

Adopting Agile Approach : A Case in Malaysia

Mazni Omar¹, Sharifah-Lailee Syed-Abdullah², Azman Yasin³

^{1,2} Faculty of Computer Sciences and Mathematics,
Universiti Teknologi MARA (UiTM),
02600, Arau,
Perlis, MALAYSIA
mazni@isiswa.uitm.edu.my, shlailee@perlis.uitm.edu.my

³ Applied Sciences Division
UUM College of Arts and Sciences,
Universiti Utara Malaysia (UUM),
06010 UUM Sintok,
Kedah, MALAYSIA
yazman@uum.edu.my

ABSTRACT

Agile methodology continues to evolve due to business and technological demands. However, there are few studies reported related to encouraging new organizations to adopt agile practices especially in Asian countries. This paper presents the initial findings from an action research study on agile-extreme programming (XP) adoption amongst software engineering (SE) teams. The results showed that the agile-XP approach received positive feedbacks amongst team members during the software project, even though the level of adoption was mixed. These findings offer valuable insights into aspects related to agile approach to suit an organizational culture.

Keywords

Agile, XP, SE teams, adoption, organizational culture

1.0 INTRODUCTION

Agile methodology emerged due to evolving and changing software requirements changing software requirements. In addition, people are becoming more interested in developing software that better fits their business requirements. Agile methodology accommodates iterative and incremental development by incorporating human values through effective communication and feedbacks.

Extreme Programming (XP) is the most prevalent in agile software development methodology (Cockburn, 2007; Salo & Abrahamsson, 2008). XP has gained a lot of attention and is a widely used agile methodology in industry. There were originally twelve major practices in XP: the planning game, pair programming, refactoring, simple design, continuous integration, test-first programming,

collective ownership, coding standards, short releases, metaphor, sustainable pace and on-site customers (Beck, 2000). These practices were based on four XP values, which are communication, simplicity, feedback and courage. However, as these practices evolved, Beck and Andres (2004) further divided XP practices into two main categories, which are primary and corollary practices. Mastering the primary practices is necessary to easily adopt the corollary practices.

The emergence use of agile practices in industry were more focused on the agile perception (Aveling, 2004; Murru, Deias, & Mugheddu, 2003; Rasmusson, 2003) and comparison studies (Ilieva, Ivanov, & Stefanova, 2004; Layman, 2004; Sillitti, Ceschi, Russo, & Suchi, 2005). Applying and evaluating the agile effectiveness in industry highlighted whether its practices can increase the quality, productivity but decrease the software life cycle time. Most studies conducted in the United States (Layman, 2004; Layman, Williams, & Cunningham, 2004) and European (Ilieva, et al., 2004; Murru, et al., 2003; Salo & Abrahamsson, 2008; Sillitti, et al., 2005) companies used XP or Scrum. However, there were fewer studies reported on agile practices in Asian countries. Therefore, this paper presents the authors' approach in educating and training four SE teams of one university computer centre in Malaysia to adopt agile practices in their new software projects.

2.0 RELATED WORKS

The variations in the rate of adoption of agile methodology mostly have been studied in the United States of America and European companies. The suitability of using agile practices in large organization was conducted by Lindvall et al. (2004) at four big organizations which were ABB,

DaimlerChrysler, Motorola and Nokia. The study showed that XP allowed developers to gain positive experiences because XP was easy to understand and therefore increased teams' morale. However, these organizations need to define an agile process tailored to their current software practices to avoid conflict of interest in developing software.

Not all XP practices were fully adopted by software development teams. Some of the practices such as continuous integration and coding standards were easy to adopt by the teams because the practices are common programming practices. However, practices that involved others parties such as on-site customer, planning game, and small release were difficult to adopt (Aveling, 2004). Positive experiences and feedbacks in applying agile approach especially XP in software organization have tremendously reported in literature review such as promoting highly collaborative and coordination teams, improving learning experiences, and receiving satisfactory relationship with customers (Murru, et al., 2003; Sharp & Robinson, 2008; Sillitti, et al., 2005)

Researchers investigated underlying reasons that prevent agile approach to be successfully adopted. Aveling (2004) claimed that level of XP adoption can be categorized into three major categories; insufficient discipline, failure to understand the practices and failure to persuade third party stakeholders, which perceived that the third category often led to project failure. Less support from the organization prevents the practices to be applied effectively. This shows that organizational culture plays important roles more than the technical issues. In addition, McAvoy and Butler (2009) indicated that learning to adopt agile is more than cognitive process, which requires software development team to change their behaviour, attitudes and opinions during the software development activities. These explain that changing people to adopt a new approach is challenging tasks as it deals with human factors. Agile is a team-oriented, therefore having a good combination of people will always lead to a successful software projects (Mazni, Sharifah Lailee, & Azman, 2010; Rasmusson, 2003).

3.0 METHODS

This study aims to investigate the adoption of agile practices in a university computer centre in Malaysia. In this study, four software development teams were observed for a period of time from May 2009 until December 2009. Each team consists of two system analysts and four programmers. This made up eight system analysts and sixteen programmers as the participants in this study. Every participant was informed about the research conducted and nature of the study.

3.1 Background of case study

A university computer centre in Malaysia was selected for this study. Initially, all staff, consisted 11 system analysts and 20 programmers, were selected to participate in 'Embracing XP Project'. The aim of this project was to explain to them an alternative approach to developing software that is simpler and lighter. This project came about because one of the deputy heads who was in charge of applications in this centre was an 'XP convert'. The manager's prior involvement with XP started when he was one of the project advisors in the earlier study (Sharifah Lailee, Mazni, Mohd Nasir, Che Latifah, & Kamaruzaman, 2009). He was impressed with the XP practices that allow a high degree of collaboration among software development teams. Collaborative teams are important to build knowledge sharing among team members, which can promote effective and creative solutions during software development.

All teams were required to use the Java Servlet Pages (JSP) language in their proposed applications, in accordance with the computer centre strategic planning. Since the language and SE approach were new to all the team members, all programmers were sent to workshops before the project started.

In this study, eight system analysts and sixteen programmers making up four teams were chosen to proceed with the agile-XP case study. The selection of these participants was based on their commitment towards completing the projects.

3.2 Agile Knowledge

Before embarking on the project, agile workshops were conducted to introduce and explain to the software development teams the agile-XP practices. This workshop on XP programming was attended by all participants. The workshop addressed the theory and the practical aspects of agile software development and was divided into three sessions. The first session was for the system analysts, which focused on the theoretical aspects of XP, where simple design document was introduced to facilitate documenting the project development work. The document consisted of a Unified Modeling Language (UML) using case diagram, entity relationship diagram (ER-D) and interface designs. The second session was for the programmers, which addressed the XP practices. In addition to the XP theory, the session also included hands-on tasks for a planning game and pair programming to assist the programmers in understanding better the XP activities. The third session on XP combined both system analysts and programmers in teams according to the selected projects. In this session, the team leader (system analyst) and members (programmers) were asked to start the project by completing story cards.

3.3 Data collection and analysis

This study used action research approach. Data were collected through a series of interviews and short-term observations. The action research approach allows a flexible and responsive strategy that helps improve the learning experiences and practices of the participants. The observations were carried out for six-months to improve the understanding of the working culture in the computer centre under a natural setting.

Interviews were carried out using semi-structured interviews with selected participants to ascertain that each team understood and applied agile-XP practices as much as possible. Every interview session lasted between five to 10 minutes. The interviews were audio recorded and transcribed after each session. Follow-up questions were asked via face-to-face meetings when clarifications were necessary. During the interview sessions, the problems that were encountered were discussed.

3.4 Data validity

Construct validity was achieved by seeking clarification of the person interviewed once the interview was transcribed. In dealing with reliability issues, all procedures such as interview guidelines, field notes, and questionnaires were documented. This is to ensure that the study can be easily replicated and same conclusions can be drawn by other researchers.

4.0 FINDINGS AND DISCUSSION

There were mixed responses when the participants were asked about the adoption of agile-XP practices. Some of the participants claimed that the agile methodology was a strict and disciplined approach that required them to follow the practices closely, but others argued that the practices were not new for them. However, it was revealed that the practices promoted effective communication and sharing values amongst the members.

“Before this, system analyst will develop system design alone, then he or she will discuss with programmers...but in this activity [planning game] we discuss it together ...so, we have more ideas to create the design and user can understand better...programmers also easily understand the system flow...” [Team member1]

“We interact each others...when one of members in team has more experience, he or she will share...” [Team member2]

4.1 Adoption of Agile Practices

During the **planning game** activities, it was observed that all teams were able to discuss and write the **story cards**. Even though some teams

claimed the activity was easy, others found the story cards to be cumbersome. It was observed that the story card acts were only as a foundation for starting the project, but as the project progressed the story cards were not used because members found them impossible to use for tracking requirements and rescheduling projects. This observation is similar to earlier findings that writing story cards without estimating the cost to implement a project was difficult (Wood & Kleb, 2003). It was interesting to observe that the team members were more comfortable using the interface design instead of the story cards because the interface designs are diagrammatic in nature. Therefore, it was much easier for members to discuss with clients and also among members concerning requirements that could be visualized by both parties. This is in accordance with the earlier research on the advantages of diagrams over text (Ainsworth & Loizou, 2003; Schnotz, 2002).

The teams realized the importance of **testing**. However, the first programming test was not fully implemented because of the teams' lack of knowledge in using an automated tool, which is JUnit. It was observed that the test cases were often in the mind of the team members. Testing was conducted manually using programmers' intuition and tested by the other team members who were not responsible for that particular module. It was not preferable to write test cases, when the projects were still in the development phase. It was found that writing test cases was time consuming, especially when the system became large and had many fields to fill in. Therefore, the members preferred to perform tests based on their intuition and record only if any discrepancies occurred. It was a practice in this company to use a standard form to record the errors in the system. The forms were used by the system analysts and programmers, and then verified by the clients. Writing test cases can be applied and is useful when the system is finished and before it is released to the clients. The members can check any faults from the test case reports for system correction.

The working environment makes an impact on adoption of **pair programming** because collocated teams are important for energizing this practice (Beck, 2000; Beck & Andres, 2004; Cockburn, 2007). However, it was impossible for team members in this study to be located in the same place. This is because the computer centre practiced on-site distributed teams to cater to the demand from various colleges, institutes and centres. The additional sophisticated networking technologies in this university also assisted in having distributed teams at various locations. It was observed that members of each team met and applied pair programming during the early sessions of project inception. The pair programming practice started with the head programmers working on a module. A

junior programmer pair programmed with each head until the knowledge was transferred to the programmer, who later returned to their respective work places to complete the module. This process was repeated until every programmer pair programmed with the head programmer. Therefore, pair programming was practiced at the initial stage only. This was how members adopted pair programming in their environment. Whenever problems on **code structure** were encountered, the respective pair met to thrash out the problems. The experienced programmers acted as drivers for junior programmers by showing how the programming tasks worked. This scenario is similarly practiced by those in outsourcing companies (Sison & Yang, 2007).

Even though pair programming was not practiced fully throughout the project, **code ownership** was achieved through the use of a server. Sharing value was upheld among the team members because members could easily check-in the code programs into the server and share the codes. The team members were able to solve programming problems without a full pair programming practice because members often **reused codes** that were uploaded into the server. Code ownership was easily utilized because code ownership had been part of the working procedures in their centre. This is in accordance with prior research showing that people in Asian countries have low individualism and strong collectivism traits, while Western countries have relatively strong individualism (Schulte & Kim, 2007).

On-site customer cannot be applied effectively because most projects were stable and small-scale size, with the teams being familiar with system requirements before the project started. Therefore, team members did not have difficulty in the use of **metaphor** in their project.

Continuous integration, frequent release and coding standard activities had been part of the working procedure for the staff of the computer centre. Therefore these activities were not new for them, even though the term XP was 'newly' introduced. **Continuous integration** and **frequent release** did not apply every day or every hour because of the organization environment that needed the SE teams to cater to other projects concurrently. **Coding standard** using appropriate naming conventions were applied but comments to the coding methods were applied only when it was necessary.

These findings conclude that not all agile-XP practices were applied effectively amongst the members. However, partial adoption of agile practices specifically XP is expected in the organization (Aveling, 2004) since to introduce and implement a new method to the SE teams is not an easy task (Nerur, Mahapatra, & Mangalaraj, 2005).

Therefore, adjustment to the latest practices (Cockburn, 2007) can be established as the teams mature and become more experienced.

4.2 Agile Challenges and Key Success Factors

Results revealed that organizational culture plays a significant factor in determining successful adoption of an agile methodology (Strode, Huff, & Tretiakov, 2009; Tolfo & Wazlawick, 2008). In this project, there was no explicit contractual agreement made because the projects were for internal clients. However, the commitment for each project was based on deadline. At the same time, the projects were categorized as low risk due to no cost constraints allocated to the projects. When this becomes the accepted norm, it was observed that the every team progress slowly and was unable to complete the project according to the deadline. In addition, it is important to note that the teams were also involved concurrently in other projects supporting the organizational goals. With several clients to be served, the teams had to prioritize every project according to management decisions. This situation led the members to manage their time carefully in order to fulfill the project requirements. Realizing the difficulties faced by the teams to complete every project while applying agile practices, computer centre management decided to meet with each team every fortnight in order for the management to monitor the progress and for the teams to demonstrate their projects. This was to ensure that the project team maintained and sustained direction towards achieving the project goals. Past research has shown that constant review and feedback can help a team and management to improve a project (Verner & Evanco, 2005). Furthermore, constant review and feedback are important aspects in agile software development activities.

In ensuring every project completed on time with applying agile-XP as closely as possible, the management initiated a rewarding scheme where prominent members with outstanding project will be sponsored to attend a professional exam. The exam is a certification for programmers that have demonstrated fundamental programming language proficiency for career advancements. Rewarding system has been proven to motivate the employees to increase work performance (Govindarajulu & Daily, 2004; Lee & Ahn, 2007) but its effectiveness depend on the organizational conditions (Perry, Mesch, & Paarlberg, 2006).

It was revealed that additional coaching was needed to improve team members understanding in applying the agile-XP practices effectively. Coaching is an attentive mechanism for the teams easier tracking their software development status. In this study, agile simple design was introduced and incorporated into teams' existing knowledge. Using use case diagram as one of the tools to model the system

make the teams easier in understanding system requirements. Allowing flexibility in adopting new approach in organization induced the members to share their knowledge with other members. Adjusting agile approach in a given environment is important to ensure a successful implementation.

5.0 CONCLUSION

Adopting agile-XP practices has been successfully implemented in this centre. Although not all practices were fully adopted, positive feedbacks amongst SE teams was received during the software project. This shows that the flexibility approach to accommodate organizational culture is important for changing people's behaviour and mindset. In addition, identifying knowledge gaps is vital for ensuring successful implementation of a healthy work culture. Effective methodology and organizational culture are important factors that must be considered to produce innovative teams and quality software. Further studies on this relationship would be valuable for creating a new generation of creative and innovative SE teams.

REFERENCES

- Ainsworth, S., & Loizou, A. T. (2003). The Effects of Self-explaining when learning with text or diagrams *Cognitive Science*, 27(4), 669-681.
- Aveling, B. (2004). XP Lite Considered Harmful? In J. Eckstein & H. Baumeister (Eds.), *XP2004* (Vol. 3092, pp. 94-103). Berlin Heidelberg: Springer-Verlag.
- Beck, K. (2000). *Extreme Programming Explained: Embrace Change*. USA: Addison Wesley.
- Beck, K., & Andres, C. (2004). *Extreme Programming Explained: Embrace Change (2nd Edition)* USA: Addison Wesley.
- Cockburn, A. (2007). *Agile Software Development: A Cooperative Game (2nd Edition)*. USA: Addison Wesley.
- Govindarajulu, N., & Daily, B. F. (2004). Motivating Employees for Environmental Improvement. *Industrial Management and Data Systems*, 104(4), 364-372.
- Ilieva, S., Ivanov, P., & Stefanova, E. (2004). *Analyses of an Agile Methodology Implementation*. Paper presented at the 30th EUROMICRO Conference (EUROMICRO'04), Rennes, France.
- Layman, L. (2004). *Empirical Investigation of the Impact of Extreme Programming Practices on Software Projects*. Paper presented at the Conference on Object-Oriented Programming Systems Languages and Applications, Vancouver, Canada.
- Layman, L., Williams, L., & Cunningham, L. (2004). *Exploring Extreme Programming in Context: An Industrial Case Study*. Paper presented at the Proceedings of the Agile Development Conference Washington.
- Lee, D.-J., & Ahn, J.-H. (2007). Reward Systems for Intra-organizational Knowledge Sharing. *European Journal of Operational Research*, 180(2), 938-956.
- Lindvall, M., Muthig, D., Dagnino, A., Wallin, C., Stupperich, M., Kiefer, D., et al. (2004). Agile Software Development in Large Organizations. *Computer*, 37(12), 26-34.
- Mazni, O., Sharifah Lailee, S.-A., & Azman, Y. (2010). Agile Documents: Towards Successful Creation of Effective Documentation. In A. Sillitti, X. Wang & A. Martin (Eds.), *Agile Processes in Software Engineering and Extreme Programming, 11th International Conference, XP 2010 Lecture Notes in Business Information Processing LNBI 48* (pp. 196-201). Berlin, Heidelberg: Springer-Verlag.
- McAvoy, J., & Butler, T. (2009). A Failure to Learn in a Software Development Team: The Unsuccessful Introduction of an Agile Method. In W. Wojtkowski, G. Wojtkowski, M. Lang, K. Conboy & C. Barry (Eds.), *Information Systems Development: Challenges in Practice, Theory, and Education* (pp. 1-13). US: Springer Science.
- Murru, O., Deias, R., & Mugheddu, G. (2003). Assessing XP at a European Internet Company. *IEEE Software*, 20(3), 37-43.
- Nerur, S., Mahapatra, R., & Mangalaraj, G. (2005). Challenges of migrating to agile methodologies. *Communications of the ACM*, 48(5), 73-78.
- Perry, J. L., Mesch, D., & Paarlberg, L. (2006). Motivating Employees in a New Governance Era: The Performance Paradigm Revisited. *Public Administration Review*, 66(4), 505-514.
- Rasmusson, J. (2003). Introducing XP into Greenfield Projects: Lessons Learned. *IEEE Software*, 20(3), 21-28.
- Salo, O., & Abrahamsson, P. (2008). Agile methods in European embedded software development organisations. *IET Software*, 2(1), 58-64.
- Schnotz, W. (2002). Commentary: Towards an Integrated View of Learning From Text and Visual Displays. *Educational Psychology Review*, 14(1), 101-102.
- Schulte, W. D., & Kim, Y. K. (2007). Collectivism and expected benefits of knowledge management: A comparison of Taiwanese and US perceptions. *Competitiveness Review: An*

- International Business Journal incorporating Journal of Global Competitiveness*, 17(1/2), 109-117.
- Sharifah Lailee, S.-A., Mazni, O., Mohd Nasir, A. H., Che Latifah, I., & Kamaruzaman, J. (2009). Positive Affects Inducer on Software Quality. *Computer and Information Science*, 2(3), 64-70.
- Sharp, H., & Robinson, H. (2008). Collaboration and co-ordination in mature eXtreme programming teams. *International Journal of Human Computer Studies*, 66(7), 506-518.
- Sillitti, A., Ceschi, M., Russo, B., & Suchi, G. (2005). *Managing Uncertainty in Requirements: a Survey in Documentation-driven and Agile Companies*. Paper presented at the 11th IEEE International Software Metrics Symposium (METRICS 2005).
- Sison, R., & Yang, T. (2007). *Use of Agile Methods and Practices in the Philippines*. Paper presented at the 14th Asia-Pacific Software Engineering Conference (APSEC'07), Nagoya, Japan.
- Strode, D. E., Huff, S. L., & Tretiakov, A. (2009). *The Impact of Organizational Culture on Agile Method Use*. Paper presented at the 42nd Hawaii International Conference on System Sciences, Waikoloa, Big Island, Hawaii
- Tolfo, C., & Wazlawick, R. S. (2008). The Influence of Organizational Culture on the Adoption of Extreme Programming. *Journal of Systems and Software*, 81(11), 1955-1967.
- Verner, J. M., & Evanco, W. M. (2005). In-House Software Development: What Project Management Practices Lead to Success? *IEEE Software*, 22(1), 86-93.
- Wood, W. A., & Kleb, W. L. (2003). Exploring XP for Scientific Research. *IEEE Software*, 20(3), 30-36.