Agile Development Methods for Developing Web Application in Small Software Firms

Faudziah Ahmad[1], Fauziah Baharom[2] and Moath Husni[3]
[1,2,3] *School of Computing*, *Universiti Utara Malaysia, Malaysia*
, [1]fudz@uum.edu.my, [2]fauziah@uum.edu.my,[3]tarawneh80@yahoo.com

ABSTRACT

Small software firms that involved with developing web application are lacked of well defined development process. Many development methods have been proposed for developing web application in these firms. However, these methods have some limitations. This paper aims to identify the agile development methods that are suitable for small development teams and determine the enhancements needed to get high quality web application. In order to achieve these objectives a comparative study was conducted on the suitable agile development methods that have been selected. Comparisons were made according to a set of criteria that include development process, project management, requirement, testing and design. The findings of this paper will be used as baseline for building a new measurable web application development methodology for small software firms.

Keywords: Comparative Study, Web Application, Agile, Small Software Firms, Measurement.

I INTRODUCTION

Web-based applications differ from other traditional applications as is it known to be of high reliability, high usability, more secured, incorporator advanced technologies, takes a shorter time to market, have a shorter product life cycles and required continuous maintenance (Rodriguez et al., 2002).

A high percentage of small software companies have been found to be involved with developing Web applications (Richardson & Wangenheim, 2007). Small software firm is any organization or company that has approximately 10 to 50 employees (Fayad et al., 2000; Hofer, 2002; Laporte et al., 2005). Currently, problems that faced by these organizations are: i) limited resources for business development; ii) limited number of available developers; iii) limited staff skills; vi) lack of well defined development method; and v) limited adopted Quality Assurance and measurements practices (Fayad et al., 2000; Dangle et al., 2005; El Sheikh & Tarawneh, 2007; Altarawneh & Shiekh, 2008; Haung et al., 2008; Tarawneh and Allahawiah, 2009; Pusatli and Misra, 2011). Thus, this important brand of organizations must use a systematic development methodology to solve these problems and get a high quality final product within the available resources.

In order to produce high quality Web applications, small software firms must ensure that the development process is short, within the specific budget and monitored by suitable measurement program. The implementation of software measurement within the development process has been considered as a significant activity for successful software development and found to produce fruitful returns such as reduce defects, decrease rework, shorten development life cycle time, minimize cost, increase productivity and improve customer satisfaction (Morasca, 1999; Solingen & Berghout, 2001; Wangenheim etal., 2003; Kettelerij, 2006; McCurley et al., 2008). Software measurement is defined as "an effective means to understand, control, monitor, predict and improve software development projects" (Kettelerij, 2006).

Many conventional development methods have been proposed for building web applications such as waterfall and spiral. However, these development methods are not adequate for developing web applications in small software firm because they cannot deal with continuous change of requirements and they are not meant for building Web applications as they require a large number of resources (skills, staff) (Haung et al., 2008; Altarawneh & Shiekh, 2008; Eldai et al., 2008). Consequently, agile development methods have been proposed to deal and solve problems that cannot be handled by conventional development approaches (Stojanovic et al., 2003; Lindstrom & Jeffries, 2004). The most popular agile methods that are commonly used for developing software in small teams and projects are Extreme Programming (XP) and Scrum (Alite & Spasibenko, 2008; Qumer & Henderson-Sellers, 2008). However, the existing agile development methods (XP and Scrum) have some limitations. These methods are found to be lacking in applying the important development practices as well as applying measurement practices during the development process (Turk et al., 2002; Fritzsche & Keil, 2007; Qumer & Henderson-Sellers, 2008; Jiang & Eberlein, 2008).

This paper aims to identify the agile development methods that are suitable for small development teams; and determine the enhancements needed.

The outline of the paper is as follows: section II describes the methodology used to conduct the research. Section III presents the findings and discussion and section IV briefly described the conclusion.

II METHODOLOGY

This study will be conducted in two phases: Identification of suitable agile development methods for small development teams; and identification of enhancements needed.

Phase A: Identification of suitable agile development methods for small development teams.

In this phase, seven methods (Scrum, Adaptive Software Development (ASD), Agile modeling, Dynamic Systems Development Method (DSDM), Crystal Family, Extreme Programming (XP) and Feature Driven Development (FDD)) were identified and compared based on the development team size. The seven methods were found to be popular agile development methods (Abrahamsson et al., 2002; Stojanovic et al., 2003; Qumer and Henderson-Sellers, 2008). From the comparison results, suitable methods were selected for the study.

Phase B: Identification of enhancements needed.

This phase was conducted to determine the enhancements needed on the agile development methods selected in Phase A. This was done by comparing the selected methods in terms of five criteria: development process, project management, requirement, testing and design. These criteria was recommended by past researchers such as McDonald & Welland, (2001), Deshpande et al., (2002), Redouane, (2004), Abran et al, (2004), Haung et al., (2008) and Qumer & Henderson-Sellers, (2008).

The five criteria are briefly explained below:

Development Process. The developing process must use systematic, disciplined methodology which clarifies the roles and responsibilities for each team member (McDonald & Welland, 2001).

Project Management. Project management activities must be performed during the development process. These activities include planning, coordinating, measuring, monitoring, controlling, and reporting (Abran et al, 2004).

Requirements. Requirment must be collected directly from the users and backtracking is needed when requirements are changed which clarify the need of iterative process to cope with requirement change (Haung et al., 2008; Redouane, 2002).

Testing. Testing process should be made on all components of web application such as page, code, site and navigation to ensure the quality of the final product (Redouane, 2002; Deshpande et al., 2002).

Design. Web application design must be simple and can produce a prototype within a short time (McDonald & Welland, 2001; Qumer & Henderson-Sellers, 2008).

After performing the comparative study, a set of enhancements were suggested to cater for the existing agile development methods limitations.

III FINDINGS AND DISCUSSION

This section describes findings on (i) suitable agile development methods for small development teams (Phase A), and (ii) enhancements needed (Phase B).

A. Suitable Agile Development Methods for Small Development Teams

Table 1 shows the findings on various agile development methdos and the recommended team size.

Table 1. Agile Development Methods.

Method name	Size of team recommended
Scrum	2 to 10
ASD	No restrictions and it used for large systems
Agile Modeling	No restrictions
DSDM	Applicable for large teams but possible many small teams (2-6)
Crystal family	2 to 40
XP	2 to 12
FDD	50-250

The most appropriate agile development methods to be used for small development teams are XP and Scrum because the development team size recommended for both is suitable for small sofware firms. Furthermore, these two methods were suggested to be used for small development team by many researchers such as Alite and Spasibenko, (2008), Qumer and Henderson-Sellers, (2008) and Qureshi, (2011).

B. Enhancements Needed

Table 2 and Table 3 illustrate findings gathered from past literature for (i) development process criterion and (ii) project management, requirement, testing and design criteria respectively.

Table 2. Development Process Criteria.

Sub-criteria	XP	Scrum
Iterative and rapid development style	Yes	Yes
Short releases (after the first iteration new versions release even daily and at least monthly)	Yes	Yes
Metaphor (guides all the development by describing how system work)	Yes	No
Simple design (unnecessarily complexity and extra code are removed immediately)	Yes	No
Refactoring (removing duplication and adding flexibility)	Yes	No
Pair programming (tow programmers + one monitor)	Yes	No
Collective Ownership (any one can change the code any time)	Yes	No
On-site customer (customer has to be available full time for the team)	Yes	No
Coding standard (coding rules must be followed by the programmers)	Yes	No
Every day meeting	No	Yes
Every iteration meeting	No	Yes

Sub-criteria in Table 2 were extracted from (Abrahamsson et al., 2002; Qumer and Henderson-Sellers, 2008; Fernandes & Almeida, 2010; Abrantes & Travassos, 2011) and these sub-criteria are considered as the most common agile development practices.

Development Process. Table 2 shows that XP fullfills more than 80% of the development process sub-criteria. Scrum only fulfills 20% of development process sub-criteria. However, both of them are recommended to be used for small development team ranging from 2 to 12 people. In addition, XP and Scrum use iterative and rapid development style and thus, speeds up development process.

Table 3 shows the results of the other comparison criteria: project management, requirement, testing and design.

Table 3. Comparison Table.

Criteria	Sub-criteria	Methods	
		XP	Scrum
Project Management	Management Practices	Planning Game	Scrum Master Sprint meeting Daily meeting
	Measurement Program	No	No
Requirement	Requirement gathering practices	User stories	Sprint backlog Product backlog
	Requirement repository for trace and reuse	No	No
Testing	Testing technique	Using the TTD technique	No
Design	Design approach	Code centered	Design centered
	Code style	Clean and simple	No

Project Management. As shown in Table 3, Scrum focuses on the management of the process more than XP. However, both of them do have not a specific measurement program to ensure the quality of the product and process. Using measurements within the development process, enable defect to be reduced, which in turn minimizes time and rework of the development life cycle (Kettelerij, 2006; McCurley etal., 2008).

Requirements. Both XP and Scrum are good on requirement gathering techniques that they use. However, both are not aware on the requirements reuse and traceability.

Testing. XP is better than Scrum on performing testing practices. XP uses Test Driven Development (TDD) technique to ensure that all implemented features are tested. However, Scrum is not using any specific technique or practice for testing.

Design. The design approach used in XP is code centric and in Scrum is design centric. XP uses a clean and simple coding style practices (i.e. pair programming and simple design). Scrum does not have standard coding style.

Based on Table 2 and Table 3 XP was found to concenrate more on the development practices but is lack of applying management practices. Whereas, Scrum was found to concentarte on the management practices but is lack of development practices such as testing, design and coding.

Both of XP and Scrum do not have a measurement program to ensure the quality of product and process. In addition, XP and Scrum do not cater for requirements reuse and traceability. Scrum does not have any specific technique or practice for testing.

In order to include the management and development practices in one development method, many researchers recommended to combine XP and Scrum together. This combination can be done by adding the important XP development practices to Scrum development methods (Clutterbuck etal., 2009; Jyothi and Rao, 2011; Qureshi, 2011). However, these combination examples still could not cater for XP and Scrum limitations: measurement program and requirement reuse and traceability are not addressed. This clarifies the need for a new enhanced measurable agile development methodology for developing web applications in small software firms.

Several enhancements are proposed for building the new methodology:

Combine important XP practice with Scrum method. This will be done by adding the important XP development practices to Scrum development method to enhance the development and management issues for both methods.

Establish a requirement repository. Simple repository can be added to the development process to save new requirements, reuse existing requirements and trace requirements.

Incorporate a measurement program. The aim of using the measurement program is to ensure the quality of web application by monitoring the process and product. The measurement program can be done by integrating set of measures (metrics) during the development process so that defects can be reduced, rework time can be minimized and the development life cycle can be shortened.

IV CONCLUSION

This paper aims to identify the agile development methods that are suitable for small development teams and determine the enhancements needed to these methods in order to get high quality web applications. Results showed that the most recommended development methods to be used for developing web applications in small development teams are XP and Scrum. However, both of them are found to be weak in applying the measurement practices and they do not apply the requirement traceability and reuse practices. The suggestions for enhancements include combining the two methods XP and Scrum, establishing a requirement repository and incorporating a measurement program. Findings of this paper will be used as a baseline for building a new measurable web application development methodology for small software firms.

REFERENCES

Abrahamsson, P., Salo, O., Ronkainen, J., & Warsta, J. (2002). *Agile software development methods*. Relatório Técnico, Finlândia.

Abran, A., Moore, J. W., Dupuis, R., Dupuis, R., & Tripp, L. L. (2004). Guide to the software engineering body of knowledge (swebok). *2004 ed P Bourque R Dupuis A Abran and JW Moore Eds IEEE Press* 1-13.

Abrantes, J. F., & Travassos, G. H. (2011). Common Agile Practices in Software Processes. *Proceedings of the Fifth International Symposium on Empirical Software Engineering and Measurement held on 22-23 Sept 2011at Banff, Alberta, Canada* (pp. 355-358) USA: IEEE Computer Society.

Alite,B. & Spasibenko,N.(2008). Project Suitability for Agile methodologies, Master.Thesis, Umeå School of Business, Sweden.

Altarawneh, H., & Shiekh, A. (2008). *A Theoretical Agile Process Framework for Web applications Development in Small Software Firms*., Sixth International Conference on Software Engineering Research, Management and Applications , 125-132

Clutterbuck, P., Rowlands, T., & Seamons, O. (2009). A case study of SME web application development effectiveness via Agile methods. *Electronic Journal Information Systems Evaluation Volume, 12*(1), 13-26.

Dangle, K., Larsen, P., Shaw, M., & Zelkowitz, M. (2005). Software process improvement in small organizations: A case study. *IEEE Software*, *22*(6), 68-75.

Deshpande, Y., Murugesan ,S., Ginige, A., Hansen, S., Schwabe, D., Gaedke, M., White ,B. (2002), WEB ENGINEERING, *Journal of Web Engineering*, *1*(1), 03-17.

El-Sheikh,A. Tarawneh, H.(2007): A survey of Web engineering practice in small Jordanian Web development firms. ESEC/SIGSOFT FSE : 481-490

Eldai O, Ahmed Hassan M. H. Ali, and S. Raviraja (2008) .Towards a New Methodology for Developing Web-Based Systems, World Academy of Science, Engineering and Technology 46 2008 (WASET)

Fayad, M., Laitinen, M., & Ward, R. (2000). Thinking objectively: software engineering in the small. *Communications of the ACM*, 43(3), 115-118.

Fritzsche, M., & Keil, P. (2007). Agile Methods and CMMI: Compatibility or Conflict? e-*Informatica Software Engineering Journal, 1*(1).9-26.

Fernandes, J. M., & Almeida, M.(2010). *Classification and Comparison of Agile Methods*.7th International Conference on the Quality of Information and Communications Technology, ICQICT.391-369.

Hofer, C. (2002). Software development in Austria: results of an empirical study among small and very small enterprises. Proceedings of the 28th Euromicro Conference (EUROMICRO'02) 1089-6503/02 $17.00 © 2002 IEEE.

Huang,W., Li, R., Maple, C., Yang, H., Foskett, D. and Cleaver, V. (2008*).Web Application Development Lifecycle for Small Medium-sized Enterprises(SMEs)*, The Eighth International Conference on Quality Software, 247-252.

Jiang, L., & Eberlein, A. (2008). *Towards a framework for understanding the relationships between classical software engineering and agile methodologies*. APOS '08: Proceedings of the 2008 international workshop on Scrutinizing Agile practices or shoot-out at the Agile corral. APM Publishing.

Jyothi, V. E., & Rao, K. N. (2011). Effective Implementation of Agile Practices. *development, 2*(3), 41-48.

Kettelerij, R. (2006). *Designing A Measurement Programme For Software Development Projects*. Master thesis, Universtiet van Amsterdam, Amsterdam.

Laporte, C., Desharnais, J., Abouelfattah, M., Bamba, J., Renault, A., & Habra, N. (2005). Initiating Software Process Improvement in Small Enterprises: Experiments with Micro-Evaluation Framework, Int. Conference on Software Development, ,153-163.

Lindstrom, L., & Jeffries, R. (2004). Extreme programming and agile software development methodologies. Information Systems Management, 21(3), 41-52.

McCurley, J., Zubrow, D., Dekkers, C.(2008). Measures and Measurement for Secure Software Development.Build security in. Retrieved august 14, 2010, from https://buildsecurityin.us-cert.gov/bsi/articles/best-practices/measurement/227-BSI.html.

McDonald, A., & Welland, R. (2001), ‘A Survey of Web Engineering in Practice’, Department of Computing Science Technical Report R-2001-79, University of Glasgow, Scotland, 1 March 2001.

Morasca, S.(1999). Software measurement Handbook. of Software Engineering and Knowledge Engineering, 2, 239-276.

Pusatli, O. T., & Misra, S. (2011). Software Measurement Activities in Small and Medium Enterprises: an Empirical Assessment. Acta Polytechnica Hungarica, 8(5),21-42.

Qumer, A., & Henderson-Sellers, B. (2008). An evaluation of the degree of agility in six agile methods and its applicability for method engineering. *Information and Software Technology*, 50(4), 280-295.

Qureshi, M. R. J. (2011).Empirical Evaluation of the Proposed eXSCRUM Model: Results of a Case Study,*IJCSI*, 8(3), 150-157.

Redouane, A. (2004). *Towards a new method for the development of Web-based applications,* in proceeding of the 3rd IEEE international Conference on Cognitive information (ICCI’04),
Washington ,DC. IEEE Computer Society, 116-122.

Richardson, I., & von Wangenheim, C. (2007). Guest Editors' Introduction: Why are Small Software Organizations Different? IEEE Software, 24(1),18-22.

Rodriguez, D., *Harrison*, R., & Satpathy, M. (2002). A generic model and tool support for assessing and improving Web processes.

Stojanovic, Z., Dahanayake, A., & Sol, H. (2003). Modeling and Architectural Design in Agile Development Methodologies. EMMSAD’03.

Tarawneh , H & Allahawiah, S (2009), Web applications Development and Software Process Improvement in Small Software Firms: a Review, Proceedings of the 4th international Conference on Information technology (ICIT 2009), al zaytoonah University of Jordan.

Turk, D., France, R., Rumpe, B. (2002*). Limitations of Agile Software Processes*, Third International Conference on eXtreme Programming and Agile Processes in Software Engineering, Sardinia, Italy, 43-46.

Van Solingen, R., & Berghout, E. (2001). *Integrating goal-oriented measurement in industrial software engineering: industrial experiences with and additions to the Goal/Question/Metric method (GQM).*7th International Software Metrics Symposium, London,UK,

Von Wangenheim, C. G., Punter, T., & Anacleto, A. (2003*). Software Measurement for Small and Medium Enterprises.*7th International Conference on Empirical Assessment in Software Engineering (EASE). Kelee, UK

.