Proceedings of the 4th International Conference on Computing and Informatics, ICOCI 2013
28-30 August, 2013 Sarawak, Malaysia. Universiti Utara Malaysia (http://www.uum.edu.my)
Paper No. 033

IDENTIFICATION OF SUITABLE WEB APPLICATION DEVELOPMENT METHODS FOR SMALL SOFTWARE FIRMS

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

ABSTRACT. Many development methods have been proposed for developing web application in small software firms. However, these methods have some limitations. This paper aims to identify the suitable development methods for building high quality web application. In order to achieve this objective, a comparative study was conducted on several current development methods. Comparisons were made according to five criteria that include fitted to 10-50 size, simplicity, flexible to change, customer collaboration and quality assurance used measurement program (QAMP). The findings of this paper will be used as a baseline for building a new development methodology for small software firms that emphasize on monitoring.

Keywords software development methods, small software firms, XP, Scrum

INTRODUCTION

Web-based applications have been known to be of high reliability, high usability, more secured, incorporate advanced technologies, takes a shorter time to market, have a shorter product life cycle and required continuous maintenance (Rodriguez et al., 2002).

Many small software firms are involved with developing Web applications (Richardson & Wangenheim, 2007). A 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). The current problems faced by these firms include: i) limited resources for development; ii) limited number of available developers; iii) limited staff skills; vi) lack of well-defined development method; and v) limited adoption of Quality Assurance and measurement 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).

The development methods for building web applications can be categorized into conventional and non-conventional methods. The two types of methods were studied because they are well-known methods and cover the majority of software development methods. Based on the two types of methods, the study will identify the most suitable development method for small software firms.

The remainder of this paper is organized into several sections. In the next section, steps of conducting the research are presented. The findings and discussion section presents answers to the research objectives, while, the conclusion concludes the overall findings.

METHODOLOGY

This study was conducted in two phases: Identification of the current development methods; and identification of the suitable development methods for small software firms.

Phase One: Identification of the current software development methods

In this phase, resources from journals, books, conferences and internet materials from the year 1970 to 2000 were studied. The aim was to identify the current software development methods. Fourteen methods categorized into two groups' conventional and non-conventional methods were identified.

The methods were Waterfall, Incremental, V-model, Prototype, Spiral, XP, Scrum, Crystal Family Methodologies (CFM), Agile Modeling (AM), Adaptive Software Development (ASD), Dynamic Systems Development Method (DSDM), Feature-Driven Development (FDD), Lean Software Development (LSD) and Rational Unified Process (RUP).

Conventional methods include methods Waterfall, Incremental and V-model. Non-conventional consists of evolutionary methods (Prototype and Spiral) and agile methods (XP, Scrum, CFM, AM,ASD, DSDM,FDD,LSD and RUP). These methods were found to be the well-known development methods in the software industry.

Phase Two: Identification of the suitable development methods for small software firms

This phase was conducted to determine the development methods that are suitable for small software firms. This was done by comparing all 14 methods attained from Phase One in terms of five criteria: fit to 10-50 size, simplicity, flexible to change, customer collaboration and quality assurance used measurement program (QAMP). These criteria were extracted from past studies that were related to software development in small software firms (Haung et al., 2008; Tarawneh and Allahawiah, 2009; Pusatli and Misra, 2011; Rodriguez et al., 2002). Brief descriptions of these criteria are given in Table 1.

Table 1. Criteria and description

Criterion	Description
Fit to 10-50 size,	The number of employees ranges from 10 to 50.
Simplicity	The development method does not require high experience and skills.
Flexible to change	The development methods should deal with requirements changes.
Customer collaboration	The development method should involve the customer within the development process.
QAMP	The quality of process and product should be ensured using a set of metrics applied by monitoring program.

Each criterion can have values of either "Yes", "Less" or "No". Each value has a particular score in which, Yes = 3 , Less = 2 and No =1. The total score of each development method was calculated by adding all scores. The lowest score that can be achieved by a method is 5 and the highest score that can be achieved is 15.

Table 2 shows the values of each criterion and the total score of each method. A criterion value was given to each method based on analysis of past literatures.

FINDINGS AND DISCUSSION

In this section, the suitable methods for small software firms are presented.

Suitable Development Methods For Small Software Firms

As mentioned earlier, in Phase Two, the two categories software development methods conventional and non-conventional were compared using five criteria: fit to 10-50 size, simplicity, flexible to change, customer collaboration, and quality assurance used measurement program (QAMP). Table 2 shows the results.

Table 2. Software Development Methods Comparison

Method \ Criteria			Fit to 10-50 size	Simplicity	Flexible to change	Customer Collaboration	QAMP	Total score
Conventional Methods		Waterfall	No	Yes	No	No	No	7
		Incremental	No	No	Less	Less	No	7
		V- model	No	No	No	No	No	6
Non-Conventional Methods	Evolutionary	Prototype	No	No	Less	Less	No	7
		Spiral	No	No	Less	Less	No	7
	Agile Development Methods	XP	Yes	Yes	Yes	Yes	No	13
		Scrum	Yes	Yes	Yes	Yes	No	13
		CFM	No	Yes	Yes	Yes	No	11
		AM	No	No	Yes	Yes	No	9
		ASD	No	No	Yes	Yes	No	9
		DSDM	No	Less	Yes	Yes	No	10
		FDD	No	No	Yes	Yes	No	9
		RUP	No	No	Yes	Yes	No	9
		LSD	No	Less	Yes	Yes	No	10

Sources:
1- Conventional methods sources: (Naqvi, 2007), (Munassar & Govardhan, 2010), (Koblenz, 2003), (Awad, 2005), (Imreh & Raisinghani, 2011), (Okoli and Carillo, 2012).
2- Evolutionary methods sources: (Alite & Spasibenko, 2008), (Koblenz, 2003), (Munassar & Govardhan, 2010).
3- Agile methods sources: (Lindstrom & Jeffries, 2004), (Stojanovic et al., 2003), (Väänänen 2008), (Beck, 1999), (Abrahamsson et al., 2002) (Schwaber & Beedle, 2001).

Based on Table 2, the first method i.e Waterfall received a total score of 7. This is calculated by substituting the value Yes with "3", Less with "2" and No with "1" and adding all the scores together. Therefore,

$$\text{No} + \text{Yes} + \text{No} + \text{No} + \text{No} \quad => \quad 1 + 3 + 1 + 1 + 1 = 7$$

The table shows that the total scores for all methods range from 6 to 13. The two lowest scores are found to be 6 and 7, deriving from the conventional type (waterfall, incremental and V-model) and the non-conventional, specifically evolutionary methods (spiral and prototype).

The highest total score is 13, deriving from the agile methods namely XP and Scrum. Other methods i.e CFM, LSD, and DSDM scored 11, 10, and 10 respectively. The rest of the methods (AM, ASD, RUP and FDD) scored 9.

All agile methods concentrate on customer collaboration and requirement change criteria. However, not all of them adopts a measurement program to ensure the quality of the process and product. In addition, four out of nine agile development methods (AM, ASD, FDD and RUP) are found to be complex development methods. XP, Scrum, CFM, LSD and DSDM are identified as either less complex or simple methods.

Table 2 shows that the most suitable methods that can be used for developing Web application in small software firms are XP and Scrum. These methods satisfy four out of five criteria, while the other development methods satisfy less than four criteria. This finding is similar to Ahmad et al., (2012b).

However, XP and SCRUM are shown to be lacking in applying QAMP (Fernandes & Almeida, 2010; Jyothi and Rao, 2011; Qureshi, 2011). Incorporating QAMP in XP can improve XP management practices and at the same time, monitor the development practices in Scrum. This findings support claims of Fritzsche & Keil (2007), and Qumer & Henderson-Sellers (2008) that affirmed that both XP and Scrum need a qualitative and quantitative metrics to monitor the quality of process and product. In addition, Table 2 shows that XP and Scrum are flexible to requirement changes by using iterative development style. However, both XP and Scrum have problems in terms of tracing and reusing requirements (Fernandes & Almeida, 2010; Fritzsche & Keil, 2007).

The design phase for both methods (XP and Scrum) is simple and this is in line claims from Fritzsche & Keil (2007) and Qumer & Henderson-Sellers (2008).

Based on the above discussion, XP and Scrum though are suitable for small software firms, still have some limitations. Thus, this shows that there is a need of a new development methodology for building web application in small software firms based on XP and Scrum. The limitations in both methods can be improved by incorporating QAMP, establishing requirement repository and merging simple design prototype. The proposed enhancement to cover these limitations is also recommended by Ahmad et al., (2011), Ahmad et al., (2012a) and Ahmad et al., (2012b).

CONCLUSION

This paper aims to identify the most suitable development methods for small software firms. Conventional and non-conventional methods were compared based on five criteria namely: fit to 10-50 size, simplicity, flexible to change, customer collaboration and quality assurance used measurement program (QAMP). Results showed that the most suitable development methods to be used for developing web applications in small software firms are XP and Scrum. However, both methods have limitations which justified the need for a new development methodology based on XP and Scrum.

REFERENCES

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

Ahmad, F., Baharom, F., & Husni, M. (2011). Investigating the Awareness of Applying the Important Web Application Development and Measurement Practices in Small Software Firms. *IJCSIT 3*(6), 147-158.

Ahmad, F., Baharom, F., & Husni, M. (2012a). Current Web Applications Development and Measurement Practices for Small Software Firms. *International Journal of Computer Science, 9*(2), 125-129.

Ahmad, F., Baharom, F., & Husni, M. (2012b). *Agile Development Methods for Developing Web Application in Small Software Firms.* Paper presented at the Knowledge Management International Conference (KMICe), Johor Bahru, Malaysia.

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.

Awad, M. (2005). A Comparison between Agile and Traditional Software Development Methodologies, Report, University of Western Australia.

Beck, Kent. (1999). Embracing change with extreme programming. *IEEE Computer*, *32*(10). 70-77.

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

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

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

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.

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

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. & Cleaver, V. (2008*). Web Application Development Lifecycle for Small Medium-sized Enterprises(SMEs)*, The Eighth International Conference on Quality Software, 247-252.

Imreh, R., & Raisinghani, M. S. (2011). Impact of Agile Software Development on Quality within Information Technology Organizations. *Journal of Emerging Trends in Computing and Information Sciences, 2*(10). 460-475.

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

Koblenz, C. (2003). *Maintenance Activities in Software Process Models: Theory and Case Study Practice*. University of Koblenz Landau Campus Koblenz., Mastre thesis

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.

Naqvi, S. (2007). *A Semi-Autonomous On-Line Chemotherapy Prescription System*, Master, Thesis. Department of Computer Science, Memorial Newfoundland Uni, Canada.

Munassar, N. M. A., & Govardhan, A. (2010). A Comparison between Five Models Of Software Engineering. *IJCSI International Journal of Computer Science Issues, 7*(5), 94-101.

Okoli, C., & Carillo, K. (2012). The best of adaptive and predictive methodologies: open source software development, a balance between agility and discipline. *International Journal of Information Technology and Management, 11*(1), 153-166.

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.

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.

Schwaber, K. and Beedle.(2001). *Agile Software Development with Scrum: Upper Saddle River*: *Prentice Hall.* New Jersey 2001.

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.

Väänänen, M. (2008). *Evaluating agile methods and their implementations*, Master thesis, Information System Competence, Applied Sciences UNI, Finland.