



ELSEVIER

Available online at www.sciencedirect.com



Procedia Computer Science 3 (2011) 893–897

Procedia
Computer
Science

www.elsevier.com/locate/procedia

WCIT 2010

A Proposed Methodology for Establishing Software Process Development Improvement for Small Software Development Firms

Mejhem Yousef al-Tarawneh^a, Mohd Syazwan Abdullah^b, Abdul Bashah Mat Ali^c

^{a,b,c} Applied Science Division, College of Arts and Sciences, Universiti Utara Malaysia, 06010 Sintok, Kedah, Malaysia

Abstract

Small software development firms represent the majority of all software firms in most countries. These firms are facing the same software engineering challenges that affect large software firms. Software Process Improvement (SPI) traditional models were developed to help large and very large firms, however small software firms could not afford these models. Furthermore, they need to manage and improve their software development processes for several reasons such as dealing with the rapid technology advances, maintaining their products, satisfying the customers' needs and sustaining their operations. This paper presents the methodology's stages of developing a suitable software development process improvement framework by using Capability Maturity Model Integration (CMMI-DEV V1.2) as the basic model for improvement and Extreme Programming (XP) method as the basic software development method.

© 2010 Published by Elsevier Ltd. Selection and/or peer-review under responsibility of the Guest Editor.

Keywords: Software Process, Software Process Improvement, Small Software Development Firms.

1. Introduction

Software industry is the most rapidly growing sector and the small software development firms in this sector play a fundamental role in most countries economies [1]. As a result, these firms need to manage and improve their software processes to satisfy the customer's requirements within the time frame at a lower cost, while maintaining the quality. SPI is "a systemic procedure for improving the performance of an existing process system by changing or updating the process" [2]. Unfortunately, there is no specific SPI model for these firms as all the current SPI models were developed for large firms, and these improvement models are difficult and not suitable to be used by small software development firms due to the fact that these models are too complicated and expensive to be implemented [3]. However, some researchers indicated that SPI can be used as a competitive advancement strategy for both small and large organizations [4]. Furthermore, there is a lack of empirical research about the small software firms. O.Pedreira et al. [5] pointed out in their survey about the empirical studies in the digital libraries such as IEEE Computer Society Digital Library, ACM Digital Library, Wiley Inter Science (computer science area), Science Direct (computer science area), Springer Link, and others that there are only 20% of empirical studies about small firms and 80% about large firms. Nevertheless, small software development firms need to improve their software development processes and this requires a suitable software development improvement framework for these firms.

2. The Need of SPI in Small Software Development Firms

In software process improvement field, many researchers have been focused on large and very large firms because most of these firms have enough investment and resources to improve their software processes by implementing the SPI traditional models and standards, where these models guide the software development improvement process [6]. Unfortunately, none of these traditional models and standards can readily be used by small software firms that are representing the majority of all firms in most countries. Small software development firms suffer from the lack of research studies to solve their problem of improving their software processes [5], where most of the empirical studies into the rate and success of implementing the SPI in small software firms are always considered as being inadequate as most studies were focused on large and very large firms [4]. Moreover, most of these firms adopt the ad-hoc approach to develop their software products, and as a consequence these firms need to manage and improve their development processes [16, 17]. Furthermore, small software development firms also require specific software development process improvement framework to be able to manage and improve their software development processes to enable them in delivering high quality software that fulfils customers' expectations faster and cheaper.

3. Extreme Programming (XP)

Agile Development Methodologies have been designed to address the problem of delivering high-quality software on time under constantly and rapidly changing requirements in business and IT environments. The emergence of agile methodologies began in the mid 1990s, and these methods are considered the newest for software development [7]. Pressman [8], pointed out that the general agile methods are Extreme Programming (XP), Scrum, Crystal, Dynamic Systems Development Method (DSDM), Adaptive Software Development (ASD), Feature-Driven Development (FDD), Lean Software Development (LSD), Agile Modeling (AM), and Agile Unified Process (AUP).

XP [9] is the most popular lightweight method in agile software development methods. This model was created to solve many limitations of traditional software development models such as: (1) most of the traditional models may not meet the customers requirements; (2) all traditional models were developed to suit large software; (3) project cancelation after a long period without ever going into production; and (4) misunderstood the business requirements and the software never solves the business problem for which it was developed. Furthermore, the XP method is called extreme because it adopts the good practices to develop the software and applies these practices extremely [9].

Since most of small software development firms adopts ad-hoc manner to develop their software product, therefore there is needed to take into account an appropriate software development method in developing the proposed framework for these firms to help them develop their software products in the right way.

The proposed methodology discussed in this paper will adapt XP as a baseline software development method for several reasons [10, 11]:

- XP is more applicable for small, medium-scale and less complex projects as it is the most widely used agile methods as well as the more prominent approaches that adhere to agile principles.
- XP is an easy model for learning.
- XP could be easily adapted with changing requirements.
- XP achieves software process improvement better than agile methods; it conforms to level two in CMMI.
- XP is a lightweight process model that can help small firms in the implementation of software process improvement.
- XP practices can work tightly together by carefully applying different practice at a time that will eventually lead to improvement.

4. Capability Maturity Model Integration (CMMI)

CMMI was developed by Software Engineering Institute (SEI) at Carnegie Mellon University (CMU) based on some emerging CMM models and these models are Capability Maturity Model for Software (SW-CMM) v2.0 draft, Systems Engineering Capability Model (SECM), and Integrated Product Development Capability Maturity Model (IPD-CMM) v0.98, and the first version of this model was released in 2000 [4]. The objective of the CMMI is to provide a guideline for improvement of the software process in the organizations, and it is written specially for the software industry to describe the software processes in details. Furthermore, CMMI Product Team pointed out that the CMMI supports the organizations to manage the development, acquisition, and maintenance of products or services. CMMI also focuses on the supplier to improve the internal software process, where it has two representations; the first one is continuous representation that focuses on process areas capability and are measured by capability levels, and the second is staged representation; which focuses on organizational maturity and are measured by maturity levels [12].

CMMI model is the most widely used compared to other software process improvement models, where CMMI is supposedly applicable for all the size of firms [18]. However, this model is not ready to be used by small software firms [3]. Nevertheless, the continuous representation can be useful and is more applicable for small software firms compared to stage representation, as the small software firms could allocate its' limited resources on the most striking problems [13]. Based on this, the CMMI-DEV V1.2 continuous representation will be used as a baseline improvement model in developing the software development process improvement framework for several reasons such as: (1) CMMI have been used to guide the software development improvement such as CMMI-DEV V1.2 ; (2) CMMI is the comprehensive software improvement model of the SEI where this model mostly comply with relevant international standards such as ISO 9000 and ISO/IEC 15504; (3) CMMI model improves upon the best practices of other improvement models in many important ways; (4) CMMI gains the international acceptance in the software engineering community ; and (5) CMMI have been broadly used for assessing and improving the organizational maturity and process capability throughout the world, where customers have confidence in CMMI because of its extensive descriptions of how the various good practices fit together [6,14,15].

5. Proposed Methodology

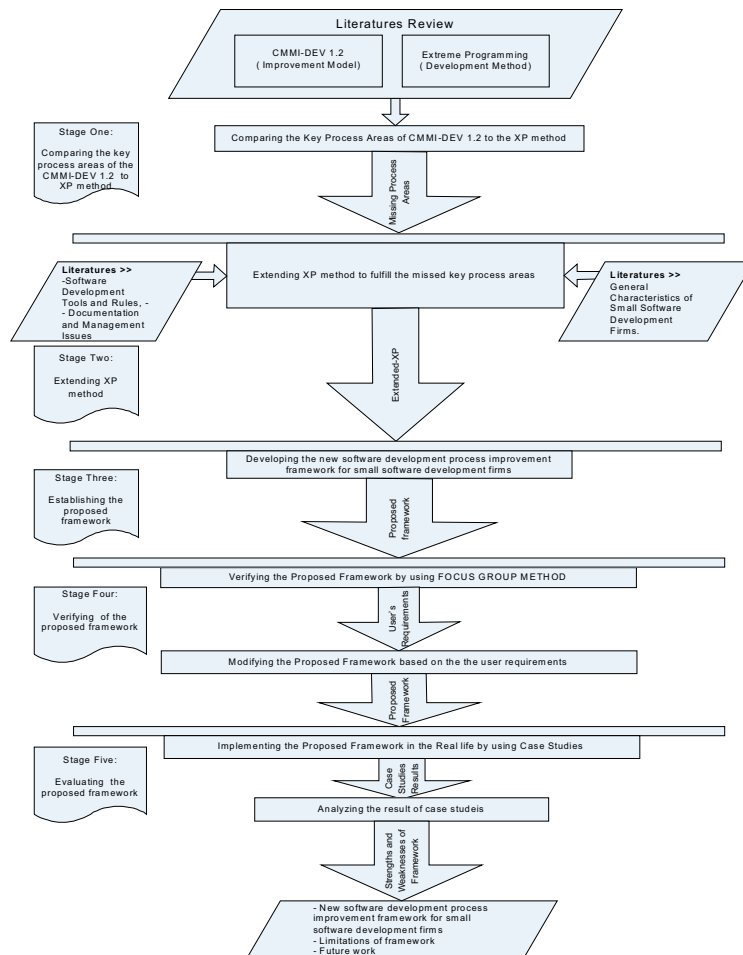
The proposed methodology consists of five stages that can be used to establish the new software development process improvement framework for small software development firms. Each stage has tasks to achieve the goals of the stage. Figure 1 highlights the stages and these are:

- **Stage One:** Comparing the key process areas of CMMI-DEV v1.2 to the XP method.
 - *Task 1:* Based on the literatures, the CMMI-DEV v1.2 key Process Areas will be compared to XP to identify the Key Process Areas of CMMI-DEV v1.2 that can be covered or missing by XP method. This can be done by comparing the specific practices of CMMI-DEV1.2 key process areas to the to XP method (practices, roles, activities, etc) to identify what are the coverage and missed specific practices of CMMI-DEV 1.2.
- **Stage Two:** Extending XP method to fulfill the missed key process areas of CMMI-DEV 1.2.
 - *Task 2:* Based on the missing specific practices of each key process areas of CMMI-DEV 1.2, it will be needed to address the required activities (from the literatures) that are required to fulfill the missing key process areas that are suitable for these firms and adding these activities to the XP method. This is done while taking into account the general characteristics of small software development firms. The outcome of this phase is the Extended-XP.
- **Stage Three:** Establishing the proposed framework
 - *Task 3:* Based on the general elements of software process improvement framework and the Extended-XP, the proposed software development process improvement framework will be established.
 - *Task 4:* Specifying the needed general requirements of the proposed framework such as procedures, rules, techniques, tools, documentation and management issues, providing guidance to order the activities, directing the tasks of the teams, individual developers, and other required requirements.
- **Stage Four:** Verifying the proposed framework and modifying it if required.
 - *Task 5:* Focus Group method will be used to verify the proposed framework and this will be done by

discussing the capability of proposed framework for achieving the requirements of the specified Key Process Areas of CMMI-DEV v1.2 and taking into account the general characteristics of small software development firms.

- **Task 6:** Analyzing the results of task 5 to determine the user requirements and based on these requirements, the activities that will be added or deleted will be determined. Then, the undesirable activities will be deleted and the activities which can satisfy and achieve the user requirements will be identified and added, taking into account the suitable specific practices of key process areas of CMMI-DEV v1.2 model and the characteristics of small software firms.
- **Stage Five:** Evaluating the proposed framework.
 - **Task 7:** Selecting small software development firms to validate the proposed framework and analyzing the results of case studies to know the efficiency, reliability, and simplicity of the proposed framework.
 - **Task 8:** Based on the results of task 7, it can be concluded the limitation of the new framework and the future work of this study.

Figure 1: Development stages of software development process improvement framework for small software development firms.



6. Conclusion

Small software firms represent a high proportion of software firms all over the world, and play a fundamental role in the economy of many countries. Unfortunately, these firms do not have the suitable software process development model to achieve the key process areas of one of SPI traditional models since these models are created to help large and very large firms. Nevertheless, these firms need to manage and improve their software processes to satisfy the customer's needs faster, cheaper and of high quality. This paper discusses the methodology stages to guide the development of a suitable software process development improvement framework for small software development firms by using XP as a software development model and CMMI as a SPI model.

7. References

- 1.F. Pino, F. Garcia, And M. Piattini, "Key Processes To Start Software Process Improvement In Small Companies," (2009), Pp. 509-516.
- 2.Y. Wang & G.King,. Software Engineering Processes: Principles and Applications. CRC Press LLC, Boca Raton, FL, USA,(2000).
- 3.D. Mishra And A. Mishra, "Software Process Improvement In Smes: A Comparative View," Computer Science And Information Systems/Comsis, Vol. 6, Pp. 111-140, (2009).
- 4.A. Cater-Steel, "An Evaluation Of Software Development Practice And Assessment-Based Process Improvement In Small Software Development Firms," Ph.D. Dissertation, Sch..Com. Info. Tech, Griffith Uni, Australia, (2004).
- 5.O. Pedreira, Et Al., "A Systematic Review Of Software Process Tailoring," Acm Sigsoft Software Engineering Notes, Vol. 32, No. 3, (2007) , Pp. 1-6.
6. D. David, "Instructional Software Analysis: Lessons from Software Development Process Improvement," Paper presented at the Seventh IEEE International Conference on Advanced Learning Technologies (ICALT 2007). Niigata, Japan, (2007).
7. O.Salo, Enabling software process improvement in agile software development teams and organisations. Academic Dissertation presented with the assent of the Faculty of Science, University of Oulu, for public discussion in the Auditorium L10, Linnanmaa, (2006).
8. R. Pressman, Software Engineering: A Practitioner's Approach. 7th international edn, McGraw-Hill Education, Newyork, USA, (2009).
9. K. Beck, Extreme programming explained: Embrace change, 3th End.Reading, Mass, addition-Wesley. Boston., (2000).
10. J. Alegre a and M. Bastarrica, "Implementing CMMI using a Combination of Agile Methods," CLEI ELECTRONIC JOURNAL, vol. 9, (2006).
11. M. Fritzsche and P. Keil, "Agile Methods and CMMI: Compatibility or Conflict?," e-Informatica Software Engineering Journal, vol. 1, (2007).
12. CMMI Product Team, CMMI for Development, version 1.2. Preface, Software Engineering Institute, Carnegie Mellon University, USA,(2006).
13. F. Wilkie, D. McFall, and F. McCaffery, "An evaluation of CMMI process areas for small-to medium-sized software development organisations," Software Process: Improvement and Practice, vol. 10, pp. 189-201, (2005).
14. CMMI Product Team, Capability Maturity Model® Integration (CMMI), Version 1.1: CMMI for Systems Engineering, Software Engineering, Integrated Product and Process Development, and Supplier Sourcing, CMMI-SE/SW/PPD/SS, V1.1, Carnegie Mellon University Software Engineering Institute, Pittsburgh PA,USA,(2002).
- 15.M. Bush and D. Dunaway, CMMI (R) Assessments: Motivating Positive Change (Sei Series in Software Engineering): Addison-Wesley Professional, 2005.
16. M. Yousef, M. Abdullah, and A. Ali, "Software development process in Jordanian small software development firms Small," MASAUM Journal Of Reviews and Surveys, Vol. 1, No.2,(2009).
17. A. El Sheikh and H. Tarawneh, "A survey of web engineering practice in small Jordanian web development firms," paper presented at of the European Software Engineering Conference and the ACM SIG-SOFT International Symposium on Foundations of Software Engineering (ESEC/SIGSOFT FSE), Dubrovnik, Croatia, Pp. (481-490), 2007.
18. F. Yuclar and S. Erdogan, "A QUESTIONNAIRE BASED METHOD FOR CMMI LEVEL 2 MATURITY ASSESSMENT," Journal of aeronautics and space technologies, Vol. 4, Pp. 39-46, (2009).