8-6-2011

Agile Development at ABC – What Went Wrong?

Cristina Karekar
University of Nebraska-Omaha

Alvin Tarrell
University of Nebraska-Omaha, atarrell@gmav.unomaha.edu

Ann Fruhling
University of Nebraska at Omaha, afruhling@mail.unomaha.edu

Agile Development at ABC – What Went Wrong?

Cristina Karekar
University of Nebraska-Omaha
(Email withheld)

Alvin Tarrell
University of Nebraska-Omaha
atarrell@gmav.unomaha.edu

Dr. Ann Fruhling
University of Nebraska-Omaha
afruhling@mail.unomaha.edu

ABSTRACT

Agile development methods continue to enjoy widespread use, with more and more companies transitioning to agile methods. Current literature suggests that most of those companies are successful in making the transition, but others are not so successful. This paper examines one such company – referred to within as the 'ABC Company' to maintain their privacy – and analyzes and discusses their struggles with implementing agile methods. In short, it appears that lack of firm leadership commitment to agile, absence of a clearly defined customer to provide clearly defined requirements or push for additional software capabilities, failure to provide adequate initial or ongoing training and support to the organization as a whole, and underestimating the change management requirements were contributing factors to ABC's struggles with implementing agile methods. These conclusions were reached based on a series of interviews with company employees, review of the relevant literature, and comparisons with other similar case studies.

Keywords

Agile software development

INTRODUCTION

Companies today deal with continual change, and those who effectively deal with that change are commonly described as being 'agile' organizations. Agility has been widely used in the software development industry for several years now, with 'agile development methods' commonly touted as solving the problems arising from use of more traditional, plan-driven development methods. Unfortunately, a successful transition to use of agile development methodologies involves much more than simply making the decision to do so. Most organizations making the switch are very successful; other organizations are not so successful at making the transition, with difficulties arising in one or more aspects of the migration. Leaders of software development organizations must understand and address the critical factors involved in determining success or failure of these efforts before making the decision to transition to agile development methods.

This article examines what we call the "ABC Company", a mid-sized software development company headquartered in the central United States. *(This company is well known in the industry, so we chose to call it ABC to maintain confidentiality. Similarly, quotes and feedback will not be attributed to specific individuals to maintain their personal privacy.)* ABC made the decision to transition to agile development methods in 2007, and shortly became very successful and increased their development efforts there ten-fold. However, one year later ABC had stepped away completely from agile methods, returning to a structured development methodology more closely aligned with more traditional techniques. This case study examines ABC's history and focuses on describing their experiences with agile development, finding reasons behind their temporary success and eventual change of heart, and development of recommendations for preventing recurrence. Information for this case study was drawn from a series of interviews conducted with several developers and managers at ABC, as well as from a review of the literature and comparisons against widely recognized best and suggested practices. Our intent is to provide practical guidance for those leaders who are considering transitioning their development organizations to use of agile methods, hopefully pointing out some of the aspects which need to be evaluated when deciding for or against moving to agile development methodologies.

This paper will also benefit the research community, as it details the specific conditions found at an organization which was unsuccessful at implementing agile development methods. This type of outcome has been underrepresented in the literature, so this case study may provide some previously missing insight for the research community.

BACKGROUND

Research Design

This research was conducted as a case study due to the limited time and resources available for the effort. We were initially looking to study unsuccessful transitions to agile methodologies in general, but we found a dearth of published literature on that topic, so in-depth study of this particular case seemed well worthwhile.

We chose to utilize interviews for much the same reasons, first talking with thirteen members of the agile development team at the ABC Company, and then doing follow-up telephone and face-to-face interviews with two of the more prominent members of the team. This singular focus on a single organization brings some recognized shortcomings – small sample size, inherent biases, and lack of generalization among them – but we believe the results are still interesting and somewhat unique. We also minimize these problems as much as possible by comparing and contrasting them to results obtained in earlier research involving eleven companies conducting agile development projects for the U.S. Department of Defense, and also by comparing them to results obtained from a recent annual survey of organizations currently applying agile methods on some or all of their projects.

Literature Review

There is not one universal methodology for system development that will work for all projects (Iivari, Hirschheim, and Klein, 2001) and all environments. The traditional plan-driven system development methodology requires extensive planning, codified processes, and rigorous reuse (Boehm, 2002). This methodology works best when requirements are known in advance – including prototyping – and when the requirements are relatively stable. The plan-driven model is often used in practice because of its straightforward and methodical, structured nature. However, in practice, the plan-driven model has a number of widely-reported shortcomings, including the inability to effectively handle changing requirements, and the tendencies to be both significantly over budget and behind schedule (Boehm, 2002; Standish Group, 2005; Watson, Kelly, Galliers, and Brancheau, 1997). As new technologies, infrastructure, and expectations evolve at today's pace, the plan-driven system development methodologies struggle to keep pace.

To address some of these shortcomings, new system development models were proposed, including the spiral model and agile approaches (e.g. Scrum, eXtreme Programming, and Crystal) (Highsmith and Cockburn, 2001). These efforts ultimately led to issuance of 'The Agile Manifesto' in 2001 by a consortium of developers active in this arena.

Leffingwell (2007) provides a simple explanation of the primary philosophical differences between traditional plan-driven methods and newer agile techniques; Figure 1 shows this differentiation pictorially. Traditional methods involve a fixed set of requirements, generally negotiated in advance, and then adjust resource levels and delivery schedules as necessary to allow implementation of those requirements. Agile methods take the opposite approach, using predetermined delivery schedules and staffing levels, and adjusting the requirements for each software release to work within those resource constraints. It is a fundamentally different approach, and one which often results in an iterative development cycle which rapidly converges on a fully functioning product.

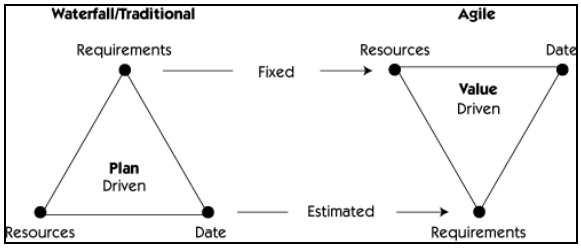

Figure 1: Traditional (plan-driven) versus Agile (value-driven) methods (Leffingwell, 2007)

This article is designed to provide information, insights, and practical, actionable advice to Information Technology (IT) managers and analysts to help them better prepare for a move to an agile system development environment. We examine the experiences of an organization that attempted the transition, and provide our interpretation of what led to failure of the effort. We also compare those ideas to the findings of an industry-wide survey looking at experiences with use of agile methods.

Company Background

The ABC Company is a mid-sized company, with branch offices spread over a five state region in the central U.S. ABC provides a variety of software- and services-based solutions, and they have a twenty-five year history in this business, with annual revenues topping $400M. ABC states that their transaction processing software interacts with over 50% of households in the United States, so they are very much a major force in the industry. ABC generally builds software for release to the industry as a whole rather than for specific customers, so products tend to be somewhat universally applicable and general rather than specific and specialized. Software development efforts at ABC are distributed across these multiple offices, rather than being concentrated at a single location. (*Specifics shown here are from the ABC website; direct citation will not be provided so that confidentiality can be maintained.)*

In early 2006, ABC acquired a successful private provider of Operations Support System software that enabled communications companies to bring bundled, advanced services to market quickly and effectively. This company had used agile methods in their software development efforts for several years, and had become very comfortable and proficient in agile software development as a result. The acquisition was meant to expand ABC's ability to support cable and satellite operators as they deliver advanced IP-based services, designed to attract and retain consumers who expect the availability of video, voice and data services over nearly any device at the time and location they choose.

Transition to agile

In 2007 ABC announced the launch of a new software product suite that would increase competitive advantage in the customer care and billing market by providing a state-of-the-art customer offer configuration, order fulfillment, and billing solution. Seeking agility, the ABC executive leadership was convinced to undertake this effort using agile development methods. ABC itself had no prior agile development experience, and the recently-acquired agile development team leaders were established as the champions of agile software development within ABC. These agile champions used the following benefits – established based on their experience at their prior company – to convince the ABC leadership to make the transition:

- Agile methods improve visibility into the development process – team members at all levels of the organization have a much clearer picture of the requirements, what has been done, and what remains to be done
- Provide earlier identification and better understanding of risks – empirical evidence shows that agile methods allow identification of schedule and cost risks up to two months in advance of that provided by plan-driven methods
- Use "change" to better motivate teams and help them recover from issues that arise in the development process. Growth comes from change, so use these issues to promote growth.

ABC recognized that this would be a major transition, and company leadership was apprehensive in many ways. They recognized that this was in fact as much a cultural shift as it was a change in development methodologies, and they were somewhat concerned about a perceived lack of control over the process. To ease the transition, ABC invested in a one-week training session for the soon-to-be agile development team – including the developers, systems analysts, product managers, quality assurance testers, and project planning personnel. Based on the "Agile Release Train" concept presented by Leffingwell (2007), this training was to form the basis and understanding of agile development methodologies for most members of the agile development teams at ABC.

In part because of leadership concerns, ABC also made the choice to not fully transition to agile methods in this development effort. The current schedule of three software releases per year to the customer was maintained, with some portions of each release cycle being completed with agile methods and the remainder continuing to be developed using traditional plan-driven methods. These release cycles – including the functionality expected and projections of the level of effort (staffing and other resources) required to attain them – were projected out two years in the future for both agile and plan-driven portions of the effort.

Early success

Agile methods caught on quickly at ABC, and they moved from a single team of 8 individuals using these methods to 11 or 12 development teams totaling over 130 personnel performing agile development in less than a year. Agile teams at ABC were organized to be internally multifunctional, but were designed to concentrate on one key area of external functionality in their efforts. Agile teams were designed to be stand-alone and autonomous, and consisted of some combination of the following:

- Product owner (may be shared across teams)
- Lead Architect
- Scrum Master
- Lead Developer
- Developer(s)
- Tester(s)
- Technical Writer

The relatively small size of these agile teams, coupled with the wide variety of skills needed across the team, required each person to be somewhat more versatile and open to new responsibilities within the development project compared to their roles on traditional development efforts. The degree of interaction and cross-training of course varied from team to team depending on the characteristics of the persons forming each team, but in general the agile teams expected their members to be much more involved in multiple facets of the operation than their plan-driven counterparts.

ABC used a customized version of Scrum in their agile development efforts. The initial training provided a starting point, but most of these customized agile processes were developed on the fly during the early stages of this effort. Interviewees provided the example of adjusting agile release cycles to correspond to the waterfall / plan-driven release cycles to illustrate how their agile processes were customized to meet organizational needs.

A common integration testing scheme and schedule was used, so it was also necessary to keep the two development efforts in synch with each other. That was the primary driver of the development schedule, although specifics within each major iteration were somewhat flexible. At least two daily meetings were held to facilitate this coordination process. The first, hosted by the waterfall development leaders and attended by them as well as the agile Scrum Masters and other leaders, ensured that the agile development teams were aware of progress on the waterfall portions of the project as well as requirements placed on the agile teams. The agile development leaders hosted a similar meeting later in the day, ensuring that the waterfall development leads were also fully apprised of the status and needs of the agile development portion of the effort.

Agile no more

In spite of these efforts, agile development efforts at ABC came to an end roughly one year later. Today they are using a hybrid method developed in-house, but it is self-described as much more tightly controlled than agile methods would allow. An additional level of control and oversight has been added to the process, and release dates and functionality are very closely monitored and controlled. Interviewees in general felt this movement away from agile methods was due to lack of firm leadership commitment to the process of converting to agile methodologies; more details and analysis will be provided in the sections below.

ANALYSIS AND COMMENTS

Failure of agile methods at ABC (they would not call it ‘failure’) can be attributed to some combination of the following:

Lack of leadership commitment

Previous research has shown that management and leadership support is critical to success of the transition to agile methods (Fruhling and Tarrell, 2008; Schatz and Abdelshafi, 2005). Respondents to VersionOne's annual survey of agile practitioners lists "management support" as one of the top 5 barriers to further adoption of agile methods within their organization, with nearly one-third of respondents including that as a limiting factor (VersionOne, 2008). Several observations in this case make it evident that ABC's leadership was not fully supportive of this effort:

- In spite of constant requests from a project manager experienced in agile methods, it took nearly two years after acquisition of a company who had successfully implemented agile methods before ABC's management would allow use of similar methods on their own projects.
- Once the decision was made to try agile methods, ABC still only went part way – they used agile methods for only certain portions of certain projects, and used traditional plan-driven methodologies for the controlling path effort. Management continued to require two-year plans for staffing and addition of functionality, in direct contrast to the concepts presented by Leffingwell (2007). Project delivery timelines were still tied to these plan-driven portions of the project, so no real "agility" was added to the process as a whole.
- This failure to adjust the development cycle meant customer saw no benefits from the movement to agile. This minimized the effect of the incremental releases that provide some of the key benefits of agile development methods, so there was no customer-driven support for the transition to agile.

Lack of adequate training

Adequate training is also an important factor in ensuring the success of the movement to agile methods. This training should cover the full range of the development staff, and should also include non-IT management, business analysts, and customer representatives if possible (Fruhling and Tarrell, 2008). VersionOne's results (2008) bear this out as well, with 42% of respondents identifying lack of training or experience with agile methods as a barrier to further use of agile methods within their organizations, and 26% identifying those same factors as the leading cause of unsuccessful agile projects. Specifics at ABC include:

- One week of initial training was provided, but this training was geared towards the development staff particularly, rather than toward the organization as a whole as recommended above.
- This training was good as a start, but there was also a need for ongoing training and monitoring. Several interviewees commented that they had to "discover for themselves" which agile practices were most applicable to their effort; some of that discovery is unavoidable, but ongoing training and coaching support has been shown to be valuable to those successfully making the transition to agile methods (Fruhling and Tarrell, 2008).
- Organizations successfully making the transition to agile methods have employed an agile champion team as well (Fruhling and Tarrell, 2008). This team consists of those most experienced with and supportive of agile methods within the organization, and is charged with minimizing roadblocks in the transition process and continuously fine-tuning the agile approach within the organization. Training is critical to the success of this team, and ABC made no real effort to ensure this success.

No "Real" Customer

In the researchers' opinion, this may be one of the primary causes of failure of agile methods at ABC. Agile literature is filled with evidence supporting the importance of an on-site customer to successful completion of agile development projects (Bradbury, 2007; Sutherland, 2005; Beck, 2000). In practice this is often difficult to achieve, however, and many agile projects are successfully completed without the benefit of a full-time customer representative co-located with the development team (Fruhling and Tarrell, 2008). ABC had some unique challenges in this area as well, specifically:

- ABC's products are generally developed for the market as a whole rather than for one specific customer. That essentially precludes them from having an on-site customer – or even ready access to an off-site customer – since "the market" is a very amorphous concept. Lack of a true and dedicated customer makes it very difficult to define precise requirements and to get feedback on ideas and implementations, both of which hamper the development environment.

- Lack of a clear and present customer also makes it difficult to demonstrate incremental progress. Delivering new or improved functionality in this environment requires actually taking a product to market – a time-intensive and potentially expensive undertaking in the commercial world. It simply is not feasible to do this roll-out on a frequent basis, further hampering the benefits seen from use of agile methods.
- ABC used a "Product Owner" as a proxy for a resident customer representative. That Product Owner needed to not only represent the customer to ABC, he/she also had to represent ABC to the customer. This dual role led the Product Owners to often be out interacting with the market to determine needs, leaving them less time to interact with and support the agile development effort. This again is a problem at many organizations practicing agile software development, but seemed more acute at ABC as they struggled to identify who the real customer was and exactly what the needs were. One interviewee – one of the agile project managers – identified this as a "huge" gap, and said that "ABC will not be successful with agile until this issue is addressed."

Failure to Address Change Management

Transition to agile development methods is very much a change that affects the entire organization, and it is important to not underestimate the degree of change management efforts that may be required. For example, Fruhling and Tarrell (2008), based on literature review and interviews with eleven organizations practicing agile development, suggest assessing corporate culture and organizational readiness for change as the first factor to consider when deciding whether agile is a good fit for an organization. ABC had some specific characteristics that may have complicated this transition, and also failed to address some more general change management concerns, both of which may have contributed to their lack of success with agile methods. Specifically:

- This effort involved a large group of people from the "original" ABC and its plan-driven culture and a smaller group of people from an agile-driven acquired subsidiary. This left ABC with a shortage of mid-level agile champions, and there may have simply not been enough advocates to help lead the migration. There was a marked difference in corporate culture between the two groups, and that mismatch and tension may have contributed to the lack of success with agile methods.
- Agile methods also require development staff to take on additional roles and responsibilities, and some developers are not willing or able to take on that extra tasking. For example, coders have much more ownership of the analysis and design when using agile methods – some developers embrace that, others do not want that extra responsibility. Other organizations have reported similar findings as well when making the transition to agile methods (Fruhling and Tarrell, 2008). Failure to recognize this phenomenon and properly prepare for it through training and/or personnel selection can impede successful transition to agile methods.
- People in general fear and resist change (Diefenbach, 2007), and that was likely a factor here as well. There may have been fear of the "new company" taking over from the "old" developers, and there was a tone of negativity towards agile from many of those involved. According to one interviewee, agile methods will help "identify" problems, but management needs to then "solve" those problems – agile cannot do that on its own, and agile will not be successful until management actively supports it sufficiently.

Distributed development

Distributed software development using agile methods is not impossible – 57% of the teams responding to VersionOne's annual agile survey reported conducting distributed development (VersionOne, 2008) – but doing so on an initial transition project may have been ill-advised. Information gathered from literature reviews and previous surveys recommends starting first on a small, in-house, non-mission critical project and then scaling up from there as success is achieved and experience grows (Fruhling and Tarrell, 2008). ABC did not proceed in that manner, and doing so may have contributed to their lack of success.

DISCUSSION

This section will address the findings of this research as well as some limitations of the case study and suggestions for follow-on research.

Research Findings

This case study revealed several key factors which influence the success or failure of agile software development projects. Specifically:

- Leadership commitment and support is vital to success of agile development projects. As previously reported (Fruhling and Tarrell, 2008; Schatz and Abdelshafi, 2005), leadership and management backing of the agile development process is vital, and needs to be communicated from the top down. There must be horizontal and vertical dedication to the agile process across the organization. Many of these factors were clearly lacking at ABC.
- Presence of a knowledgeable, empowered, and decisive local customer representative has been shown to be critical to agile software development projects (Adkins, 2008; Fruhling and Tarrell, 2008; Mohammadi, Nikkhahan, and Sohrabi, 2008). Problems develop when any of these capabilities are lacking, including requirements which are inadequately developed, unclear, or worse yet wrong (Cao and Ramesh, 2005). This is the situation that ABC found itself in – they had no clearly defined customer, and so had no one to clearly define precise requirements. They used a 'Product Owner' as a proxy for a local on-site customer, but in this case it appears the association was too esoteric to be successful. One of the basic tenets of the agile movement is 'collaboration with the customer over contract negotiation' (Beck, et al, 2001), and it is difficult to meet that goal without a readily available, clearly defined customer.
- Agile software projects must also provide some increased benefit to the customer relative to more traditional methods. In fact, at least two of the twelve principles of agile development attest to that: 'Our highest priority is to satisfy the customer through early and continuous delivery of valuable software' and 'Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale' (Beck, et al, 2001). ABC's agile development effort chose not to adhere to those principles, as they embedded the agile processes within their existing plan-driven methods, and so provided no evidence to the customer that agile methods were in use at ABC. This appears to be one reason behind their lack of success in implementing agile development methods.
- One must also not neglect the organizational changes brought about by use of agile methods. Responsibilities are pushed down the management chain in agile projects, so traditional managers may feel a loss of control, and some developers may resist taking on the extra responsibilities associated with agile development. It was not apparent that ABC was prepared for this aspect of the transition, and that lack of preparedness contributed to their lack of success in transitioning to agile methods.
- Finally, one must recognize that ABC attempted to make this transition by embedding agile development within traditional plan-driven teams, and also within distributed development teams. This is certainly not a unique approach, but is arguably one of the more difficult development environments available, and is direct contrast to the sage advice of starting on small, in-house, 'simple' projects when first attempting agile development (Diefenbach, 2007). There is no guarantee that following that advice would have made ABC successful in this effort, but not following it made their task significantly more difficult.

Figure 2 summarizes these findings:

Lessons Learned from ABC's Lack of Success
Lack of Leadership Commitment can be fatal
No clearly defined and on-site customer creates issues
Need to provide additional benefits to the customer
Do not neglect change management aspects of the transition
Start small – grow from there

Figure 2: Lessons Learned from ABC's Lack of Success

Limitations

This case study has some limitations. It was conducted at a single site and involved interviews with only a small number of agile team members. That small sample size may limit generalizability, and may also introduce bias into the results. Persons interviewed were also directly involved in the attempt at agile software development at ABC, so they likely have their own internal biases, exacerbating the potential bias brought about by the small sample size.

Follow-On Research

It would be interesting to continue this same research at ABC, perhaps expanding it to get more of a management perspective. It would also be instructive to conduct similar research at other organizations who have been less than successful at transitioning to agile methods. There are many examples of positive outcomes in the literature, but not nearly so many with negative outcomes, and it would be interesting to see if this is a true representation of the state of agile development as well as to look for similarities in those organizations who were not successful in their transition to agile development methods.

REFERENCES

1. Adkins, M. (2008) Interview.
2. Beck, K. (2000) eXtreme Programming Explained, Addison-Wesley, Boston.
3. Beck, K., M. Beedle, A. van Bennekum, A. Cockburn, W. Cunningham, M. Fowler, J. Grenning, J. Highsmith, A. Hunt, R. Jeffries, J. Kern, B. Marick, R.C. Martin, S. Mellor, K. Schwaber, J. Sutherland, and D. Thomas (2001) Manifesto for Agile Software Development, Snowbird, UT: Agile Alliance. Available on-line at www.agilemanifesto.org (retrieved October 2005).
4. Boehm, B. (2002) Get Ready for Agile Methods with Care, *IEEE Computer,* 35(1), 64-69.
5. Bradbury, D. (2007) Scrum Down to Get a Project Moving, *Computer Weekly*, 00104787, September 4.
6. Cao, L. and B. Ramesh, Agile Requirements Engineering Practices: An Empirical Study, *Software IEEE*, 25(1), 60-67.
7. Diefenbach, T (2007) The Managerialistic Ideology of Organizational Change Management, *Journal of Organizational Change Management*, 20:1, 126-144.
8. Fruhling, A. and D. Zhang (2007) An Empirical Study Examining the Usage and Perceived Importance of XP Practices, *Proceedings of the Americas Conference on Information Systems*.
9. Fruhling, A. and A. Tarrell (2008) Best Practices for Implementing Agile Methods: A Guide for DoD Software Developers, IBM Center for the Business of Government, Washington, DC.
10. Highsmith J., and A. Cockburn (2001) Agile Software Development: the Business of Innovation, *Computer*, 34(9), 120-122.
11. Iivari, J., R. Hirschheim, and H.K. Klein (2001) A Dynamic Framework for Classifying Information Systems Development Methodologies and Approaches, *Journal of Management Information Systems*, 17(3), 179-218.
12. Leffingwell, D. (2007) Scaling Software Agility: Best Practices for Large Enterprises, Addison-Wesley, Boston.
13. McMahon, J. (2003) Five Lessons from Transitioning to eXtreme Programming, *Control Engineering*, 50(3), 59-65.
14. Mohammadi, S., B. Nikkhahan and S. Sohrabi (2008) An Analytical Survey of On-Site Customer Practice in Extreme Programming, *International Symposium on Computer Science and its Applications*, 1-6.
15. Schatz, B. and I. Abdelshafi (2005) Primavera Gets Agile: A Successful Transition to Agile, *IEEE Software*, 22(3), 26-42.
16. Standish Group (2005) The CHAOS Report into Project Failure, West Yarmouth, Massachusetts: The Standish Group International Inc. Available on-line at http://standishgroup.com/visitor/PDFpages/chaos1994.pdf, (retrieved November 2005).
17. Sutherland, J. (2005) Future of Scrum: Parallel Pipelining of Sprints in Complex Projects, *Proceedings of the Agile 2005 Conference*, Denver Colorado.
18. VersionOne (2008) Annual Survey on The State of Agile Development, www.versionone.com.
19. Watson, R.T., G.G. Kelly, R.D. Galliers, and J.C. Brancheau (1997) Key Issues in Information Systems Management: an International Perspective, *Journal of Management Information Systems*, 13(4), 91-115.