Internatıonal Journal of Natural and Engineering Sciences 8 (3): 07-11, 2014
ISSN: 1307-1149, E-ISSN: 2146-0086, www.nobel.gen.tr

Customer Oriented Requirement Engineering By Using Scrum Methodology

Syeda Mariam GILLANI[1*] Salman QADRI[1] Muhammad FAHAD[2]
[1] Department of Computer Science and Information Technology, The Islamia University of Bahawalpur, Pakistan
[2] Department of Computer Science, Virtual University of Pakistan, Lahore, Pakistan

*Corresponding Author:
Email: mgillani39@gmail.com

Received: August 04, 2014
Accepted: September 13, 2014

Abstract

This research aims to find the influence of rigorous requirement engineering activities in helping the developers to change over from the traditional 'heavyweight' methodologies to the Agile methodology. Four software projects were chosen as case studies from different software houses and research was conducted during their project life cycle. A decision support system assisted the development team to analyze the requirements gathered in requirement elicitation phase and recommended suitable development methodology for fulfilling them. Scrum methodology was recommended for all four case studies. In the next stage the progress of these projects were evaluated to determine the effectiveness of this combination of Scrum and Requirement engineering. A general improvement in the performance of the development process was reported after applying Scrum methodology under the umbrella of customer oriented requirement engineering.

Key Words: SDLC, Agile, Scrum, Requirement Engineering

INTRODUCTION

Agile development methodologies have enabled developers across the globe to produce user centered products with flexibility and higher customer satisfaction. But there are serious reservations regarding the changeover as Agile methods generally lack the proper documentation and strict planning, which gives confidence to the developers during the project lifecycle. This research aims to investigate the effectiveness of Scrum which is the project management tool under the Agile umbrella using case studies from Pakistan's software industry. The study will make use of requirement engineering phases to evaluate the nature of a software product and provide decision support system to the developers and team leads to choose appropriate development methodology based on the requirement context. Keeping the developers reservations in account, the Scrum methodology used for this research will also incorporate well-planned and thoroughly documented requirement engineering practices which will provide further support to address the issue of effective capturing of requirements. With this approach the study will help in introducing Scrum methodology to the local software developers and also highlight its effectiveness.

The traditional software development methodologies have been around since beginning of the software development and have become de-facto. The waterfall model is incremental in nature as it divides the development process into five steps carried out one by one. Due to this the waterfall methodology lacks to incorporate the ever changing requirements of the customer[1]. The basic nature of a software product is to provide a quick solution to a specific problem in any kind of environment. The problem may evolve as it takes its course hence the software product should also adapt to its changing nature and ensure that it performs its tasks well in any environment. To overcome the deficiencies of the traditional methodologies a more flexible approach was introduced which focuses more on user's perspective of the software product.

Among many implementations of Agile manifesto the decision was made in the favor of Scrum methodology as it was more suitable for further empirical study due to its effectiveness in achieving high level of customer satisfaction in user centric software development[2],[3],[4]. Scrum framework is designed to incorporate stakeholder's requirements during the development process allowing the development team to choose and prioritize these requirements in sync with the development task. It defines roles, events, artifacts and phases to deliver working components of the software product in Sprints[5], [6].

Researchers have paired Scrum with other traditional and emergent methodologies to further exploit its potentials. In the work by Felker *et al.* Scrum has been integrated with user experience development methods which were reported to have resulted in end product achieving high usability and customer satisfaction[6]. Similarly Salinas *et al.* in their research map Scrum practices with CMMI Level 2 goals to test the compatibility of the two approaches. A need was felt for developing a generic framework for Agile organizations that can help them achieve all CMMI levels [7]. Memmel *et al.* propose to bridge the gap between Human Computer Interaction and the Software Engineering practices by using Agile principles to bring about rapid development keeping the cost and effort under check [8]. The research by Reichlmayr proposes the practical demonstration of Agile methodology to the software engineering students by incorporating the Agile techniques in SE course projects[9].

This research aims to provide confidence to software house management and team leads to leave their comfort zones and move towards innovations in development to achieve high standards with the help of in depth analysis of requirements using requirement engineering practices to determine which software methodology is best suited for the fulfillment of these requirements in efficient and timely manner. Through RE activities the customer's needs are identified, analyzed and documented for further processing. The basic aim of attaching an engineering process with the concept of requirements is to bridge the gap between the real-world problem and the development process[10]. This is achieved by five processes of the RE activity namely requirement elicitation, analysis and negotiation, documentation, verification and validation[11].

Paetsch *et al.* and Batool *et al.* discuss in detail the similarities and differences between Agile practices and requirement engineering activities. Scrum supports RE practices such as customers involvement, requirements prioritization and validation. Product backlog has special importance in context with requirement tasks and they enable continuous user involvement and change incorporation [12][13]. For successful project development it is of utmost importance to get the requirements fully understood and rationalized. Requirement Engineering and its processes meet these needs to bring stake holders and developers together. As indicated by Bose *et al.* in their research work, needs to be done in the direction of better requirement elicitation by taking view point of all stakeholders and not just one Product Owner [10]. The research presented by Williams *et al.,* Sillitti & Succi and Vlaanderen *et al.* gives results from three case studies to demonstrate the enhanced performance of Scrum when used in combination with software engineering and project management practices[14],[15],[16]. With the link between effective requirement engineering and Scrum practices established, the research moved forward towards comparison of software development methodologies to form the basis for the critical choice factors that contribute to the selection of a particular methodology [17],[19].

MATERIALS AND METHODS

Product Context and Requirement Analysis

For the purpose of developing a decision support system for developers and team leads that are new to the Agile Scrum methodology, existing work in this context was searched.

The research by Geambașu & Jianu points out key factors that can contribute to selection of methodology and evaluates the contribution of these factors to successful project delivery in RUP, RAD and XP approaches[18].

In a similar context the research by Faridani propose a decision tree approach for deciding which methodology is best suited for the project needs. The guidelines provided in this research have been gathered under the two categories [19].

First category is the "project characteristic analysis" in which project scope, stakeholder's roles and level of involvement are listed.

The second category is the "methodology decision analysis" which looks closely at project requirements and features to determine which course of action is most appropriate.

After analysis of both researches a decision was made about the critical factors that can significantly affect the success of the software development process.

Critical Choice Factors

CCF1 Project scope

This factor realizes the importance and impact of the project. During the requirement elicitation phase the stakeholders and customers are asked to define the scope and objective of the end product. A clearly defined problem scope and list of objectives can help the requirement engineer to gauge the nature of the project. System criticality and complexity are also to be evaluated as they play a key role in project success or failure.

CCF2 Stakeholder/Customer need for communication

During the requirement elicitation phase the requirement engineer will find the medium of communication that will be adopted during software development process. Will they require extensive documentation to convey the progress at each stage or will they prefer more casual face to face talk approach.

CCF3 Expected change in requirements

In the requirement elicitation and requirement analysis phase the requirement engineer will probe the customer and stakeholders to find if they are confident of all the requirements they have stated or further clarity is required as the project will shape up.

CCF4 Timeline constraints

Each project has to meet its deadline to ensure high level of customer satisfaction. An accurate estimation of deployment deadline can have positive affect on the development team.

CCF5 Nature of requirements

During the requirement analysis and specification phase the requirement engineer will determine the nature of requirements that have been initially provided. Requirements can be broadly classified as functional and nonfunctional. Some methodologies work better with one type of requirements only. The nature of requirements will also specify the completeness of the statements. Changeable and unclear requirements can be better handled in Agile methodologies where the requirements take shape as project builds itself.

CCF6 Size of development team

The last factor affecting the software development process is the size of the development team that will be employed to develop the project as traditional methodologies work better with larger teams and Agile methods works well for small to medium sized teams. Initial estimation of project cost can help decide the team size in a better way.

Decision Support System

The decision support system divides the task in three requirement engineering phases. In the requirement elicitation phase software project context is evaluated. In the requirement analysis and specification phase the requirements gathered in elicitation phase are tabulated in a simplified SRS document. In the verification and validation phase the team leads are conveyed about the methodology that is best suited for the project and decision is reached. The proposed framework for decision support system is shown in Figure 1.

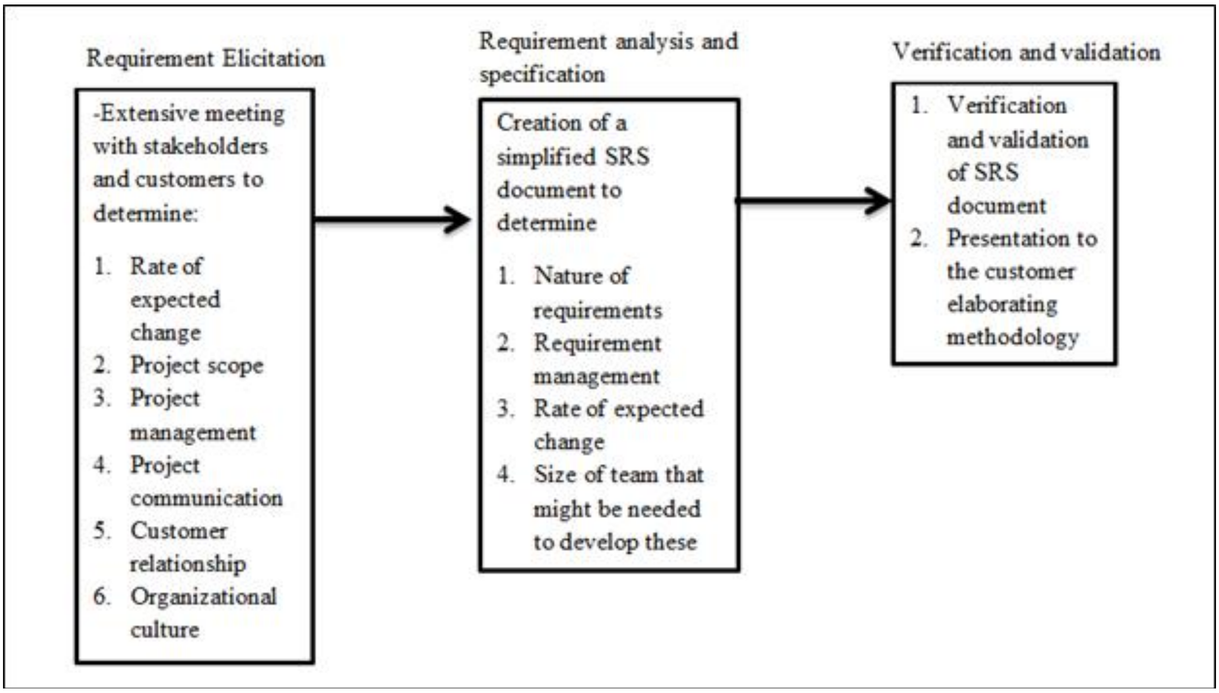
Requirement Elicitation
-Extensive meeting with stakeholders and customers to determine:
1. Rate of expected change
2. Project scope
3. Project management
4. Project communication
5. Customer relationship
6. Organizational culture
Requirement analysis and specification
Creation of a simplified SRS document to determine
1. Nature of requirements
2. Requirement management
3. Rate of expected change
4. Size of team that might be needed to develop these
Verification and validation
1. Verification and validation of SRS document
2. Presentation to the customer elaborating methodology

Figure 1. Decision Support System

RESULTS AND DISCUSSIONS

Case Studies

The case studies chosen for this evaluation framework belong to four different sectors of user centered software. These being web development, game development, mobile application development, windows desktop application development. User centered software are the trickiest when it comes to requirement elicitation and product satisfaction. All case studies belong to developers who have been using traditional waterfall model previously and find themselves contended with it. A number of meetings were conducted in sequence for both purposes of requirement elicitation and to make observations. Table 1 shows the summary of each case study with respect of each of the six critical factors and proposed methodology based on these findings.

Observations

After the requirement engineering process was completed the results of these studies were conveyed to the respective team leads. Appropriate training and orientation was provided to the developers who were unfamiliar with the Agile methodology. After a period of one month the progress of each project was checked observations were made based on three questions. The observations are summarized in Table 2.

Table 1. Summary of the Case Studies

	Case Study 1	**Case Study 2**	**Case Study 3**	**Case Study 4**
CCF1	High assurance of stability required	Less focus on stability and more on delivery	End product should be stable and predictable	High level of user satisfaction and medium stability required
CCF2	Customer involvement through frequent briefings	Customer involvement is minimal	Customer involvement through face to face meetings	Customer involvement in form of weekly meetings
CCF3	Fairly stable requirements	Unstable requirements	Fairly stable requirements	Stable requirements but subject to change and additions
CCF4	No timeline constraint	One month's limit	Three months	Working module to be deployed over a period of 4 months
CCF5	Functional requirements are dominant Clearly stated set of initial requirements	Nonfunctional requirements are dominant with functionality left to the developer Short story type requirements	Mostly non-functional requirements Clearly stated set of rules	Mix of functional and non-functional requirements Clearly stated set of requirements which can be changed or altered by customer
CCF6	Two to four	Two	Four	Four
RM	Scrum	Scrum	Scrum	Scrum

Table 2. Summary of Observations

	Case study 1	**Case study 2**	**Case study 3**	**Case study 4**
How difficult was the changeover	Fairly difficult	Very difficult	Not difficult	Fairly difficult
Level of customer satisfaction	Fairly satisfied	Unsatisfied	Satisfied	Satisfied
Is project timeline being followed	Yes	No	Yes	Yes

Comparison with past performance

A marked increase in overall speed of work was reported in three case studies (Case Study 1, 3 and 4), however it was also brought into notice that in the initial days the development task lagged behind. But once the new concepts sunk in the work, the speed also picked up. Case study 2 on other hand reported a decrease in performance. These comparisons are shown here in the form of figure 2, 3, 4 and 5.

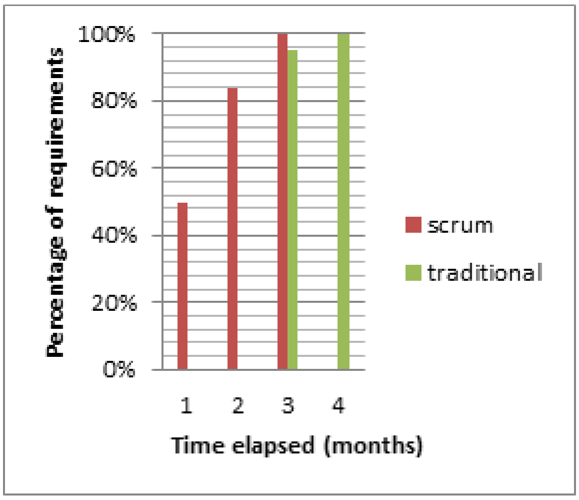

Figure 2. Performance Comparison for Case Study 1

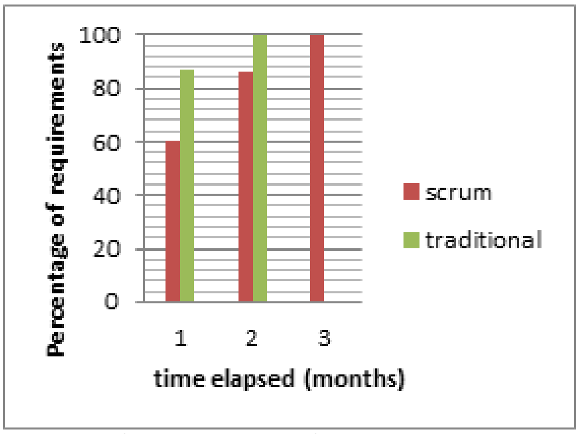

Figure 3. Performance comparison for Case Study 2

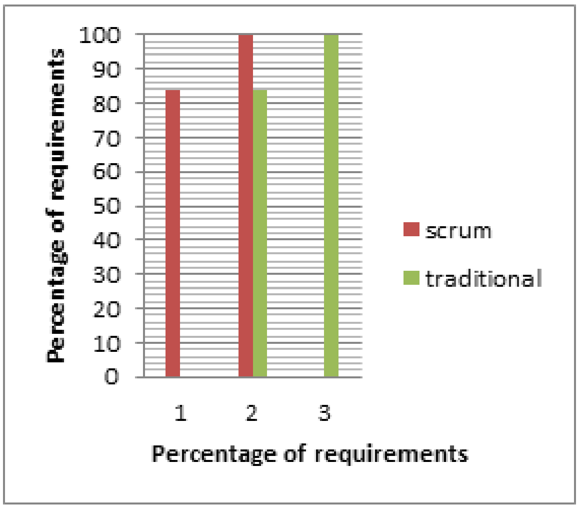

Figure 4. Performance Comparison for Case Study 3

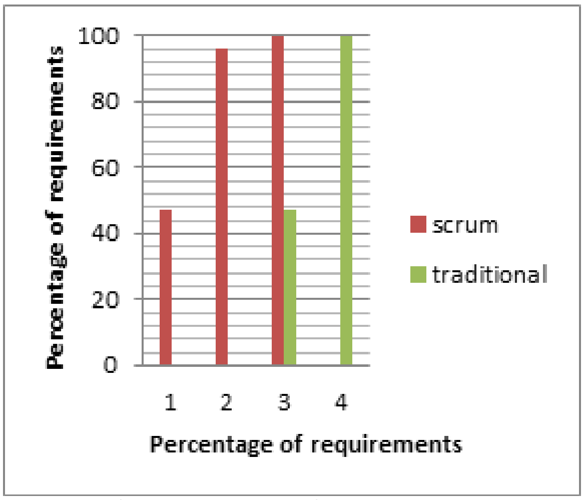

Figure 5. Performance Comparison for Case Study 4

CONCLUSION

The aim of this research was to introduce Agile methodologies in Pakistan's budding software industry which usually develops small to medium user centered soft wares. We have introduced the requirement engineering processes to precede the software development process in order to act as a decision support system. In the course of this research four case studies were employed. These case studies belonged to user centered software which require high customer satisfaction and are characterized by unstable requirements. The most appropriate methodology for such projects is the Agile scrum as indicated by the decision support system which analyzed the requirements gathered in the initial phase based on six critical choice factors. In the next step the performance of the software development has been monitored in each of the four case studies. The final results showed satisfactory implementation of Agile scrum in three case studies which increase in performance and user satisfaction.

Limitations and Future Work

The results generated by this research were subject to a general unwillingness to adopt to change from larger and more established software houses. All the case studies in this research are from small and medium sized companies hence these results cannot be applied to larger more sophisticated software. Most of the developers were inexperienced in development hence conversion from one methodology proved to be either too simple or difficult for them. For future work we recommend employing the projects of all types and sizes to better assess the working of our decision support systems. Working with large and well established software houses with experienced and skilled developers will also bring an interesting insight into the findings.

REFERENCES

[1] Farrel, A. 2007. Selecting a Software Development Methodology based on Organizational Characteristics. An Essay Submitted in Partial Fulfillment of the Requirements for the Degree of "Master of Science in Information Systems", Athabasca University. (Athabasca, Canada)

[2] Qasaimeh, M., H. Mehrfard and A. Hamou-Lhadj. December, 2008. Comparing Agile Software Processes Based on the Software Development Project Requirements. In Proc. of IEEE Int. Conference of Computational Intelligence for Modelling Control & Automation (CIMCA'08). (Vienna, Austria). pp. 49-54. ISBN: 978-0-7695-3515-2.

[3] Sletholt, M. T., J. Hannay, D. Pfahl, H. C. Benestad and H. P. Langtangen. May, 2011. A Literature Review of Agile Practices and their Effects in Scientific Software Development. In Proc. of the 4th Int. Workshop on Software Engineering for Computational Science and Engineering (SECSE' 11). (New York, USA). pp. 1-9.ISBN: 978-1-4503-0598-3.

[4] Waguespack, L. J. and W. T. Schiano. January, 2012. SCRUM Project Architecture and Thriving Systems Theory. In Proc. of 45th Hawaii Int. Conference on System Science (HICSS' 12). (Maui, Hawaii). pp. 4943-4951. ISBN: 978-0-7695-4525-7.

[5] Sutherland, J. and K. Schwaber. 2010. The Scrum Papers: Nut, Bolts, and Origins of an Agile Framework. SCRUM Training Institute.

[6] Felker, C., R. Slamova and J. Davis. February, 2012. Integrating UX with Scrum in an Undergraduate Software Development Project. In Proc. of the 43rd ACM Technical Symposium on Computer Science Education (SIGCSE'12). (New York, USA). pp. 301-306. ISBN: 978-1-4503-1098-7.

[7] Salinas, C. J., M. J. Escalona and M. Mejías. December, 2012. A Scrum-based Approach to CMMI Maturity Level 2 in Web Development Environments. In Proc. of the 14th Int. Conference on Information Integration and Web-based Applications & Services (IIWAS'12). (New York, USA). pp. 282-285.ISBN: 978-1-4503-1306-3.

[8] Memmel, T., F. Gundelsweiler and H. Reiterer. September, 2007. Agile Human-Centered Software Engineering. In Proc. of the 21st British HCI Group Annual Conference on People and Computers: HCI... but not as we know it. (Swinton, UK). 1: 167-175. ISBN: 978-1-902505-94-7.

[9] Reichlmayr, T. November, 2003. The Agile Approach in an Undergraduate Software Engineering Course Project. In Proc. of 33rd Annual Conf. on ASEE/IEEE Frontiers in Education (FIE'03). (Westminster, CO, USA). 3: 13-18. ISSN: 0190-5848.

[10] Bose, S., M. Kurhekar and J. Ghoshal. 2008. Agile Methodology in Requirements Engineering. SETLabs Briefings Online.

[11] Mahmud, I. and V. Veneziano. December, 2011. Mind-Mapping: An Effective Technique to Facilitate Requirements Engineering in Agile Software Development. In Proc. of 14th Int. Conference on Computer and Information Technology. (ICCIT'11). (Dhaka, Bangladesh). pp. 157-162). ISBN: 978-1-61284-907-2.

[12] Paetsch, F., A. Eberlein and F. Maurer. June, 2003. Requirements Engineering and Agile Software Development. In Proc. of 12th Int. Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises. (WET ICE'03). pp. 308-313. ISSN: 1080-1383.

[13] Batool, A., Y. H. Motla, B. Hamid, S. Asghar, M. Riaz, M. Mukhtar and M. Ahmed. January, 2013. Comparative Study of Traditional Requirement Engineering and Agile Requirement Rngineering. In Proc. of 15th IEEE Int. Conference on Advanced Communication Technology (ICACT'13). (PyeongChang, South Korea). pp. 1006-1014. ISSN: 1738-9445.

[14] Williams, L., G. Brown, A. Meltzer and N. Nagappan. September, 2011. Scrum + Engineering Practices: Experiences of Three Microsoft Teams. In Proc. of Int. Symposium on Empirical Software Engineering and Measurement (ESEM'11). (Banff, AB). pp. 463-471. ISSN: 1938-6451.

[15] Sillitti, A. and G. Succi. 2005. Requirements Engineering for Agile Methods. In Engineering and Managing Software Requirements. The Springer Berlin Heidelberg Companies, Inc. pp. 309-326. ISBN: 978-3-540-28244-0.

[16] Vlaanderen, K., S. Jansen, S. Brinkkemper and E. Jaspers. 2011.The Agile Requirements Refinery: Applying SCRUM Principles to Software Product Management. Int. Journal of Information and software technology. (MA, USA). 53(1): 58-70. ISSN: 0950-5849.

[17] Archer, N. P. and F. Ghasemzadeh. 1999. An Integrated Framework for Project Portfolio Selection. Int. Journal of Project Management. 17(4): 207-216. ISSN: 0263-7863.

[18] Geambașu, C. V., I. Jianu, I. Jianu and A. Gavrilă. 2011. Influence Factors for the Choice of a Software Development Methodology. Int. Journal of Accounting & Management Information Systems. 10(4): 479-494.

[19] Faridani, H. 2011. A Guide to Selecting Software Development Methodologies. http://www.gtislig.org/HamidFaridani_GuideToSelectingSWMethodologies_SOC_PDD_ 20110305.pdf