Scientific Research Publishing

Journal of Software Engineering and Applications, 2018, 11, 1-11
http://www.scirp.org/journal/jsea
ISSN Online: 1945-3124
ISSN Print: 1945-3116

Agile Development: Exploring What Practitioners Want to Know

Nanda C. Surendra, Salman Nazir

Management Information Systems, College of Business & Economics, West Virginia University, Morgantown, WV, USA
Email: nanda.surendra@mail.wvu.edu, salman.nazir@mail.wvu.edu

How to cite this paper: Surendra, N.C. and Nazir, S. (2018) Agile Development: Exploring What Practitioners Want to Know. *Journal of Software Engineering and Applications*, **11**, 1-11.
https://doi.org/10.4236/jsea.2018.111001

Received: December 7, 2017
Accepted: January 7, 2018
Published: January 10, 2018

Abstract

Agile development has its origins and roots in practice rather than academia. Hence, in this research, we go to the practitioners' world to explore what they want to know about Agile development. We conducted our study using a multi-methodological approach, a combination of a survey and an interpretive case study. We learned that Agile development is yet to reach a mature phase with: a) relatively limited numbers of experienced Agile practitioners; b) a large number of software developers who were trained in plan-based Waterfall development trying to transition to Agile; c) some companies and practitioners continuing to be skeptical of the benefits of Agile development over plan-based development; and d) tools in the process of being developed to support Agile development. Based on those factors, we learned that practitioners are wanting to find answers to the following questions: a) What should good Agile practitioners and teams know and do? b) How to train developers and teams to become good Agile practitioners? c) How to transition from plan-driven Waterfall development to Agile development? d) What tools are available to practice Agile development? How to use them to support Agile practices? The following are the answers we derived from our interpretive case study for the above four questions: a) Good Agile practitioners define their job in terms of the client's mission; they don't define their jobs as analysts, designers, programmers, testers, or project managers; b) For training, Agile practices such as test-driven development and continuous integration are best understood not in theory but by the act of doing; for organizational adoption of Agile development, Agile practices should be used not only in software development but encouraged in all functional areas; c) The best way to introduce a plan-based Waterfall practitioner to Agile is to have him think of Agile in terms of something familiar, as a series of mini-waterfalls that have very quick iterations; d) Simple tools, even non-software tools, that help manage Agile practices intuitively are preferable to complex Agile management software.

Keywords

Agile Development, Case Study Research, Agile Practice, Scrum, Agile Teams, Software Development, Waterfall Development

1. Introduction

Traditional plan-based software development methodologies have been used by development teams for several decades now. Practitioners have a solid understanding of the techniques and tools used while working with plan-based software development methodologies [1]. However, despite having a solid understanding of these tools and techniques, software development teams often struggle to produce software that is on time, within budget and with all promised functionalities [2]. The primary reason might be that software development is essentially a complex process with the needs to continuously coordinate moving requirements and to manage inputs from multiple stakeholders having non-overlapping knowledge sets [3].

To address these concerns, some companies have incorporated Agile development practices. Although Agile development practices have also been around for several years, companies and practitioners still do not have a clear understanding of how Agile practices can be beneficial to them [4] and this has led to limited adoption of Agile practices [5]. This paper strives to understand the concerns that practitioners might have about the adoption and continued use of Agile development practices. Agile development practices started in industry, several years before Information Systems researchers started studying Agile development [3] [6]. Compared to academia, industry practitioners are much more interested in the practical aspects of innovations. Academia's interest is more of a theoretical nature (producing testable theories, testing models, coming up with definitions), whereas practitioners struggle with issues related to implementation and the pros and cons of adoption of innovations. We believe starting from practitioners gives us a clearer understanding of the issues that practitioners face. With this information, we intend to answer the concerns of practitioners with aspirations to improve adoption of Agile development and use of Agile practices. So, instead of imposing research ideas or theories on Agile development formulated in academia, we decided to start from the source by asking Agile practitioners what Agile development questions were most important to them. Then, we tried to answer these questions by conducting an interpretive case study at an organization that was practicing Agile development.

2. Agile Systems Development

Software developers' interest in Agile software development has grown over the past ten years for two reasons [3]. Developers have become frustrated with the shortcomings of traditional plan-driven development that include inflexibility in

responding to changing requirements, emphasis on unproductive documentation, and slow systems development cycles. They have also become more aware of Agile development since the release of an Agile Software Development Manifesto by a group of Agile development pioneers [1] [3].

There are number of Agile development methodologies with Scrum and extreme Programming or XP being the most prominent and widely practiced [1] [3]. However, all of them share a set of common characteristics: a) acceptance that requirements will change; b) an adaptable development process; c) short iterations that deliver tangible functionality; d) frequent, open communication between software developers and customers; and e) system designs that are as simple as possible.

3. Research Methodology and Data Collection

We conducted this research using a multi-methodological research approach: a combination of a survey and an interpretive case study [7] [8] [9] [10]. Using a multi-methodological research approach helps researchers understand the research question being studied from multiple perspectives. In addition, Information Systems Development is a socio-technical process, involving a complex combination of software, people, development practices and organizational procedures [11]. An interpretive case study helps understand and disentangle this complex process. This research used a combination of a survey of 21 practitioners followed by an interpretive case study of an organization transitioning from plan-driven development to Agile development. We have named this organization Alpha, a pseudonym to help protect the identity of the actual company and its employees. We understand that the relatively small sample of 21 practitioners might be a concern. However, it is less of a concern in this study since we use a qualitative/interpretive approach rather than a quantitative/positivist approach [7] [9].

We started by surveying a group of 21 software development practitioners from several different companies who were attending an intensive three-day workshop on Agile development. One of the authors was a participant-observer at this workshop, attending this workshop in the roles of a software developer doing a deep-dive into Agile practices as well as a researcher learning about what practicing software developers want to learn about Agile development. Since we knew that the practitioners attending the workshop were interested in Agile development, we asked them one simple question: What is your most important question about Agile development you wish answered during this workshop? This question was posed to the practitioners by the workshop leader who stated that, following the Agile practice of putting customers' requirements at the forefront, he would address each of the questions during the course of the workshop. The practitioners were given 15 minutes to think and articulate their question on a notecard. This is approximately the length of time to complete survey questionnaires. However, the problems with questionnaire-based surveys are the fol-

lowing: a) the uncertainty as to whether the respondent is in the correct target group; b) the respondent tends to spend less time and thought answering each question as the number of survey questions increase; and c) there usually isn't any motivation for the respondent to provide reasoned responses. By conducting this survey in person at an Agile development workshop, we avoided the first problem of choosing respondents outside the target group. We overcame the second problem by having the respondents focus on one key question of most importance to them in attending the workshop. The respondents were motivated to think and provide good responses to ensure that their question (the reason for them attending the workshop) was addressed during the workshop.

We analyzed and identified the key recurring themes among the responses (the questions) from these 21 practitioners. We used these key recurring themes to formulate meta-questions. Then, we used the meta-questions as the basis to conduct an interpretive case study of the Agile development practices of a small (about 100 employees) software consulting and development company, Alpha. For our case study, we conducted three face-to-face in-depth interviews with the Chief Executive Officer (CEO) and Chief Information Officer (CIO) of Alpha and exchanged several email messages. The CEO and CIO of Alpha were responsible for initiating a transition over the previous two years from plan-driven Waterfall practices to Agile practices.

3.1. Alpha Company Profile

The following are some of the salient characteristics of Alpha, the company in which we conducted our interpretive case study:

- Alpha was founded in 2007 by two software developers.
- One of the founders is the current CEO while the second is the current CIO.
- The company is based in a large city in eastern United States.
- Alpha has clients from both the private sector and government agencies.
- Alpha has experienced rapid growth and currently has about 100 employees.
- During the first two years after starting, Alpha developed all their software using plan-based Waterfall development.
- Their first project using Agile practices was in 2009.

3.2. Data Collection Methods

The following table lists the data collection methods we used and the duration of each of those methods:

Survey of 21 software developers at a 3-day intensive Agile development workshop.
Participant-observation for three days at the Agile development workshop. Observing, participating in Agile practices with the developers, and communicating with them.
Three face-to-face meetings with the CEO: two at the author's university campus and another at the company's offices (a total of about 5 hours).
One face-to-face meeting with the CIO at the company's offices (about 3 hours).
Six email messages discussing and clarifying Alpha's practice of Agile development.

4. Analysis

We first carefully analyzed the 21 questions provided as responses by the practitioners during the survey about what each practitioner wanted to learn about Agile development. From this analysis and observations during the Agile workshop, we determined that their questions fell into the following three broad categories. Based on these categories, we derived four meta-questions (MQs) that we could use during our interpretive case study. We did not want to overwhelm our interviewees in Alpha with 21 questions, especially since several questions were related.

4.1. First Category of Responses (The Workshop Attendees' Questions)

1) How to train developers to be Agile?

2) How to use an Agile process, such as Scrum, on a daily basis?

3) How can I become a more efficient developer using Agile practices?

4) How to train developers to test and ensure system quality?

5) What is the developer's role in producing releasable code at the end of each sprint?

6) How can my team become more efficient at Scrum?

7) How do teams work towards releasable code each sprint?

8) How does Scrum impact development practices and lifecycle?

9) What is the impact of Agile practices on quality improvement?

10) What are the long-term benefits of Agile practices versus short-term sprint goals commitments?

The questions in this category are best represented by the following two meta-questions:

MQ1: What should good Agile practitioners and teams know and do?

MQ2: How to train developers and teams to become good Agile practitioners?

4.2. Second Category of Responses

1) How to introduce Scrum to an existing team or project?

2) What is a good elevator pitch to people new or resistant to Agile development?

3) How to best encourage buy-in to uncomfortable practices such as test-driven-development and paired-programming?

4) How can scrum teams best collaborate with waterfall teams?

The questions in this category are best represented by the following meta-question: MQ3: How to transition from plan-driven Waterfall development to Agile Development?

4.3. Third Category of Responses

1) How do automated builds fit into the Scrum process?

2) How can we automate production deployments? What kind of tooling is

available?

3) What are the efficiency gains from tooling?

4) How to use automated testing tools as a part of Agile development?

5) How to automate testing using tools?

6) How to perform unit and integration testing, especially in data driven apps? How to use Dependency Injection to perform testing?

7) How to practice source code sharing and control?

The questions in this category are best represented by the following meta-question: MQ4: What tools are available to practice Agile development? How to use them to support Agile practices?

4.4. Interpretive Case Study

We used the meta-questions we derived as the basis to conduct our interpretive case study. Our case study consisted of three in-depth face-to-face interviews with the owner and chief information officer of a software consulting and development company, Alpha. Each interview was preceded and succeeded by the exchange of several email messages to clarify questions and responses.

4.4.1. MQ1: What Should Good Agile Practitioners and Teams Know and Do?

Good Agile practitioners and good Agile teams must understand and define their job in terms of the mission of the end user. As the CIO put it, "They need to understand that everything they are contributing to a project should go toward improving the end users ability to execute their mission: whether that mission is protecting the country's borders or selling more toys."

In plan-based Waterfall development, members of the development team could focus on one specific specialized role such as analyst, programmer, tester, etc. and not have to understand what needed to be done in the other roles. The CIO stated that good Agile practitioners and teams "must understand the full lifecycle of development, which begins at an idea (no requirements) and ends with operations and maintenance". Every member of a good Agile team should understand all the roles, "every developer must understand all facets not involving code and every scrum master/project manager must understand code." Each member of a team should be able to communicate with all other members of the team using the other's "language" (such as requirements analysis, design modeling or testing) and be able to understand what all other members do.

The CIO stated that good Agile developers need to ask questions: "If a user story doesn't make sense, say so. If acceptance criteria seem different than the description, say so." Further, according to the Alpha CIO, good Agile developers should "be able to break a user story into smaller manageable tasks".

4.4.2. MQ2: How Do You Train Your Developers and Teams to Become Good Agile Practitioners?

Most of Alpha's developers were not Agile developers prior to joining the company. The CIO told us that "We prefer to train them, similar to the military…

we want to make sure that everyone is doing it the Alpha way. Then, they can contribute to the improvement of the process." The company had decided to infuse the Agile culture into the entire company by "applying Agile practices to not only software development but all functional areas, Human Resources, Finance and Company Management". In addition, Alpha's CIO told us that "we've also created an Agile Czar who is responsible to work with all employees, from leadership to the newest junior employee to have us all on the same page [for following Agile practices]."

4.4.3. MQ3: How Do You Enable Developers and Teams Used to Plan-Based Waterfall Development Transition to Agile?

Alpha tells developers and teams who have only done Waterfall development previously to think of Agile development as though it is Waterfall development except applied to a few user stories identified in a single sprint. As the CIO explained, "Agile is many small waterfalls"—having the components of requirements, design, development and testing. The difference is that "these mini waterfalls allow us to course correct much faster."

4.4.4. MQ4: What Tools Do You Use to Practice Agile Development?

Alpha followed Scrum for their Agile practices. Since Alpha's developers did most of their development using Microsoft programs such as SharePoint, C# and SQL Server, they first used Microsoft's Visual Studio Team Services as their Agile project management tool. However, after a few months, they had started experimenting with an open source Agile Project management tool, Jira. The CEO updated us after a few months stating, "Since Jira is much more focused on Agile, we have switched from Visual Studio Team Services to Jira." In addition to the software tools, the CEO said that he considers "the Sprint Ceremonies such as Sprint Planning, Daily Scrum, Sprint Demo/Review, Sprint Retrospective are also tools used to practice and manage Agile development."

5. Discussion

The key lesson we learned about the characteristics of good Agile practitioners (MQ1) is that they define their job in terms of the end user's or client's mission. Good Agile practitioners don't define their jobs as analysts, designers, programmers, testers, or project managers. They don't focus on a specific role and are willing to accept any role that is needed at a given time to help fulfill the client's mission. This means a good Agile practitioner should have multiple, diverse skills and a knowledge base that allows him to function as an analyst, designer, programmer, tester or project manager. Having multiple skills allows the development team members to get involved in all phases of development, which motivates them to take up responsibility of the overall project rather than one individual module which may not integrate well with the rest of the application [2] [12]. This capability to take on multiple, diverse skills has the potential to reduce the three myopias of learning: temporal, spatial and failure [13]. By being

involved in the overall project, team members become aware of the larger scope of the project, rather than one small module, thus facilitating course corrections to fix minor issues before they become major pitfalls. In addition, working with a diverse skillset allows team members to enhance their overall absorptive capacity [14], thus allowing them to understand novel customer requirements and providing innovative solutions to development issues.

Another lesson we learned about being a good Agile developer is the ability to break a large user story into specific tasks. Since a key element of Agile development is testing, it is not possible to know if a test succeeds or fails if an Agile developer is unable to define when a task is "done". Hence, the need for good Agile developers to be able to break a user story, which could be large and not have a specific definition of "done" into smaller tasks, each of which have a clear definition of "done".

Regarding training employees to become good Agile practitioners (MQ2), Alpha said that experienced Agile developers were relatively rare to find and expensive to hire. So, Alpha hired mostly developers who did not have an Agile background and then trained them in Agile by mentoring them on the job. Alpha has an "Agile Czar" who communicates and works with all teams and members of each team to ensure that each team and every member of a team fully understand Agile development principles and practices. We learned two main lessons from exploring MQ2 at Alpha. One lesson is that Agile practices such as test-driven development and continuous integration are best understood not in theory but by the act of doing. The second lesson is that for an organization to truly benefit from Agile development it should try to infuse the Agile culture into the entire organization as Alpha did by applying Agile practices to not only software development but all functional areas.

The lesson we learned about helping plan-based Waterfall teams and practitioners transition to Agile development (MQ3) is to reduce the intimidation factor and lower the barriers to entry. The best way to introduce a plan-based Waterfall practitioner to Agile is to have him think of Agile in terms of something familiar, as a series of mini-waterfalls that have very quick iterations, measured in days to a few weeks compared to the time scale of months or years they are used to. Lack of knowledge about how Agile practices can improve the software development process or how they can be used to benefit the user's mission is cited as a key barrier to the adoption of Agile development [4]. The transition to Agile development from traditional Waterfall approach is akin to changing the team's mindset which causes a natural resistance to the transition [15] [16]. It is very likely that during such a transition, managers might be wary of giving up control as they are more used to the control and command based kind of approach to handle the software development project. At the same time, other team members might feel overwhelmed by the additional responsibility of being involved in the overall software development project [17]. It is, therefore, suggested that handling the transition by thinking of Agile in terms of a series of

mini-waterfalls that have quick iterations might help with the transition.

We derived two lessons from our analysis regarding tools used to practice Agile development (MQ4). The first lesson was that an Agile management tool that in Zuboff's terms is "ready-to-hand" is going to be preferred over a tool that is "present-at-hand" [18] [19]. A ready-to-hand tool can be used to accomplish a task or objective without thinking about the tool, while a present-at-hand tool forces the user to constantly figure out how to make the tool accomplish the task or objective. Since Agile development is very much oriented toward "ready-to-hand" practice (the Agile practitioner's focus being on achieving the client's mission), any tool that is not ready-to-hand is likely to be discarded by Agile developers. Alpha started preferring an open source tool, Jira, that was ready-to-hand over the Microsoft tool, Visual Studio Team Services, that was deemed present-at-hand even though they used Microsoft products for all their software development. The second lesson we learned was that we should not think of tools used for Agile development just in terms of software tools. Practices that did not necessarily involve a software tool, such as Sprint Planning, Daily Scrum and Sprint Retrospective, are also to be considered important tools for enabling good Agile development.

6. Conclusions

Conducting an interpretive multi-method study of Agile practitioners, we found that they have several questions about how to realize the full potential of adopting Agile development. From our survey, we derived that practitioners want to know the how and why of Agile development as well as Agile best practices. They would like to know how they can train newly hired employees and newly formed teams to succeed in Agile as well as learn how to help transition team members from plan-based Waterfall development to Agile development practices. Another recurring question was about tools that enable the practice and management of Agile projects.

Our interpretive case study allowed us to learn several important lessons. It was found that in order to be strong Agile practitioners, team members must shed their old habits of defining their jobs by roles. They should be willing to put client's needs as their priority and take on any role for the team that helps successfully develop the client's functionality. They should have diverse skillsets that allow them to take responsibility of different phases of the project. Practitioners should also be able to divide user stories into smaller sub-level tasks that can have a clear definition of "done" such that they are easily testable. We also learned that the organizational culture is an important facilitator of successful adoption and use of Agile practices. Since Agile is learned best by doing, organizations that foster a culture of mentorship and collaboration will have the benefit of experienced team members working with junior colleagues, showing and teaching them how Agile can be used in practice. Infusing Agile practices into all functional areas of the organization helps achieve a smoother transition to Agile

development and educate skeptics about the benefits of Agile. Team members that are familiar with a plan-based development methodology can be helped in transitioning to Agile practices by allowing them to think about Agile as a series of mini Waterfalls done in short intervals. Tools that are "ready-to-hand" rather than "present-at-hand" are deemed more beneficial in adopting Agile practices. Since Agile development practices are focused on providing value for the client, the additional cognitive burden of a present-at-hand tool acts a barrier to the achievement of that goal.

Limitations and Future Research

As with any study, this work has some limitations that must be noted. The first limitation is related to sample size. Our survey was administered to 21 practitioners which is not a significant number. However, it must be noted that for a qualitative study this sample size is sufficient. Since the survey was open-ended, and the respondents were given a prolonged response time, we believe we were able to elicit quality responses that helped shape our understanding of the phenomenon better. Future works can extend this study by testing the findings with larger sample sizes. Quantitative studies can also extend the findings of this research by assessing the relative importance of each of the practitioners' questions in terms of its effect on successful Agile adoption.

Another limitation of the study is the fact that it focuses only on Agile development issues and does not compare those findings with the issues related to the Waterfall approach of software development. Future work should extend this line of work by simultaneously exploring the questions and issues related to both, the Waterfall approach as well as the Agile approach.

Finally, our interpretive case study at Alpha was conducted after it had already adopted the Agile approach of software development. This did not enable us to see the actual process followed by it while transitioning from the Waterfall to the Agile approach. Future works should strive to address this issue by studying this process of transitioning as it would be extremely beneficial seeing first-hand the problems as they occur in organizations during the transition process. Longitudinal studies of the actual process of transition from plan-driven Waterfall development to Agile development would help better understand the process and thereby provide better insights into what can be done to help ease the transition.

References

[1] Cohen, D., Lindvall, M. and Costa, P. (2004) An Introduction to Agile Methods. *Advances in Computers*, **62**, 1-66. https://doi.org/10.1016/S0065-2458(03)62001-2

[2] Maruping, L.M., Zhang, X. and Venkatesh, V. (2009) Role of Collective Ownership and Coding Standards in Coordinating Expertise in Software Project Teams. *European Journal of Information Systems*, **18**, 355-371. https://doi.org/10.1057/ejis.2009.24

[3] Sutherland, J. (2014) Scrum: The Art of Doing Twice the Work in Half the Time. Crown Business, New York.

[4] Gandomani, T.J. and Nafchi, M.Z. (2016) Agile Transition and Adoption Human-Related Challenges and Issues: A Grounded Theory Approach. *Computers in Human Behavior,* **62**, 257-266. https://doi.org/10.1016/j.chb.2016.04.009

[5] Maruping, L.M., Venkatesh, V. and Agarwal, R. (2009) A Control Theory Perspective on Agile Methodology Use and Changing Requirements. *Information Systems Research*, **20**, 377-399. https://doi.org/10.1287/isre.1090.0238

[6] Lechler, T. and Yang, S. (2017) Exploring the Role of Project Management in the Development of the Academic Agile Software Discourse: A Bibliometric Analysis. *Project Management Journal*, **48**, 3-18.

[7] Myers, M.D. (2009) Qualitative Research in Business & Management. Sage Publications, Thousand Oaks, CA.

[8] Tsang, E.W.K. (2014) Case Studies and Generalization in Information Systems Research: A Critical Realist Perspective. *The Journal of Strategic Information Systems*, **23**, 174-186 https://doi.org/10.1016/j.jsis.2013.09.002

[9] Walsham, G. (1995) Interpretive Case Studies in IS Research: Nature and Method. *European Journal of Information Systems*, **4**, 74-81. https://doi.org/10.1057/ejis.1995.9

[10] Yin, R.K. (2003) Case Study Research. 3rd Edition. Sage Publications, Thousand Oaks, CA.

[11] Oates, B. and Fitzgerald, B. (2007) Multi-Metaphor Method: Organizational Metaphors in Information Systems Development. *Information System Journal*, **17**, 421-449. https://doi.org/10.1111/j.1365-2575.2007.00266.x

[12] Beck, K. (2000) Extreme Programming Explained. Addison-Wesley, Reading, Massachusetts.

[13] Levinthal, D. and March, J.G. (1993) The Myopia of Learning. *Strategic Management Journal,* **14**, 95-112. https://doi.org/10.1002/smj.4250141009

[14] Im, G. and Rai, A. (2008) Knowledge Sharing Ambidexterity in Long-Term Inter-Organizational Relationships. *Management Science*, **54**, 1281-1296. https://doi.org/10.1287/mnsc.1080.0902

[15] Cohn, M. and Ford, D. (2003) Introducing an Agile process to an Organization. *Computer*, **36**, 74-78. https://doi.org/10.1109/MC.2003.1204378

[16] Pikkarainen, M., Salo, O., Kuusela, R. and Abrahamsson, P. (2012) Strengths and Barriers behind the Successful Agile Deployment-Insights from the Three Software Intensive Companies in Finland. *Empirical Software Engineering*, **17**, 675-702. https://doi.org/10.1007/s10664-011-9185-5

[17] Nerur, S., Mahapatra, R. and Mangalaraj, G. (2005) Challenges of Migrating to Agile Methodologies. *Communications of the ACM*, **48**, 72-78. https://doi.org/10.1145/1060710.1060712

[18] Zuboff, S. (1985) Automate/Informate: The Two Faces of Intelligent Technology. *Organizational Dynamics*, **14**, 5-18. https://doi.org/10.1016/0090-2616(85)90033-6

[19] Zuboff, S. (1988) In the Age of the Smart Machine: The Future of Work and Power. Basic Books Inc., New York.