Journal of Engineering and Applied Sciences 12 (16): 4222-4227, 2017
ISSN: 1816-949X

Exploring the Effect of Agile Methodology and Team Effectiveness

[1]Mazni Omar and [2]Sharifah-Lailee Syed-Abdullah
[1]Human-Centered Computing Research Lab, School of Computing,
Universiti Utara Malaysia, Changlun, Malaysia
[2]Faculty of Computer and Mathematical Sciences, Universiti Teknologi MARA,
404050 Shah Alam, Selangor, Malaysia

Abstract: Agile methodology emerged in response to the importance of humanistic aspects in Software Engineering (SE). However, there is lack of empirical evidence that supports its effectiveness in SE. The lack of empirical evidence demands more research in this field to generate more empirical data. Therefore, the aim of this research is to investigate empirically the effect of agile methodology on the members of software development teams. An empirical inquiry in the form of a case study was carried out in a computer centre in Malaysia. The case study focused on four software development teams working on different applications in an organization. Both quantitative and qualitative analyses were used to triangulate and strengthen the empirical results. This study demonstrates that there is a relationship on the degree of agile application with team effectiveness.

Key words: Index terms-agile methodology, team effectiveness, software team, adoption, centre, relationship

INTRODUCTION

The research conducted on the use of agile practices in industry typically focused on the agile perception (Aveling, 2004; Tuli *et al*., 2014; Rasmusson, 2003) and comparison studies (Ilieva *et al*., 2004; Layman *et al*., 2004; Layman, 2004). Nonetheless, studies aimed at applying and evaluating the agile effectiveness in industry indicated that its practices can increase the software quality and team productivity as well as decrease the software life cycle time. Most studies conducted in the United States (Layman *et al*., 2004) as well as those carried out in European countries (Ilieva *et al*., 2004; Sillitti *et al*., 2005; Salo and Abrahamsson, 2008) were based on data gathered from companies using XP or Scrum. However, there were fewer studies reported on agile practices in Asian countries, specifically in Malaysia. Consequently, due to the lack of culturally relevant empirical evidence of these practices, implementation of the agile methodology amongst practitioners in Malaysia has been slow (Asnawi *et al*., 2011). Therefore, there is a need more empirical studies to investigate the impact of agile approach amongst software developers in Malaysia.

Literature review: A study focusing on appropriateness of using agile practices in large organizations has been carried out by Lindvall *et al*. (2004) who conducted a pilot study at ABB, Daimler Chrysler, Motorola and Nokia. In this study, the software developers in these organizations incorporated XP practices in their regular software development activities. The study findings suggest that all developer teams agreed that XP had allowed them to gain positive experiences in terms of ease of understanding and applying the XP practices and the appreciable increase in the team morale. Furthermore, the practices helped improve customer satisfaction, software quality and team productivity. This study has shown that XP practices such as pair programming, continuous integration, on-site customer and planning game, contributed towards the project implementation success. However, it should be noted that these organizations had to modify some aspects of agile methodology in order to incorporate the activities within their current software practices to avoid conflict of interest in developing software.

Ilieva *et al*. (2004) carried out a case study on an e-Business application whereby two similar projects were compared-one that used agile (XP) methods which served as a pilot project and the other based on traditional methods as baseline project with respect to productivity and defect rate. The empirical findings imply that agile team managed to increase their development productivity as well as reduce effort and defect rate. Moreover, constant requests for feedback from the customer

Corresponding Author: Mazni Omar, Human-Centered Computing Research Lab, School of Computing,
Universiti Utara Malaysia, Changlun, Malaysia

increased customer satisfaction as they felt in control of the development progress. However, although the team found working in pairs useful, the participants reported that the practice required great concentration to conform to coding standards and as a result, the developers became exhausted.

Another study by Layman *et al.* (2004) indicated that XP project increased productivity and improved pre-release and post-release quality compared to plan-driven project. The case studies were conducted at IBM and Sabre Airlines. However, post-release quality results are limited due to the lack of customer usage measurement during the study. Sillitti *et al.* (2005) conducted a survey by interviewing 16 project managers of Italian companies where half of the participants reported on the use of agile and the remaining half on using documentation-driven methods. Based on their survey, Agile companies were found to be more customer-centric and flexible and therefore able to have satisfactory relationship with the customers as compared to document-driven companies. However, the survey was conducted with participants working for different types of companies and it is thus likely that sizes of projects and general working conditions were too diverse to allow for valid comparison. Therefore, further investigation must be carried out in order to obtain more reliable findings.

Positive experiences and feedback in applying agile approach, especially XP in software organizations are increasingly being reported in literature with the emphasis of the value of promoting highly collaborative processes, coordination and communication amongst team's members, improving learning experiences and receiving satisfactory relationship with customers (Sillitti *et al.*, 2005; Sharp and Robinson, 2008). However, literature review identified that 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 as these are common programming practices. In contrast, practices that involved other parties such as on-site client, planning game and small release were difficult to adopt (Aveling, 2004). This selective adoption of agile methodology stems from the fact that learning to adopt agile practices is more than cognitive effort but a complex process that requires software development team to change their behaviour, attitudes and opinions during the software development activities (McAvoy and Butler, 2009). Thus, given that full implementation of agile practices requires fundamental changes in characteristics and attitudes of all involved individuals in order for this challenging task to be successful, human factors that underpin SE development must be fully understood and addressed. Moreover as agile approach is a team-oriented, understanding the diversity of human personalities would help to ensure that teams are comprised of a good combination of individuals which will in turn, always lead to a successful software project (Cockburn and Highsmith, 2001; Mazni *et al.*, 2010).

MATERIALS AND METHODS

In this study, a case study was carried out to seek empirical inquiries that investigate contemporary phenomena within real life contexts (Yin, 2003). This case study applied a combination of interpretative and descriptive case study approach. An interpretive case study attempts to investigate phenomena through the participant's interpretations of their software development activities whereas the descriptive allows the researcher to gain new insights into the effectiveness of agile practices amongst software developers in industrial setting. In this study, four software development teams were observed for six months.

To increase the research validity, multiple data sources were used-questionnaires, interviews, obser vations and project documents review-to determine the effectiveness of agile practices used by the teams. Team effectiveness was measured based on the assessment made by the manager of a computer centre. The manager has <20 year's experience in managing software project.

In this research, an Agile (XP) practices and activities table was used (Syed-Abdullah *et al.*, 2006) to measure the degree of agile application by each team. Although, this measurement was based on a qualitative approach using interviews, participative observations and document analysis, the data was subsequently quantified.

A computer centre in Malaysia was selected for this study. This computer centre was responsible for providing and supporting information technology systems and services to campus communities to automate and computerize its major activities. This project came about because the deputy director who was in charge of applications in this centre was an 'Agile (XP) convert'. The director's prior involvement with agile methodology started when he was a project advisor in the earlier study. The director was impressed with the agile practices which were simple and practical and thus in his view the existing situation demanded that the computer centre adopt this approach. Initially, all staff members including 11 system analysts and 20 programmers were selected to participate in the 'Embracing Agile (XP) Project'. This project sought to obtain an alternative approach for developing software using agile methodology. The teams were required to apply agile practices and produce simpler agile documents during the project (Table 1).

Table 1: Profiles of four software development teams

Team characteristics	Team A	Team B	Team	Team D
Types of projects	Upgrading a current system A to a web-based system	Upgrading a current system B to a web-based system	A new web-based system C	A new web-based system D
Team composition	1 system analyst 4 programmers	2 system analysts 4 programmers	2 system analysts 4 programmers	2 system analysts 4 programmers
Experience in software development	5-10 years: 1 10-15 years: 4	5-10 years: 3 >15 years: 3	5-10 years: 5 >15 years: 1	5-10 years: 3 10-15 years: 1 >15 years: 2

Before embarking on the project, agile workshops were conducted to introduce and explain the agile methodology to the software development teams, focusing on XP practices. All participants attended the workshops on agile methodology which addressed the theory and practical aspects of Agile Software development during three sessions.

The first session was aimed at the system analysts and focused on the theoretical aspects where simple design document was introduced to facilitate the production of the documentation needed for the project development work. The document consisted of a UML use case diagram, Entity Relationship Diagram (ERD) and interface designs.

The second session was aimed at the programmers and addressed the agile practices. In addition to agile theory, the session also included hands-on tasks for planning game and pair programming to assist the programmers in better understanding the agile activities.

The third session on agile methodology combined 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.

RESULTS AND DISCUSSION

Analysis of results: In this study, levels of application of agile practices were assessed qualitatively and quantitatively as guided by Agile (XP) practices and activities table (Syed *et al*., 2009). The number of agile practices used was measured through the interview sessions, analysis of the project documents and observations. The evidences were documented quantitatively using the adapted agile activities table (Syed-Abdullah *et al*., 2006). Each practice was evaluated and assigned a percentage of the number of agile practices used. The overall percentage of agile practices used by each team is shown in Fig. 1.

In this study, quantitative measures of software quality were not used because the element of understanding and fulfilment of the requirements were sufficient to rank the team's effectiveness. The manager used his expert experience to rank the team's effectiveness as shown in Table 2.

Table 2: Team effectiveness result

Teams	Software quality results
A	Ineffective
B	Effective
C	Effective
D	Ineffective

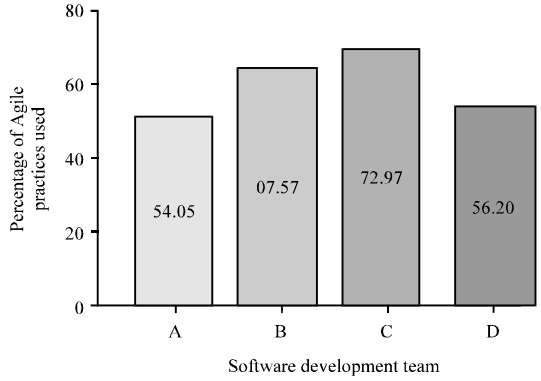

Fig. 1: Percentage of agile practices used by each team

With reference to Fig. 1, the highest percentage of agile practices used is achieved by Team C (72.97%) followed by Team B (67.57%), Team D (56.76%) and Team A (54.05%). The results of Spearman correlation test show that there is a significant positive relationship between software quality and the number of agile practices applied by each team which are planning games ($r(23) = 0.57$, $p = 0.01$), pair programming, ($r(23) = 1.00$, $p = 0.00$), continuous integration ($r(23) = 0.69$, $p = 0.00$), frequent releases ($r(23) = 1.00$, $p = 0.00$) and sustainable pace ($r(23) = 0.95$, $p = 0.00$).

These results indicated that teams which apply agile practices as closely as possible are able to achieve higher software quality thus they are categorized as effective teams. The results showed that the effective teams are Team B and C. The = details application percentage of each agile practice used by each team is shown in Fig. 2.

In Fig. 2, there are differences in the level of application of Agile practices in each team. All software development teams had dedicated on-site clients that were responsible for monitoring and providing feedback for each system application developed by the teams. It

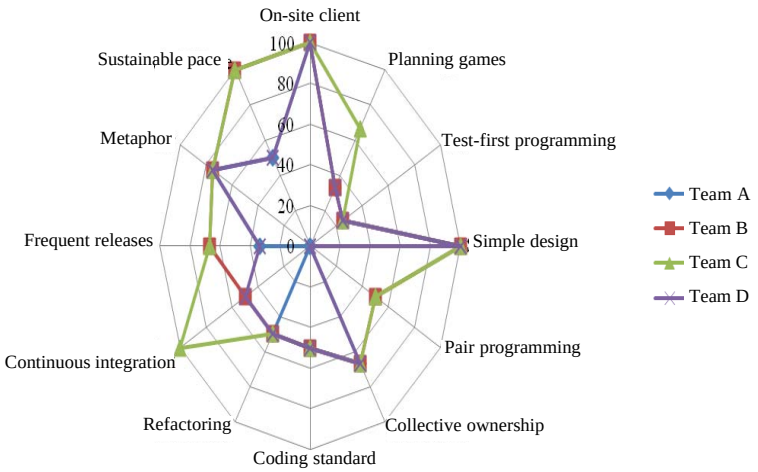

Fig. 2: Percentage of each agile practice used by each team

was observed that Team C was able to apply the planning game practice approximately 66.67% of the time. This team was able to write story cards which were then used to divide their tasks. Each of the team members was responsible for each of the story cards written. The other teams only managed to apply this practice 33.33% of the time because although each team could write the story cards there were no specific developers in charge of them.

In terms of the test-first programming practices, each team only managed to apply functional test activities. This is because the developers did not have experience in using automated testing tools. In addition, all teams were able to produce a design document which was simpler than their previous documentation. Thus, positive feedback was received by each team regarding the use of simple design document.

Only Teams B and C managed to apply pair programming activities by having two programmers responsible for each module. These teams were able to apply this practice because their programmers had the same programming style, making it easier for them to communicate in programming tasks whereas the programmers in Teams A and D were more comfortable working alone to solve programming tasks. Each team had a dedicated server and was able to have a testing partner to achieve collective code ownership practice. Thus each team had the same level of collective ownership practice of approximately 66.67%.

In addition, each team also applied a specific coding standard in assisting every member to understand their coding style. However, it was observed that none of the teams followed the standard consistently. This is because of time constraints and the different programming styles between team members. In terms of the refactoring practice, reusable code was the activity most frequently applied by each team to ensure that their programmers could code faster. However, refactoring to remove smelly code was not applied because the teams wanted to keep their code for future use and save programming time.

For the continuous integration practice, only Team C managed to apply incremental integration and testing because members of this team were consistently able to work together. This practice allowed them to view the system easily. Teams B and D managed to apply incremental integration and testing of the system as a whole. Thus these teams managed to apply this practice approximately 50% of the time. Team A was unable to apply continuous integration because their programmers were more comfortable working alone, only integrating system modules at the end of the project. By applying continuous integration, Team C managed to obtain a continuous review and early feedback from their client. This also happened to Team B. Therefore these teams were able to achieve a frequent release practice approximately 66.67% of the time. Teams A and D did get a continuous review from the client but not as early as Team B and C. From these observations, all teams had stable system requirements and experience of developing in the same application domain. Therefore, all teams had a similar degree of understanding of the vision and architecture of the system being developed.

Based on each team's progress, it was found that only Team B and C managed to schedule fewer activities during the system evaluation week and deliver a completed, tested and integrated system. They, thus achieved 100% for the sustainable pace practice. Team D only managed to deliver a completed, tested and

integrated system and thus achieved 75% whereas Team A only delivered a completed and tested system, achieving just 50% for the sustainable pace practice level.

These findings showed that not all agile practices were applied effectively amongst the members. Earlier research on application of agile practices only showed partial adoption 1 since to introduce and implement a new method to the software teams is not an easy task (Nerur *et al*., 2005; Korkala and Maurer, 2014). Even though some teams loosely followed the agile practices, it was calculated that each team managed to apply <50% of the agile practices (Fig. 1). Agile practices such as continuous integration, frequent release and coding standard are part of the working software practices in this computer centre thus enabling them to easily understand and accept the practices. In relation, past studies have shown that for a new technology to be easily accepted, it influenced by several factors such as compatibility with past experiences (Riemenschneider *et al*., 2002), support from top management (Sultan and Chan, 2000), organizational culture (Tolfo *et al*., 2011) and teamwork (Ceschi *et al*., 2005; Moe *et al*., 2010; Yu and Petter, 2014).

CONCLUSION

This case study was confined to the government agency of Malaysia where software developers involved in this study developed in-house software projects. Past research has demonstrated that introducing and implementing a new method to the software teams is not an easy task (Nerur *et al*., 2005). Nevertheless, this study was successfully performed where support from top management and employees were given. Results revealed that organizational culture plays a significant factor in determining successful application of an agile methodology (Tolfo *et al*., 2011). The first challenge is the working procedure of the organization. It was noted that every team members was also involved concurrently in other software projects that supported the organizational goals. With several clients to serve, the teams had to prioritize every project according to management decisions.

RECOMMENDATIONS

This situation led members to manage their time carefully in order to fulfil the project requirements. Realizing the difficulties faced by the teams to complete every project while applying agile practices, the computer centre management decided to meet with each team every fortnight to allow the management to monitor the progress and for the teams to demonstrate their projects. This was to ensure that the project team maintained agile project momentum and sustained the direction towards achieving the project goals. This is important because constant reviews and feedbacks help the teams and management to improve projects (Hayes *et al*., 2011) (Verner and Evanco, 2005).

ACKNOWLEDGEMENT

The researchers wish to thank the Ministry of Education Malaysia for funding this study under Fundamental Research Grant Scheme (FRGS), S/O project Code: 12818.

REFERENCES

Asnawi, A.L., A.M. Gravell and G.B. Wills, 2011. Empirical investigation on agile methods usage: Issues identified from early adopters in Malaysia. Proceedings of the 12th International Conference on Agile Processes in Software Engineering and Extreme Programming, May 10-13, 2011, Springer, Madrid, Spain, pp: 192-207.

Aveling, B., 2004. XP lite considered harmful?. Proceedings of the 5th International Conference on Extreme Programming and Agile Processes in Software Engineering, June 6-10, 2004, Springer, Garmisch-Partenkirchen, Germany, pp: 94-103.

Ceschi, M., A. Sillitti, G. Succi and S.D. Panfilis, 2005. Project management in plan-based and agile companies. IEEE. Software, 22: 21-27.

Cockburn, A. and J. Highsmith, 2001. Agile software development, The people factor. Computer, 34: 131-133.

Hayes, D.R., F. Grossman, C. Knapp and L. Rising, 2011. The impact of project retrospectives on process improvement initiatives: A case study. Proceedings of the 2011 IEEE Conference on Long Island Systems, Applications and Technology (LISAT'11), May 6, 2011, IEEE, Farmingdale, New York, ISBN:978-1-4244-9878-9, pp: 1-7.

Ilieva, S., P. Ivanov and E. Stefanova, 2004. Analyses of an agile methodology implementation. Proceedings of the 30th Conference on Euromicro, September 3, 2004, IEEE, Rennes, France, ISBN:0-7695-2199-1, pp: 326-333.

Korkala, M. and F. Maurer, 2014. Waste identification as the means for improving communication in globally distributed agile software development. J. Syst. Software, 95: 122-140.

Layman, L., 2004. Empirical investigation of the impact of extreme programming practices on software projects. Proceedings of the 19th Annual ACM SIGPLAN Conference on Object-Oriented Programming Systems Languages and Applications, October 24-28, 2004, ACM, Vancouver, British Columbia, Canada, ISBN:1-58113-833-4, pp: 328-329.

Layman, L., L. Williams and L. Cunningham, 2004. Exploring extreme programming in context: An industrial case study. Proceedings of the 2004 Conference on Agile Development, June 22-26, 2004, IEEE, Salt Lake City, Utah, ISBN:0-7695-2248-3, pp: 32-41.

Lindvall, M., D. Muthig, A. Dagnino, C. Wallin amd M. Stupperich *et al*., 2004. Agile software development in large organizations. Computer, 37: 26-34.

Mazni, O., S.A. Sharifah-Lailee and Y. Azman, 2010. Agile Documents: Towards Successful Creation of Effective Documentation. In: Agile Processes in Software Engineering and Extreme Programming, Sillitti, A., X. Wang and A. Martin (Eds.). Springer-Verlag, Heidelberg, Berlin, pp: 196-201.

McAvoy, J. and T. Butler, 2009. A Failure to Learn in a Software Development Team: The Unsuccessful Introduction of an Agile Method. In: Information Systems Development, Wojtkowski, W., G. Wojtkowski, M. Lang, K. Conboy and C. Barry (Eds.). Springer, New Yok, USA., ISBN:978-0-387-30403-8, pp: 1-13.

Moe, N.B., T. Dingsoyr and T. Dyba, 2010. A teamwork model for understanding an agile team: A case study of a Scrum project. Inform. Software Technol., 52: 480-491.

Nerur, S., R. Mahapatra and G. Mangalaraj, 2005. Challenges of migrating to agile methodologies. ACM. Commun., 48: 72-78.

Rasmussen, J., 2003. Introducing XP into greenfield projects: Lessons learned. IEEE. Software, 20: 21-28.

Riemenschneider, C.K., B.C. Hardgrave and F.D. Davis, 2002. Explaining software developer acceptance of methodologies: A comparison of five theoretical models. IEEE. Trans. Software Eng., 28: 1135-1145.

Salo, O. and P. Abrahamsson, 2008. Agile methods in European embedded software development organisations: A survey on the actual use and usefulness of extreme programming and scrum. IET. Software, 2: 58-64.

Sharp, H. and H. Robinson, 2008. Collaboration and co-ordination in mature eXtreme programming teams. Intl. J. Hum. Comput. Stud., 66: 506-518.

Sillitti, A., M. Ceschi, B. Russo and G. Succi, 2005. Managing uncertainty in requirements: A survey in documentation-driven and agile companies. Proceedings of the 11th IEEE International Symposium on Software Metrics, September 19-22, 2005, IEEE, Como, Italy, ISBN:0-7695-2371-4, pp: 1-10.

Sultan, F. and L. Chan, 2000. The adoption of new technology: The case of object-oriented computing in software companies. IEEE. Transac. Eng. Manage., 47: 106-126.

Syed, A.S.L., M. Omar, M.N.A. Hamid, B.C.L. Ismail and K. Jusoff, 2009. Positive affects inducer on software quality. Comput. Inf. Sci., 2: 64-64.

Syed-Abdullah, S., M. Holcombe and M. Gheorge, 2006. The impact of an agile methodology on the well being of development teams. Empirical Software Eng., 11: 143-167.

Tolfo, C., R.S. Wazlawick, M.G.G. Ferreira and F.A. Forcellini, 2011. Agile methods and organizational culture: Reflections about cultural levels. J. Software Evol. Process, 23: 423-441.

Tuli, A., N. Hasteer, M. Sharma and A. Bansal, 2014. Empirical investigation of agile software development: Cloud perspective. ACM. SIGSOFT Software Eng. Notes, 39: 1-6.

Verner, J.M. and W.M. Evanco, 2005. In-house software development: What project management practices lead to success?. IEEE. Software, 22: 86-93.

Yin, R.K., 2003. Case Study Research: Design and Methods. 3rd Edn., Sage Publications, USA., ISBN-13: 9780761925521, Pages: 181.

Yu, X. and S. Petter, 2014. Understanding agile software development practices using shared mental models theory. Inf. Software Technol., 56: 911-921.