Rolling out Agile in a Large Enterprise

Gabrielle Benefield
Yahoo!
gabriellebenefield@yahoo.com

Abstract

Yahoo! is a large enterprise with a $32 billion market cap and has one of the largest Agile implementations in the world. The adoption of Scrum and Agile practices has been steadily growing over the past two years, and now encompasses more than 150 Yahoo! teams in the United States, Europe, and Asia-Pacific. The projects range from new product development for properties such as Yahoo! Autos to heavy-duty infrastructure work on Yahoo! Mail, which serves 250 million users each month around the globe.

1. Introduction

In the highly competitive Internet space, getting products to market quickly while being both flexible and adaptive to change is critical. Yahoo! needed a software development process that could support an Internet startup culture within the structure of a company that provides products and services to more than 500 million users worldwide. I joined Yahoo! in 2005 to help Yahoo! adopt and utilize the highly effective frameworks of Scrum and Agile throughout the organization. We started with Scrum, using its lightweight framework to create highly collaborative self-organizing teams that could effectively deliver products to market. Next we started to add in Agile engineering practices and Lean fundamentals to deliver greater business value and reduce organizational waste. In the two plus years since this effort began, there have been tremendous successes and valuable lessons learned. This is a retrospective look at my experiences in implementing Agile at a large company.

2. Background

Yahoo! went from being a small startup to a large enterprise company quickly. As such, the culture at Yahoo! is very much like a large startup. There is a constant stream of innovative ideas and product launches as the company strives to be the first to market with new services, while continuing to meet the everyday needs of its users. The company wanted to preserve the feeling of being a startup, but also recognized the need to adopt standard process and practices to help teams deliver better products faster.

Yahoo!'s first effort to meld its culture with a managed software development process began in 2002 with the release of a globally mandated waterfall process called the "Product Development Process" (PDP). Unfortunately, many teams simply ignored the process or, where they couldn't ignore it, paid lip service and made it look like they had adhered to the steps retroactively. The teams that did follow the PDP found that it was heavy, slowed them down, and added little real value.

Some grass-roots efforts to experiment with Agile practices began to emerge within the company and in November of 2004, Tobias Mayer, an engineer on an Agile team invited Jeff Sutherland, one of the inventors of Scrum, to speak at Yahoo!. Pete Deemer, the VP of Product Development attended Jeff's talk and was very enthusiastic about what he heard. He asked Jeff to return and speak to the senior executive team at an offsite. This talk was very well received, and sparked a lively internal discussion about how Yahoo! should embrace Agile development methods. The debate was whether to initiate a large-scale, top-down "forced adoption", or to try to cultivate adoption from the ground-up in a grassroots fashion. The decision was made to proceed with the latter approach, and Pete Deemer was given a budget and mandate to lead this effort.

3. Kickoff

At the time, the most visible symptoms of dysfunction in Yahoo! product development were at the project and team layer (centered around issues of planning, project management, release management, and team interactions), rather than at the technical practices or tools layer. As a result, Yahoo!'s initial focus was on the adoption of Scrum. There was active debate about whether Agile engineering practices should also be adopting in parallel; in hindsight, it would have

accelerated the benefits had they been. But the constraints of time and budget, as well as a nervousness about "trying to make the elephant dance too fast, too soon" led to the postponement of those efforts initially.

Yahoo! launched its Scrum pilot program in February, 2005. Four teams volunteered to try Scrum and share their experiences with the rest of the company. The teams covered a broad set of products and services including the Yahoo! Photos 3.0, a new backend for Yahoo! Mail, internal tools for managing small business sites, and a media site redesign.

The initial commitment made by the teams was (i) to complete comprehensive Scrum training (which translated into Certified ScrumMaster training for most members of the team); (ii) to work with outside Scrum coaches during the first several Sprints; (iii) to use all the standard Scrum practices described in Ken Schwaber's "Agile Project Management with Scrum"; and (iv) to complete at least one Sprint (the term of art for a 30-day iteration in Scrum). It was made clear to the teams that after the first Sprint they could at any time choose to abandon Scrum if they found it unsatisfactory.

At the end of the two months, the feedback was for the most part very positive; the teams liked the process and experience, and management saw positive results. What's more, the positive word-of-mouth was spreading within the company, and other teams were beginning to express interest.

3.1. Roles and Responsibilities

The pilot teams received some early coaching and training from leading Agile thinkers, Ken Schwaber, Paul Hodgetts, Mike Cohn, and Esther Derby. I was initially brought in as a consultant as well. As the program expanded, I accepted a permanent position to build an internal coaching team, working with Pete Deemer to evangelize the benefits of Scrum throughout the company, and to train, coach and support the teams that wished to use it. With just two dedicated coaches in-house and burgeoning demand, we continued to work with external consultants to ensure that teams had as much support as possible in the often difficult early stages of their use of Scrum.

The coaching at this early stage consisted of facilitating key events in the Scrum methodology, such as iteration planning and retrospectives, attending daily stand-up meetings and working with key team members to help answer questions and provide guidance on an as-needed basis. Also, because the early Scrum teams uncovered broader systemic impediments, we tried to knock them down as we went so that the issues we solved for one team would be solved for many. These solutions included working with facilities to secure meeting rooms and take down cube walls, removing governance gates where processes were overly bureaucratic, and changing the way we conducted resource planning and portfolio management. We did not want individual Scrum teams burdened with breaking down resistance from upper management on extraneous issues and were keen to reduce the paperwork and process gates that these early teams would face.

3.2. Tracking Progress

Throughout the transition, we were very active in reaching out across the organization to solicit feedback. We tried to be as honest and transparent as possible, capturing and presenting the challenges along with successes.

Many teams didn't want to try something new unless it was tested and proven a success with Yahoo! teams already. To help combat this problem, at the end of their first month of using Scrum we had all team members and their managers participate in an online survey to anonymously provide their feedback. As an incentive, the responders received a custom-printed Scrum t-shirt for participating.

The overall response rate to the survey was 71 percent (85 percent for Scrum pilot team members). The respondents rated their experiences against the previous processes they had used. The data revealed important insight into both how Scrum had benefited the Yahoo! teams that had participated so far and also their greatest pain points. The survey demonstrated the power of Scrum and provided a strong incentive for other teams to overcome their doubts and get on board.

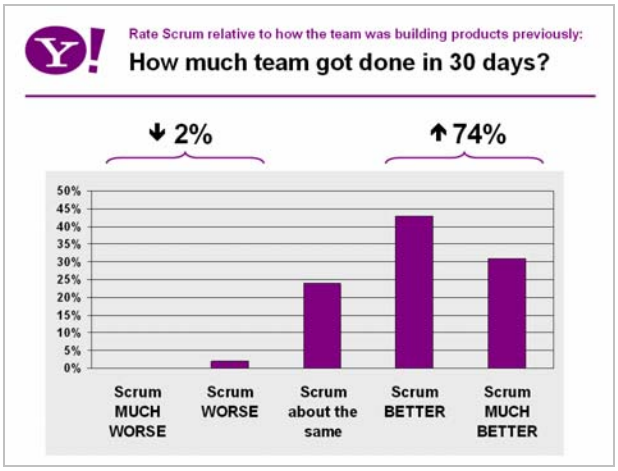

Figure 1. 74 percent of respondents said Scrum improved thirty-day productivity

Figure 2. 80 percent of respondents said Scrum helped clarify team goals.

Figure 5. 89 percent of respondents said Scrum helped collaboration and cooperation with the team.

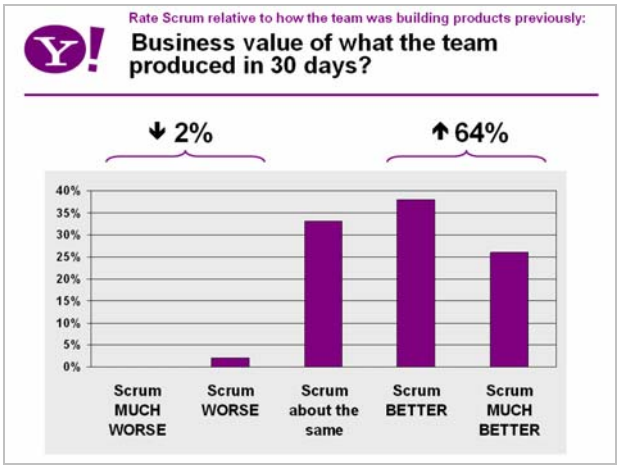

Figure 3. 64 percent of respondents felt Scrum improved the business value of their product at the thirty-day mark.

Figure 6. 68 percent of those surveyed said Scrum helped reduce the amount of time wasted.

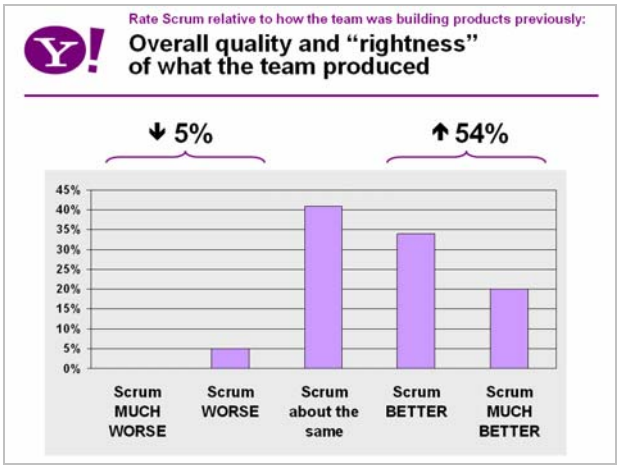

Figure 4. 54 percent of respondents said Scrum improved overall quality.

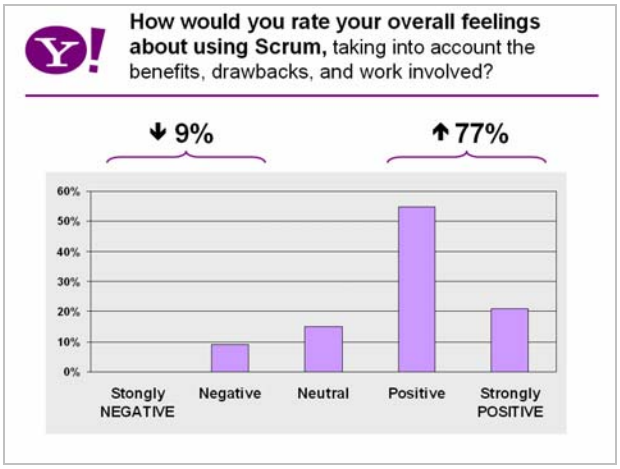

Figure 7. 77 percent of those surveyed had positive feelings concerning Scrum.

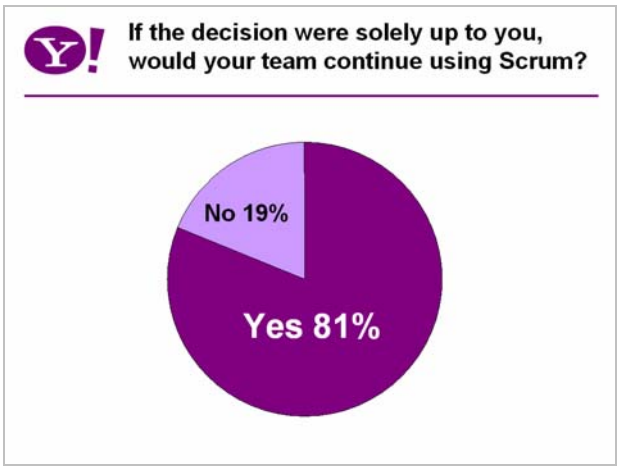

Figure 8. 81 percent of respondents wanted to continue using Scrum.

We continued to track the overall happiness of teams who were using Scrum as a baseline metric to show how we were trending as we scaled. The number has remained fairly consistent over the last few years.

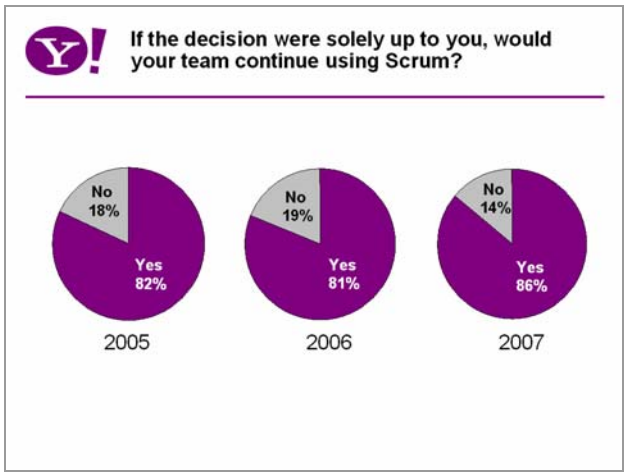

Figure 9. Over the past three years, the number of respondents who want to continue using Scrum has remained consistent.

3.3. Management Support

We noticed across the board that management felt more comfortable with our progress when they heard real feedback from their peers. We tended to focus on the benefits rather than on the mechanics of Scrum, as general managers are typically not interested in the specifics of the process as long as it generates positive results. This in hindsight may have led to some problems; those senior managers that had a clear understanding of the principles and practices of Scrum tended to be much better at supporting their teams use of Scrum, not surprisingly; and we found that the worst time to be trying to explain Scrum to a senior manager was in the midst of a difficult situation where Scrum was the apparent source of the problem; since Scrum tends to make all dysfunction visible, it is often unfairly blamed for the bad news it brings up. Similarly, we tried hard not to "oversell" Scrum to management, because we were concerned that Scrum might be misinterpreted as a silver bullet. We wanted to ensure that people understood the hard work required by such a major change.

3.4. Coaching Model

Once we laid the groundwork for the pilot program, we experimented with an engagement model that allowed us to coach multiple teams effectively. We wanted to work closely with teams until kickoff and then slowly wean them off our services.

While designing the model we tried to be sensitive to the fact that no two teams would face the same challenges in implementing Scrum, nor would they have the same solutions. Some teams might need revolution and rapid change to get rid of major dysfunction, while other teams might need gentle persuasion and understanding. We needed to allow time for coaches to learn the individual needs of a team and be able to help them tailor the Scrum framework specifically for them. We also made the conscious decision not to try to prescribe Scrum throughout the organization all at once, or to allow teams to be forced into Scrum by others; the only teams that would use Scrum were the teams that wished to use Scrum.

Our initial rollout strategy consisted of the following engagement model:

Initial discussion

- Meet with people interested in Scrum and discuss their context and challenges.
- Schedule an overview for key members of the team.
- Organize training and coaching, including Scrum Master training and team training.

Preparation

- Work with the Product Owner to prepare the Product Backlog (Scrum's prioritization of work items).

Training

- Conduct two-day Scrum training for the whole team.

Coaching

For the first sprint, the Agile coach would facilitate the following events:

- First sprint planning meeting
- First sprint review (including setup)

- First sprint retrospective

During the second sprint, the Agile coach would mentor the ScrumMaster as he or she facilitated the meetings. The coach would be available to observe, give feedback, and answer questions. Where possible, the coach would encourage ScrumMasters on different teams to facilitate the retrospective for each other. This would allow them to observe other teams and share knowledge.

We found that teams forgot things over extended periods of time so we stayed in contact and continually re-engaged to lead some master retrospectives and give the teams objective advice and coaching. Teams need to pursue improvement relentlessly.

Teams seemed to follow a standard adoption curve pattern. The first three months were turbulent and challenging. Scrum seemed to work well as a way to get the team self-organizing. During the second three months, many teams were ready for more advanced information. If you have the resources, this is a great time to focus on technical practices and advanced product planning.

4. Scaling Beyond Our Means

By the end of 2005 we had twenty-five teams using Scrum and ongoing feedback checks showing that a consistent 84 percent of team members supported the new Scrum framework as compared to what their teams had been using previously. We had high hopes of increasing the coaching team size by the end of the year so we could continue the momentum. Unfortunately, changes in the budgetary cycles delayed hiring by a few months. The entire organization was affected by the cycle change. The development team was now on the line for meeting their planned roadmaps with reduced resources. The obvious solution (to them at least) was to use the magic bullet of Scrum to make everything go faster. Most people didn't really know what Scrum was, but they heard it was good. They were scaling up before we were ready to scale with them. In the first week of January 2006 our pipeline increased dramatically (figure 10). They all wanted training and coaching. The coaching team now consisted of only two people, Senior Agile Coach JF Unson and me. Our executive champion, Pete Deemer, had left for India to be the Chief Product Officer of our Bangalore office. and tried Scrum anyway, getting their help from attending a training class or reading a book. This had long term negative implications that we are still grappling with.

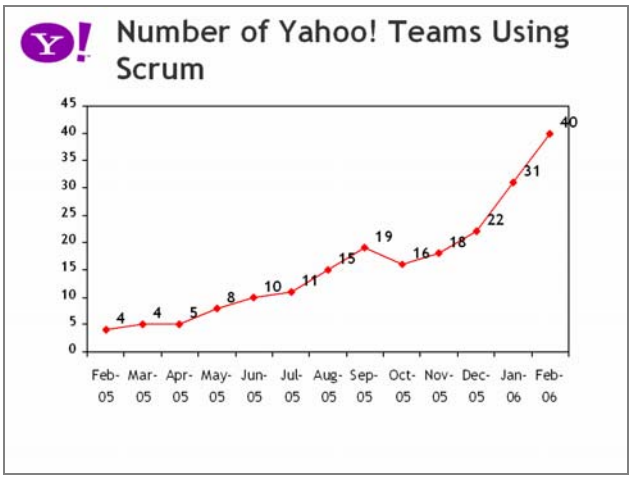

Figure 10. In one year, the number of Scrum teams rose from four to forty.

We were spread so thin that we lost track of the team trainings in progress and were unable to keep up with demand. The whole program was in danger of imploding. We said "no" to teams who were requesting help thinking that might stem the tide. Many of those teams went ahead We frequently ran into one reason that many Scrum implementations fail; Scrum looks simple but it causes change, and change is hard. Scrum works by making dysfunction visible, so that the team can take steps to improve; and this improvement often involves fundamental change by individuals, teams, and the entire organization. Unfortunately, both of these take a toll of the people involved, with extra work, conflict, fear, stress, pain, risk of failure, and other things that people ordinarily try at all costs to avoid. Teams without adequate training and coaching will fail to see the "difficulty" they experience as what it usually is -- Scrum challenging them to improve and they either abandon the practices that challenge them most, or fall back into familiar habits when times get tough. We have discovered teams who believe they are Agile, yet are really doing mini-waterfalls. We have heard managers using Scrum terminology while they are assigning tasks and signing teams up to unreasonable demands (such as making people work nights and weekends).

While these teams are exceptions, they are contrary to Agile values, and they give the practices a bad name. It is generally more work to correct these misguided implementations of pseudo-Scrum than it is to do it right from the beginning.

5. Scaling the Budget

As the organization's embrace of Scrum progressed, we saw again and again how training could make the

difference between success and failure for a team. Unfortunately, our funding and headcount just weren't sufficient to support the accelerating adoption. We had funding from various parts of the organization, including the internal professional development organization, but the demand for training and coaching was simply outstripping supply. We needed to show that well-coached Scrum teams were worth the cost of coaching. For that, we would need metrics.

We began by completing internal case studies and surveys to show the difference between teams that were coached versus teams that were not. We found differences in performance and satisfaction between the two groups. The teams with solid foundational training and coaching thrived, while teams with little or no coaching still had many challenges to overcome. We saw the differences in productivity gains, in survey comments, and in individual team's management satisfaction levels.

The executive team was interested in these results, but needed more concrete data regarding how Agile would affect the bottom line in order to justify funding. Based on this management requirement, the coaching team spent a lot of time debating ways to measure success and productivity. We had no established baseline and didn't want the metrics to be used to skew behavior in a way that could potentially be dangerous. For example, if we measured features built as a success metric, we might encourage people to deliver more features, rather than less of the right features. This would increase complexity and the possibility of more defects and ongoing maintenance costs.

"On-time and on-budget" measures might also be meaningless. How useful was it to ship on time and on budget if we built the wrong thing and released it at the wrong time? If we rushed to hit a promised date we would likely miss key opportunities to learn and take advantage of changing market conditions.

The more we debated, the more complicated the solutions became. We weren't getting anywhere. To move forward, I simply went and asked all the Product Owners on Agile teams to estimate their teams' productivity using Scrum as compared to their previous process. I surveyed thirty-three Product Owners, defining productivity as "how much work the team completed per unit of effort expended." The degree of improvement ranged from 0-200 percent. I captured their comments along with the numbers so we could learn what affected their responses.

I was very upfront about the fact that this data was very subjective. I told the management team that if they had a better suggestion we would be happy to try it. The good news for the program was that the two teams who felt there was no improvement with Scrum felt that their continued lack of productivity was related to events outside of their control, not the process itself. On the other hand, the teams with high productivity increases directly cited Scrum as the reason for their improvement. For example, the Yahoo! mail team uncovered major architectural issues early enough for the team to fix them before launch and were able to quickly respond to a competitive threat during development. They released a major storage improvement ahead of schedule, saving the company millions of dollars.

Overall, the average productivity increase was a 34 percent improvement. An experienced Agile coach might look at this number as see massive opportunity for additional improvement; however, for a management team that was coping with budgetary belt-tightening, an improvement of a one third was dramatic; "Wow…" one executive said, "Imagine if we could increase the productivity of the entire company by 30 percent!" Small improvements can make a big impact when done at such a large scale. We now build the productivity question into the online survey and run it with all team members and their managers. The number has tracked fairly consistently over time. Our most recent survey in July 2007 showed a 39 percent improvement.

The productivity number helped make a case for more coaching resources. I've always been disinterested in numbers until they are related to buying power, then I am very interested in them. Here's what we found:

- One Agile coach can coach about ten teams per year.
- Each team averages ten people, a ratio of approximately 1:100.
- Based on our survey, productivity should increase by an average of 30 percent across those ten teams (we factored down to be conservative).
- Using these assumptions, we calculated that that one coach gave the value of about thirty people per year. We used the average productivity improvement number although we saw a far greater productivity for the coached teams versus the non-coached teams.

We took these numbers to our finance team and constructed a simple spreadsheet model. It was built very conservatively, but still showed that one coach saved the company around 1.5 million dollars per year. It was compelling data. As Pete Deemer aptly put it, "Based on these results, we should have an army of coaches!" We were one of the few teams that doubled its size in the budget rounds that year.

6. Refining the Process

With a more robust coaching team, we were able to scale more effectively and correct some of the mistakes

that were made while we were stretched too thin. Throughout the process, we have not gotten too hung up on having the perfect tracking tools, training materials, coaching program, etc. in place. We have made a lot of mistakes but we have also improved quickly based on iterative feedback. Our philosophy has been that it is better to make the flight than have our bags packed perfectly and still be waiting on the ground.

6.1. Employee Support

We have continued to keep the program voluntary. Although Agile was initially championed by people at the top of the company as well as people lower in the ranks, the fact that Scrum was never mandated meant it had genuine bottom-up support. It had to be "by the people, for the people" for the process to work.

While we built relationships at all levels and marketed the successes to the management team, the real driving force was letting the word spread virally. The teams using Scrum spread the word and people moving throughout the company seeded new teams.

We leveraged the experiences of the people in the trenches to create a very effective promotion engine. We have an internal mailing list called "Agile-people" where people share Agile-related information. We have pockets of committed communities throughout the company and want to build this out further so that the teams can support each other.

A designer on an Agile team, Matt Fukuda, helped to create an Agile t-shirt. It had a "Controle el Caos" emblem with a star on the front and "Viva Agilista" on the back and was part of our guerilla marketing strategy. We used them as incentives to encourage people to complete our survey. We didn't go through corporate branding or put any advertising on them at all. People just knew they were part of the Agile movement.

6.2. Feedback

We tried to keep the process and feedback we were receiving transparent. By being upfront with the challenges, we were able to confront difficult issues and improve. We asked several Scrum teams to conduct panel discussions and "tech talks" to share their experiences in transitioning to Agile and how they dealt with various issues.

Top-down mandates that tried to enforce Scrum practices in a by-the-book fashion always backfired for us and it didn't work to force people against their will. Some teams simply weren't ready or willing to use Agile and we had to respect that. We conversely had teams that followed the practices so religiously that they lost perspective. We had one overly zealous ScrumMaster "protect" the team to the point that management felt shut out of the process. This led to conflicts that threatened the Scrum pilot for the business unit. The ScrumMaster was removed and, through the use of retrospectives and active coaching, the group recovered, but being too extreme can have extreme consequences.

6.3. Coaching

To kick off an enterprise Agile rollout, we found it really helped to have people with experience in the field. The foundation of the overall strategy was built on acquiring and leveraging lessons learned and understanding how to deal with change.

We built out a centralized team of coaches who were passionate and good at building relationships. The team had a mixture of skills, including product management, QA, design, Extreme Programming, Scrum, and Lean principles. It was useful to have people with specializations in addition to generalized coaching skills so they could build bridges into different functional groups.

Personality was also key. Having people who were overly zealous or abrasive would have quickly bought the program to a halt. One important aspect of hiring was to find people with strong skills in collaboration and building consensus.

Lastly, we found that the best Agile champions (evangelists and influencers) were the people already in the teams, from all levels and disciplines. These people knew the context and the challenges of their particular situation and could adjust the process to meet their team's needs. Finding good people who really understand the Agile principles and training them to help their own teams are keys to scaling effectively in a large organization.

6.4. Training

Our training continues to evolve. We have experimented with different class lengths (partly in response to many requests in the early days to do one-day training sessions—two-day sessions were seen as a big imposition for a busy team). We have found that it takes at least one day for anyone to internalize the Agile mindset and then another day to instill the principles. As such we rarely do one-day training sessions anymore.

In the beginning, we were a small coaching team, so everyone learned by the "sink or swim" method. We booked training sessions, developed our own materials, delivered them to the class, and then used rapid feedback to evolve them on an ongoing basis. We now have a "Train the Trainer" program to extend our coaching to people working on teams so they can coach and supplement our group. We also like to bring fresh ideas

into the organization and continue to invite Agile thought leaders onsite. Examples include, Mike Cohns "Certified Scrum Master class" and "Estimating and Planning" workshops. Mary and Tom Poppendiecks "Implementing Lean" for Managers and Practitioners, Jean Tabakas "Five-Levels of Planning" workshop, Jim Copliens "Agile Architecture" and Esther Derby and Diana Larsens "Agile Teamwork" workshop. Gaps in external expertise resulted in us creating workshops and tutorials such as, "Design and Agile" for Visual, Interaction and User Experience Practitioners. We also cross-train by pairing internal trainers with each other. This helps improve and keep materials and messaging consistent across the group. As a result, the trainers on the internal team consistently receive high evaluations, averaging a student satisfaction rate of over 80 percent. Because our trainers are internal, they have the added advantage of knowing the Yahoo! environment and being able to address context-specific challenges for their teams.

We currently offer a two-day ScrumMaster training engagement every month open to anyone at the company, and we also offer customized Scrum team training regularly. We do a lot of planning within the two-day sessions and our classes are very hands on and interactive, with little reliance on slideware. I run Certified ScrumMaster classes every quarter to provide certification for people who require it, as well as to keep me close to the organizational issues the teams are facing. We offer some technical classes but hands-on coaching is still the best way to learn the technical practices. We also run planning workshops from the initial product vision stage through to the sprint plan itself. This level of hands-on coaching is extremely effective but does require a large investment in terms of time.

6.5. Hiring Coaches

Finding great coaches is a Herculean task. Solid, deep Agile experience is thin on the ground. Outgoing, engaging personalities with a mixture of integrity and lack of ego are also rare.

Whoever leads the internal coaching team will be the one who attracts the candidates, so it helps to start with someone who knows what they are doing or hire in a consultant who is very connected within the community. My best hires were through word of mouth or people I already knew and had worked with.

No one we hired came through traditional job sites. We posted the job openings to groups such as ScrumDevelopment and BayXP and XPjobs (Yahoo! groups).

Agile coaches can be very expensive, but we observed that you get what you pay for; as a result, our preference was to sacrifice quantity for quality, even if that means having fewer coaches overall. It is worth spending the money on consultants if you lack internal expertise, as they can ultimately save the company a lot of time and money if you apply them wisely.

We mentor new coaches by pairing them with other coaches on the team. We embed new coaches as dedicated ScrumMasters for teams during their first one to two months so that they can learn the ropes from the trenches. It is a faster ramp up than putting people in as a drop-in coach. The coaches also prefer embedding with a team over "drive-by" consulting.

6.6. Tracking Tools

At Yahoo! we apply the principle of being flexible by not mandating one standard tracking tool. If the process a team is working with is so complex it requires a tool to track, then likely the process has too much complexity in the first place. However, if a team does need a tool (for example, when tracking a large-scale project), we realize that the features of a tracking tool we might use at the enterprise level would probably be overkill for a smaller team. One size does not fit all.

7.0 Lessons Learned

The development team at Yahoo! approached the transition to Agile with pragmatism and adaptability, and has experienced great success with the program as a result. Nevertheless, there are always things you can change. The whole process has been and still is all about learning and adapting as we go. It is important to allow teams to understand that failure is itself an effective learning mechanism. The failures we experienced propelled us to new levels.

7.1. Coach. Don't Dictate.

We received some early lessons in humility. At one point we noticed that many teams were saying they were using Scrum, but were actually dropping key elements, such as the daily Scrum meeting. One team was dropping its daily meeting down to twice a week. As we didn't have much bandwidth we were unable to dig into the underlying reason for the dropping of the daily meeting. A well-meaning coach mandated that the team follow the process and wrote up the core Scrum steps, saying that they "must" be followed. Even though there are great reasons that the meeting should be daily, the coach's language turned some people off. They thought the coaching team was filled with unreasonable zealots. We changed the language to "should," which helped, but the damage had already been done. In a large organization you don't always see the ramifications of your actions

immediately, but over time we discovered that some teams would simply not come to us for coaching. They had decided independently that we were the "process police," and this in turn hindered their adoption and learning curve.

The line between being firm about the value and structure of Scrum and sensitive to the freedom and independence of teams is a difficult one to walk. On one hand, if you dilute the core practices and modify Scrum too much it simply doesn't work; on the other hand, if you try to enforce "by-the-book" adoption across the board you can generate a backlash and some people won't want to work with you.

7.2. Privacy Is Important

Another error we made was to disclose team names when sharing the results from the productivity measures we had gathered. As mentioned earlier, we had two teams show no increase in productivity; the reasons for their lack of improvement were due to technology platforms issues outside of their control. Unfortunately, data hits a competitive nerve and in our culture it is seen as a failure if low scores are shown. One team was fine with the data being shared, as they felt it was an accurate reflection of their situation, but another team felt it reflected badly on them.

We learned how important it is to keep the teams safe. The team who felt exposed ended up working out its own process and backing away from the Scrum coaching team at a critical time when it most needed our help. We now keep data anonymous so there is no risk of inadvertent exposure.

7.3. Align with Management

We did well initially in convincing the management team to try Scrum; however, we would have faced less resistance if we had been able to educate them well on the Agile values while it was still new. Now that managers have been exposed to the terminology and hear about it often, they are not interested in going to dedicated training as they assume they know enough.

Some managers feel left out when the team becomes more self-organized and find it difficult to transition from a traditional command and control model. They sometimes lash out or subvert the process out of fear or lack of understanding. Where we came across people who were anti-Agile, we tried to get them to understand their changed role and to give them more responsibility and involvement. For example, one very anti-Scrum engineering manager who was subverting the process was given the role of ScrumMaster for the team. The manager became the biggest Scrum advocate once he felt engaged and was adding value.

We need to be more effective at selling the benefits to management and teaching them that the values and principles of Agile can help them solve larger organizational problems. We intend to create more case studies and encourage greater sharing of experiences within the management layers. We are creating dedicated management training using Lean principles as a base, as they speak to upper management challenges effectively.

7.4. Design: Find Common Ground

We have a strong design group at Yahoo! Our products are heavily consumer focused so design is very important to us. Design is made up of user experience experts, interaction designers and visual designers. The designers' initial reaction to Agile was similar to the way engineering architects react when faced with the idea that you don't design everything up front, that you constantly refactor, and that requirements will change. They were quite taken aback.

Our design teams were coached in a very by-the-book fashion with coaches from an engineering background. To gain the trust of a design team you need to have credibility in the field and speak their language. Working to understand the challenges and finding common ground in this way helped improve the situation. There are some things about Scrum that naturally fit with design thinking, for example early customer feedback and requirements written in a customer-centric way.

We tried to be flexible, to listen to the design viewpoint, and to help the whole team find a way to work together in a way that made sense to them. If the whole team was not able to find common ground and subsequently worked at the expense of key members, then the whole team failed.

We have found that the primary reason designers like Agile is the collaboration aspect. We still haven't resolved all of the design team's concerns but are incrementally improving.

7.5. Don't expect everyone to like it

We had to deal with the reality that not everyone is willing or able to change and that, ultimately, the new environment may no longer be a good fit for them anymore. From experience it seems that 10 to 15 percent of people will not like the status quo at any given time. You will never be popular with everyone. People sometimes react to change in a negative way. If no one is reacting badly, perhaps you are only telling them what they want to hear, not what they need to hear.

7.6. Fund the Coaching Team Adequately

It would have been great to have the internal coaching team staffed adequately so we could have had a good scaling strategy in place earlier. I was working by myself for a period of time, and the group only had two full time coaches consistently for the first year. We gave new meaning to the term "lean." It took a long time to get more resources assigned to the central coaching team and this only came after financial analysis helped to prove each coach's value. With more coaches, not only would more teams have been more successful, but we also could have scaled more quickly.

7.7. Coach Deep, Not Broad

Due to demand, it was impossible to work as deeply with teams as we would have liked. It is preferable to coach teams more intensely, rather than being so broadly spread across many teams, as a shallow coaching engagement makes it difficult to solve the deeper more challenging issues.

7.8. Implement Solid Engineering Practices

We would have seen far superior results if we had been able to implement solid engineering practices from day one, helping teams configure their environments and learn core Agile engineering concepts. It is very challenging to deliver incremental products without good engineering discipline and this has definitely held back the productivity and quality of many of our teams.

7.9. Align with Project Management

In any large organization like Yahoo! there inevitably exist Project Management Offices with entrenched process control practices. We had conflicts arise in our international offices as they had continued to roll out the original PDP during the U.S. transition to Scrum. The teams attempting to use Scrum were often at odds with the process group and, as a result, the two groups were not communicating well. We tackled this challenge by building relationships and identifying the common goals that both groups were focused on achieving. The Agile teams worked hard to map the very real needs of the process group to the Scrum process. It is a difficult and ongoing challenge but the issues are being resolved through communication and a willingness to be flexible.

7.10. The Organization Must Also Adapt

We did a good job coaching at the team level and instilling Agile principles and values. We now have many high-performing, self-organizing teams. However we face the constant threat that restructuring and the instinct from management to track and leverage resources at the individual level (for example organization-wide resource pools) will destroy the team dynamic and adversely affect productivity. Furthermore, proposals such as time-tracking systems, HR incentives that reward individuals over teams, and matrix structures that encourage local optimization at the expense of company wide goals are constant threats to the Agile program.

8. Conclusion

The Agile team and strategy continues to evolve. Our only constant is change. The Agile team has more than doubled in size and continues to grow. Although we have over 150 Agile teams at Yahoo!, we still have a long way to go. Some teams are very Agile; others do mini-waterfalls and call it Agile. Change is difficult, and to change a company as large as Yahoo! sometimes feels like trying to steer a giant ship with a small paddle. We have learned that patience is important, as is remembering that even the smallest of incremental improvements can have a massive payoff when you do them at a large scale.

9. Bibliography

[1] Cohn, "Agile Estimating and Planning", Prentice Hall, 2006

[2] Schwaber, "Agile Project Management with Scrum", Microsoft Press, 2004

[3] Schwaber & Beedle, "Agile Software Development with SCRUM", Prentice Hall, 2001

[4] Poppendieck & Poppendieck, "Lean Software Development", Addison-Wesley, 2003

[5] James O. Coplien and Neil B. Harrison "Organizational Patterns of Agile Software Development", Prentice H