is that it is very visual, which makes it easy to see what is happening to the tasks. If things are stuck, then the column turns red, and everyone can sees that there is a blockage and where it is. This allows the entire team to have clarity on where the issues are and when they need to drop what they are doing to pull together to resolve a problem and move things forward. Sarah also feels that it reminds everyone that all of them are working on the same thing. The fact that team members are able to work like this is a testament to their ability to work in a highly collaborative and communicative way. It is important to understand that the ability to work in such a natural rhythm is a result of this culture. It would not be possible for a larger company with big projects teams and a history of communication issues to work this way and be effective.

The team has a pretty cool way of managing the features. Since there are no branches and code submission is automatically part of the production code, they put in a simple "if" statement that allows them to control the visibility of and access to a feature. The team built in a simple UI to allow product management and customer support people to turn flags on and off, although the simplicity of the dialog started to decrease as more and more feature flags were created. They also found that, once the feature was released and the feature flag turned on for a customer, it might be forgotten about for broader distribution. This happened partly because the task would no longer be in AgileZen, since it had been completed and the features required a certain amount of time to solidify and have the bugs worked out. There also needed to be an opportunity to notify customers about new features, so that the ground was not constantly shifting beneath them. Given the potential for a lag between the task leaving AgileZen and being turned on for everyone, the company needed a way to track the items. The team gives the feature flags their own Kanban board. This board has three lanes: "UX," "Dev," and "Ready." A card is created at the same time as the feature flag. A date is associated with each item as it moves through the columns, so it is easy to see if an item is stalled in one section or another. Once a feature is robust, Sarah has a list of beta testers to exercise the feature. They agree to use the feature for a week, and if their feedback is positive, then she can turn the feature on for everyone. Once a feature is "on," information about it needs to be posted to the company's blog and the customer support team also does a write-up. See Adzerk's blog post on managing feature flags for more details on this technique (http://team.adzerk.com/post/20129347973/a-word-on-managing-feature-flags).

Since Sarah was one of the earliest employees, she has been able to be a very active participant in defining the process and how UX fits into things. She acknowledges that she has been quite lucky and management has been very supportive, essentially telling her to figure out what she needs and go ahead with it. She also admits that it always helps to be in a place where UX is valued. While she is currently the only UX person, she is fairly confident that the process currently in place could extend to accommodate more UX team members. When asked if she would practice Agile techniques if she ever found herself in an environment with a more traditional development process, Sarah thinks that she would. She feels that there is still room to fit in Lean UX, even in a waterfall environment. She thinks that, if she were in that situation, she would talk about it in terms of timeline to get it accepted into the process. Since waterfall methods move so slowly, she feels that there is plenty of room for doing smaller cycles and fitting iterations into that.

When I asked Sarah what advice she would give to UX folks trying to find their way with Agile, her first piece of advice is not to read a lot of Agile books or worry so much about your process. She feels that if you worry about your customers and how to meet their needs and figure out to support that, you will work out the process that enables it. She quotes the Agile Manifesto, "People over Process." The other advice she gives is for the UX team to be constantly communicating and chatting with the rest of their teammates. This means engaging each other in emails, virtual chatting, and face-to-face discussions to keep the flow of information going and support relationship building. She recommends instituting team lunches and getting outside of your department. Getting to know each other as people and connecting in ways outside the project work helps the team to gel and be better able to work together. Sarah acknowledges that Adzerk is really lucky about the blend of people that it has, although this has been a fairly conscious part of the hiring process. It has been able to hire for fit. The resulting team is a group of self-motivated individuals who are friendly and close knit. They have team lunches and spend time together in addition to executing work projects. Their CEO has been very clear about the culture he wants to create. He wants a company of friendly, easygoing people and has mandated that there be no more than one formal meeting a week. She also thinks that it helps that everyone on the team has worked in other places before Adzerk, and many of them had negative experiences with project management and want to avoid recreating those experiences.

Sarah likes that working in an Agile way virtually removes all the friction from her job. She feels empowered to do what she needs to do without having to justify it. She has laid out the process, everyone who works with her knows what it is, and she is able to just do it. Such a transparent process helps build the trust around how the team works. It is a very collaborative way to work. Nothing goes into the backlog without a little discussion. The company is so committed to this transparency that it extends that to its customers as much as possible. However, when you empower your users so much, it can be easy to get derailed by their requests. The way that they manage this is to just be honest with the customers. If customers have asked for something that is in the backlog but not likely to be released for a few months, the company just tells the customer that it is actively being worked on. It acknowledge that the request is a good idea but that the customer probably will not see it for a while. This works, because the company has credibility with the customers from the work done previously. Sarah is also able to leverage these customer requests to recruit for usability testing. The transparency that the company creates with its customers through the constant interactions, blog posts, and the collaborative spirit builds a high levels of trust, which makes this dynamic possible. The team blogs not just about product features but also about the process;d these blog posts can be found at team.adzerk.com.

The level of transparency that this team has about its process, both internally and externally, is a testament to its commitment to creating a circle of trust. The posts doe not just advertise their successes but include candid discussions that speak to the evolution of the process and acknowledge things that it realizes require refinement. This kind of self-awareness and sharing of information shows how deeply ingrained Agile values are in the culture. This is why it can take a very lightweight method like Kanban, use it, and modify it to fit its needs. By nature, thecompany is highly collaborative, Agile, and lean; it just needs a framework around what it does so that its natural process is extensible and can accommodate a growing team.

Key points

- Often, we focus on the mechanics and tactics of the process. We look to meetings and tools to help build a certain culture. However, Sarah's point that the team is cohesive because the members spend time together getting to know each other as humans is a something often overlooked. Sometimes, instead of having a retrospective or hashing out issues, we should invest time getting to know each other as people and talking about things other than work. Have lunch, maybe even off site, as a team and talk about what interests you. Team building occurs in many ways, and it always contributes to a better working environment.
- Do what feels right for your team, especially if your team members really know who they are. It might be a little unusual to have a separate Kanban board to track feature flags, but it works for the team and came out of identifying a gap in the process. They could been hung up on whether doing the second board was Lean enough or properly Kanban, but that would have been wasting energy instead of solving the problem.

CASE STUDY—ANONYMOUS 3, A COMPANY SPECIALIZING IN DIRECT MARKETING PRODUCTS

Success stories are educational, but it can be equally informative to look at teams having more difficulty to see what contributes to their difficulties. This case study tells the story of a team that is missing most of the success criteria, and the UX staff member finds herself in a situation that is confusing and chaotic.

Anonymous is a senior designer at a company that develops Web-based products for small customers to facilitate interactions with their own customer bases. On the surface, it might sound as if her company is employing the same Kanban techniques that Archie Miller describes in his story; however, the result is drastically different. This story stands in stark contrast to Snagajob.com and serves as a reminder that the process is not the biggest driver of success but the people and the culture.

When she started with the company a few years ago, the teams had already adopted Agile and some had moved to Kanban, or at least a loose interpretation of it. Despite internal experts in the methods, most people had no formal training and a lot of individual interpretations of how the process should work. While varying interpretations are fine, if they are done to improve the flow of work, the lack of training can be a problem. It increases the likelihood that some of the variations are not thoughtful adaptations but come from a lack of knowledge. The

UX team at this company is centralized and supports multiple teams; this is challenging for many reasons, but the variability among the approaches to implementing Kanban added an unnecessary level of complexity for staff members who were not able to generalize to one approach.

Each UX designer is responsible for his or her own time and is a project manager. (At the time we spoke, the team had no manager and had been in the process of looking for one for quite some time. This is a pretty unusual and extenuating circumstance that contributes to some of the chaos around the UX team.) The designers really just try to keep up and go with the flow, since they are dealing with a fairly large workload.

In describing the overall process, she says that a request would come in and the designer would get scope from the product manager. The product manager would break down the tasks and do this as an isolated activity, without the involvement of the development team or any other team members. The PMs also set task priorities based on their vision, and other functional areas are not involved in this activity either. The result of working this way is that the team has no opportunity to voice its opinion and share their insights. A development team could easily identify items that might be risky or difficult and help the PM to make decisions based on this information. Even more unfortunate is that, by keeping such close control over the process, the PM actually decreases the likelihood that the work will be implemented as he or she intended. With no conversation around these items when they are created or prioritized, the team loses out on understanding the rationale behind the requirements and the priorities and has less insight into what the PM is trying to achieve with that item. Additionally, the company has no analyst to help define and clarify scope and requirements based on real metrics. This makes her feel like she is working blind, in many ways, especially since there is no use of metrics to inform any decisions. She also worked on a single project that had 200 or 300 user stories with no epics to tie them together or establish any relationships among the stories. Trying to track that many user stories using a Kanban board is pushing the limits of the technique and makes things harder for the team to manage. To have so many stories with no overarching epic or clearly identified relationship reduces the amount of understanding that the team has of the work and makes it more challenging for it to have an overview of the project.

At the project kickoff, these user stories would be in place as created independently by the Product Manager. At the meeting, the rest of the user stories would be defined by the development team and UX staff. A project-level sticky note would go on the Kanban board. This is great for visualization and a quick overview but does not really show the scale of the work needed for each item, and she found that a lot of complexity would be hidden by that simple sticky note. This makes it harder to use the boards to get a real sense of the progress of the project, since six sticky notes representing small efforts may have made their way through the process, but one sticky note stalling at the beginning represents a very large piece of work. Plenty of tools are in place to support the project. JIRA is used to manage and track tasks, but this is a development-centric activity, and the UX team's work is excluded from the system. Wiki (Confluence) pages are used for sharing information. Balsamiq is used for design sketches. Visio is used for wireframes, and Visio .pdfs are often attached to the Wiki page. The QE team uses these .pdfs as a reference.

The main form of communication for the UX people with their product teams is conference calls, since many developers are located throughout the United States and the United Kingdom in addition to the development staff colocated with the UX team. These calls are often used as wireframe review session. The wireframes are posted and communicated out as soon they are created, with review sessions scheduled afterward. The main deliverables for the team are wireframes and sketches, although there seems to be some confusion around those deliverables on some of the teams. The UX people are also responsible for research and usability testing. When they conduct usability testing, they document the results and post them. They also generate an email containing a short summary with links to the more detailed report.

She also struggles with the fact that there is no master Kanban board to provides an overview of all the projects. The closest thing to an overview is the UX team's Kanban board, as it documents the efforts of all the team members, but it is not granular enough to provide high-level insights in to the states of the projects or how they fit together. It might not be necessary to have a summary Kanban board, but there is a clear need in her organization for some overview of the work, showing where the efforts are at and how they relate to each other. This is especially valuable for her and her teammates, because they support multiple projects, need a better sense of the big picture, and are not able to get it.

In assessing how the process is working for the organization, she says that it is really difficult for her to work this way, as it feels like the worst of waterfall. Everyone is doing his or her work but without seeing the big picture, and the design team is just trying to stay current with all the work but is never able to get ahead. She is frustrated when she tries to plan for a release and finds that no one is thinking in terms of a phased approach and considering whether or not design just needs to happen in the end game, at the beginning, or all along. The teams interprets this process as relieving them of the burden of planning.

It is hard for her to put her finger on what is keeping the process from being successful, she thinks that part of the trouble for the design team is that the organization is not really sure what design is and thinks of the whole process from an exclusively development-centric perspective. There is no real integration among functional areas, which may be the root of the problems, so it is hard for her to say specifically what these teams should do differently to achieve a more positive outcome, because there is tremendous room for improvement.

The UX team that works on the company's website had a contrasting experience: That development team also practices Agile but is having more success. The projects are smaller, and that team relies on usage metrics to drive what features it work on. This makes the workload more manageable, because the number of overall user stories is smaller and easier to track. Having metrics to guide the work also allows the team insight into how the users actually use the site and the site's relative importance to the user community.

It can almost be more frustrating to see that the Agile process can work at the company, because it feel less like a broad organizational issue. In fact, in looking at her story, it does seem pretty clear that the trouble with the company's adoption of Agile relates to the product team organization, communication of high-level goals and priorities, and a broader scope of work than some of the projects might be able to accommodate. The first warning flag for me is the idea of the user stories being written with no involvement from other functional areas. This might give the PMs the illusion of tighter control over the release, but it really just obscures their general intent and leaves things open for interpretation by the development team. It also completely undermines the focus on communication and transparency at the core of Agile values. From the start, the teams are disconnected from what it is that they are supposed to achieve.

There is also a lack of colocation among the development teams, but I do not necessarily see this as having significant impact on the overall success of the team. While not ideal, geographical diversity is a reality for most people and generally with much larger gaps in time zones than this team has to deal with. In this case, colocation may simply be exacerbate existing communication issues but is probably the least of the problems to address for the process to work more effectively.

These projects seem a questionable fit for the Kanban technique, or at least the way that technique is being applied is not quite right. It seems that the Kanban boards are cover such a broad scope of work that they are not effectively communicating effort and progress as they are intended to. It also would seem that the projects might bear some breaking down a bit, so that a single Kanban board does not represent such a large amount of work. More important, it seems that the culture at the company, or at least within her part of the organization, is not well suited to such a lightweight Agile method. There are clearly gaps in communication and silos between functional areas. It would seem that they might benefit with a form of Agile that has more framework to facilitate the resolution of such issues or at least more training and coaching to refine the existing process.

Key points

- A process is useless without a culture to support it. The success of an any Agile project depends highly on frequent discussion and teamwork. The product managers here are effectively undermining the process from the start by controlling the most critical artifacts, developing them in isolation, and throwing them over the wall to the product team. When the first step of the process is done in such a non-Agile way, it is very difficult for the team to successfully move forward and balance out the effect of that first misstep.
- Training matters and should be supplied to every functional area that supports Agile. While not every employee requires such training, there needs to be a critical mass of consistently trained "experts" who can speak up when the process starts to run off its course and no one has a great sense of how to fix it. People put their own spin on something they did not understand very well in the first place, which can be dangerous. It is okay to deviate, but you should have a solid understanding of what you are changing (and why) before you do so.

SUMMARY

Every Agile team, no matter how experienced, faces different challenges as it tries to work as efficiently as possible to produce a wonderful user experience. While the specific techniques and activities might vary from team to team, a few things clearly contribute to a group's ability to successfully engage in Agile and Agile UX efforts. Focusing the team's attention on the project and not the bells and whistles of the process, healthy team interactions and communication, ensuring there is a sufficient level of training, and continuing to refactor and improve the process lays the groundwork for a high-functioning team.

References

10-Week Residency, n.d. Retrieved April 2, 2012, from http://luxr.co/programs/lean-ux-residency/.

Cockburn, A., 2002. Agile software development. Boston, Addison-Wesley.

Gothelf, J., 2013. Lean UX: Applying Lean Principles to Improve User Experience. O'Reilly Media.

Kahn, S., 2011. November 3. Agile UX. and Kanban at Adzerk. Adzerk Team Blog. Retrieved April 28, 2012, from http://team.adzerk.com/post/12289837931/agile-ux-and-kanban-at-adzerk?goback=%2Egna_3315113.

Kahn, S., 2012, March 29. A word on managing feature flags. Adzerk Team Blog Retrieved April 15, 2012, from, http://team.adzerk.com/post/20129347973/a-word-on-managing-feature-flags.