Abstract

Yahoo! is a great proving ground for Agile. Since the introduction of Agile methods and practices into the company over five years ago, many contests of the Agile process have played out across the expansive Yahoo! software development landscape. Throughout its history, the spirit of Agile has survived, often in surprising and unexpected ways. Whether being mandated from the executive-level or arising from self-motivated small teams, one theme is constant - Agile has embedded itself into the DNA of Yahoo!. And to the present day, Agile continues to emerge and re-emerge in many forms from the dedicated individuals who use the tools of Agile to create some of the most innovative user experiences on the Internet.

This paper offers an informal retrospective of the relative successes and failures of enterprise Agile adoption at Yahoo! from the period 2004 to 2009.

1. Introduction

Previous experience reports on Yahoo! have recounted Agile organizational practices either as singular case studies [1] or perspectives from the leadership [2]. The perspective that we take in this paper is to recount the stories from the Agilistas who experienced Yahoo!'s Agile adoption first hand “from the trenches”.

To determine just how well Agile has succeeded in Yahoo!’s diverse development culture, we split the evolution of Yahoo!’s Agile adoption into seven key phases. Within each phase we discuss one or more critical examples that illustrate the adoption strategy at that time and whether those strategies succeeded or failed. Specifically, we look at the tension between the top-down mandates and the bottom up grass-roots efforts and who or what was controlling the Agile adoption process. We also reveal surprising flips and reversals, and the (often contentious) interpretation of what “to be” Agile really meant.

2. Yahoo!’s Product Development Culture

From outside Yahoo!, visitors to Yahoo!'s expansive network of web-based experiences may assume that the software development process that has produced such highly creative user experiences has had a logical and well-managed path of evolution. However, Yahoo! is a global company of great diversity containing thousands of innovative thinkers and passionate doers, many of whom are specialists or experts in their domains. Many have joined Yahoo! through its many acquisitions. Each person brings into the company their own ideals of the creative process. The idea to do better or try something new is embedded into the culture.

At a larger scale, Yahoo! has often been described as a developer nation of "city-states" of nearly independent teams of developers gathered under one flag. These team-based initiatives cover a very broad area of computer and Internet technology, such as social networking, content delivery, search, advertising, mobile, and cloud computing to name just a few. Yet, in this great diversity of individuals and projects, there are common needs of flexibility, adaptability, fast time to market, and delivering applications that hit the sweet spot of end user acceptance. Prior to Agile, no one process had come close to succeeding in unifying these city-states. Beginning with its introduction, Agile quickly became a lightning rod to many across Yahoo! satisfying these core needs. But the way to Agility for Yahoo! has been and continues to be tumultuous.

3. Pre-Agile – the PDP

In 2004, Yahoo!’s internal software development landscape was exploding. Two years earlier, the Chief Product Officer (CPO) had mandated that a single product development process (known as the PDP) should be followed corporate-wide. The senior leadership believed that a common process framework
that was centrally regulated was the best practice. By 2004, the PDP was seen to be even more vital to coordinate and control the rapidly growing array of software development initiatives across the company. However, changes needed to be made to the PDP to retain the timely product deliveries necessary to remain competitive in the rapidly growing global market place in which Yahoo! was a major player. But the PDP had become outdated and therefore a revision of the PDP was in progress.

The current PDP had many visible deficiencies. It was very Waterfall with many gates and sign-offs required by upper levels of management whose time was becoming increasingly scarce. But, project scope and dates were still dictated by upper management. If a team was able to adhere to the project dates, it was commonly met by exhaustive heroics from the team. The result was high levels of burnout and turnover rates. At the lower levels, the real value of the PDP was being questioned by the development teams that needed to accelerate their schedules and accommodate their unique situations. In the end, many were simply ignoring the rigid structure, while others were giving token observance to the gates that couldn’t be ignored. Exceptions were often granted. A communication divide widened between upper management and the development teams.

Recognizing that change was imperative, the Senior Director of Product Development responsible for the PDP revision invited help from selected program managers. In negotiations with one program manager in particular, Agile-like iterative development processes were provisionally introduced, and adjusted to fit individual teams. In addition, some teams were allowed to restructure their organization to better achieve business goals. The stage had been set for something dramatically better to come.

4. First Agile Experiences – By the Book

By mid to late 2004, mounting dissatisfaction with the PDP was causing independent activity from within the developer ranks. One engineering manager, passionate about exploring new ways of thinking about software development for his team, was instrumental in seeding the idea of Agile practices into Yahoo!. His belief was that localized decision-making at the team level held more value than a centrally controlled process, and importantly, that true change best happens at the grass roots level within an organization. To ignite interest, he invited recognized Agile experts such as Ken Schwaber and Jeff Sutherland to come talk at Yahoo! in widely publicized open sessions. The senior director revising the PDP was among those who attended these sessions.

The general reaction to these Agile sessions was an uncoordinated mix of disinterest and skepticism, and unilateral interest from specific controlling parties. The general developer community was very reluctant to try Agile as it was untested and seemingly prescriptive, or “By the Book”. Negative reaction was especially strong among the user experience and design specialists who were very wary about the applicability of Agile to their discipline and the loss of freedom it might entail. Engineering also had doubts about it actually working. Conversely, upper management heard the resonating message of “make products faster”.

The senior director saw solutions to the PDP issues in the possibilities of Agile too, especially in addressing the command and control brittleness, fundamental organization issues, and empowerment of individual contributors. Thus, as a result of these introductory Agile sessions along with his own research into Agile, he soon became a convert to the general principles of Agile and began to push for Agile adoption as a viable alternative to the PDP. However, he had to directly deal with the initial negative reactions, namely the perception (whether right or wrong) of Agile as a one-size-fits all solution regardless of project, team dynamics and personnel, and the emphasis on mechanics and a rigid prescription. To work at Yahoo!, it had to be pliable.

To assist him in this effort, an Agile specialist was hired as a full-time on-site consultant and senior coach, who was at once adaptable, approachable, dynamic, visible and embodied the best principles of Agile.

5. Adapting Agile to Yahoo!

By early 2005, the senior levels of management were receptive to the Agile message. Although initially considered as a mandate to take the place of the PDP, the senior Agile coach chose instead to work with volunteers and teams who were amenable to trying Agile Scrum for their development projects. By working directly at the developer level, the senior coach was able to readily gauge interest, readiness and cultivate success. The teams who volunteered were entered into a corporate sponsored pilot program with an accompanying multi-day full-team training in Scrum. Experienced Agile coaches, who were certified Scrum Master trainers, were brought in to teach basic as well as advanced Scrum training classes on a regular basis. Interested employees outside of the pilot program were also allowed to attend the classes. These Scrum classes offered practical real-world examples, and a more adaptable approach tailored to the culture of Yahoo!. They became popular and quickly
established a common basis of understanding across the company.

The pilot and the accompanying training program are considered to be one of the most important success factors in the early stages of the Agile adoption process. Moreover, the individuals and teams that went through this experience contributed to the longer term growth and sustainability of Agile by founding other Scrum teams in future generations of projects.

In addition to the Scrum teams initiated by the pilot program, many teams spontaneously started their own Scrums, encouraged by the company’s sponsorship of the pilot program. Without proper coaching however, the success of these informal teams met with mixed results over the longer term. In retrospect, many of these teams did damage to themselves by improperly applying Agile techniques, particularly when command and control mechanisms were still in place.

Despite the fanfare of the pilot program, there were still those that remained skeptical of the Agile promise of superior results. But the principles of the Agile manifesto had clearly taken hold within the company and there was no turning the tide at this stage.

6. The Pilot Program

The corporate opt-in pilot program was created to show the promise of Agile and to address its applicability to the various types of projects then active at Yahoo!. Four product teams volunteered to be a part of the program and share their experiences with the rest of the company. The four projects that volunteered covered different types of development projects, and included a revision of a popular web-based image sharing application, a back-end infrastructure project, a tools project, and a media site redesign [2].

To contrast the difference between teams that participated in the program versus those that did not, we examine the characteristics of a candidate team from each set.

6.1. Case Study #1 – Ready and Eager

One team that entered the pilot program had previously delivered a very successful web-based image sharing application. They were very eager to try Scrum. Their current development path was Waterfall driven with many checkpoints. The team had a high ratio of user experience interaction and visual designers over other team members. The user experience and engineering subteams had large gaps in their communications, spanning several months at a time as each prepared their own roadmaps and completed their designs, and executed their specific tasks. Integrations happened late in the project. Finally, the project had a new version of the product to launch given a very fast timeline.

The pilot was started with both the product management and engineering teams using Agile methods from the beginning. The user experience team was skeptical at first, but they were willing to give it a try, and they participated in all phases of Scrum. The team executed their first sprints with the help of outside coaching staff.

The results of the pilot were clear. The trial exposed deficiencies in the team ratios. There was a too high ratio of user experience designers to software developers. The shared product owner was only partially committed and spent most of his time working on maintaining the existing site instead of helping guide the new release. The user experience team and the software developers kept iterating through multiple sprints on just one component of the product. The developers simply did not have enough engineering resources to get the job done.

The pilot had proven many good points however. The pre-existing separate work silos were broken down, and team members talked directly with each other instead of communications traveling up through manager level. At the end of the pilot, the entire team was generally favorable to the new process. Unresolved at the end of the pilot, however, was the issue of how to manage external dependencies with other teams. And, the user experience specialists’ main difficulty was how to keep the entire user interface design in perspective, and still iterate on specific components.

6.2. Case Study #2 – Not Ready for Scrum

Many teams, however, declined to participate in the pilot program. One team saw Agile as a way to correct deficiencies in their current process, but because of the outstanding risks involved, it chose not to participate. Their project had many challenges to address. The product had a heavy design presence with more than one half the product about the user experience design. It was a legacy product with highly intertwined code bases. The product would be launched simultaneously worldwide, a first in the business unit's history, and a tight timeline was to be met. The product was already highly visible. It was also the first project in the business unit's history to have the team compose the timeline instead of being mandated from the executive level; the ability to deliver according to the deadlines was under high scrutiny.

The proposal to go Agile was met with mixed acceptance by the team. The message that resonated
with the product management was the selling point of being "able to launch a product faster". The engineering staff was neutral in opinion. The user experience designers were skeptical about a process that allowed changing requirements, and obviated complete up front design and instead asked for frequent refactoring. Program management had reservations because of various business mandates that the team had to deliver, and unique challenges that the team faced. At that time, Agile practices did not have any known tried-and-true examples to satisfy those constraints. After some deliberation with the Agile coaching staff, the mutual decision was made that there were too many risks involved to make an attempt at using Agile at that time.

As an epilog to this no-go decision, many months later, an offshoot development team formed within the same product organization that did adopt Agile. They based their decision on the successes of the other pilot teams, and were able to successfully integrate user experience design work into Scrum. This secondary team then influenced the original team to finally try Scrum.

7. The Golden Age of Agile

The years of 2006-2007 brought a high rate of adoption of Agile within the company. Agile activity in product development was becoming commonplace. Within two years of its introduction, there were well over 150 development teams reporting that they were using Scrum. The phrase “to be agile” had become an accepted and desired part of the Yahoo! software development nomenclature. Company-wide talks on Agile topics, interest group gatherings and other special events were offered. Yahoo! Agilistas were regularly participating in external Agile community events -- a small company presence attended the Agile 2006 conference and substantially increased in number for the Agile 2007 conference. Yahoo! papers were accepted and presented at other venues as well. Agile ambassadors fanned out across the company and Agile was considered a key innovative process. Local Agile “champions” within business units were recruited to help spread learned knowledge and techniques. The central Agile coaching staff was hiring. Agile was in rapid expansion and to be on a Yahoo! team using Agile was invigorating.

Along with the success of Scrum adoptions in the United States corporate home office, Agile practices were also being deeply integrated within the international offices as well. A significant portion of software development was being performed in the European offices, where established protocols and regulated standards had already been in place for some time. For these teams, Scrum practices were retrofitted into already working gates and checklists. As the Bangalore, India office was expanding, Agile coaches were on site to help not only local teams embrace Agile, but also to help them work successfully with their U.S.-based Agile counterparts in distributed collaborations. Agile had indeed many dialects within the global company.

But there were significant challenges. Agile was becoming a new mandate within many areas of the company. The demand for in-team Agile coaching was far outstripping the supply of coaches. Qualified coaches were hard to find and the central coaching staff had only a small handful of well trained people to handle the building requests for assistance. Many teams had to “make do” with limited contact from the coaching staff, or simply struck out on their own paths. This caused some fragmentation and widely varying interpretations of Agile practice. Also troublesome was the continuing notion, particularly with upper management, of Agile’s purpose to be primarily “make products faster”, and Agile as the panacea for all sorts of organizational issues. Agile had developed the “silver bullet” syndrome. Alternate Agile methods were also being tested, such as Kanban, which stressed the still fledging Scrum rollout, causing some friction and fragmentation.

7.1. Case study: The Game Changing Transformation

The positive results from the pilot program and self-reported surveys led to some product teams to being mandated to use Agile by their engineering management, even though that was not how the Agile staff intended it. For one team, the entire membership was trained in the now established 2-day Scrum training, and the team immediately started to use Agile. The team also had available an in-house Scrum coach in an advisory capacity. For this team, the application of Scrum proved to be so successful in just the first quarter of use, that the productivity increase was specifically called out by the Chief Product Officer in his 2006 first quarter address to the company. From inside the team, the Scrum adoption helped solve several key organizational and communications issues. It turned a previous skeptic into a believer, for the team’s program manager “it changed my opinion and mindset permanently.”

While the significance of this success was widely profiled within the company, there were still issues to resolve. The team had increased their adeptness at short term execution, but there were big gaps in long term planning. This gap was later addressed by applying a more advanced Scrum technique of
multilevel planning [7]. Focus group lunches were initiated for the business unit’s Scrum Masters to share stories and issues.

But by far, the most difficult issue was scaling the success of Agile beyond single instances of teams -- which was proving to be the adoption nemesis for Yahoo!. Trust with upper management was building because of Agile’s ability to deliver and consequently Agile was mandated to other teams of the same business unit. However, the continued lack of available qualified coaching impeded repetition of success and the efficiency of implementation for other teams.

To address the scaling problem, several approaches were tried in parallel. Hiring additional coaches was obvious, but finding qualified top notch candidates with deep experience proved exceedingly difficult. Being truly adept at Agile Scrum techniques is a skill not easily taught and requires an uncommon mix of personal attributes: technical aptitude (quite necessary for Yahoo!), teaching and planning abilities, and especially listening and synthesizing. Also, the budget for hiring was minimal. The senior Agile coach also initiated a program to train trainers who would operate within their local business units, but again, finding suitable candidates that had enough spare time to participate outside of their normal duties was rare. This idea was appealing to those at the trench level, but actualization was hard. What resulted were ad hoc networks based on people connections that work well in a local context but fails at the enterprise level.

8. Transference to the Trenches

Beginning in late 2007, the greater business state of the company began to ripple with forebodings of reductions in budgets and staffing. The central Yahoo! Agile coaching staff was not immune to the coming changes and was directly impacted by events coinciding with the company-wide staff reduction that began in early 2008. The coaching team was disbanded, with team members moving into specific business units of the company or exiting the company altogether. Communications from a centrally led Agile leadership went silent.

To the agilista in the trenches, the impact wasn’t immediately felt. Many of the teams had learned to become self sufficient earlier because of the scaling issue concerning the availability of coaches, and now more so as the company turned inward to reinvent itself. As time went on, calls for coaching and classes were routed to outside vendors. Agile didn’t stop as the DNA of Agile was now firmly rooted and teams carried forward on their own. But, unresolved issues from before surfaced and became more acute.

9. Agile Goes Multigenerational

The period following the departure of the central Agile coaching staff was the first test of the durability of Agile. The first generation agilistas who graduated from the pilot program or who attended the sponsored Scrum classes and then gained multiple iterations of experience became the pathfinders and teachers to the following generation of Agile teams. Those that understood and embraced the true meaning of Agile were generally successful at creating productive and sustainable teams. Those who bent the intention of Agile, whether knowingly or not, had mixed results.

To better understand the differences between Agile teams and those that purported to be Agile, we examine three case studies of teams. All of the case studies are from the same business unit, viewed from the level of the agilista within the team.

9.1. Case Study #1: Command and Control

“Agile”

Teams that transition from command and control processes face steep challenges and often don’t make the full conversion. Teams on this adoption path are often characterized by what they take from Agile to suit their own needs, applying local interpretation, which typically results in acute imbalance and protectionism of their unique processes. These teams are not sustainable on their own and need external forces to keep them operational. Micromanagement and employee burnout resulting from the misuse of the increased visibility into the team are known hallmarks. Survival rates of team members are very short.

In this first case study, we look into a prominent and long-lived web-based product that is supported by several interconnected teams. Each subteam had asked for and received initial Scrum training for all of its members, but afterwards declined further coaching. Internally, the program and its management had adopted some, but not all, of the essential parts of the Scrum discipline. Of the Scrum practices that were adopted, they were substantially altered. The overall product backlog became a task “funnel” controlled and evaluated by one person, from which iteration tasks were assigned. User stories were lacking or difficult to trace. Delivery dates were handed down to the team. Daily stand-ups were status reviews. Sprints were fixed scope and fixed dates. Task estimates were questioned by managers, “you committed to 6, why is it taking you longer?” Feature completion was paramount and measured primarily on the amount of code checked into the repository, often lacking unit tests. Requirements from regional sister teams were heavily deprioritized, and not included in planning,
thereby splintering overall incentives and efforts. In essence, mis-adaptation of Scrum had provided them a glorified and highly scrutinized task tracking system and the last factors of control left to the team members were the quality of the code and the quality of their lives.

9.2. Case Study #2: “Agile” Cowboys

For the second case study, we examine a new development team which had been formed from senior developers, some of whom had been exposed to Scrum in previous teams. As in the first case study, the team had adopted the scaffolding of Scrum, but because of entitlements of developer rights implicitly gifted by management, certain high profile individuals were allowed to skip core elements of the Scrum discipline, especially when stressed by looming deadlines. The impact was that the team started out following Scrum, but over time the team’s decision making process mutated into a variant of command and control.

The team was initially attracted to Agile via the promise of empowerment to “get things done quickly.” Multi-level planning was embraced because it gave the image of “team involvement”, but never became a fully team driven exercise as the engineering management was very willing to step in and be “hands on.” Moreover, when individual developers wanted exceptions, they could side-step the Scrum process and appeal privately to upper management to grant the exception, which often occurred. Additional misinterpretations included managers assigning tasks directly to individuals, eschewing documentation altogether, user stories lacking acceptance criteria and similar activity added to the Scrum dysfunction. And, from the trenches, team members complained of micromanaging.

9.3. Case Study #3: True Agile DNA

For our final case study, we examine a team that has been Agile since its inception in late 2007. Although operating in an organization in which command and control is predominant, the team has not only been able to maintain its agility, but withstand challenges and thrive. Furthermore, this team has been given continually more responsibilities as well as management of a new project of higher profile with more critical value to the company’s future.

This team has benefited from strong Agile leadership. The engineering director, a veteran of one of the early pilot programs, brought to this team his experience and Agile oriented disposition. To create a proper foundation, he led by example by volunteering to perform as the initial Scrum Master of the team.

In a dramatic demonstration of the power of team-led management, when the executive management mandated a specific release date and accompanying feature set for an upcoming release, the team responded by collaborating on a multi-level release plan. The team created a sufficiently realistic release plan with enough detail to cause the executive management to change the date. The team then delivered as planned.

In another example of successful team self-organization, when a new Scrum Master attempted to introduce Waterfall-style templates into the team workflow, the team convened and petitioned management for a different Scrum Master. They produced requirements for what they must have for a person in that role and the petition was granted.

10. Conclusion

Agile is still evolving at Yahoo!. In this five year retrospective, there has been much diversity of experience yet certain themes stand out, particularly when looking into the teams or “trenches.” Once an individual participates in a successful implementation of Agile and understands the true benefits, that person becomes a strong devotee and locks on to the Agile “DNA”. This is a necessary element for any long term agile adoption as that Agile genetic material will provide the resiliency for individuals to survive in even very harsh process-oriented conditions.

11. Acknowledgements

We would like to thank Gabrielle Benefield for her Agile mentorship while at Yahoo!, and JF Unson for his Agile trench survival stories. Also, we sincerely appreciate the guidance provided by Jeff Patton, who helped iterate our paper to completion.

12. References


