Challenges of Agile Adoption and Leadership Roles in Agile Adoption Success

Rachel (Thu) Le

University of Washington

Abstract

The rapid development of technology and the increase of complexity in designing software systems in the past decade has caused the shift from traditional approach of Waterfall or life-cycle development model to Agile methodologies such as Scrum and Extreme Programming (XP). In many projects, Agile methods have shown to be successful and reduce wasted effort, time, money and the product is delivered sooner in frequent increments that are easily adapted to change. However, Agile brings the most success when it is used for small and individual teams. When adopting agile, larger scale companies with multiple teams, especially teams in geographically distributed sites often face many challenges that can ultimately lead to failure of the effort. In this paper, I present a literature review of the challenges, success factors of Agile transition in large organizations as well as leadership roles in its success. A total of eight papers regarding challenges, success factors and leadership roles in Agile adoption and scaling that are published in the past 5 years are examined. Most of the reports are based on interviews and surveys from agile adopters and practitioners which shows that there is currently insufficient effort to do thorough academic research about the topic. The most common challenges are human-related issues such as resistance to change due to misconceptions about Agile and wrong mindset, the difficulty of creating self-organizing teams and the lack of management support. These multi-dimensional and interlinked challenges can potentially be addressed starting from leader roles in teams and organizations.

Keywords: agile transition, agile adoption challenges, agile leadership

INTRODUCTION

Software Development Life Cycle (SDLC), with its earliest and most well-known version Waterfall model, is a framework for developing software that helps deliver software in one single project by going through multiple sequential development stages such as requirement analysis, design, implementation, testing, deployment and maintenance. On the other hand, agile approach with its most popular versions, Scrum and XP, builds product by repetitions of the same development stages in multiple increments in a shorter period of time. SDLC requires upfront specified requirements that are hardly changed once the development cycle starts while agile approach provides flexibility by allowing changes to be introduced during or in between iterations. Customer participation and feedback during development also contributes to more successful and high-quality products as it better reflects customer needs and visions. Last but not least, the early discovery of defects and adaptability to changes in product and direction by doing smaller frequent iterations help deliver product with high value and lower cost, time and effort.

Due to these advantages of agile, SDLC is losing its popularity and more companies are adopting agile for developing software products. Startup companies with small teams are well-known for using agile methodologies such as Scrum. On the other hand, larger and more established organizations attempt to become agile through adopting frameworks such as Scaled Agile Framework (SAFe), Scaling Scrum with Large Scale Scrum (LeSS) and Scrum-of-Scrum. However, they often face a great deal of challenges when adopting agile to fit their scope. In addition, compared to large scale companies that utilize agile since starting up, ones with long history of using SDLC have even more difficulties due to existing structure that is often in conflict with agile values and principles.

Agile adoption comes with the change in people's perception about agile values and processes, with understanding of agile values as the most important aspect before adopting agile processes and tools. The focus on agile values shows that change requires a huge shift in mindset, attitude and behaviors. This is often difficult, takes a long time and the attempt can fail due to humans' natural tendency to resist change and maintain our old way of working. This paper will discuss common challenges among all selected papers for literature review such as resistance to change, wrong mindset, poor Agile customization, cultural and organizational issues as well as team-related problems such as lack of collaboration, commitment and teamwork. The discussed solutions are team-related approaches such as shared view on values and practices, effective communication and coordination as well as solid engineering practices. The paper also provides my own self-reflection regarding the Gacha Game development experience.

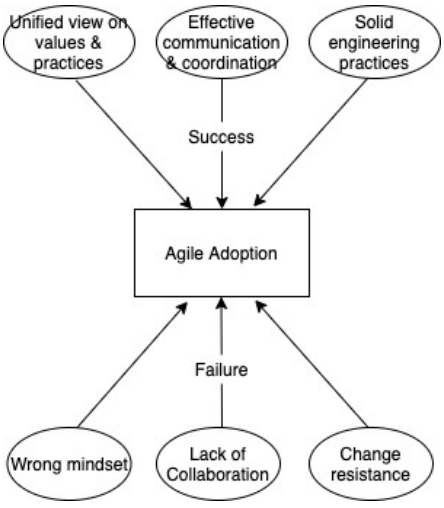

Figure 1. Challenges and Success Factors of agile adoption for Software Management class CSS 566 (excluding Poor Agile Customization and Cultural/Organizational Issues as we did not have these challenges in our class environment)

CHALLENGES

Change Resistance

In selected studies, authors have their own way of sorting challenges of agile adoption into different categories. However, one common theme is that a challenge can be a result or a cause of another challenge as these challenges are often multi-dimensional and interlinked. For example, every selected study mentions change resistance as an impediment to successful agile transformation (Kalenda, Hyna, & Rossi, 2018, p. 8; Dikert, Paasivaara, & Laasenius, 2016, p.92; Gandomani & Nafchi, 2016, p. 261). In Dikert's publication, resistance to change is the number three out of nine most commonly reported challenges (Dikert, Paasivaara, & Laasenius, 2016, p. 95). There can be multiple reasons behind change resistance. First and foremost, it is human nature to resist change and maintain old way of thinking and working. Second, some members do not see the benefits of adopting agile and are skeptical if agile would even work when their current way of working is already "good enough" to deliver product to customers. From the team aspect, many team members are used to being told what and how to do things due to managers' command and control style so they do not know how to operate in self-organizing teams which comes with new responsibilities. Many also believe the increased team transparency means they are being watched by others. For team members who want to contribute their creativity and decision making, fear of job insecurity and risk taking can prevent them from change. On the management side, managers have to transition from micromanaging and giving orders to giving freedom to team and as a result, they fear they would lose power and not be needed anymore (Dikert, Paasivaara, & Laasenius, 2016, p. 96; Gandomani & Nafchi, 2016, p. 261) Additionally, when agile adoption is forced upon the team by management without first

inspiring and helping them realize the values and benefits behind agile, team is likely resist change.

Besides reasons mentioned above, change resistance can also stem from past experience of failed agile adoption. An example of this is my classmate, James. He had a bad experience with Scrum implementation at his workplace that resulted in him quitting his job. During James' presentation, he mentioned the lack of support from Scrum Master, Product Owner, middle and upper management as well as the customers. In fact, everyone was "working against the developers" (Zhang, 2020). With Scrum implemented in the wrong way, James did not believe that it would work and his skepticism and loss of hope in Scrum might impede his acceptance of Scrum process (that could have been likely to succeed) at another workplace.

Wrong Mindset

Another challenge that easily results in failure of agile adoption is having the wrong mindset. Wrong mindset can manifest as adopters being overly enthusiastic about agile methodologies and having unrealistic expectations such as thinking agile would solve every problem (Gandomani & Nafchi, 2016, p. 262-263). Other manifestations can be managers relabeling old practices with agile terminology or teams utilizing agile tools and processes with no understanding the underlying purpose and values while keeping their own way of thinking. Consequently, this creates a false sense of doing agile. Sometimes, doing agile wrong is even more harmful than not doing it at all as wrong mindset and behaviors are difficult to correct once they are established. James' experience again could be an example of Scrum Master and management having the wrong mindset about Scrum. The reason is they tried to implemented agile methodology but did not embrace its value. Particularly, the Scrum Master cared about his

or her personal interest and did not establish trust with the developer teams while the Product Owner accepted whatever customers asked and wanted developer teams to satisfy all customers' needs (Zhang, 2020). There was no support and promotion of freedom for self-organizing team; leadership roles were still in the command-and-control mode. A quote from a Head of Development Department in Gandomani's (2016) article summarizes the wrong mindset by many different roles:

> "The number one and huge cultural problem is pretty much everyone having the wrong mindset. Managers think command and control which is excellent for managing work that is about execution [but] is unsuitable for design and development. Customers expect to purchase new product design, development work with fixed price, fixed scope contracts. Legislators assume the same. So basically, all parties related to software product development mostly get it completely and hopelessly wrong." (p. 262)

Poor Agile Customization and Cultural/Organizational Issues

Agile being difficult to implement as the number one reason for unsuccessful agile adoption includes lack of literature research in agile scaling guidance (Dikert, Paasivaara, & Laasenius, 2016, p. 96-97 & 101). There exist frameworks for scaling agile such as SAFe, LeSS and Scrum-of-Scrum but following by-the-book agile processes and practices without consideration for organizational environment and dynamics unique to current team(s) will likely result in failed agile adoption. In addition, it is not easy to improvise agile approach without the expense of weakening agile principles and this requires a high level of agile experience as well as deep understanding of specific company and team's culture and politics.

This problem is not present in our class but it is somewhat related to James' experience as well. James discussed having too many meetings that were unnecessary. Instead of fostering communication, it was used as progress report for Scrum Master. The report solely discussed what was done and not done and Scrum Master did not ask if teams need help or proactively reached out to remove impediments for teams. (Zhang, 2020) In his case, the agile processes were being followed but the true agile values were lost.

Organizational culture also plays an important role in the success of agile transformation process (Dikert, Paasivaara, & Laasenius, 2016, p. 98; Gandomani & Nafchi, 2016, p. 261-262). For example, a large company wants to experiment Scrum on one specific team and if successful, it will replicate the progress gradually to more teams. Despite the company's leaning towards agile adoption, middle and higher management are still using command and control style instead of giving freedom and allowing team to self-organize. This results in a misalignment and conflict between new agile values of teams and current old leadership/management style. The lack of understanding and support by management to move towards agile can be a tremendous impediment to successful agile adoption. A Scrum Master/Project Manager reported in Hoda's (2016) article:

> He (reporting manager) was not involved, but he [comes] along with the team, finds out how things are going. (p. 250)

Lack of Collaboration, Commitment and Teamwork

The transition to newer way of thinking requires extra workload on top of current development work such as collaboration and work coordination inside and across teams. Adding other layers of difficulty such as teams located in multiple sites, learning new skills to be cross-

functional and implement continuous integration, the constant pressure to show meaningful results from agile adoption as well as the lack of support and resources, it is hard to make agile work for multiple teams and easy for teams to lose motivation and commitment to change (Kalenda, Hyna, & Rossi, 2018, p. 8). Last but not least, the lack of transparency, effective communication and information sharing still poses as one of the team-related challenges in agile adoption (Dikert, Paasivaara, & Laasenius, 2016, p. 102; Gandomani & Nafchi, 2016 p. 262).

At the beginning of Gacha Game Developement, the Product Owners (me, Carla and Prianka) worked together to generate requirements and had requirement clarification during class in one session. There was not much communication between teams as well as between Product Owners, Scrum Master and the teams after that. Some teams reached out for questions but chose very specific Product Owner. Most went to Carla and some went to me. I did not see any team representatives asking Prianka. Product owners also did not discuss with each other the interactions they had with the teams when questions came. As we moved to Zoom meeting due to the coronavirus outbreak and the teams developed more solid parts of the product that were testable, communication happened more frequently and naturally. In addition, at this point, everyone had worked together, knew each other more and had become friends outside of class. This helped establish interpersonal relationship and built trust between the team members and other roles. Initially, the development started in a chaotic scene but became progressively more stable with an increase sense of collaboration and coordination.

SUCCESS FACTORS, LEADERSHIP ROLES AND SELF-RELFECTION

Success Factors

Fortunately, we did not have to deal with challenges related to company culture and politics in our class. Therefore, the discussion will focus on success factors related to team as well as other roles such as Product Owner and Scrum Master.

First, having a united view on values and practices is important to successful agile teams. This includes everyone sharing the same product vision and understanding of Scrum methodology, their roles and responsibilities as well as communicating using the same language (Kalenda, Hyna, & Rossi, 2018, p. 10).

Another success factor is solid engineering practices (Kalenda, Hyna, & Rossi, 2018, p.10). Some examples are standards for documentation and README file. During one Zoom meeting, I suggested improving instructions for Product Owner to test product to be more readable and easier to follow while showing an example GitHub repository. Most teams were receptive to my feedback and changed their README files accordingly.

Last but not least, effective communication and coordination that allows cross-team cooperation and consistency before even establishing team autonomy is crucial to agile adoption (Kalenda, Hyna, & Rossie, 2018, p. 17). As mentioned before, as communication increased in frequency over time, teams collaborated more effectively, trust was established, and team culture started to emerge. Additionally, communication also promoted early feedback and product improvement. When I told the teams about my testing experience (with some functionalities not working), they shared they had not merged the branch they were working on with the master branch and suggested testing that specific branch instead (as it had more working functionalities). If I did not share my testing feedback, I would have missed that.

Leadership Roles and Self-Reflectio

Reflecting on the experience, there are many things I want to do differently or reconsider next time. According to Alahyari et al, effective team work is importance for success of agile teams (Alahyari, Horkoff, Matsson, & Egenvall, 2018, p. 1). In team collaborative, the first thing is communication, especially communication tool(s). As we only met in person twice a week, most interactions happened outside of class. Early establishment of communication tools that everyone agrees on using and how often it is used are important. From checking Slack channels often (without intruding and asking about teams' work progress), I found most teams were active in communicating what each member was doing as well as if anything broke or needed fixing. However, database team did not utilize Slack nor have online interactions with other teams. The agreed-upon tool of communication was not fully utilized across teams. Another thing I would change with communication as a Product Owner is proactively and frequently reaching out to teams and asking if they need help or further clarification, instead of waiting for them to come to me.

Hoda mentioned that lack of acceptance area is one of the challenges in software management and good requirement generation and clarification is a skill that takes time to develop (Hoda & Murugesan, 2016, p. 252).

This brought me to the second point. As a Product Owner, I would want to gain more knowledge about user story creation and estimation to be prepared for making the Product Backlog and requirement refinement during Sprint Planning. In fact, this task of creating user stories and requirement is one of the requirement engineering challenges for scaling agile (Dikert, Paasivaara, & Laasenius, 2016, p. 98). Towards the end of the quarter when we Product Owners started the testing process, checking if database requirement was satisfied became a

difficult task as they were vague and not technically detailed enough. We only specified database as maintaining data security and integrated with the Gacha Game application.

Third, as Product Owner, I realized we needed to develop acceptance test along with the game development and we did not accomplish that in this class. For future classes and projects, we can improve by meeting more frequently outside of class in person or online with all the teams. During the meeting, we can walk through the build, integration and the usage of the product.

Last but not least, even though we thrived to be cross-functionality, in reality, teams still selected requirements based on domain such as user interface, backend and database. From my observation, the user interface and backend teams collaborated effectively and were engaged outside of class. However, the database team seemed to work more on their own and were not as engaged during in class and online. I was wondering if their database work resulted in less overlap and opportunity to collaborate compared to the other three teams. We also did not evaluate if all teams were happy with the work they chose (or had to choose due to other teams selecting it first). If I could redo the game development, I would suggest excluding database from the first Sprint so everyone could work on other aspects that require lots of contact between teams for successful product delivery. In addition, most students met each other for the first time and a sense of trust and friendship was not yet developed. This takes time. Designing the first few Sprints to allow team culture and norms to emerge, I think, is crucial for future success of agile teams and projects. Lastly, for some members who had bad experience with Scrum, resulting in skepticism (which can potentially decrease likelihood of successful agile adoption), making sure the first few Sprints are positive and empowering experiences is extremely important.

CONCLUSION

In summary, agile adoption is mostly a long and challenging transition due to many factors such as change resistance, wrong mindset and lack of effective communication and team coordination. With these in mind, we can increase the likelihood of successful agile adoption by having shared view on values and principles, establishing solid engineering practices and effective communication.

References

Alahyari, H., Horkoff, J., Matsson, O., & Egenvall, K. (2018). What Do Agile Teams Find Important for Their Success? *2018 25th Asia-Pacific Software Engineering Conference (APSEC)*. 474-483. https://doi.org/10.1109/APSEC.2018.00062.

Dikert, K., Paasivaara, M., & Laasenius, C. (2016). Challenges and success factors for large-scale agile transformations: A systematic literature review. *The Journal of Systems and Software*. 87-108. https://doi.org/10.1016/j.jss.2016.06.013.

Gandomani, T. J., & Nafchi, M. Z. (2016). Agile transition and adoption human-related challenges and issues: A Grounded Theory approach. *Computers in Human Behaviors, 62.* 257-266. https://doi.org/10.1016/j.chb.2016.04.009.

Hoda, R., & Murugesan, L. K. (2016, March 18). Multi-Level Agile Project Management Challenges: A Self-Organizing Team Perspective. *Journal of Systems and Software*. p. 245-257. https://doi.org/10.1016/j.jss.2016.02.049.

Kalenda, M., Hyna, P., & Rossi, B. (2018, March 15). Scaling agile in large organizations: Practices, challenges, and success factors. *Journal of Software: Evolution and Process*. https://doi.org/10.1002/smr.1954.

Zhang, J. (2020). Why Scrum Doesn't Work. *Week 10 Book Cafe Presentation.* Seattle, WA.