CHAPTER 3

Case Studies

CONTENTS

Agile User Experience Design. http://dx.doi.org/10.1016/B978-0-12-415953-2.00003-0
Copyright © 2013 Elsevier Inc. All rights reserved.

INTRODUCTION

When I started interviewing people, I hoped to be told that the magical sprint number was precisely *x* number of days, the best tool for managing tasks is this one, and the secret sauce to making Agile work is as follows. Turns out, there is no magic or secret sauce. What I learned was that the successful teams, the middle-of-the-road teams, and the teams still struggling all talked about the same issues. There were no discussions that did not touch on the importance of communication, transparency, culture, and training. However, everyone described an experience that was a little different and the way that each group approached the problem reflected its unique situations. Techniques used in one company may not apply to another, but hearing their stories proves that the road to Agile is not always smooth. But, the trip can certainly be worthwhile. It also illustrates that if your adaptation of Agile methods is different than those of someone chatted with at a conference, it does not mean that you are doing it incorrectly. Keep the values of Agile present in your mind and visible in your work. If you do that and your team is successfully delivering a good, or great, product, then what you are doing is working.

As you read these cases studies you will recognize some names, either the person or the company might be familiar to you. Quite a few of the people I interviewed are active in the discussion of Agile UX and have given presentations or dedicated blog posts to the subject. You will also see a few anonymous case studies. Many people were willing to share their stories, but not everyone is on the record. It can be difficult to being so open and honest about the experience in a public way, especially when your story is not all rainbows and sunshine. While I am grateful to those who were able to speak on the record, I am equally appreciative of those who would only speak if they had anonymity. I felt that their stories were incredibly valuable, in part because it can sometimes be more helpful to hear about teams that are challenged or struggling than only the success stories. Some comfort can be taken in knowing that you are not the only one having difficulty finding your way. I believe that these case studies help present a broader picture of the Agile universe, because we can include tales about where Agile has not been successful for a team. I am thankful for all of my colleagues who shared their stories, and especially to those who did it simply to help others learn from their experience without receiving credit for doing so.

SUZANNE O'KELLY, APPNEXUS

Suzanne manages the UX team at AppNexus, a platform for real-time advertising. She was able to receive the most intense formal training of anyone with whom I spoke and was able to apply the training immediately to the project on which she was working. Suzanne's story provides a good illustration of a small

UX team given very solid Agile training and rolling out the methodology in a focused way, starting with a single project. Often, the discussion is around wholesale changes and entire organizations moving from traditional development processes to Agile, but Suzanne's story reminds us that it is possible to concentrate deeply and thoroughly on one project and leverage that success to influence a bigger change.

Suzanne test drove Lean UX on the development of the AppNexus's "Auto-tagger" feature. This project required that the resulting design be simple to use and transparent in its behavior as it executed its functionality. Suzanne kicked off the project with the product manager then worked with the UI developer, the API (application programming interface) developer, and one other UX person to realize the project.

The development team had been agile for quite some time, but an element of the UX work was still a bit waterfall, and it was not really working to throw the designs over the wall. In her research, Suzanne found lots of information about how different teams were doing UX with Agile. Taking the time to do this kind of research helps make the UX team well-informed about its options for adopting Agile UX. This is fairly critical to success, since the development teams most likely understand Agile only as it affects their process. Unless they had access to a coach who specifically raised the issue of bringing UX into the process, the UX team will be responsible for defining this interaction. Suzanne also wrote a manifesto, to help guide the UX work. This is a really interesting idea and not something commonly done. The Agile Manifesto is so central to all its methods, it makes sense to write a manifesto for your UX team to define the values and principles of how the team will practice Agile UX. While Suzanne did this herself, the creation of a manifesto might be a very interesting group activity. It could be very empowering for the members of a larger UX team to craft a statement of their intention for becoming Agile UX. This is something easily shared outside the UX team, and can help create a much better understanding for everyone about how UX activities fit into an Agile development process.

In addition to Suzanne's efforts, the company sent her team to LUXR (Lean UX Residency) and they used real projects as homework, so they were able to get coaching on the actual project as they went through the 10-week program. The program was a great training, but it did not yield immediate results because the team still had to do the background work of defining personas, engaging customers, and so forth. However, the goal was to have good results at the end of the project and to be able to make traction with the UX integration into Agile; and the training set the team up for that. This approach has a lot of appeal to it.

Suzanne was working with a small amount of UX resources and needed to be strategic in her efforts, so it made sense that the effort to integrate into an Agile environment be done one step at a time. This tactic could work well for larger UX

teams and bigger projects, because it gives the UX team an opportunity to test drive a well-informed approach and leverage its subsequent success, rather than making a big statement about how the UX team will work then needing to refine it. While nothing is wrong with the second approach, if it is done in an organization where not everyone is fully bought into UX, the act of refining an approach may be viewed as a failure rather than a healthy evolution. Starting with a success, of any size, makes for an easier sell of the UX team's approach to Agile.

The whole Autotagger project took about three to four weeks, which was about how much time the team would have expected it to take without using Lean UX. Originally, some had been concerned that the UX team could slow down the development speed. That the UX team had no negative impact on the overall schedule was a big win and recognized as such.

Suzanne was the only permanent full time UX person on her team. She had one full-time UX contractor and two engineers, one of whom had gone through the LUXR training. Additionally, one of the engineers worked remotely and was on site only one week a month. The small team size might be ideal for an Agile process, but the lack of colocation is certainly less than ideal. On the other hand, one benefit of not being really formal in their Agile application is that the members had no specific stages or sprints, which allowed them to follow the natural rhythm of the project. some loose Omnigraffle functional specs were developed to facilitate communication. Suzanne acknowledges that, if the whole team had been local, they might have been able to just whiteboard many items. But, since that was not the case, they chose to produce fully interactive prototypes. She and her remote developer would plan together. They managed remoteness by making sure there was a lot of communication. They did informal daily check-ins, constant emails and phone calls bridged the distance and fostered team dynamics. Sometimes, the engineer had questions, because he was working off of a more lightweight spec, so he would reach out to Suzanne and ask which solution to go with. Sometimes, Suzanne would ask him to implement a few solutions, and they would test them to see what worked better.

She was doing UX testing every day or two. Often, it would be done with internal users. In the very early stages, they did the testing this way while they worked out the big kinks. This high frequency of feedback might be more challenging to manage on a larger project, but it allowed this team to function as if it was bigger, since many more sets of eyes were on the design than the handful of people directly supporting the work. When there was a consensus that the overall design made sense, the team began to take it to customers' sites and gather more robust feedback. Having this level of engagement with the customers really allowed them to build stronger customer relationships. Asking for feedback from the customers earlier in the process also instilled a sense of trust and confidence in the process for the end users. Suzanne found a lot of value in this activity,

beyond just getting direct customer feedback. All customers enjoy having their opinion solicited, but on this project, customers were often able to see the impact of their comments on the project and really know that they were being taken seriously. This gives an end users a new kind of appreciation for the company and allows them to see the relationship as being more of a partnership.

In looking back at her experience, Suzanne felt that communication was a critical factor for the success of the team. They were very informal in how they handled the avenues of communication and the developers were proactive about reaching out to the team. She was not sure that the type of collaboration they experienced was necessarily something that could be formalized and repeated more broadly across different teams. She noted that they were very casual about daily standups, since the team was so small and in such constant communication with each other. Skipping a daily standup would be less viable in the context of a larger team, where it could actually contribute to a communication breakdown. It is true that when a team grows, it can often necessitate formalizing team interactions and more closely following events like those prescribed by Scrum. However, it is important not to lose some of the natural rhythm, and create a camaraderie that makes it comfortable for a remote worker to reach out or, even better, have your voice in the back of his or her head while doing the work. A smaller team, especially one that attends training together, is better able to connect with each other and facilitate collaboration. A larger team should look to see what it can steal from these smaller groups to create the same feeling. Encouraging frequent emails, IM chats, and screen sharing sessions to review work—any of the things that small teams do naturally can be leveraged by a bigger group.

Having gone through the process, Suzanne acknowledges that she is still in the early stages of Lean UX. Despite her formal training, she feels that tackling Agile does not have to be a really formal process. She recommends engaging in Agile in any form, even if it is by the seat of your pants; doing a little piece of Lean UX is better than not doing any Agile at all. This is a very interesting thought, since so much of the discussion around Agile is very all or nothing. However, the idea of getting leaner is just as valid, as being Lean UX is very powerful. It also might make it less intimidating for teams to move in this direction if they approach it as something that can be tried on a per-project basis or on a small scale rather than adopting it across the board. And she is right; it is better to try to get more user feedback into the cycle or spend some time discussing the design rather than focusing exclusively on producing the specifications. Taking those baby steps does not qualify your team as a Lean UX, but they may reduce some wasted effort and possibly encourage a move to a different process.

While Suzanne's experience is certainly a testament to the power of strong training, her ability to work well with her colleagues and customers allowed

her to be effective in such a short amount of time. She was truly able to treat her team and her customers as design partners and produce a better user experience because of that. To read more about Suzanne's experience (O'Kelley, n.d.), expressed in her own words, check out the AppNexus tech blog online at http://techblog.appnexus.com/2011/autotagger-a-case-study-for-lean-ux/.

Key points

- The communication strategy needs to fit the team dynamics. It does not need to be formal for the sake of being formal, but if the team chooses to skip standups or another Agile event, then it should be done consciously and only if there are no negative consequences. The team must find opportunities to speak frequently outside formal meetings and use whatever technology is available to support that—video calls, chat clients, online workrooms, Wiki pages.
- Formal training can be a fantastic opportunity to learn and get coaching on a project. It can also encourage team bonding and getting the team on the same page. Funding for such training also indicates organizational support for the effort and creates more of an investment in the process in general. It should be pursued within your organization, as much as possible. And, if there is no budget for these things, be creative in seeing what you can do to replace it.
- Engaging in Agile does not have to be intimidating. While corporate initiation of the process can contribute to a broader, more successful adoption of the process, introducing Agile into the process a little bit at a time or in a more grassroots manner can create interest in taking bigger steps and help introduce Agile concepts in a nonthreatening way. A little

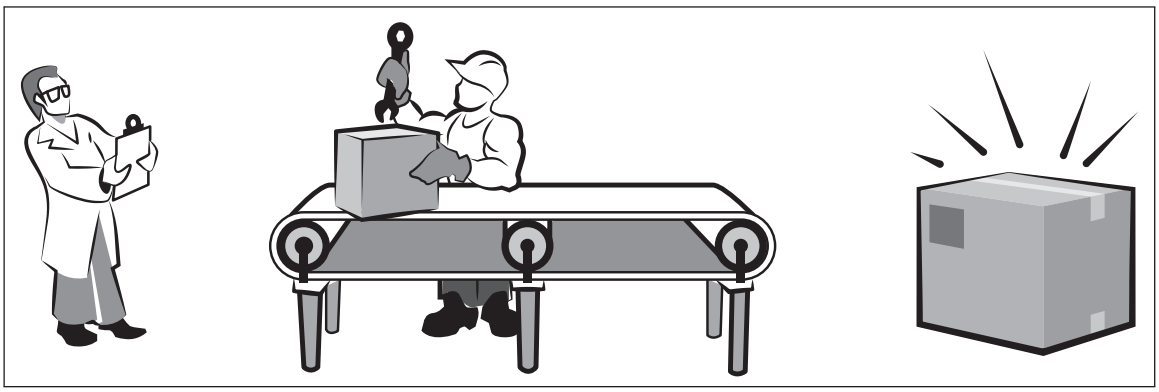

FIGURE 3.1
Lean UX.

Agile, when it contributes to the success of a project, can go a long way in creating interest.

THYRA RAUCH, IBM

Thyra is a user experience leader at IBM, and her story gives insight in to how a practitioner can manage to support an Agile project that faces the challenges of multiple locations and time zones. Her approach is an example of how to adapt to fit without compromising the process.

On her first Agile project, none of the team members really knew what it meant to be Agile. Then, the entire team went to Agile training together. They were able to use their real project as the in-class example and get coaching as they did their training. (This was seen as a success and has been done for other teams since then.) Once they were done with training, they were very strict about their application of Agile. Thyra says it was a bit awkward at first, as is any new process, but it worked. Successfully releasing the product in three months also made people feel very positive about the experience. At the next release, they made some minor adjustments to the process, based on brainstorming sessions they had during the postmortem about how to make the process work even better. As a UX person, she had not come into the process with great expectations but was really surprised at how much she liked it. She was additionally surprised to see how much more effective she was. Previously, Thyra found that she was hung up on all of the work around the up-front design. With Agile, she was able to put that aside and keep an open mind as she engaged in the design process. She describes the process as more fluid but feels it gives her more control over what she is producing. Every sprint you plan again and can feed all the user research back in to the cycle rather than waiting months or years for the next release because it was too late in the cycle to react.

Thyra feels that the Agile environment changes the dynamics so that the place no longer feels as though development's voice primarily gets heard. Everyone, including those in UX, is an integral part of the team. There may not be an opportunity to do big research ahead of the cycle, but UX people create scenarios, refine them, and check in with customers as they go along. She acknowledges that it is not necessarily going to be comfortable to go Agile at first. The team has to believe it is the right thing, and management needs to support it for people to really invest in making it work.

At the time of the interview, Thyra was working on a text analytics product, which provides visual representations of a large amount of structured and unstructured data that the user can explore. Unlike most Agile teams, the makeup for this particular team is not colocated. Development is in Japan, QA (quality assurance, testing) is split between Japan and North Carolina.

Documentation is in Maryland, the visual designer is in Vancouver, while the product manager is in southern California. She is either in North Carolina or northern California. There are few convenient common time zones, so daily standups for the entire team are not a practical option. Communication is very critical to any Agile team, but they had to approach this in a nontraditional way. Thyra works with the visual designer, and they post their work on a Wiki page. Designs get linked or attached to individual story items and can be viewed 24 hours a day. The group tried to have a weekly scrum meeting with the whole team, with development having a daily scrum meeting, but they have since refined the process to eliminate the weekly get together. Development still meets daily, and the scrum master takes notes on the discussion and posts them to the Wiki page. If other functional areas have anything to share, they post it to the Wiki page.

To facilitate communication, she posts a variety of visuals, such as scanned pencil sketches, screen captures with callouts, wireframes, and high-fidelity prototypes. The development team does prototypes and end-of-iteration prototypes and keeps them up on a server. This is critical to communication, as she can access the server any time and see if the interactions are what they need to be. She can also use these end prototypes for user testing. Thyra notes that many methods of communication are available now and the communication itself is more important than the event of a face-to-face meeting. While Thyra is the only official UX person, she describes herself as being on a UX team. The team is composed of the visual designer, technical writer, product manager, project manager, QA leader, development manager, and UI lead developers. This group of people meets once a week for an hour, more if necessary. They might discuss the items one iteration ahead of when they will be worked on. The meeting topics and prototypes are usually distributed ahead of time for easier communication and a more effective working session. Using this forum, all the stakeholders are able to provide their feedback

Incorporating user feedback has been different on each Agile project. On brand new products, where she was less familiar with customers, they were able to test every sprint. That worked well for her when she was trying to understand the problem space. With her current project, frequent rounds of usability testing allows them to identify the most successful features and quickly identify issues, such as the need for the users to access "Help" despite their reluctance to do so. As a result, the team was able to build more help into the design and create mini-tutorials that facilitated the end user's ability to navigate through the product.

When supporting a more mature product, she can cut back to one user session every other sprint. She has more design partners, which are critical, since she needs a certain amount of feedback from expert users on what works for them and what needs improvement. She will continue to test with novices as well,

because making it easier for them makes functioning easy for everyone. All user feedback is extracted into requirements and put into the backlog.

Thyra thinks that it is possible to take some of the Agile nuggets and drop them into any process. If you learn something in Month 3, you should be able to react to that and incorporate into a release. She notes that there are many Agile methods and not really one way to do it. The important thing is to consider the best way for a team to work given the cost, speed, requirements, the like to which they are beholden.

Key points

- Be adaptive. Part of going Agile means working more collaboratively with the team, stakeholders, and end users. Do not get hung up on having a particular event, like a daily standup, if it is impractical for the team. Instead, be creative and find a way to use technology to replace the communication that the event is intended to support.
- Get comfortable with things being more fluid. It is unlikely that there will be opportunities to have huge research projects or to do massive amounts of up-front design. On the other hand, it is more likely that user feedback need not be tabled until the next release cycle because it was received at the wrong stage of development.

ARCHIE MILLER, SNAGAJOB.COM

Archie's experience is a little different than the other success stories. Rather than being the initiator or on the first wave of a UX adoption of Agile, he came into a company that was already using Kanban quite successfully. This case study is not only a study of a culture that is a perfect fit for Agile, but of what it means to fit in to that culture after it has already been defined.

When he first arrived at the company, an online job site for hourly workers, he did not even know what a Kanban board was. This is not unusual, since most UX people are not exposed to Agile methods until they find themselves on a project that uses them. It can also be unsettling to be on such unfamiliar territory, especially if you are well into your career. Adding to the learning curve for Archie was that each team also had its own interpretation, so he had work hard to orient himself to the process quickly as he worked on the job seeker part of the site. In a situation like this, the best thing to do is first understand the lay of the land and how you can contribute to it. Once that has happened, it is important to get training or education on the process, so that you can be part of identifying potential opportunities for improvement. If you are coming into an existing Agile project, the way Archie did, you are likely to receive your training later anyway, since the company might wait until its coach is available or there

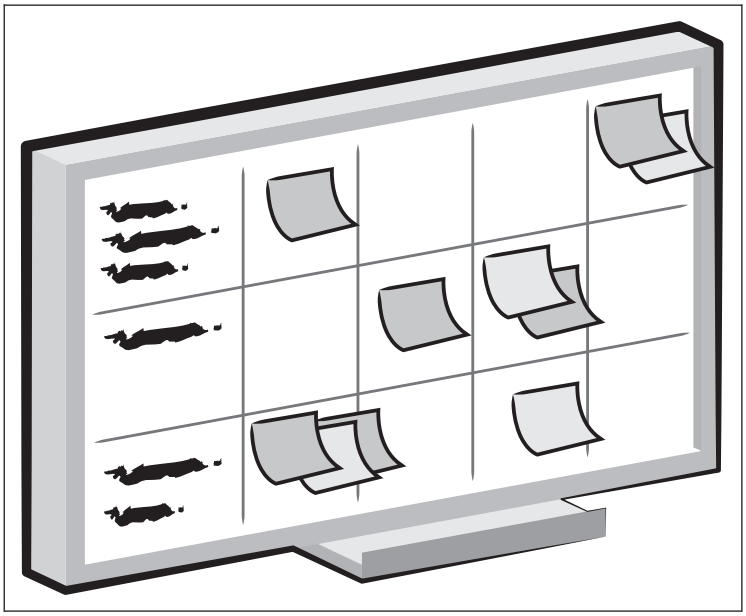

FIGURE 3.2
Kanban.

are enough new hires to put through a class. This is not necessarily a problem. While it is important to get a handle on the process itself, the most critical thing is to understand how your team or teams are working and how you fit with that.

The transition for Archie was undoubtedly made easier by the fact that Agile principles dovetail very well with the corporate culture at his new company. Their culture is so strong, that it practically radiates from the pages of their website. Instead of the typical dry content that one often finds in the "About …" sections of a corporation's website, the Snagajob site has a culture blog, a "How We Roll" page and stories that embody its corporate values. The company defines itself as "Collaborative. Accountable. Passionate." Just from looking at the website, it is very clear that it is intentional in creating a certain environment and wants that to be known. In comparing its site to other job search sites, it certainly has much more content describing the company and giving insight into what it is, what it does, and how it does that. Obviously, this is a great recruiting tool, but it also humanizes the company for anyone who will be doing business with it. The description of what it is provides a certain transparency into the workplace and creates trust with potential customers. Archie describes the culture as being carefully curated and intentional. The recruitment process is geared toward identifying candidates for fit. Once you are an

employee, the company establishes a balance between personal growth goals and specific project goals that recognizes and supports the needs of the individual and the needs of the work that individual supports. If ever a corporate culture was best suited to adopting Agile practices, it is this one.

On its Kanban board, Archie's team typically has the columns "Not Started," "Discovery," "Clarify Context," "Design Solutions," "Plan Delivery,' and "Done." The "Context" column represents tasks in which problems and goals are still being defined, target users are being determined, and so on. The "Design Solutions" column includes sketching, prototyping, and testing activities. It is important to have this column, as it creates an awareness of the need for these activities and helps teach the team how long these efforts might take for a given feature. The "Clarify Context," "Design Solution," and "Plan Delivery" columns each have "To Do" and "Doing" subcolumns to identify the items ready to be acted on and those that in progress for the given activities. This allows the team to differentiate between items that have moved into this phase but not been worked on and items in progress. Not all teams might find such notation necessary, but it is a great idea for larger teams and larger projects, as it makes it easier to track the real state of a card and identify slow moving or "stuck" items. The rows of the chart large represent chunks of work like "Active leads with job tips" or similar higher-level task. The granular tasks within that row are the ones that make their way across the board until they hit the "Done" column.

Archie's team is responsible for the seeker portion of the site and is made up of a core team of 20 or so people, with a handful of those people working on maintenance projects. The team is colocated, or as Archie describes "extremely colocated," which is a function of both the culture and the company growing at a rate causing it to reach capacity in its office space. Other teams in the company are larger and more dispersed, so those teams rely on tools like Trello to replace the physical Kanban board and help track the work.

The rhythm of the cycle can vary, but the company tends to release every week once the site is up and averages around two weeks during the discovery phase. Archie works a sprint ahead of the development team and his deliverable during the discovery phase is to populate the backlog and create the tasks. This is a great role for a UX person n the process. Not only will you have a great understanding of all of the items in the backlog, you can ensure that the items are customer centric. During the production cycles, he creates Axure prototypes and distributes them to the developers by sending them .urls using AxShare. Archie describes the delivery of the prototypes as organic. The developers can easily access the interactive prototypes that communicate Archie's design intentions. The team is also able to incorporate user feedback into the process as often as it wants by recruiting users, using Ethnio, to do remote testing. This

means that a user who visits Snagajob.com might see a screener from Ethnio to be recruited for a remote usability study. The team has also been able to conduct larger pieces of research via diary studies and site visits with users.

Retrospectives are conducted faithfully and taken very seriously. Archie describes them as being like a religious experience. Someone who cannot attend in person makes sure to communicate input via email. This kind of dedication to self-reflection and process improvement supports Agile but very clearly comes from Snagajob's corporate culture. It is also a clear part of the success of Agile in the company. A sprint or even a project can always go wrong, but if the team discuss it in a retrospective and can make course corrections or learn something for next time, then the process and the team have succeeded. It is also important to note that, even with such an effective process in place, the company continues to invest in training and coaching. It works with Jeff Patton to provide training and coaching to new hires and experienced teams, and he comes in on a recurring basis to work with everyone. This is essential for the newer staff members, because it means that they get the same understanding of the process as the rest of the team and can get some insight into the specific implementation at their new company. It is also very helpful for the existing team members, as it keeps them active in re-evaluating and refining their methods and techniques as the work and the team evolve.

Archie enjoys working in an Agile environment because he feels that he gets a better design more quickly than if he was just working by himself, because he is forced to collaborate with more people earlier in the cycle than he would in a more waterfall process. It is certainly true that, in waterfall, you can go months without showing your designs to anyone, but it would be nearly impossible to go more than two weeks without sharing the work in an Agile workplace. Archie acknowledges that learning to work more collaboratively was a process. He majored in fine arts as an undergraduate and considers himself to be an artist. He is also used to coming up with all the answers, which is typically the role a UX person plays in a non-Agile setting. Having successfully made the transition to Agile, he is a convert. Waterfall feels clunky to him now, and he says that the word *handoff* is not even in his vocabulary anymore.

Archie's story is an example of a culture and a process that fit together l hand in glove. Obviously, corporate culture can be slow to change, and the Snagajob.com environment is fairly unique. Depending on where you work, it may not be a realistic goal for your corporate culture to be as carefully cultivated as that one is. However, an organization that is genuinely adopting Agile values will move in that direction by fostering communication, engaging in healthy retrospective, valuing the contributions of individual people, and fostering an empowered team environment.

Key points

- Everyone needs a little help. Even in a company with an exceptionally Agile-friendly culture, there is still room for coaching. There is always an opportunity to refine and evolve your approach. Having the coaching available to new hires makes their transition much more productive.
- Agile changes the UX role. Archie is not the only to mention this, but being a UX practitioner in an Agile process requires you to think differently about your role and work more collaboratively. If your mind is open to that, the transition is easier.

CAROL SMITH, PERFICIENT

In her role as a lead UX business consultant, Carol has the opportunity to see Agile implementations at many companies and cultures. Carol says that she has not worked with any groups that are doing ideal versions of Agile, but she also points out that she has not worked with anyone who is practicing an ideal form of UX either. Because of her role and expertise, she tends to be involved more in research and usability testing and experiences the companies' Agile experience from that viewpoint.

She recalls one project where she was working for a large insurance company in the Midwest that had a team of 60 UX people. This team comprised the interaction, visual design, and research groups. Carol's project was for the research group. In that company, most projects used Agile. The UX group was also well established, very strong, and played a well-defined role within the organization. However, the level of UX integration into the Agile projects varied by team. This is not uncommon with big design teams, as it can often be hard to standardize across such a large number of people. The UX team members themselves were very open to practicing Agile, but their ability to do so successfully tended to be specific to the team with which they worked.

She noticed that, to fully adopt Agile, the organization has to be at a point where it sees that what it is does not work. She feels that this is fairly analogous to the successful adoption of UX practices in a company. This insight raises an important point about the motivation of a company in engaging in Agile. If the company is doing it because Agile is the new and modern thing or to make things happen faster, the problems that require correction have not necessarily been acknowledged, implementation of Agile may move forward while keeping broken things "as is," and no real change will occur. However, if any organization recognizes that it could be more efficient or relies on a process that has become unwieldy, then it may have an engagement with Agile that is more focused on shifting the culture and have a sense of where things are or are not working well in the current environment.

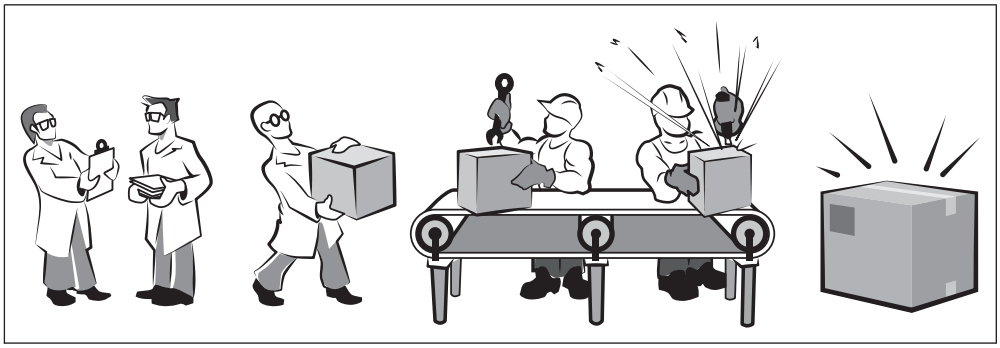

FIGURE 3.3
Agile.

She finds that there is a still a huge amount of ignorance about Agile. At a recent conference on computing, she was surprised that no one she spoke with had even heard about Agile. For those of us in the midst of Agile, it seems like it is all anyone is talking about, but the reality is that, while Agile has gained significant traction over the years, it is not quite ubiquitous. Within the UX community especially, many practitioners have not experienced Agile in their careers and do not see it on their horizon. Carol never encountered any resistance to Agile processes but found plenty of ignorance. When she mentioned this, I wondered if there were geographic clusters in which Agile has been adopted. As someone who lives on the East Coast, I can say that it seems to have taken longer for the conversations about Agile to begin happening out here, and even now Agile methods may not be as commonly used here as out West, where Agile was born. It is also a great sign that Carol says that she has never encountered resistance to Agile, because I can recall quite a bit of resistance or fear of the unknown among UX teams when they started to realize that this development process would have impact on them. It seems that enough UX people now have had direct experience with Agile that the conversation has gone from being based in irrational fear to trying to figure out how to make it work and sometimes about how to make it work better. This really indicates a maturity for both Agile processes and UX practitioners.

Carol acknowledges that ideally you would need a running start to get the research done and ramp up on background information. She suggests that this could be done while development is coding the architecture. She thinks that, once this is completed, UX team members can set up regularly occurring usability testing and do activities like a card sort to hash out the big questions, then just continue the design as they go. Using collaborative techniques, such as a card sort, is a great way to involve the team in early design activities and get

everyone to agree about the design goals. It can also be done without investing significant time into the activity, which is critical in the fast-paced Agile environments. It can also serve as a team building activity, thus supporting another element that contributes to a team's success.

Carol points out that, "Agile doesn't mean you aren't planning. You have to know what the user needs and what your goals are, so you know when you done." Sometimes, this message gets lost as teams put their heads down, work from sprint to sprint, and look at only the next step they have to take. This breakneck speed can even affect designers, as they focus on producing all the work they need to for a sprint and forget to take a breath and seek out customer validation. Because it seems like the work can happen without it, some teams do not document a vision or an overall strategy, but this is a mistake. Even though the work can happen, the outcome may not adequately meet the customer's needs. As Carol's comment illustrates, if you do not know where you are going, then how do you know if you have gotten there? A list of features that need to be in a release is not exactly a plan or a product vision. While the team may be able to execute pieces of work, there is a cost to not having an overall vision, goals, or plans. Throughout the cycle, formally or informally, priorities must be identified and trade-offs made; without a plan or guiding principles to use as a touchstone, these efforts may not result in the most strategic decisions having been made.

Carol likes to use RITE as a usability testing technique. She feels that it works very well and gets everyone really involved. In her experience, you can bang out the prototype, and by the third day of testing, you have something close to being on the right track. This is technique allows the team to really embody the value of having customer collaboration as part of the production cycle, since the design work is based on sessions with the customers. The technique adds value, since it is quick and easy to do, which makes the RITE technique a good fit for an Agile cycle.

Carol also values the use of personas and thinks that the group should at least define the users at the beginning of the project, and the creation of personas is the most appropriate way to do so. She does not feel that the group necessarily needs to define fully fleshed-out personas, just discuss them and get to some initial agreement. The UX team can use that outcome to create draft profiles, which gives the development team something to hang its hat on. She feels that this activity also gets everyone on the same page. Carol points out that if we all are not thinking of a customer in the same way, then we are not really working as a team. Some organizations may be reluctant to have visions and high-level goals in their version of Agile, but there is likely to be little resistance to the creation of personas or user profiles. These can s keep the team working toward a common goal without stepping on any toes and, in most cases, with very little

effort required. Although she will often work off of the draft personas, if more time is available, Carol might flesh them out a bit more. When she does this kind of work for teams, she makes sure that the main customer gets a persona, but a secondary user might get a more lightweight profile. Sometimes, she relies on profiles and flesh them out in more detail during the usability testing. Also, she likes personas posted on the walls. She feels that is important for them to be visible and accessible. Showing them is an important part of the process. One of the most powerful tricks a UX team has is putting its work on the wall to show what it does and spark conversation. It can often be important to create awareness around UX activities to help different functional areas better understand what the UX team does and how it contributes to the product. Carol's idea of posting the personas not only makes sure that everyone is constantly reminded of who their customers are but is a tangible representation of the work done by the UX team.

Now that Carol has spent some time working with Agile, she finds that she likes that it is so focused on users. She recognizes that, with technology changing so fast, designing ahead does not work, even within a six-month project. She sees that you risk not being relevant by the time that you are done with a release. She also likes that an Agile environment makes things transparent. She thinks that, in general, when you are doing waterfall, you go off and it becomes a black box to the rest of the team. This not only inhibits collaboration but reduces the amount of moments to educate the team about good design. The job is not to just produce research or designs; it is to help create an understanding about our customers and what good design is. She sees some challenges and acknowledges that the pace can be frightening. She realizes that, for some UX practitioners, not knowing what you are designing up front can be frightening. She likes deadlines, so it does not bother her. The one piece of advice she would give to a UX person who is about to go Agile is this: Try to become very close to the project managers and influence how UX gets incorporated into the process, especially if it is Kanban. This influence is critical because the UX team is in the best position to identify the work it needs to do and figure out how to work that in to the proposed process. It also affords an opportunity to create visibility around certain key activities and ensure that they are accounted for in the timeline.

Agile is nothing to be afraid of and does not fundamentally change what the UX team needs to do to involve the customers and keep the development team conscious of the customer. It is really the speed at which things are done, the frequency with which they happen, and the amount of communication that occurs around those things that are different. Familiar techniques like research, usability testing, and persona creation can all effectively be put to use in an Agile environment. Additionally, techniques like persona creation and RITE can be leveraged, because they help achieve an outcome at a quicker pace.

Key points

- Research and planning are still necessary to provide a framework and guidance for the team's efforts. These activities can still happen, but UX people might find themselves having to try new techniques, rather than the ones they relied on in a more traditional environment. For more advice on this, check out Carol's presentations on slidehare: "Faster Usability Testing in an Agile World", (Smith, 2011) and "Getting Started with User Research" (Smithe 2010).
- Agile is increasingly common, but it is not so pervasive that there are no longer any software professionals who are unaware of it. Even among those who are familiar with the term, plenty do not know what it really means to "be Agile." This may include team members with whom you work, so be educated on Agile and prepared to educate others. Take any opportunity to express your understanding of Agile and how UX fits into it.

KAYLA BLOCK, PAR SPRINGER MILLER

When I spoke with Kayla, she was a user interface leader at PAR Springer Miller, a company that specializes in software for the hospitality industry. She faces the ever-more-common challenge of geographically distributed teams, often with team members across several time zones. Her team has seven or eight members in Las Vegas and four or five offshore. Since Kayla has a background in psychology, she examined the challenge of working remotely and its impact on team dynamics through that lens. Her intriguing blog post "Geographically Distributed Teams in Agile Software Development" (Block, 2012) can be found on her blog at http://kaylablock.com and offers a different take on what is lost by working remotely. We spoke about these challenges as well as the rest of the work of fitting UX efforts into an Agile environment.

She acknowledges that, as a UX person, working in an Agile manner brings with it a fair number of challenges. She feels that so much of UX design requires collaboration with product management and engineering and the most effective way to do that is with whiteboarding. The quickest way for any team to get on the same page is to sit in a room together, hash out design issues, and take turns grabbing the whiteboard marker. Not only will all have an opportunity to express their opinion, but the team will have a shared understanding of why and how a particular decision was reached. Kayla's team has been creative and tried many things to replace in-person whiteboard sessions. One method has been to use Skype with screen sharing, which allows interactions like reviewing a UI together and being able to say "can you move that a few pixels to the left." It is not as dynamic as working together in the same room, but it is certainly a more rich interaction that emailing back and forth over the changes to a screen. This also seems to be the best way of interacting with remote team

members. The team experimented with aiming a web camera at a whiteboard for a remote colleague but it did not really work it and ended up being a fairly frustrating experience for the person who was not in the room.

The more time zones involved, the more complex things can get. The team was having a daily standup that would occur at 10 at night for the team members in the Kuala Lumpur office. Not only is this a challenge because it forces one location to work outside the rhythm of the normal workday, the remote staff often is taking the call from home, with all of life's potential interruptions, like crying babies in the background. Additionally, time zones that are further away often also come with cultural differences that need to be considered when attempting communication. In addition to the typical challenges of an average conference call, speakers need to be cognizant of language differences and the comfort level of remote teams in speaking up during a call. Kayla also worked with a developer who was based in Los Angeles and found that worked fairly well. Since they were in the same time zone, it was easier to be able to Skype and screen share during the typical business day. It was fairly easy to replace normal collaboration techniques this way, partly because they were dealing with only a single remote person in the same time zone.

In addition to location, Kayla sees that the ability to work collaboratively can be affected by team size. Her ideal team would be closer to three or four rather than seven or eight. Once there are more members on the team, facilitating the additional communication that supports collaboration becomes harder to scale. Skype and screen sharing work best when it is one to one, although it can be done with more people and still be beneficial. Conference calls can start to be challenging when they are too many people or just a few are on the other end. It is very easy for people to talk over each other and very difficult for someone on the phone to participate in a conversation or even completely follow the flow of conversation. When Kayla was working with a product manager (PM) based in Los Angeles, even though they were in the same time zone, the PM often felt frustrated on conference calls because it was hard to participate. The type of role may factor in to the ease of remote integration as well. The Los Angeles–based developer was likely a bit closer to the day-to-day project work and needed to have interactions on specific issues, which could be easily handled by a chat. The product manager likely needed to know and understand higher-level decisions and interact with the team as whole, which requires a meeting with multiple people and is harder to do as the only person not physically in the room.

Scrum itself can be an obstacle; it often feels as if there are so many meetings, that it seems like the team never actually gets to do any work. The rhythm of the Scrum cycle can present some pretty serious challenges in providing a high-quality, consistent UX. Kayla feels that you need to be consistent and methodical in your approach and not just constantly shooting from the hip. It

is important to have a plan and a baseline. Kayla often worked one to three sprints ahead of the development team in what she refers to as *AgileFall*. She is not sure if doing so is in keeping with Agile values and finds that it often did not work. When you work ahead on something slated for a future sprint, things can change and the work either no longer needs to be done or might have changed in the interim. This can be one of the problems with AgileFall, in trying to keep some of the predictability and static planning of waterfall and interleaving it with the fluid nature of Agile, you can end up constantly shooting at a moving target.

One of her favorite tactics, which is very helpful in creating a baseline in a fast-moving environment, is to build a framework like a pattern or design library. Whatever its name, it documents what a combo box is, looks like, and how it behaves. She originally did some of those artifacts in Adobe Illustrator and produced a .pdf that she shared with the team. She would like to try doing a Wiki equivalent, but since that was not something her company was doing; she instead created a SharePoint list with searchable keywords. Each item contains a .png, .xaml code and any additional information she might need to add. This version was recently rolled out to the team and was received well. Not only do other people use it, she finds it to be a helpful reference for herself. She also uses other tools for communicating about designs to the team. Since her team uses Team Foundation Server (TFS) for the project work, she likes being able to add her deliverable directly into TFS. People need not go looking for the deliverables that way; they can just find them with the task or the user story. She produces lightweight specifications and images and simply attaches them to the user story. She does not want people to have to dig through multiple resources to find the information about the design, since it decreases the likelihood that they will actually see the artifacts.

In the first 18 months of the project, the work was done externally by contractors. This had two impacts on the project. The first was that there were often variations in the UI for similar tasks. This tends to happen when the focus is heads down on hitting deadlines and no pattern library is in place. A scrum of scrums can help manage some of that variation, as can a single UX person working on several projects. However, a reference of some sort that is easily accessible is probably the more effective solution. In general, the chaos of Agile and its schedule can make consistency a challenge. Even if you have a Sprint 0, the design is iterative and will continue to evolve throughout the release. If you have no vision to ground the work, then it can easily go in different directions. The other impact that it had was that, with other factors, it contributed, along to a cultural of literalness. When doing work with contractors, you must stay very close to the requirements and the spirit of the contract, especially if deviating means more work than the contractors agreed to do. This created a habit of the business analysts wanting to stick to the exact conditions of satisfaction in

a user story, even when the work started being done in-house. Part of this may have been because the conditions of satisfaction are more vague when the work is done internally, leaving more room for interpretation. When the work is done by the contractors, the business analysts needed to be more specific, often creating functional specifications and workflows.

Kayla says that she would never want to go back to waterfall. Agile may be imperfect, but she feels that it is a better variant. She really appreciates that Agile provides the ability to react to changes in the marketplace and with technology. Warts and all, she loves an Agile environment. She cites an example that not everyone would see as a success. Her company recently had a meeting with a significant customer to discuss a potential large sale and, based on their feedback, the project needed to change direction fairly dramatically and the sprint fell apart as the team adjusted. Kayla rightly sees this as proof that Agile works. It is fantastic to be able to react to a customer request instead of having to say "not in this release." The sprint may be derailed, but when a significant sale is at stake, the development team's quick reaction creates a greater opportunity to close that sale.

Kayla's advice to UX practitioners beginning to work in Agile is to loosen up. If you are used to having every interaction nailed down before the developers get to them, you can let go of that and still get a good or better UX than you might have otherwise. She also recommends being okay with being imperfect and even taking advantage of the new way of working. She says to get your testing into the cycle and iterate away. She has even been able to make the process work for her and gotten buy-in to have user stories for "UX cleanup" as a part of a sprint that occurred before a trade show. She used this to create a punch list of the little cleanup items, set priorities, and get it in to the sprint. This advice can also help when you are unable to get everything done that you need. Kayla was not able to do any usability testing or site visits, for a variety of reasons. In her situation, as with most of us, competing demands for time and resources, so usability is not able to be a priority at all times. At the same time, Kayla recognizes that their product is so far ahead of its competitors that having an ideal design may not always be necessary, because the company's design is the best that there is. This is an interesting point, since as UX professionals, we are used to aiming for an ideal design or user-centered design process. In the end, we often have to compromise these ideals to fit into a messier real world. It might be more practical to look at the "ideal" design with respect to a product's competitive landscape and be as strategic as possible with those compromises. She notes that software is not perfect, so you have to pick where you want your imperfections.

Kayla also wrestles with the tension of producing deliverables that are very clear but lightweight and still work for remote teams. On the one hand, you want

deliverables that are very clear and interactive like Axure. The need for communication is heightened in an Agile environment, since the documentation is so light and needs to be supplemented by a certain amount of immediate interaction to support a common understanding. On the other hand, you do not want to spend time producing documentation that works well only for colocated, collaborative team. She learned that, if the deliverable is sparse, you need to make sure that you had enough verbal communication around it to support that. You cannot expect to email a remote team a quick sketch on a napkin and expect it to walk away with a clear understanding of your design intention. There might not be so much tension between these goals if you can quickly and easily produce interactive prototypes. If creating these artifacts is not time consuming, they can satisfy the need to be lightweight in their nature, since they bear no resemblance to the heavyweight specifications of waterfall. Additionally, their interactive nature provides some of the clarity around design intent that would normally be communicated in face-to-face meetings full of hand gestures and whiteboard sketches.

Working remotely can present additional challenges in the already tricky Agile environment. Recognizing what is lost without constant face-to-face communication and experimenting with different ways of compensating for it is critical to building a cohesive team environment. It may mean stepping away from or supplementing textbook Agile techniques, but if the focus is on people over process, then you are heading in the right direction. It can be hard for remote team members to feel like a part of the team and even harder for them to "feel" the real difference between working in waterfall and working in an Agile way. Any effort that can be made to include them ultimately improves the speed at which the team can work.

Key points

- UX requires intense collaboration no matter what process is in place. With Agile and the increasing reality of remote teams, ther need to maximize the ability to collaborate under any circumstances and make that happen more quickly are more urgent, as the production cycle leaves little time to work out the kinks. Trying different tactics and adjusting as necessary as well as being creative about how to tackle the problem are skills that need to be in every Agile UX practitioner's toolbox.
- Even when it seems like Agile is falling apart; it might actually be working as intended. The sprint schedule may be blown up because you need to respond to a customer, but bear in mind the alternative. In a waterfall environment, regardless of the size of the customer or the severity of the issue, it might take an executive order to shift the release date or the course of the product development. Letting the schedule go where it needs because of customer feedback is a sign that the process is working.

ANONYMOUS 1, AT AN ENTERPRISE SOFTWARE COMPANY

Anonymous 1 is the manager of a small UX team at an enterprise software company. The development team had been Agile for quite some time, but the company started investing in UX a year ago and it was still sorting out how to integrate UX into its process. She is having a good experience with the teams tshe is supports, but finds that not all of the UX people are having the same success and is concerned about making it a more generalizable experience.

The company is Agile, teams have a lot of leeway to choose which method practice. Scrum, Kanban, and Scrumban are all employed, with varying degrees of faithfulness to the methods. This creates some confusion, as she thought her team was practicing Scrum but was recently told that it was in fact using Kanban. This can indicate a need for more consistent training across the teams so that the operational definitions of the methods, their differences, and their relative benefits are clear to everyone in the organization.

Overall, she feels that the process of integrating UX into the process has gone well. The teams have chosen to embed the UX person and make that individual a full-fledged member. Sometimes, the UX people may also serve as an on-call resource for teams that require less support. The UX people are doing well with the tactical UI work and are an active part of the team discussions; they create wireframes for the teams and conduct UX reviews of work. She is still working on building out a bigger, parallel track, where user research, usability testing, and strategic and high-level design would occur. Things are a little more ad hoc as they currently try to conduct research by piggybacking on preplanned events. If a customer visit is scheduled by the PM, UX people tag along. The company's customer advisory board meets several times a year, which allows for customer contact and research. Outside these events, it can be fairly challenging to get customers to participate in usability testing. Given their situation, the UX people are doing a great job of getting engaged with customers and finding out the information they need to inform requirements, user stories, and personas.

Originally, the product manager wanted to own the high-level design. Prior to the UX team becoming part of the process, the product manager had been running Sprint 0, but UX took over that responsibility. She feels that it is really important to have some amount of up-front design, so that the end result is part of a coherent whole and not just different-colored Lego blocks stacked together. This is especially true for large projects, where the seams of the product can start to show if every task is done as a separate piece of work. Not surprisingly, some politics have arisen, with the product managers wanting to be more hands-on and the development team wanting to keep them at a distance. Organizationally, UX is part of development, so these people are

seen as part of the team and do not face the same challenge as the PMs. This is good for the UX team, but ultimately the issue needs to be addressed globally so that requirements and user stories are not caught in a tug of war over influence.

In her role as a manager, she serves as the UX architect and team leader but also supports product teams directly. She feels that she has done well integrating into her product teams but is starting to see that not every UX person is having the same experience. On other teams, the development managers are less experienced and may still be finding their way to managing an Agile process. One UX person is supporting a team that has the only development manager colocated with him; the rest of the development team is in China. Obviously, this presents different challenges in terms of communication and relationship building than when everyone is on-site. For her UX team members that are struggling, it seems to be that they are still working on some of the communication issues she has been able to avoid.

To plan her work, she meets with team leaders every week and discusses the stories that are coming up and what the guidance is from the product manager. If there is consensus that they need to write down what they talked about, then she authors scenarios or a matrix of features vs. needs, then the development manager will scope accordingly. If necessary, she may even work up some hand sketches to facilitate a shared understanding of the design intention. It the work is just a revamp of existing screens, she probably will not generate more than a screenshot with markups to call out the changes. For a large piece of new design work, she generates wireframes for the team. Having a discussion about the work allows her to set expectations with the team around what kind of deliverables or support they can expect from her for a given piece of work. Not only does this reduce potential surprises about what she will deliver, it can help give the team insight into what kind of UX work is necessary for what type of user story.

Once the design solidifies and people provide feedback, the developer is ready to execute it. While she has been fleshing out the design, he has been thinking about how to implement it. Once the design is firm, he completes coding and it is ready for review. The sprint cycle is not much of a driver for her teams; while each sprint is roughly a month, some stories do not fit well into that time frame, and they do not try to force it. While this may not match a textbook description of the methodology, she feels it works and that the team has found a good rhythm.

She thinks her efforts have been effective because the UX people have been willing to integrate with teams. She looks at another team in the organization, where the development people chose to be more closely aligned with the PM and they kept UX external to the team. She sees the UX people having a much

harder time getting their designs implemented. She notes that, because they support multiple projects, the UX people do not go to standups and some of the other time-consuming events. This may not affect relationship building for them, because they are organizationally part of the team and are accepted as such without having to prove their commitment by attending these meetings.

The company continues to refine how UX fits in with the overall process. At the beginning, it used separate UX stories and tasks. But, this made the UX artifacts too separate and removed from the rest of the work. She would find that developers did not know how or where to look for the wireframes the UX people were producing. She thinks that it was helpful to start out this way; it allowed them to get their bearings and learn the process. Now that the development team understands how to include UX into its tasks, there is no need for separate UX stories.

One thing she did that improved the odds of success for the UX team was sitting down with the development managers and creating a diagram of how UX fit in to their Agile process. She then iterated on the diagram with the managers until everyone was happy with it. The diagram was both realistic and a bit aspirational, but in the end, it was something that all parties were comfortable with. Getting this kind of buy-in up front and at the same time managing expectations in a participatory way are tremendously valuable and eliminate many potential misunderstandings down the road.

Key points

- Recognize the variability within an organization. Different development teams within a company have different personalities and logistics, and the approach of the UX person needs to suit each team. She is very successful with her team, but her techniques might not work for other team members, so she must work with them to find the right tactics.
- An ounce of prevention is worth a pound of cure. Collaboratively defining the role of the UX team not only achieves buy-in for the agreed-upon definition and prevents future issues, it also shows in a very tangible way that the UX group is committed to being a partner in the process. It also creates a certain amount of comfort among the other functional areas, because they understand what to expect from UX, once they see the plan for how the UX staff will be working with them.

CHRISTINA YORK, ITHAKA

Christina is a UX manager for ITHAKA, a not-for-profit organization that supports the academic community to adopt and use new technologies. She has a team of two strategists, two interaction designers, and one researcher. She has

been practicing Scrum for about two and a half years and has been able to integrate her team very effectively into an Agile environment.

The initial transition to Agile was a baptism by fire. The company decided to Agile and committed to a full-scale change. It brought in trainers, including Jeff Patton, to do training. Since there was not a lot of guidance at the time on how to integrate UX efforts into the process, the company specifically chose Jeff Patton because of his focus on user experience design. The organization continued its commitment to educating staff members, supporting their efforts to staying current on Agile practices, and encouraging them to be active members of the Agile community. The company even sent one person from each functional area to the Agile Alliance Conference, so that they could learn and bring these insights back to the company. When the transition first began, the UX team was able to get funding to attend a User Interface Engineering (UIE) online seminar featuring Jeff Patton. This was a fairly inexpensive training session and was run virtually, so the whole group could attend. As time went on, the UX people have been able to leverage their successes and get funding for the team to attend other UIE online seminars and to subscribe to online journals—all to help the team keep fresh and find new tools and methods without travel or significant expense.

Despite the organizational support and enthusiasm, there still were bumps along the way. There is a still amount of difficulty in estimating size and velocity. However, since the teams are, in Christina's words, "great at retrospectives," improvement in this area is likely over time. At first, sizing was a development-only effort, and it would just apply a ratio to the QE and UX efforts (e.g., if a task has a development velocity of three, then the QE effort might be one and a half, and UX one). Since a simple ratio to apply to QE and UX efforts relative to coding effort is not always apparent, this did not really work out. Some tasks might have a velocity of three but a QE effort of six, if it requires manual testing or rewriting automated testing. In the same way, that a velocity of three for the development team could be a ten for the UX team, if it requires significant design effort and user validation. The team also experimented with the Spike process to manage the design work. (A "spike" is a time-boxed piece of work, the outcome of which allows the team to define the user story and understand it well enough to provide an estimated effort.) The intention of spikes is to manage unanticipated development work that requires some up-front research and could work well to manage unforeseen design problems but might not work well to handle standard design efforts.

The UX team started with creating a one-year research plan and reviewed and refined it monthly but since evolved to revisiting the plan quarterly. Since the plan is fairly high level and the work is broad in scope, things do not necessarily change often enough to warrant frequent review and adjustment. The strategists

constantly review the backlog and focus on high-level things as they move through the cycle. Additionally, the UX team works to communicate with the business owners who are not colocated. UX travels to them and does design jams to involve them in the design process, get their buy-in for design work, and increase the collaboration. The UX people generally pick one big or important item and get the business owners to contribute in a controlled way. They make these trips monthly, quarterly, or as part of the preparation for an upcoming sprint that has a lot of business logic associated with it. These efforts have been helpful in giving the business owners a forum in which they can get involved in the design and are fairly strategic activities for the UX team to engage in.

One really interesting innovation at ITHAKA is that, instead of having a single product owner, it has a product owner team. This team consists of a UX person, a PM, and a lead senior developer. There is no question that having a cross-functional perspective at this level allows the organization to make more-informed decisions than with a single owner, regardless of the owner's background. It may seem like extra overhead to have three people in a role that could be filled by a single person, but efficiencies are to be gained by doing this. Having multiple disciplines represented may eliminate the need for some team discussions and allow team meetings to focus more on resolving specific issues than hashing through every detail. The product owner team does planning ahead of the sprint and are also part of the scrum team. It breaks the epics into stories and the stories into tasks. It does all this work via sticky notes. It also uses JIRA and Greenhopper to manage the tasks, but it had a problem keeping the sticky notes in sync with the tasks in JIRA, so there is room for refinement. The interaction designer works with the owner team to identify the big issues, negotiate the timing, and distribute the effort. The designer is a member of the product owner team so he or she has insight into the decisions as well as the UX work. The process relies heavily on the UX team being highly communicative. While it is working very well for the organization, Christina thinks that it might not scale well, since it is so dependent on the communication abilities of the team members. Scalability is a very legitimate concern, especially in situations where the UX team is large or growing at a fast rate. However, if this skill is kept in mind during the hiring process or cultivated with existing staff, there may not be any problems extending the process currently in place.

For the most part, the Agile project teams are a colocated. However, the interaction person sits there only part time, because there is also a central UX team location in an open area, so that the members have a space and opportunity to work together as a design team. Spending time together as a unit allows them to create what Christina describes as "a hive mentality." The ability to sit away from the Agile team is also helpful, because the interaction designers also support non-Agile projects and need space to get away from the nonstop

action of an Agile team. At first, the UX team tried to set up office hours but got rid of that. There is a facilities concern around having two spaces, so the group created a general rule that chickens (especially if they support multiple projects) can have two spots, but pigs cannot. The company is planning construction of a new space and planning to still have privacy cubes for phone calls, offices for ad hoc use, and lockers for personal items. A lot of investment by the company and the facilities organization made sure that these changes really work for everyone. Christina sees this as a real show of commitment by the organization to supporting an Agile environment. She sees their ability to continue to deliver as proof of the return on investment into this setup. It also helps that the QA organization really supported the idea of colocation and kept metrics to show its value. The organization was able to point to the number of bugs, the time to fix them, and the like as evidence that the colocation supports greater efficiency. It started making these metrics visible to customers, who in turn lobbied to support whatever was contributing to the metrics.

The team also did participatory design with customers and worked to define their mental models. When a customer was in town, they did F.I.D.O (Freehand Interaction Design Offline, see Tedesco, Chadwick-Dias, and Tullis, n.d., http://legacy.bentley.edu/events/agingbydesign2004/presentations/tedesco_chadwickdias_tullis_fido.pdf for more information about this technique) with the customer, discussed the results, and presented their findings. Previously, there was some tension between UX and the business owners, but using participatory design methods essentially made the customers the advocates for the designs and the publishers would negotiate on behalf of the UX and request UX involvement.

Since UX team members straddle both waterfall and Agile projects, it is left to the individual to manage a personal schedule and negotiate his or her time. The members tend to go to all the project meetings, the sprint planning, the retrospectives, and the standups. However, if they have a conflict or need to drop something, they will go to the team and say, "I have eight hours for your project this week, how do you want me to spend it?" The team decides if the time is better spent attending standups or skipping some other team event. Often, it will frontload the UX issues in a planning meeting so that the designer can leave once the conversation turns to development-centric issues. This is evidence of the trusting relationship that the UX people have been able to build with their teams. They have shown their commitment by constantly showing up, and they allow the team to help determine where the trade-offs need to occur when they cannot make it to everything.

The healthy communication that the teams established depends on having a stabilized team. One designer stays with a given team and that team is kept together although the projects that they work on might change. This improves

retention and breaks down knowledge silos. This also allows each functional area to focus on hiring based on skills, because a process is in place to train people on the product knowledge. This speaks to the fact that the most important factor in going Agile is the people. Christina notices that there is a tendency to focus on the process, but there are so many different flavors of Agile and UX that the people end up being what really makes it work. This is as true for her as it is for every case study in this chapter. She points out that "People over Process" in the Agile Manifesto is right. Probably, the most frequently referenced part of the Agile Manifesto when talking to UX practitioners about how they are integrating into Agile is this concept. A team that does a really rigid, by-the-book adaptation might not incorporate Agile very well at all. A willingness to accommodate UX work shows a predisposition to the idea of "People over Process."

She feels an advantage to being Agile is that there are more shared decisions. She thinks that UX teams often stand alone because they want to. It can be easier to be the bearer of bad news and messages like "you didn't do it the way I wanted" than to put the effort into working collaboratively. This us vs. them mentality can often be made worse in waterfall environments, where the UX team ends up calling out where its designs were not implemented to specification. Agile does not really allow that kind of separation and distance, since it demands that everyone work together. She prefers solving problems together and sees that as the path to innovation. She says it is not about being right, it is about understanding. The team has no design divas; it is really a marketplace of ideas. Because of this, Agile is not for everyone. Christina can see how someone who is awesome at waterfall might not fit into Agile. Indeed, for many people, the transition can be a challenge, as it can require them to change the way they worked for many years.

When they went Agile, Christina added a section to the weekly UX meeting called "F*ckups and Wins" at the end. At first, only Christina would volunteer her mistakes, then it became a joke on the team, and finally other people started participating. Christina was really inspired by the UPA Munich presentation by Steve Portigal (2010), "Culture: You're Soaking in It" (Portigal's slides on this topic can be found at www.slideshare.net/steveportigal/culture-youre-soaking-in-it), and focused on creating a specific culture for her team.

Christina was conscientious in her approach to creating an Agile UX team and did a lot of research to inform her decisions. As a result, she could draw on a lot of inspiration as she navigated the transition. Just as important, she modeled the values that she wants to see and is able to influence change without being preachy. She does it from a leadership position, but her techniques can be equally effective when used by an individual contributor and can be used to change the style of dialog in planning meetings or retrospectives.

Key points

- Manage the culture of the UX team. Being conscious and deliberate about creating a culture of collaboration begins with your own team. Although Christina worked within an organization that was able to conduct effective retrospectives, her team would have been well-equipped to model this behavior and spread it throughout any organization. Embodying the culture you would like to see allows the team to model the right behavior and encourages others to do the same. It also creates a more uniform expectation from people about what it is like to work with members of the UX team. They may not know person A, but they know the UX culture, so they are better prepared for what to expect.
- Grow your Agile skills and your UX skills. While it is important to keep current on new Agile trends, especially as they relate to design, it is also critical to stay current on design and usability techniques and methods. Turning this into a team activity with design workshops and skill sessions not only allows the entire team to benefit from the knowledge but creates a more cohesive team.
- People over process. In the early days of a transition, it is easy to get caught up in the minutiae of the process. Some of detail is important, but the people and the culture end up having the greatest effect on the success of the effort. Learn what you need to about the process, but focus your energy and effort on coaching the people.

ANONYMOUS 2, A LARGE DESKTOP SOFTWARE COMPANY

This is a story about an adoption of Agile that is still struggling to find its way. In no way is it a case study about a complete failure. In fact, the company did many of the right things and even invested in training. In certain areas of the development process where Agile is working well. However, in other places, the organization is not really engaging in Agile values. This case probably resonates with many people for whom Agile is a mixed experience , neither an overwhelming success nor an abject failure. The UX person has chosen to remain anonymous to be as candid as possible.

Anonymous 2 has been working with Agile since 2006, after previously doing "heavy waterfall." When doing waterfall, she was writing huge specifications. After the move to Scrum, the company brought in a local person as an Agile coach. The organization also provided development training and product owner training. This taught people about writing user stories and being more user focused., which was not something they had previously been doing. These are certainly promising first steps on the Agile path and consistent with what many of the successful adopters do.

The motivation for the change was that the time between releases was too long, around two years. This is a legitimate motivation, as such a kind of gap in releases is not typical anymore, even for large software packages. Customers have different expectations, and even if they are not necessarily able to adopt more frequent releases, it allows the company to see the direction in which it is heading. Waiting upwards of two years can not only cause the company to miss important opportunities, it can also create some anxiety in the marketplace. Choosing Agile to address this particular issue is a sign that the organization had identified that its current process is not working, and it needs to make a change. Her company started out cautiously by beginning the efforts with pilot projects to test Agile. To minimize the risk, it used an independent project as the environment for the first test. She was happy to see some of the waterfall elements go away, especially the huge specifications that would take so long to go through approval that they became obsolete. They also did not really have an established UX process under waterfall, so the UX team saw this change as an opportunity to define the process. Even teams with years' worth of legacy at a company often have no formal process in place and a move to Agile can be a chance to put in place a thoughtfully designed UX process. Her team came to the table with a proposal about the UX process, suggesting parallel sprints. The members had done their research and thought that this would be the best approach for them.

Overall, the team has been pretty successful in integrating more frequent user feedback than it previously was able to do. The original plan was to have what she describes as "revolving door" testing. The goal was for every scrum team to test at the same time every few sprints. That way, the team would have a predictable opportunity for feedback and would know that testing would occur at the beginning of every other sprint. This is something many teams, especially teams that support websites, are able to do; and it seems like an ideal approach. The team found that it was not able to achieve that level of predictability and instead has been doing frequent ad hoc user feedback sessions. It is not the ideal it was aiming for, but tit gets feedback much more often and earlier in the cycle than before. During the time with waterfall, it would usually not get user feedback until a month before the quality testing phase, toward the end of the release cycle. The Scrum process definitely helped create support for including more feedback into the cycle, because it is an accepted part of Agile. The team is also able to show more features earlier in the process, because he sprint cycles produce workable features early on.

Something has not gone quite as planned is that the UX team has not been able to get on a parallel track and work ahead of the development team. The members found that they have been pulled into projects once the projects already have started, and it is very hard to get ahead if you start out playing catch up. This is a situation that UX teams often find themselves in during the first

time out with Agile. In the rush to get all the work off the ground, it may not always be clear when and where UX fits in, and the UX team may often is brought in after the fact. When the work is already underway and the UX person required to produce designs yesterday can be an impossible position to be in and very difficult to change as the sprint cycles barrel ahead. Regardless of whether someone are working in parallel sprints or within sprints, it is never good to be behind. It may not always be possible to avoid these situations, but it is critical to minimize their occurrence. She notes that defining the big picture ahead of time is great, but that does not always work. She feels you can tell when it does not work, because the work is affected. The challenge with planning ahead is to plan to a finer level of granularity than is appropriate, given the level of certainty, or uncertainty really, around some items. You can also plan work that never materializes, which is fine if you have not made a significant investment and can easily change direction. As she points out, the real indication that time has not been well spent is if the work is compromised or derailed.

The company has over 30 Scrum teams and a UX team of six people to support them. As one might expect, the high-priority projects get attention and UX support. Any given UX person may have to juggle a few projects. She had once been assigned as a dedicated resource to a project, and while she liked it, this is a rare occurrence. On some projects of a sufficiently high priority, it was possible to give that focused attention, but on others, the UX team functioned more as consultants. Working like this can make it even harder for the UX team to fully engage in projects and figure out how to master their new process. So much of Agile is about building trust and transparency and being really involved with the day-to-day work of the team. If the UX people cannot do that, it might mean that they are taking on too much work. It may be hard to push back, especially since no one likes to say no to project work. Especially when the team is orienting to a new way of working, there needs to be some space to take a pause and reflect on how things are going and where changes need to be made. Supporting too many projects and working at a very fast pace can prevent the necessary self-reflection from happening.

The UX team was excited to make the move to Scrum, but the transition turned out not to be so easy for everyone. Chunking the working into pieces was a new skill for some, but everyone on the team got through it. The team members also started using new tools, like Balsamiq for sketching designs. They were used to using Photoshop, but switching to Balsamiq made it easier to churn out design ideas quickly. Since the learning curve for Balsamiq is fairly low, especially relative to Photoshop, this is an easy skill to acquire and can be almost as natural for sketching as working with pencil and paper. The team also started using Axure because it was having trouble really showing complex interactions, especially for controls like trees and tables. She found that it is a good tool. The members got great feedback, and their PM is impressed with the quality of the

prototype they can produce quickly. They have also used it a few times to produce a prototype for usability testing. They do testing in their own lab and also go on site with customers or at conferences.

The UX team members have a solid idea of where they want to go and just need to get there. However, she feels that cultural issues still need improvement. She thinks that they need an organization more focused on user experience. While the company talks about user experience quite a bit, things have not yet evolved to the point where UX is such a priority that the company would be willing to hold the release because of a UX issue. In very few companies could a UX issue hold up a release. Many of us have heard leadership say that UX is important but not necessarily follow through with actions that show this commitment to be true. This can occur for so many reasons. Sometimes, the organization wants the customers to think it is concerned with improving their experience but is not entirely prepared to spend time and resources on figuring out how best to do this. What often can help resolve this issue is creating visibility around UX activities, their benefits, and how little investment is required to effect significant change in the products. In other situations, leadership is genuinely interested in making the UX of the product better, but it does not know what to do so or how to get there. When facing this dynamic, the UX team can gain traction by doing some education around UX methods and having a specific proposal to pitch to leadership. She also finds that both the company and the release cycle are still very feature driven. This means that if a trade-off needs to be made for any reason, it is likely that UX work will be dropped in deference to features.

Another issue that has arisen since the move to Agile is that the organization has become very big on metrics. Development is completely focused on their velocity, and that function often is very averse to changing the UI because it thinks that would make it look like their velocity was decreasing. As often happens when metrics become a primary driver, people work toward the metric rather than the spirit of the metric. Part of the problem here is that, if reworking the UI is having a negative impact on the metrics, the work is not being tracked properly. Refactoring work, which is supported by Agile, is still work and should show up positively in the metrics. Another frequent problem is that the project team would come to the end of the sprint and end up with stories not having being done. This in itself is a very big red flag, especially if it is a recurring issue. It means that the team is either overcommitting to the amount of work that it can do or churning too much over the work during the sprint cycle. The fact that it continues to happen is an indication that the team is not raising or resolving their issues during retrospectives.

In addition, she has seen a struggle between boundaries of product owner and the UX team. The views have a lot of natural overlap, but the product owner is

a single person. This tension is not uncommon, especially if the product owner defined many of the UI elements in the past and is reluctant to give that over to the UX team. It can also simply be an issue of the product owner feeling as though, in defining the interfaces, the UX people are stepping too far into the territory of "owning" the product. She has seen a lot of arguments over who had the final say on a disagreement. This resulted in a rule that the product owner has the final say, but this has not always worked out well for the UX team. The UX team members feel they do not have as much of a voice as they would like. This can be very frustrating for a UX team working hard to adapt to a new process and work at a faster pace, only to see that if the members disagree with how something is being done, it is unlikely that their viewpoint will be supported.

Overall, it still feels really waterfall and not very Agile to her. She thinks if development could see the usability sessions and the users actually using the software, this might help. Since so much testing is done on site with the customer, only one or two developers can come with them. Involving the development organization more in feedback sessions might be quite effective, as it seems that much of the needed changes are at the team level. She finds hope in the fact that smaller projects that get big wins tend to influence change at their organization, and this represents an opportunity for her team. She also thinks that it is good to know where you are and where you want to go, so you can figure out how to get there. This means identifying a goal for the team and how it wants to support the organization then working toward it. She recommends this as a first step to moving to an Agile environment, the plan can be revisited, but it is important to at least start with a specific vision and end state to guide choices along the way. Another piece of advice she gives to UX teams is that they need to recognize that you cannot boil the ocean in a single day. Change is an ongoing process, and you need to be patient but proactive as your team switches gears. She definitely sees benefits to working in an Agile way. In the past, shehad to justify everything and mo longer has to. She can see progress, even if they are far from where they want to be.

As with most organizations, there is certainly room for the organization to mature in its Agile application, but the UX team has been able to make the most out of the transition. Members can streamline their work process, and they discovered new tools like Balsamiq to support their efforts. They see many of the challenges as opportunities and approach these daunting changes with a positive attitude. The team is also able to be tremendously effective, despite their relatively small size and fairly extensive workload. This helps it build credibility and support for its efforts. The UX team has certainly not failed to integrate into the Agile process, it is simply in a situation where the organization needs to continue to evolve and find a way to increase collaboration.

Key points

- If your organization still has a lot of room to grow into a mature adoption of Agile, focus on the small wins and the places where the UX team is able to gain traction. Change does not always happen overnight and identifying the small successes can not only keep up morale but help affect change elsewhere. Be strategic, and target the projects where you anticipate you will be successful; leverage these to get more support, more resources, or whatever else is needed by the team.
- When people hide behind the Agile process, work around it. Agile is not about metrics and hiding incomplete user stories. That does not mean that Agile teams do not often get caught up in things like this because of organizational pressure from above. If you happen to be in a situation where this occurring, taking the issue head on may not be the path to success. (Although, if it seems like it is possible to resolve it, by all means go for it!) Instead, work with the development team to figure out a way to track the velocity in a way that supports reworking the UI while meeting the metrics.

AUSTIN GOVELLA, AVANANDE

Austin is an experience design manager at Avanande, a provider of business technology and managed services. He has practiced Agile for more than five years in his current role and at a previous employer. He has a very well-thought-out philosophy on Lean UX and has been very successful in adapting his approach to situation needs. While still executing on the deliverables for the release, he focuses much of his energy on strategy and gives priority to his time spent on strategic activities.

When discussing his experience with Agile, the first thing he points out is that no two Agile teams are ever the same. This statement comes from Austin's experience of having worked with different teams and at different companies, but it is generalizable to the broader Agile world and is worth calling out, because it does not got expressed often enough. Many of us will work on Agile projects at more than one company or with different teams within the same company. If you approach each project as if it is the same, you may not be as successful as if you adjust to fit the team at hand. This is not unique to Agile, it is something that UX practitioners often do to be effective. However, since Agile methods depend so much on the project team being as high functioning as possible and able to communicate well enough to promote teamwork, it is especially important.

Austin describes himself as practicing a form of modified Scrum. He has learned from experience that, if he is on more than one project, he cannot go to

all the meetings for each project. It is really important to know your limits when working in Agile. As designers, we always want to be able to help produce as many good designs as possible. Unfortunately, trying to support multiple projects and attend all of the meetings compromises our bandwidth. Since the process works in cycles, measured in weeks and not months (or years), it becomes apparent very quickly when you are in over your head. He feels that the key to Agile is really committing to a sprint's worth of work, delivering on it, and maintaining constant communication. He thinks that, compared to waterfall, it is a big commitment. Instead of agreeing to a set of work that may, or may not, take place over a long period of time, the team is agrees to do very specific things in a short amount of time. Working this way means so much more accountability. There is also an implicit agreement that the team will work together, which is never a factor addressed in a waterfall cycle.

He has a three-month release cycle and lots of meetings, although maybe not daily ones. He finds that the urgency of everything increases in an Agile environment. This is certainly true since the most time you have to turn around a single task might be three weeks. On the project he was working on at the time of our interview, his team built up a backlog and was doing preplanning activities. Austin was able to write some of the stories, which is very helpful, as he can make sure that they are written with the appropriate customer focus. The planning meeting that they is more high level; they do not hash out the tasks during this event but rather do that outside the meeting. This serves to keep the planning meetings somewhat focused and able to finish in a reasonable amount of time.

The development team for the project he is working on is in Buenos Aires, and he estimates that at least 50 percent of the work that he did the last few years has been done remotely. He uses Campfire, a persistent chat room, where he can post files. He and his visual designer are on that site all day, working together virtually. (The visual designer is in the remote location.) They spend time commenting and critiquing each other's work, and the site is a place for him and the designer to work collaboratively. It is a space where the UX team can talk openly about the design issues. It works better than any kind of instant messaging, which is too immediate to use across time zones, and is better than email because it has a dedicated and focused purpose. Austin is the architect and constructs wireframes, the visual designer is focused on the front end, and together they produce the overall design.

Prior to the kickoff of his most recent project, he put together a strategy for the architecture and how the pieces fit together. He likes doing this, because having that kind of document and presenting it to the team provides a common language and framework from which everyone can work. This up-front work saves time and reduces the number of course corrections needed later. Business

needs and user needs are embedded in the document, and it is a compact way to communicate all these things. He describes the strategy document as "just a description of the Lego pieces," in essence saying, "these are the things we'll have." He posted the document and did all of this work ahead of staffing the development team, so the document really drove the definition of the project. Essentially, he took advantage of the downtime before the project really ramped up to produce the document and get buy-in for it. Not only did this lay the groundwork for the project, but it is much easier to do before the work is underway.

He developed two personas for his current project. He presented it generally and casually, and since there was no pushback, they are now they are part of the process. Every story in the backlog is built around these personas. Having a high-level artifact, such as personas, and making sure that it feeds into the more detailed user stories are excellent tactics for creating a coherent deliverable that fits together in a consistent way. He also uses the story as a basis for the wireframes and references it. Each wireframe has a description that contains a mini-story, which includes for whom it is targeted and how it will be used. This fleshes out the wireframe and gives it enough context that the other functional area can have a solid understanding of not only the design intent but its rationale. He was able to do a bunch of user interviews for competitive research. He also reread *Designing Social Interfaces* (Crumlish and Malone, 2009) to prepare for working on the designs. He did workshops with business and technical stakeholders to get insight into their expectations and customer needs. Business stakeholders said that they wanted him to produce a list or a backlog, but they got a strategy presentation and document instead, which was more useful in setting a clear direction for the project work. He notes that UX practitioners sometimes have to provide what is needed and not necessarily what is asked for.

There are no explicit "stories" for the wireframes and visual designs. Since he wrote the stories, he feels that there is an inherent relationship between them and the design work, so it is not necessary to create separate items. He and the designer do have tasks for the design work, but Austin manages those independent of the rest of the team. His velocity is not tracked because he straddles different projects and tracking his work would provide no insight into how the team is progressing. For deliverables, the UX person is actually producing the front end .html and handing that off to the development team. Austin also uses wireframes to illustrate workflows and creates site maps. He finds that the site maps are usually a great communication tool because everyone can understand them. Those always get bang for the buck relative to other efforts.

When it comes to user testing and validation, he finds that he can just schedule testing, and it happens. He was also planning to get the original interviewees'

reactions to the wireframes. After the first release, he will run through basic tasks and get feedback. In a typical sprint, the team spends a half day planning, three and a half to four days of coding, three to four days quality testing, then the team has a retrospective. The first release happens after three sprints, and that pattern going continues two or three cycles. For tools, the team is considering using TeamForge, but it had not deployed it at the time of our interview. He is still using his original Excel document to manage the backlog.

Austin acknowledges that Agile heightens all the communication issues any team might have. In a waterfall environment, communication occurs at a slow rate and problems may go unnoticed or get lost in the noise. However, a problem, such as a team member who prefers to work in isolation and is uncomfortable working cooperatively, will reveal itself very quickly. The same is true for other dysfunctions that might exist. He also recognizes that "it is easy for someone to hop on the backlog train and drive it wherever they want to." He emphasizes that Agile itself is neither good nor bad, but it works best if everyone has a shared vision.

He also likes to track the improvement in UX across releases. He incorporates this into the process by including, in every story comparison criterion, that this has to be as good as, better than, not as good but useable than before or than a competitive feature. He finds that the developers are not very interested in these comparisons. They do not tend to participate in this activity, but it is very effective to use when communicating with the product owners. Executives love it, and charts and graphs can illustrate the progress or the lack of it. Sometimes, there is no real world comparison; then he uses the wireframe or prototype as the baseline. Austin notes that you can also use something like a "UX health check," which is inspired by Jon Innes's (2011) UXI matrix (see www.slideshare.net/balancedteam/uxi-matrix-jon-innes for more information on this matrix and its application). Both techniques are ways to analyze the overall robustness of the design and can identify areas for improvement.

One thing Austin learned along the way is that it is not designers or developers who create products, it is the organization as a whole. He points out that every project problem is a team problem. If someone does not want to do testing it is not because the person is lazy, but because the value of doing it is not clear to them; and it is the responsibility of the UX team to show its value. It is important to realize that, if you want to change the way things work, you have to influence the organization. (See his presentation on this topic on Slideshare; Govella, n.d.: www.slideshare.net/austingovella.) This is really an important consideration when deciding what strategy to use with the team on a given project. The role of the UX person is not to simply churn out sketches but contribute to a good customer experience in every way possible. This means helping to define the strategy, the personas, and the user stories as much as possible. It also means considering how

to contribute to a high-functioning team and promote an understanding on the team of the users and how to meet their needs through design.

He also advises to make your process visible. This can be the most effective way to create understanding of and support for your work. If you do a project well, then people will copy your process and you will influence the culture. He suggests putting your work on walls and engaging in collaborative design. In a meeting he had once, everyone sketched ideas for a page. With nine people in the room, they wound up with eight pictures (one person abstained). This kind of activity can produce an appreciation for and awareness of the design process, as well as generate multiple design solutions very quickly. It is also an effective way to propagate UX techniques, since part of the goal should be to help give developers the skills to make the best design decisions they can. Austin also advises that focusing on strategy is more important than the production of wireframes. He feels that, if you can put an idea in to the team's head, it will infect everything the team does. The wireframes can be implemented in ways over which you have no control, so it is more important to focus on influencing the mindset. This is a better investment of UX effort and will contribute to a better outcome. He also recommends that UX practitioners engaging in Agile read the Balance Team (n.d.) blog (www.balancedteam.org/) and Jeff Gothelf's (n.d.) blog www.jeffgothelf.com/blog) to learn more about Lean UX. These blogs tend to focus on strategic efforts and how the UX team can work as efficiently as possible, all very important when working in an Agile environment.

When starting with Agile, it can be very easy to get hung up on or caught up in all the details of the process and lose sight of the big picture. It can certainly be easy to get fixated on the progress of burn-down charts, obsess over the details of the process, and keep your eyes focused exclusively on your next piece of work. However, if you wait until it is time to generate deliverables to try to guide or influence the process, your efforts will be much less impactful. Austin's story serves as a good reminder that defining the way people speak about the work by using personas and driving the overall strategy of the effort is the most effective way to affect the culture and the design. This is an especially valuable message for those of us with limited UX resources.

Key points

- Fancy tools are not required. The fact that Austin is managing his backlog with an Excel spreadsheet shows that the tools are much less important than the intention. You cannot judge the maturity of an Agile implementation by the tools the team uses. And, if the tool of choice is getting in the way of the team's work, consider keeping things simple. Plenty of fancy tools are available to Agile teams, and many of them provide great benefit, but in the end, they are just tools. If the team has problems, they need to be solved in a way other than by purchasing new software.

- While it is easy to get caught up in the deliverables, the wireframes, and the sketches, the most influential work is done on the front end. Being a part of the story definition process has the most bang for the buck, especially since the design tasks will fall out of that. Austin makes a very good point that you cannot control the implementation of the wireframes. However, if you can help drive (or win) the user story definitions, you may be able to worry less about some of the smaller details.

JOSH O'CONNOR, NATIONAL COUNCIL FOR THE BLIND, IRELAND

Josh is a senior accessibility consultant at National Council for the Blind Ireland's Center for Inclusive Technology and often works as a consultant to evaluate the accessibility of a client's website. He does usability testing with visually impaired people on websites and public domain sites. From the vantage point of a consultant, he has the opportunity to view many organizations and the way they approach Agile.

Josh is a big fan of Agile, although he acknowledges that, on paper, it can look like more work than traditional methods and it needs a champion within the development organization for it to be successfully adopted. Plenty of executives are excited by the idea that it just makes things happen faster, but those who are closer to the production of software worry about the additional overhead and process changes required to go Agile. He thinks the products produced via Agile methods tend to be better because of the iteration and customer involvement. However, he has seen quite a few environments that claim to be Agile, but once you scratch the surface, it is clear that this is just lip service. This is a fairly common phenomenon, since there are places that want to be Agile but have not yet figured out what that really means or how to get there. They may co-opt a given method in a superficial way and leave it at that. Suffice it to say, these organizations are likely to bear a stronger resemblance to waterfall processes than to Agile. Organizations that have taken on Agile in such a shallow way are likely to have the same approach to accessibility.

Since he is brought in as a consultant and companies are not obligated to implement his advice, he is often in the position of having to negotiate with cultures that are closed to complaints about accessibility. In his experience, the degree to which a team is proactive about accessibility issues has to do with the corporate culture's awareness and its history with supporting people with disabilities. While Josh is speaking specifically to accessibility, this is can be applied to broader user experience issues. If the company recognizes the value in improving these things, either from a revenue standpoint or simply because it is the right thing to do, there are fewer battles in which to engage. Since this is not always the case, he has

found that the best way to get support and buy-in for accessibility design ideas is to show the developers cool solutions. Everyone likes cool things and being shown a better way to do things. He advises being really aware of what is new and how the technologies work. This goes hand in hand with the idea that UX people should not just point out all the flaws in the work. He recommends reinforcing what is good and doing it in a really, really supportive way. When you say what is wrong, have a solution, a way to fix it. This advice can be applied not only to design issues but also Agile adoption. Acknowledge that the team has done good work, but that it could be better if *x*, *y*, or *z* is done. Make the team your partners in producing a great product. Stay aware of the current technologies and be able to show the development team the cool things hey could be doing and help them make the change possible.

Josh has an interesting perspective as a consultant, because he comes into a culture that is already defined and he is not in a position to directly influence the organization. However, because he is brought in as an expert in accessibility and as an outsider who does not carry the same baggage as a team member, he is remarkably positioned to influence the people with whom he comes in contact. Focusing on solutions and showing developers a cool way to do things is a technique that any UX person could benefit from using.

Key point

- Just because an organization says itis Agile does not mean it really is. If an organization is only giving lip service to the method rather than genuinely engaged, you end up with the worst kind of process. When working in this situation, it is best to try to get what you can implemented, model good Agile behavior as much as possible, and focus on the small wins.

ADRIAN HOWARD, QUIETSTARS

Adrian is a consultant with Quietstars, in the United Kingdom, and specializes in Agile. His thoughts on Agile show up in many forums, at many conferences, and a quick overview of some of his answers on "how to be Agile" can be found online (Howard, n.d.; to see what Adrian has to say, go to www.quora.com/Adrian-Howard/answers/Agile-UX).

He recalls first reading about Extreme Programming and thinking that it would never work. He came from an architecture background and had been taught that you design your system first. The idea of designing as you go along ran opposite to everything he had learned. He decided to give it a try despite this and thinks that the appeal of working this way is that the process is iterative,

feedback driven, and teamwork oriented. Once he started working in an Agile fashion, he felt the process just worked better.

He feels that there is often a divide between the coders and the "artists"/ designers, which is harmful to the product. Having a pretty skin and an ugly inside does not work to, they have to connect. After all, the gorgeous graphical treatment looks much less appealing to a user who is completely frustrated by the quality or performance of the application. Agile can help bridge this divide by increasing communication and encouraging more conversations. Rather than engage in a more classic "us vs. them" dynamic, the team needs to work as a team and collaborate.

He feels Agile makes more explicit the need to do the design work throughout the project. There may have been a latent need for this in waterfall, but it may have only occasionally actually occurred. However, with Agile, the need for iterative design is the basis of making the process work and is, in fact, required by Agile methods. He points out that the design problems do not occur only before you build and Agile supports that reality Too often in traditional processes, the issues are identified after the code is written, and it is almost always too late in the cycle to change anything but the most catastrophic of issues.. He acknowledges that Agile can be more overwhelming, but sometimes, you end up doing less work, because you can get feedback more quickly and spare yourself from heading in the wrong direction or spending time on designs that will never see the light of day. In more traditional environments, you might never get feedback that the design is flawed until late-cycle usability, when only minor design refinements are feasible. If you get significant feedback in one sprint, you can completely rework the design and validate it in the next round. As chaotic as the pace may feel, efficiencies are to be gained by working this way.

He likes Agile, because what he likes best about design is not the creation of wireframes but seeing the design in the user's hands and seeing it going well.(Although he acknowledges that, sometimes, you watch them curse the application out of frustration, but that provides important feedback as well.) He thinks that getting customers involved with the design early on happens more often in Agile environments. The process is more feedback oriented and customer focused and, done right, can be better positioned to achieve a good user experience. Certainly, Agile has built-in support for involving the customer, so if you previously found resistance on this front, it is likely that you will no longer have that problem. He also finds less resistance to Agile than there used to be and wonders if the people who are anti-Agile are really criticizing Agile itself or are reacting to someone's bad implementation of Agile. It can be hard to divorce a single bad experience from the process itself. And, since many people have only the experience of a single company or team, it can be

easy to write the whole thing off as a disaster if that is all a person known. It would also seem that more and more people are having positive experiences with the process, if the resistance to it is going away. Even for those working with imperfect adaptions, it is clear to see that there are benefits. Few UX people will be sad to say goodbye to the large specifications they were responsible for writing.

When talking with Adrian about Agile and UX, it seems like such an obvious fit that it is almost surprising that anyone struggles to make the two work well together. Obviously, many of us do struggle to be Agile, but Adrian's words are a good reminder that the values of Agile and UX are a natural pairing. If there are tensions or resistance, it has to do with people and the dynamics of the team. The processes themselves are a naturally good fit.

Key points

- Agile is very immediate and can seem like more work than traditional waterfall. In reality, because of the speed, the whole process tends to have less waste; and if feedback is a big part of the cycle, it might be possible to get to the best design quicker.
- Up-front design has been the norm because of the prevalence of waterfall, but it is not the natural state of things. While it may seem exhausting to commit to constant iteration formally, it tends to be what happens anyway. But, because Agile recognizes this, it is more likely that the iterations occur under the guidance of a designer rather that in isolation in a developer's cube.

ELISA MILLER, SENIOR USER EXPERIENCE ENGINEER GE HEALTHCARE

Elisa is a senior user experience engineer; in her role she supports multiple teams and serves as the sole researcher in an Agile environment. When I first heard about her situation, I fully expected to hear a story of frustration and difficulty. After all, working research into a tight sprint cycle can be a challenge for usability people who support only one project; I could not imagine how difficult it would be for someone juggling multiple teams. What I actually heard was a story of someone who embraced the challenges and the pace of Agile and made it work for her, managing to conduct frequent usability tests.

Her teams are composed mostly of systems engineers, who are a blend of user interface designers and business analysts. They create the requirements, define the user interface, and identify what graphic elements they will need for the project. The UX team is made up of 17n people who work on multiple products. Three are managers and the rest are designers who also do design research.

She is the only one doing usability testing on her product. The other UX staff members work very closely with the systems engineers on a project and with Elisa.

She supports two main product areas, each with multiple projects. One product represents the features associated with electronic medical records (EMR) and the other is practice management software. A third product area that she supports addresses scheduling and registration, which allows users to manage the use of rooms and equipment as well as streamline the patient registration process. Those modules are currently stable, so most of her focus has been on the EMR and practice management modules. She alternates one sprint on EMR efforts and the next on practice management software.

When we had our discussion, Elisa and her teams were about a year into their Agile journey, far enough along to have established a good rhythm in some places but early enough that they were still trying new things. They were about to embark on an eight-week project with Elisa supporting multiple teams and having a one-week Sprint 0. The previous rounds of sprints were two weeks long, and Elisa had sorted out a way to divide her time. The sprint reviews for each team occurred every second Thursday. The company's sprint reviews do not serve as a means of communicating to stakeholders but rather function as an integration meeting for all the teams working on the product.

On Friday, she would meet with the systems engineers and the product managers of a team that she would be supporting in the following sprint to discuss their needs and concerns and identify the areas for testing. On Monday, she would create a test plan. She would spend the rest of the week recruiting and walking through the tasks on her own, then with other people to make sure that the test plan was solid. She would also make sure that she had access to a virtual machine with the right product build, including the set of features, needed in the test environment. On Monday and Tuesday of the following week, she would run the usability sessions with customers. Her time on Wednesday and Thursday would be spent writing up the results, and she would deliver the findings on Friday.

In the Friday meetings with the product owner and the systems engineer, she will present a set of slides to summarize the usability testing results. The presentation will give an overview, the top issues, some screenshots, and recommendations. She sometimes gives demographic information about the participants to provide context and always includes user quotes, so that the audience can hear the voice of the customer. While she invites the team to attend the session and makes the notes from and recordings of them easily available via OneNote, it is not always possible for the team to attend sessions, so hearing the actual voice of the customer during the presentation is very valuable. All in all,

she will present 10 or 12 slides and make the presentation short and sweet. The feedback is handled in different ways, depending on the team and the type of issue. If the problem is minor, needs time to be designed, or is unlikely to be addressed in the next sprint, then most teams put issue in the backlog to be worked on later. If the issue is serious and its resolution is clear, then most teams fold it into their next sprint.

She has tested on everything including paper prototypes, Axure prototypes, and development environments. Her favorite is to use html or Axure prototypes, because they allows her to test the interactions before they are coded. She might put together a prototype for the test session, but she does not always have the time to do that. Her UX colleagues or the systems engineers may also produce a prototype using Axure, Expression, or even PowerPoint, depending on the time they have and which tool is the most comfortable to use. Her tests are done remotely, using WebEx and a virtual machine. In two days, she generally can test with 8—10 customers and tries to achieve a mix of people from small and large practices. Elisa does not find remote testing to be the most informative, but it produces great data. Testing remotely also allows her access to a broader pool of users and a mix of representation. Her customers are very interested in providing feedback. Some of them have been using the product for 10 or 15 years and are very invested in the products. Recruiting is not that difficult for her, especially since she can leverage the working group for referrals and the user group meetings.

In addition to her efforts within the sprint cycles, she has touch points with the customers. In April and October, they have user group meetings that last for three days and are attended by as many as 600 customers. She is able to get customers in a conference room and test features and functions with 10 customers a day through 30 or 40 minutes that are quite focused in their scope. She has been able to use these opportunities to test out design ideas on areas that contain known issues and try out potential design solutions. She had a feature set that they were trying to refine and was able to have the customers perform four tasks on the existing UI then do the same four tasks on a prototype and received great feedback. The product also has working groups that meet every two weeks to review prototypes. Since this audience is reviewing prototypes early and often, she does not have them participate in the usability sessions.

Since she is juggling in-sprint testing and other activities and may not be doing work for a specific project team during the course of a sprint, Elisa adopted office hours. Every day, for a few set hours, her door is open for everyone, regardless of what project they are on or what kind of question they have. This allows her to be constantly available to all the teams she supports, but in a manageable way. Having the office hours not only helps to control when and

how to support multiple teams, it sends a really clear message to those teams that she is committed to ongoing and continual communication.

As all of these sprint-based activities are going on, the UX team is looking at ways in which it can be more efficient and streamline the process. Because some of the products were acquired from different companies, there are natural differences in how certain things are done in the UI. Between the EMR and practice management applications, there are four or five different calendar controls, which is not an unusual situation for many legacy software systems. Since they are moving the code over to WPF (Windows Presentation Format), the team had to build a new calendar control. Elisa took this opportunity to make sure that the new control had all the functionality of the previous controls, so that when the old code is updated, the team can move to the new widget without a loss of functionality. Similar opportunities have arisen for standardizing column controls, tab controls, and other widgets. In addition to guiding the design of new components, the team is working on putting things into a pattern library, with snippets of code, so that the developers can access the detailed information about the design and get the code all in one place.

Her advice to UX practitioners who are starting out with Agile is to be flexible. If the team that she is scheduled to support during a sprint has nothing that needs her attention, she finds something else to do. When she first found out that she was going to work on an Agile projects, she simply pulled out a calendar, took a look at what the sprint schedules were, and figured out how to fit into that. Once she had an initial strategy, she pitched it to the product managers and product owners to see if they were comfortable with it and thought it was feasible. She also sat down with the systems engineers and asked them if they thought it would work. Elisa has been superflexible throughout her career and feels that this skill has served her well. When she looks at how to fit her work into Agile, she takes the perspective that it is all about risk and managing that risk. If something in the product needs attention, she flags it for testing and gets it in front of the users. As for being textbook Agile, there is some flexibility around that as well. The reality is that her company is fundamentally a manufacturing firm not a software firm, and certain processes are in place because of that. The Agile methods that the team uses must fit into this larger framework, although if there are places where it must see a change to work more effectively, there is some room to request changes.

Elisa's other piece of advice is to make friends and bring cookies. She finds it critical to the process that everyone knows who she is and that she is there to help. It is important to communicate that the UX person and the process are not about being punitive but really about working together and making the

product better. It is important to make everything feel inclusive. The team needs to be invited to the usability sessions, even if their schedules are unlikely to allow them to attend. It is important to convey the attitude that "you did not do it wrong" and focus on what the requirements are and that they can be satisfied in many ways. Sometimes, it might be best to make sure that the UX team is involved, since their job is to represent the user. She really tries to let her team members know that she wants to make them look good. She feels that her role allows her to see how things are built across products, which keeps her aware of what everyone is doing. This helps her guide and advise her team members more effectively. Now that she spent a certain amount of time working in Agile, she loves the chaos of it. Since she has a very low tolerance for boredom, working in a highly adaptive, fast-paced environment suits her very well. The fact that Agile UX is in its experimental stages in so many companies makes the process more exciting for her.

She wishes she had been able to have had her formal Agile training earlier. Most of the team was able to take its training in December, but she was unable to attend. She did not take her training until April. She had worked on an Agile project at a previous company, but that team was very small and cohesive. The team had official sprints, and their daily standups were often as short as five minutes. The team members were working on a tabbed interface, and she would design the bulk of the tab ahead of time, then refine it based on the requirements. She describes it as a fabulous experience working with a really smart group of people who worked well together. The outcome was a very good product. However, this did not necessarily give her the more formal training that she needed to support her current products, and she felt as if she could have had the answers to a lot of process questions sooner if she had been able to train with the rest of the team. When she did take her training, she also opted to attend the product owner training, just to have a broader perspective. She will not be a product owner but appreciates having insight into what their role is and what that means to the project. She found it tremendously helpful that the company doing the training had extensive experience with her company and was able to tailor its instruction to provide guidance on how to be Agile in that environment.

A lot of effort in general centers around communication and creating a better understanding of how to provide a better user experience for the customer and how to work with the UX staff. She recalls working with one developer who had more of a system focus and was trying to work with her to identify tasks for usability testing. In the end, she was not really able to get the information she needed, so she was inspired to write some guidelines for the systems engineers about how they can help support the usability testing. The team is building itsw pattern library, which allows everyone to contribute to a better design. They are also looking into doing a quarterly newsletter about the UX team and what it

does to help create a better understanding of how the team works. This is still in the early stages, but it is an important tool for communicating internally. The team is also taking advantage of the interest in and growth of the UX team to do activities like creating personas. It has not always been able to do this, and this has not necessarily been a problem, because the company has a good understanding of its customers. However, Elisa sees the existence of the personas as a way to personalize the process for the systems engineer and developers. Instead of them just knowing that some users work this way, they can hear her referring to "Joe" and think of a specific person instead of an anonymous set of customers.

So much inspiration can be taken from Elisa's example. She may not be in an ideal Agile situation—she supports multiple projects as the sole researcher and works in an environment that cannot be completely Agile. But, she clearly embraces the values of Agile and is working on creating a process that works for her and her team. And, she has been successful. Her example should serve as inspiration that it is possible to do research in an Agile environment, as part of the cycle. Focusing on how to make it work and moving past any apprehension or preconceived notions about what it will be like frees you up to be a more effective participant in the process. At the time of our discussion, Elisa also served as the directory of professional development for the User Experience Professionals' Association (UXPA), so I hope to hear more about her Agile evolution at future conferences and in presentations.

Key points

- Focusing on how to solve the problem will lead you to a solution. It would have been very easy for people in Elisa's role to wring their hands and complain about how impossible their task was, they certainly would have found many sympathetic ears. That would have done absolutely nothing to change the circumstances, however. Taking a candid assessment of the situation, figuring out what could work, and getting buy-in for a strategy is a path to success, and it is exactly what Elisa did.
- Never forget that you are a part of a team and make sure that everyone knows that this is your philosophy. It would be very easy for Elisa's client teams to feel ignored or that she was not fully invested in their efforts. But, by establishing a collaborative tone, having a specific schedule of support that was vetted with her teams, and supporting office hours, she has made it clear that she is there for her teams no matter what she is focusing on in a given sprint. This is important in any environment, but critical in Agile environments where there may be less time to work through trust and relationship issues.

SUMMARY

The most successful Agile UX teams do not start out that way, but they put effort into identifying what they are doing well and work to correct the things that are going wrong; this self-reflection leads to success. In hearing the stories of how other teams practice Agile UX, it is clear that a wide variety of tools, techniques, and methods can be used. There is no one-size-fits-all solution for Agile UX teams, and the trick is to find the right fit for your project. Some UX teams find the transition easier than others because the pre-existing corporate values fit well with Agile, being more comfortable working with short time frames, or simply because the team has a natural affinity for working the Agile way.

Often, UX teams first engaging in Agile projects do so with a lot of trepidation. Some of that fear is healthy; after all, it might require some fairly drastic changes in how you do your work on a day-to-day basis. However, the skill to make it work is one of the core competencies that most UX people share. We are highly adaptive, out of necessity, and are always trying to fit our work into the culture around us and figure out how we can do more usability testing or increase customer involvement in the design. These are the same issues faced when moving to Agile. Sure, the pace and the rhythm and the events of the process might be different. Overall, moving a UX team to Agile is a design problem—gather your research, understand your user, propose your best solution, and iterate until you get it right.

References

10-Week Residency, n.d. LUXr: The Lean UX Company. Retrieved April 7, 2012, from http://luxr.co/programs/lean-ux-residency/.

Adrian Howard's Answers on Agile UX, n.d. Retrieved April 7, 2012, from www.quora.com/Adrian-Howard/answers/Agile-UX.

Balanced Team, n.d. Retrieved April 7, 2012, from www.balancedteam.org/.

Block, K., 2012, April 5. Geographically distributed teams in Agile software development Retrieved April 12, 2012, from http://kaylablock.com/agile-geographically-distributed/?goback=%2Egmp_3315113%2Egde_3315113_member_105723713.

Crumlish, C., Malone, E., 2009. Designing Social Interfaces Principles, Patterns, and Practices for Improving the User Experience. O'Reilly Media, Beijing.

Ethnio: Recruiting for User Research, n.d. Retrieved April 11, 2012, from http://ethn.io/.

Gothelf, J., an.d Beyond Staggered Sprints: How TheLadders.com Integrated UX into Agile. Retrieved April 2, 2012, from http://johnnyholland.org/2010/10/beyond-staggered-sprints-how-theladders-com-integrated-ux-into-agile/.

Gothelf, J., n.d. Jeff Gothelf—Lean UX Advocate, UX Designer, Information Architect, Usability Specialist, Agile, Interface Designer, Interaction Designer, Design Team Leader in New York City, New Jersey, Connecticut, tri-State Metro Area. Retrieved April 7, 2012, from www.jeffgothelf.com/.

Govella, A., n.d. Abandon all [Agile] ye who enter here Retrieved April 7, 2012, from www.slideshare.net/austingovella.

Innes, J., 2011, September 26. UXI Matrix (Jon Innes). Retrieved April 7, 2012, from www.slideshare.net/balancedteam/uxi-matrix-jon-innes.

Life at Snagajob—About Snagajob, n.d. Retrieved April 7, 2012, from www.snagajob.com/about-us/life-at-saj/.

O'Kelley, S., (n.d.). Autotagger: A Case Study for Lean UX | AppNexus Tech Blog. AppNexus Tech Blog. Retrieved April 2, 2012, from http://techblog.appnexus.com/2011/autotagger-a-case-study-for-lean-ux/.

Portigal, S., 2010, June 3. Culture: You're soaking in it. Retrieved April 7, 2012, from www.slideshare.net/steveportigal/culture-youre-soaking-in-it.

Smith, C., 2010, August 12. Getting started with user research, Presented at Agile2010. Retrieved April 7, 2012, from www.slideshare.net/carologic/getting-started-with-user-research-presented-at-agile2010.

Smith, C., 2011, August 11. Faster usability testing in an Agile world. Presented at Agile2011. Retrieved April 7, 2012, from www.slideshare.net/carologic/faster-usability-testing-in-an-agile-world.

Tedesco, D., Chadwick-Dias, A., Tullis, T., n.d. Freehand Interactive Design Offline (F.I.D.O.): A new methodology for participatory design. Retrieved April 7, 2012, from http://legacy.bentley.edu/events/agingbydesign2004/presentations/tedesco_chadwickdias_tullis_fido.pdf.