The Nucleus **55**, No. 1 (2018) 17-21

www.thenucleuspak.org.pk

The Nucleus
ISSN 0029-5698 (Print)
ISSN 2306-6539 (Online)

Adapting Agile Practices to Improve the Quality of Small and Medium Scale Organizations

H. Naseer[*], Y. Hafeez and W.A.A. Arshad

University Institute of Information Technology, Pir Mehr Ali Shah Arid Agriculture University, Rawalpindi, Pakistan

ARTICLE INFO

Article history :
Received : *29 April, 2017*
Accepted : *25 April, 2018*
Published : *27 April, 2018*

Keywords:
Agile methodologies
Software process improvement
Information systems
Management,
Agile system
Scrum

ABSTRACT

Most of the Software industries are demonstrating absorption in Software Process Improvement (SPI) in Small and Medium Enterprises. In the modern world, software development and improvement is an important area that permits organizations to develop applications in house. Traditionally, waterfall system is used to administer software development. The traditional, waterfall model counted upon linear, sequential stages, well explained features, sign of documentation and widespread coverage. The outcome of the study is to formulate a holistic model for customization of agile practices for small and medium organizations. This paper has demonstrated a case study emphasizing on tailoring agile methods including tailored XP (extreme programming) focusing Small and Medium Enterprise (SME) for software process improvement. This framework is for software processes being implemented by SME merging agile methods XP, Scrum and defines the handover in pre-transition and post order.

1. Introduction

Over the last few decades, it has been observed that the software development has gained an immense importance globally. Traditional development failed due to its complex nature and rigid behaviour and practitioners moved their development trends from typical to modern development approaches such as agile development practices. Agile is the most popular software development methodology in the current era. Traditionally, waterfall system has been used to administer software development process. Waterfall model focuses on linear and sequential stages, well explained features, sign of documentation and widespread coverage. Agile software development prefers a rapid development approach as compared to its heavily-planned counterparts. In agile software development, the context of an underlying development project has a strong relation with the selection of appropriate agile methodology [1]. This selection requires tailoring of existing methodologies to suit a given context.

Method Engineering is the process of systematic analysis, comparison, and construction of new methods from the existing methods. Moreover, method engineering focuses on "The design, construction and evaluation of methods, techniques and supporting tools for software development" [1]. Using method engineering, new methods evolve from existing methods to cater for emerging critical situations. For reader’s understandability, a situation is defined as a combination of circumstances at a given moment in an underlying organization [2].

Methods are stored in a hierarchical format in an electronic database. Leaf nodes are features of a given development paradigm. A feature is a description of a software engineering method used to build new methods according to the assigned situation [3]. This method leads to the other methods or relevant features on the higher hierarchy methods. Method engineer is a person whose role in the situational method engineering, is to formulate method’s hierarchy. Later, the method engineer will tailor the software according to an assigned situation or given context.

Software researchers and practitioners have contributed sufficiently in the field of Software development. The waterfall model is planned-driven and process oriented traditional approach, referring to this paradigm. Boehm [4] depicted that CMMI (Capability Maturity Model Integrated) is one of the renowned standard. Most of the software organizations applied CMMI in order to measure the quality of software development processes. This standard is also helpful for practitioners to improve the performance and efficiency of daily software development practices. This standard has two different representations, first one is stage representation which is systematic approach; whereas, second one is continues representation which is bit flexible approach for practitioners. In the last decade, software change management evolved frequently in small and medium size projects. Software continues to develop in size and complexity. Developers and users focus on the time to market, software quality, and ease of use [5]. For flexible development methods, major emphasis has focused on customization of different types of projects [6]. Considering the effectiveness of Agile-system methods, the present paper shows its application in a Small to Medium Enterprise (SME) as a case study. Combination of two agile methods, i.e. Scrum and XP, are applied in a SME. Software configuration management is a field of study to handle the advancement of computer software products, both during the early phases of development and during all phases of maintenance [7]. Nowadays agile is one of the popular paradigms globally; however practitioners have

[*]Corresponding author : hafsanaseer89@yahoo.com

faced problems in versatile situations. Hence, it is necessary for a comprehensive model in order to fulfill the versatile nature of the software industry

The first section of present paper highlights introduction to agile methods and their possible customization for small and medium industries, specifically designed by authors. In addition, Scrum and XP are often used in the software industry are thoroughly explained. The subsequent section of the paper encompasses the proposed approach. Later, the proposed work will focus on the research method reinforcement. The following section states the framework of tailor Scrum in software development process within SME's. The next section provides comparison of the studies along with results and discussion to analyze and evaluate the proposed solution. Last section contains concluding facts of the research paper. Literature review helped to identify, classify, assess and understand the contents of research topics. Table 1 shows issues related to reduce the complexities and for intensifying the software improvement.

Table 1: Issues related to software improvement.

Issues	Description
Disproportionate focus on governance [2,8]	Less focused on enhancing governance
Reduced productivity [7]	Not an effective way of enhancing the productivity
Larger development phases [9]	Not focused on proposing shorter developed phases
Larger development time tables [12]	Not focused on developing shorter development time tables

Researchers have made many discoveries which need to be taken into consideration when evaluating the proposed methodology of this study. Generally software development was performed the dominant waterfall system, but the researchers did not focus on governance improving quality, flexibility and most importantly reducing productivity. The major aim of this research to focus on the software situations to tailor the agile base product. In addition, it includes the goal of maximizing gain by enabling situational factors related to improve the agile practices, which leads to Software Process Improvement (SPI).

No appropriate methods are suitable for variant situations in software development projects. The approaches used to develop methodologies that fit all situations occur in projects named Situational Method Engineering (SME). Taromirad and Cefam [9] have proposed a method based on software project engineering Meta-model (SPEM 2.0) plugged into Computer Aided Method Engineering (CAME) tool. The aim of this tool follows assembly based SME approach to develop the customized agile approaches. However, the validity of this framework has not yet been examined.

Ramsin et al. [10] depicted a Meta-model $\mathrm{M^2M^2F}$ that developed through integration between RUP and ISO/IEC 24744 Meta-models. It allows defining method fragments with agile, planned and hybrid use contexts when selecting fragments project risk was not examined.

Jeffries conducted a case study that indicated the use of Agile Software Solution Framework (ASSF) in a SME [11]. ASSF uses agile and formal practices for the creation of hybrid methods for a situation in a large software development organization [12]. However, this framework was particularly designed for large organization and is not capable for small organizations. Table 2 presents summarized review of literature.

Table 2: Summary of literature review.

Papers	Strength	Weakness
Agile data modelling & design thinking approach to information system requirements Analysis [13]	Proposing requirements analysis process about the application of agile and design for business oriented data modelling.	Focuses on Non-technical factors e.g people and organizational culture
Multi-Dimensional Success Factors of Agile Software Development Projects [14]	Designed successfully agile features demonstrate success factors of Agile Sofware development projects.	Difficult to analyze Un-realistic schedule Lack of user involvement
Comprehensive Evaluation Framework for Agile Methodologies [9]	Proposed the concept of meta models	The validity of this framework has not yet been examined
Defining Agile and Planned Method Fragments for Situational Method Engineering [5]	Meta-model$\mathrm{M^2M^2F}$ that developed through integration between RUP and ISO/IEC 24744 meta models	When selecting fragments project risk are not examined
Construction of an Agile Software Product-Enhancement Process by Using an Agile Software Solution Framework (ASSF) and Situational Method Engineering [2]	Conducted case study based on Agile Software Solution Framework (ASSF)	This framework was particularly designed for large organizations and is not useful for small organizations

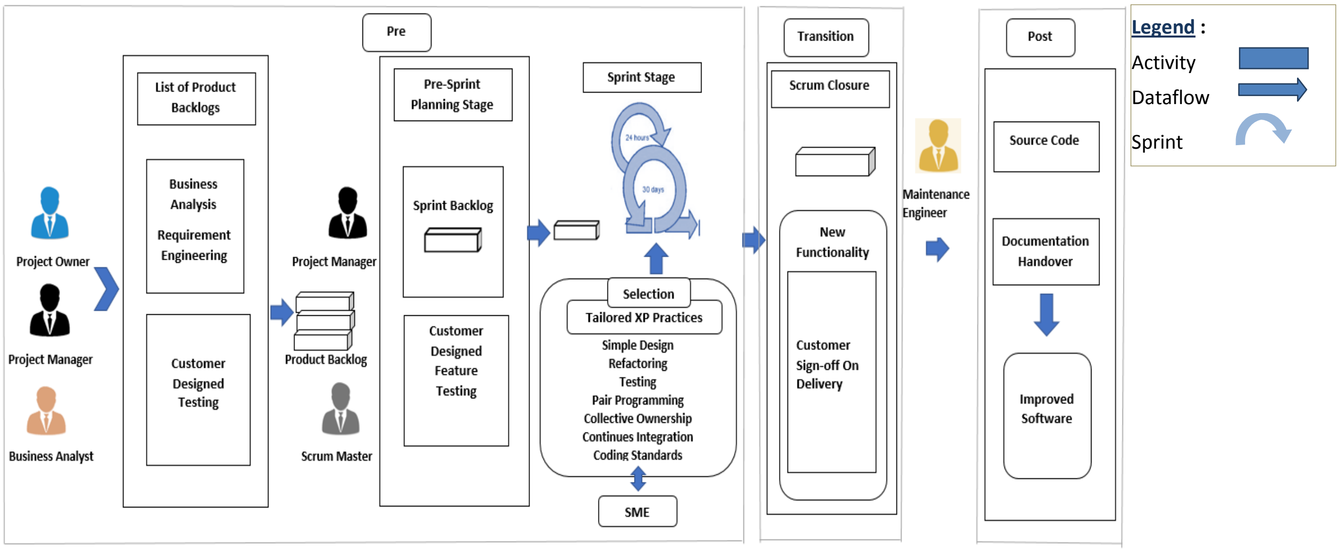

Fig. 1: Software development process tailoring Scrum/XP within SME.

2. Materials and Methods

An exploratory and interpretative case study research technique was adopted for this study. Using this case study through qualitative research methods, data collection was carried out. Primary interviews were conducted in a two-hours period. Secondary interviews consumed fifty minutes in order to investigate the phenomena, as shown in Fig 1. In this proposed solution, firstly Scrum is tailored in software development process within SME's. Each stage includes the handover process that also helps to enhance and improve the efficiency of the software process.

2.1 *List of Product backlogs*

In this phase list of product backlog is prepared by the business analyst, which are is elicited from the product owner.

2.2 *Pre-Sprint Planning Stage*

In this stage, cross functional team plans were made and prepared a list of possible tasks which will be performed in the upcoming sprint.

2.3 *Sprint Stage*

In this phase, XP practices involving Pair Programming, Simple Design, Coding Standard, Continuous Integration, Collective Ownership, Testing and Refactoring are performed.

2.4 *Scrum Closure*

In this stage, a list of possible artifacts is prepared as per client specification. The customer sign-off is applied to the unit and final product delivery, which is shown in Tables 3 and 4.

2.5 *Post-Sprint Stage*

In this stage, when sprint has been completed all the stakeholders attend the post sprint meeting in which product owner decides to accept/reject or prioritize the presented product.

3. Results and Discussion

There are many research methods as action research, grounded theory and case study, which are normally used to evaluate empirical work. The researchers have evaluated the proposed framework through industrial experimentation.

3.1 *Industrial Experimentation*

To measure the satisfaction level of the customers, the present work presents Assessment Factors for industrial experimentation. The implied factors used are Domain Knowledge (AF1), Effective Communication (AF2), Managing Documentation (AF3), Change Management (AF4), Proper Planning (AF5), Shared Knowledge (AF6) and System Knowledge (AF7). In this experiment, a total of eighteen participants with the appropriate background knowledge participated. Above mentioned factors were used to evaluate the user's satisfaction towards the proposed framework. Results of the user's satisfaction are measured using assessment factors and are shown up in Table 5.

4. Comparison with Existing Models and Frameworks

Previous models and frameworks i.e. Meta Model (SPEM 2.0) [5], Meta Model M2F [5], ASSF [2] and ISO/IEC 24744 [5] are less focused to incorporate scrum/XP practices, whereas our proposed framework demonstrates the handover process with the incorporation of Scrum/XP practices. This proposed framework has also supports the software industry to implement the handover process in Scrum/XP practices. From the satisfaction level shown in Fig. 2, it can be seen that the participants were extremely satisfied by proposed framework including the handover process for Scrum/XP practices.

Table 3: Research assessment (Tailored Scrum process).

Stage/Practices	Efficiency	Risk
List of Product backlogs	At the earliest stage, strengthening trust between customer and developer	Customer, Project manager disengagement: cultural, social or business level
Pre-Sprint Planning Stage	Individuals align with the overall project milestones. Plan (based on minimum 20-day sprint duration) within the next sprint	More complex plans, Project personnel difficulties, Lack of customer knowledge of business processes.
Scrum Closure	Completely standard part of every project confirmed.	In customer organization, business process change or environment.

Table 4: Research assessment (Tailored XP practices).

Stage/ Practices	Efficiency	Risk
Continuous testing	Business functionality and software standards alignment	Not associated with the project component
Pair programming	Improved quality and productivity	Pair programming is not industrious for simple coding tasks.
Simple design	Produces important and sufficient product	Documentation on the code can create multiple dependencies.
Continuous integration	Overall, a greater knowledge of the architecture of the project	No project team or not to scale in terms of project size and complexity
Coding standards	Software architecture to ensure readability	Acclimatization time for new project members
Refactoring	Refactoring Reduces debugging, straightforward software architecture	Inadequate design, time delay problems.

Table 5: Evaluation through assessment factors.

Assessment Factor		Strongly Agree	Agree	Satisfactory	Disagree	Strongly Disagree	Satisfaction %
Domain Knowledge	AF1	6	4	5	2	1	83.3%
Effective Communication	AF2	7	5	4	1	1	88.8%
Managing Documentation	AF3	5	6	4	2	1	83.3%
Change Management	AF4	4	6	7	0	1	94.4%
Proper Planning	AF5	7	6	4	0	1	94.4%
Shared Knowledge	AF6	6	4	4	2	2	77.7%
System Knowledge	AF7	7	5	5	0	1	94.4%

The results have clearly reflect that the most of the participants strongly agreed with all the assessment factors and very few disagreed against each assessment. The results of this industrial experimentation are shown in Fig. 2 which helps to build our argument that the participants have shown their interest and their satisfaction level also increased after the successful execution of our proposed such as tailored Scrum/XP method within SME's software development process. Fig. 3 represents the user satisfaction towards proposed framework. Results of the user's satisfaction are measured using assessment factors. The results clearly show that majority of the participants are satisfied with the proposed framework.

5. Conclusions

This research resulted in effective tailoring of Scrum and XP practices in an SME based organization. Software developer assessed the risks and deficiencies from tailoring that lead to software process improvement (SPI). Considering the effectiveness of Agile-system methods, this study has signified its applications in Small and Medium Enterprise (SME). A software process has been implemented in this case study with the help of SME principles through the grouping of two agile methods, Scrum and XP which have defined the handover in pre-transition and post order. This case study has resulted in an

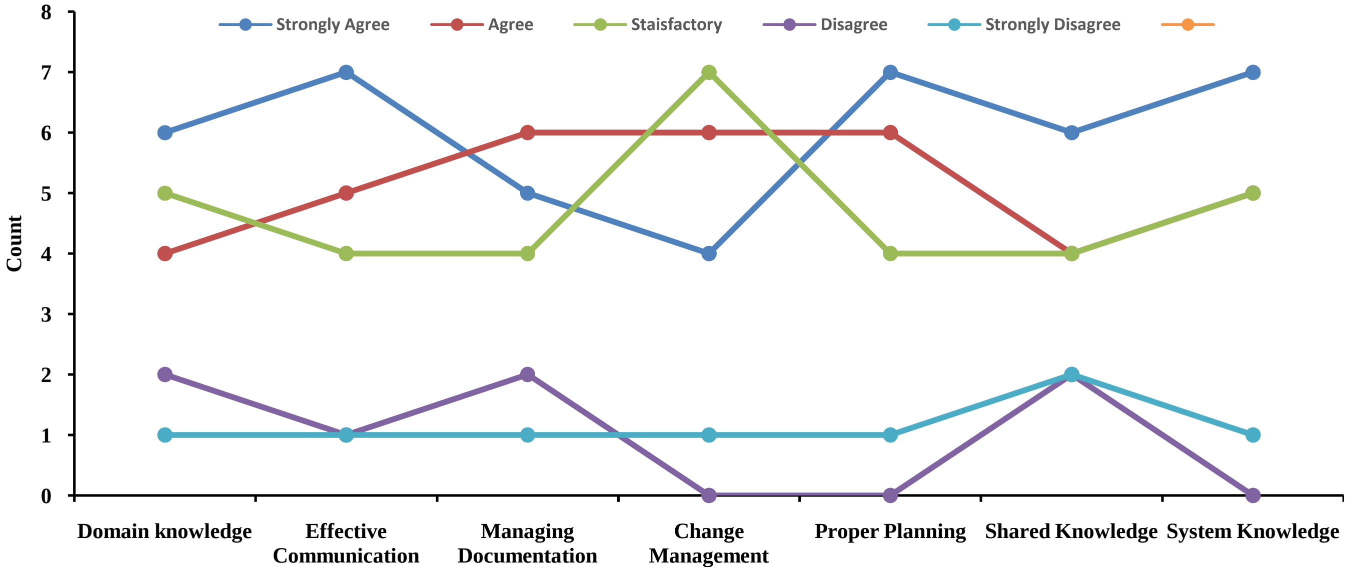

Fig. 2: Satisfaction level.

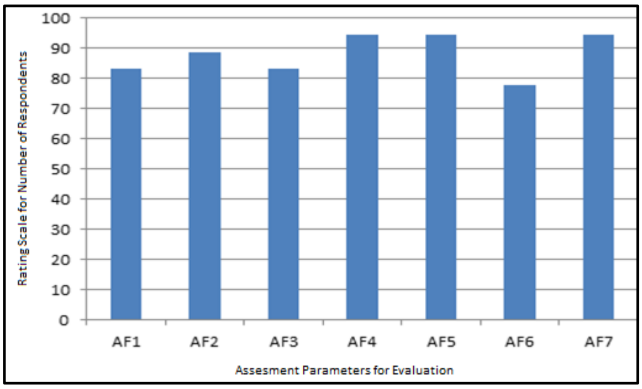

Fig. 3: Users Satisfaction towards Proposed Framework.

agile system approach for the single developer's software process, which may be tailored or fine-tuned using tailored agile based approach that influences SPI. This tailoring depends on the developer's assessment of best practice. In future, inclusion of several other features such as feature-model and software product line engineering for tailoring of the situational method engineering will be studied to provide ease in appropriate method selection. Further more agile practices also customized for global software development after incorporating cloud based approach may be used.

References

[1] Z.S.H. Abad and R. Ramsin, "Towards tool support for situational engineering of agile methodologies", 17th Asia Pacific Software Engineering Conference (APSEC)", vol.1, pp. 326-335, 2010.

[2] A. Qumer and B. Henderson-Sellers, "Construction of an agile software product-enhancement process by using an Agile Software Solution Framework (ASSF) and situational method engineering", 31st Annual International Conference on Computer Software and Applications (COMPSAC), Beijing, pp. 539-542, 2007.

[3] V.C. Gerogiannis and P.G. Ipsilandis, "Multi objective analysis for time boxing models of software development", ICSOFT (ISDM/EHST/DC), pp. 145-153, 2007.

[4] B.W. Boehm, "Software and its impact: A Quantitative Assessment", Datamation, vol. 19, pp. 48-59, 1987.

[5] G.V. Pereira and L.M. Fontoura, "Defining agile and planned method fragments for situational method engineering", VIII Simpósio Brasileiro de Sistemas de Informação, Fortaleza, Brasil, pp. 660-665, 2012.

[6] G.C. Succi and M. Marchesi", "Extreme programming examined", Addison-Wesley Longman Publishing Co., Inc, 2001.

[7] J. Koskela, "Software configuration management in agile methods", VTT Publications, pp. 29-33, 2003.

[8] L. Crispin and T. House, "testing extreme programming", Addison-Wesley Professional, pp: 153-154, 2003.

[9] M. Taromirad and R. Cefam, "Comprehensive evaluation framework for agile methodologies", Software Engineering Workshop, SEW'08, 32nd Annual IEEE, Kassandra, Greece, pp. 195-204, 2008.

[10] R. Ramsin, L. Claudia and R.F. Paige, "Process-centered review of object oriented software development methodologies, ACM Computing Surveys (CSUR), vol.40, pp. 3, 2008.

[11] R. Jeffries and T. Hendrickson, "Extreme programming installed", Addison-Wesley Professional, pp. 107-113, 2001.

[12] T. Dingsøyr and G. Hanssen, "Extending agile methods: postmortem reviews as extended feedback, International Workshop on Learning Softwarer Organizations", Springer Berlin Heidelberg, Berlin, Germany, pp. 4-12, 2002.

[13] K. O'Driscoll, "The agile data modelling & design thinking approach to information system requirements analysis, J. Decision Systems, vol. 25 (Supl), pp. 632-638, 2016.

[14] J.G. Taghi, Z. Hazura and Z.N. Mina, Agile transformation: a multi-dimensional process, IEEE, pp. 89-96, 2015.