Proceedings of the Pakistan Academy of Sciences 50 (4): 263–270 (2013)

ISSN: 0377 - 2969 (print), 2306 - 1448 (online)

Pakistan Academy of Sciences

Research Article

A Scrum Framework for Requirement Engineering Practices

Asma Batool, Yaser Hafeez*, Sohail Asghar , Muhammad A. Abbas and Muhammad S. Hassan

University Institute of Information Technology,
PMAS Arid Agriculture University, Rawalpindi, Pakistan

Abstract: Nowadays, software development industry has departed from traditional development to Agile software development. In this paper we have proposed a conceptual framework. The main emphasis in this framework is to make the requirement engineering process more effective and add to the suppleness of it. The research method we have used in this research is case studies, expert reviews from the software industry, existing published reports and articles. The proposed framework has resulted in improving the previous related work. This framework provides guidelines and an easy to adapt approach for the software development teams.

Keywords: Requirement engineering, traditional requirement engineering, agile, scrum

1. INTRODUCTION

Requirements are the desired specification functions that illustrate what will be the software product [1]. Traditional requirement engineering is based upon some key factors, and these steps include identification, analysis, documentation, validity and management of requirements for the system to be developed [2]. In requirement elicitation activity, an analyst gathers requirements along with any other information from the stakeholders, and organizes them to develop the software requirement specification document [3]. Raw requirements are collected from the stakeholders. Various techniques are used to collect the requirements, like system prototyping, model-driven techniques and cognitive techniques [2]. Requirements prioritization and traceability issues are resolved in the requirement analysis. Requirements are organized in some consistent and accessible way in requirement specification. After documentation of the requirements, there is a need to verify the requirements in the requirement verification phase. Requirements are verified through writing test cases and checklists.

Changes in requirements are tackled by requirement management. Each anticipated change needs reviewing and estimation before it is accepted by the change control board [4]. Agile methods belong to a family of software development processes. It became popular during the last few years [5]. Agile framework emphasizes on two main principles, the first one is active communication between team and users whereas the second principle deals with iterative, incremental, continuous delivery and flexible approach to development. Agile RE is an approach different from the traditional RE because Agile RE focuses on iterative approach as requirements engineering activities like elicitation; negotiation, documentation are carried out in small development cycles [6]. Both the approaches have similar goals, the main difference is that the traditional RE depends more on documentation but Agile RE has reduced concentration on documentation and keeps focus on face to face communication with customers. RE involves the customers only in early stages of the development process but Agile RE involves the customers throughout the process [7]. Agile requirement engineering decreases the communication gaps

Received, April 2012; Accepted, September 2013
*Corresponding author: Yaser Hafeez; Email: yasir@uaar.edu.pk

between customers and developers. Consequently, it can easily manage the change in requirements at any stage of the software development life cycle. Requirement gathered at early stage creates problems during later stages because requirements evolve throughout the development process due to customer needs. So there is a need for an iterative process that may communicate with the customer at any stage. Continuous change in process is a salient feature of agile as the software being developed consists of small releases and change is essential to agile methods [8]. Agile processes replace the other old processes to speed up the development process and cope with the changing requirements of different organizations [9]. Different methods of agile are very helpful in the software development that focuses on abilities, communication, roles and relationships between customers and developers [10]. A scrum is a well-known agile method that focuses on the project management [11]. It has a unique framework that consists of best practices which have been successfully applied in software development process [12]. Scrum is quite favorable agile approach when applied to small and medium projects, rather than a full process or methodology; its framework does not provide the detailed description of the project because the team involved in the project better knows how to tackle every problem [10]. A scrum is a development method in which software is delivered in increments called sprints [13]. It starts with the vision and rises with core requirements and set their priorities in the form of the product backlog. The team plans the requirements in the form of sprint backlog. Each development cycle completes in sprints that involves the designer, developer, tester complete functionality to complete an increment in functionality. The team demonstrates the completed functionalities to stakeholders to get their trust in the sprint review meeting and sprint retrospective. The scrum process consists of three artifacts i.e. Product backlogs, sprint backlogs and burn-down charts. Backlogs are about the customer requirements and daily burn down charts demonstrates the collective work left behind. [14]. It became popular because its an iterative, flexible and reliable process in which all development phases are completed in shorter sprints and change is easy to manage [15].

2. RELATED WORK

Many researchers have presented the difference between traditional requirement engineering and agile requirement engineering [16]. Requirement engineering is a multidisciplinary activity that consists of multiple techniques and tools which is a very important phase in the software development life cycle [2].

The traditional RE process is a very time consuming process so modern software industry demands an iterative and speedy process [5]. Lucy et al has also mentioned some recommendations to solve the problems related to documentation and critical requirements because functional and non-functional requirements are difficult to handle in the requirement document [6].

Paetsch et al [7] have listed some important features of RE process and have critically examined their applicability in agile methods [8]. Michael Coram et al evaluated the effects of agile methods in project management, people involved in the project and their applicability. Agile Methods recommended a realistic approach to overcome the changes. Agile methods are helpful when applied under the right situation [17].

Sen and Hemachandran [3] presented an Agent Based Goal Elicitation (ATABGE) system that consists of agile approaches. The system involves the agents and stakeholders for decomposing and refining the high level goals into low level goals [3]. Liu Jun et al presents the comparison of Agile RE and Traditional RE and addresses the issues related to the Traditional RE and concluded that Agile RE practices are best for the modest-sized developments [18].

In Veerapaneni Esther Jyothi et al proposed a collaborative and innovative framework for the solution of difficulties faced by developers in agile software development [19]. In another study, Emam Hossain et al. Identifies the risks related with the scrum used at GSD by conducting a systematic literature review of primary papers. It captures some kind of risks and presents their solutions in the form of a conceptual framework [20]. A framework is introduced by Sven Overhage et al which is based on extended Technology Acceptance Model (TAM). It presents the developer's approval of scrum and

sets up guidelines to understand the methodologies [21].

Rohit Ramanujam et al describes the study about the multi vendor challenging situations and propose a collaborative framework for agile scrum development. Subject Matter Expert (SME) plays a vital role in producing successful and collaborative environment. Each team must have its own scrum master who is responsible for removing the barriers [22].

Shvetha Soundararajan presents the Agile Requirement Generation Model (Agile RGM) that is structured to support the agile requirement engineering process. This model helps to choose the different existing agile methods provides a successful agile requirement engineering process. Agile RGM consists of five phases, education phase, feature development, story development phase, task development phase and development phase. These five phases are combined together to support a vigorous agile requirement engineering process [23]. The Standish Group surveyed IT executive managers for their opinions and found out that why projects succeeded. The project success is based upon three major factors which are user involvement, managerial support and a clear declaration of requirements [24].

3. MATERIALS AND METHODS

As many organizations move to agile development and switching from traditional requirement engineering to agile requirement engineering because of issues and problems faced in the traditional requirement engineering. In large organization's requirements are evolving throughout

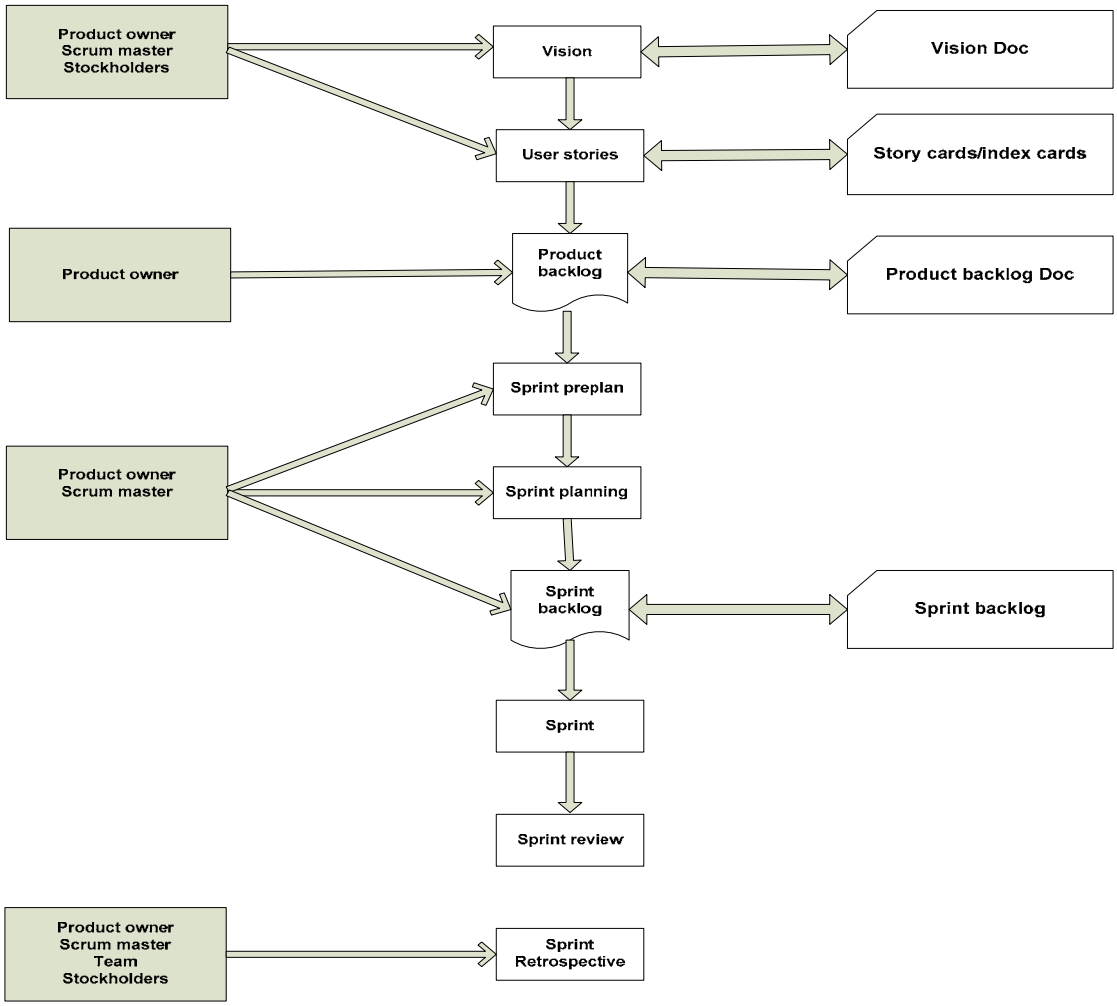

Fig. 1. Scrum process flow with respect to roles, activities and artifacts.

the development process and due to the iterative nature of agile development. In this context, we have decided to present a framework that combines the requirement engineering and scrum practices. A scrum is a very useful and efficient agile approach that helps to make the requirement engineering process flexible. The proposed framework makes certain the efficiency and flexibility of the requirement engineering process, software development removes the issues that are faced in the traditional requirement engineering process due to changing requirements, lack of communication and feedback of customers. Our proposed research method consists of two parts. First part in Fig. 1 describes the flow of scrum activities and the second part in Fig. 2 presents the proposed requirement engineering framework with the scrum practice.

3.1 Scrum Process Flow with Respect to Roles, Activities and Artifacts

This process hierarchy in Fig. 1 describes the flow of scrum activities in the form of its Roles, Artifacts and Activities. Gray boxes on the left side in Fig. 1 are representing the roles, white boxes in the middle are showing activities and white boxes on the right side of figure are showing the artifacts of the scrum approach. The detail description of this process is given below:

The product owner is responsible to define the desired outcomes of the project and prioritize the features of the product. Regulate feature and priority in every iteration, as needed. Product owner considers the Return On Investment (ROI) and responsible for accepting or reject work results.

Scrum master represents the management of the team and ensures that team is efficient and creative. Resolves the problems and protect the team from external intervention.

Cross-functional team consists of 5-9 people including programmers, testers, and designers. The team is responsible for failure or success of the product.

Product Backlog Contains the requirements that are prioritized by the product owner. Requirements are prioritized at the start of each sprint.

Sprint Backlog consists of the requirements that are ready to develop. Each Item on the sprint backlog includes description and estimate.

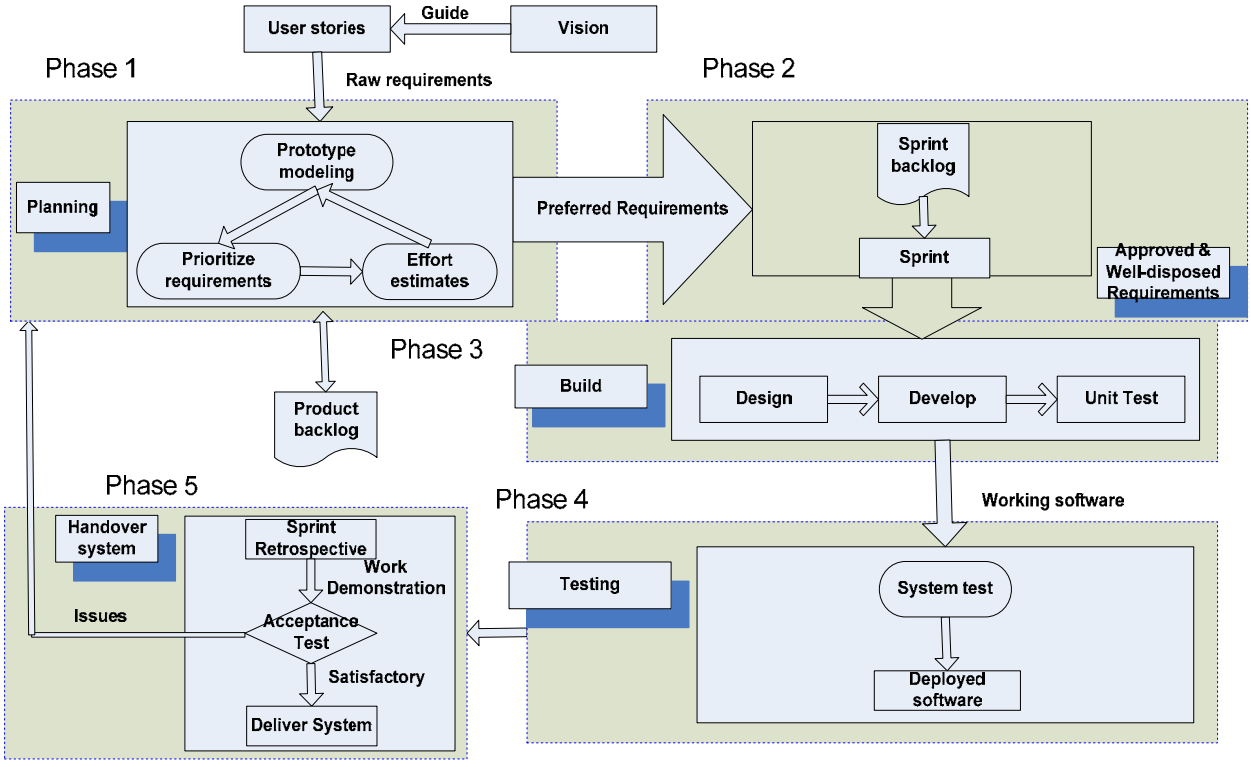

Fig. 2. Scrum requirement engineering process.

User Story is the initial requirements in the form of story cards and index cards.

In Sprint Planning selects the items from the product backlog and creates the sprint backlog.

Sprint Review presents what has been completed during the sprint in the form of demos of new features.

Sprint Retrospective involves all the members i.e. scrum master, product owner, team and stockholders. It has occurred at the end of every Sprint after review meeting and discusses the experiences and problems faced during development.

3.2 Proposed Scrum framework

The overall structure of our proposed framework consists of 5 phases as shown in figure 2. Basically these steps show the flow of our work. Each of these steps, perform a valid functionality in the proposed framework. Each and every step is detailed below:

Vision: In the vision we describe what we want to do with the system.

User stories: User stories are the description of requirements expressed by customers that consists of sufficient information needed by the developers for the effort estimation and implementation. Product owners elicit the requirements by consulting with the stakeholders in the form of user stories. The requirements are taken on story cards. All requirements are not gathered at this stage; they are flexible and can be changed at any given stage. After gathering the requirements in the form of user stories these raw requirements enter into the first step which is planning.

Phase 1: Planning

Planning consists of prototype modelling, requirement prioritization and effort estimation. Prototype modelling is not a scrum activity; we integrate it into scrum to make the requirements clearer. It helps to understand the requirements in the form of user stories. With the agile point of view Product owner prioritizes the requirements and then estimates the efforts of these prioritized requirements. The input of this phase is the raw requirements and output is the preferred requirements. The product owner engages the stakeholders in every step in the planning phase.

Phase 2: Approved & well-disposed Requirements

After the identification of preferred requirements, these are collected in sprint backlogs. In this phase product owner and scrum masters select the team for constructing the selected requirements. Sprint backlog requirements are ready to implement in running sprints.

Phase 3: Build

In this phase, approved requirements are designed, coded and tested by a cross functional team. The approved requirements are designed by preparing their class diagram to make them more specific and understandable for programmers. Then these designed requirements are coded by choosing a language in which programmer can easily code them. After implementation requirements are tested. Tester applies unit tests to check the functionality and reliability of the system. A daily meeting of 15 minutes duration is arranged to check the efficiency of the team members, only product owner, scrum master and team members can discuss the status of work. The output of this phase is the working software.

Phase 4: Testing

In this phase, the team itself applies the system test to finally check the working software. In the system test team assure that the system is fulfilling the specified requirements and checks its overall performance by deploying software.

Phase 5: Handover System

The main objective of this phase is to verify the product that meets the specified requirements and assesses the response of the customers. Product owner, scrum master, team and stakeholders gather to test the deployed software. In sprint retrospective, the team demonstrates the performance of working software. Acceptance testing is done by the product owner and stakeholders. If the product meets its specified requirements and raises the level of satisfaction then the product will deliver to the stakeholders otherwise pointed issues are resolved in the planning phase in the next cycle.

4. RESULTS AND DISCUSSION

We shall consider a case study of Management Information System (MIS) of a university where we will check the effectiveness of our model by the expert opinions of the software industry.

A public sector university X has been planning to launch a Management Information System (MIS). It has recently computerized its examination process based on some off-the shelf software. The university also has a computerized library management system. The university management team decided to develop a single large MIS because the University has loaded itself with too many independent information systems.

The university has assigned the task of developing an MIS for the University to industry. This MIS system consists of 3 systems:

The Student Information Management System (SIMS) is a system for managing the student information. It contains the record about enrolled students, their personal information, respective courses, classes and teacher schedules, absences and attendance, fees and fines, health conditions, boarding house details, communications with students and their parents.

Examination Management System (EMS) contains the information about student exams and their results. Its features include examination schedules, no of students appearing in exams, no of students not appearing in exams because of valid reasons, available classrooms, results, student grades and scores.

Library Management System (LMS) consists of a graphical user interface. All users can login and register though this interface. It contains the information about registered students, books regarding their particular subjects, issued books, returned books, and generate a fine list. The librarian can login, register, add category, add/ remove book, and can do add/ remove issue and return books information through the system. Our evaluation method is based on eight requirement engineering success factors. These are efficiency, flexibility, visibility, cost effective, concise documentation, adequate and constant communication, project status demonstration, responsive to change [25]. We select these factors from previous research and by conducting surveys of the software industry. These factors are very important for the effective requirement engineering process. We get the response from ten industry experts who are involved in this project.

Some of the existing scrum frameworks improved in the software development but our proposed framework has produced is the requirement engineering process. Evaluation of the proposed framework is shown in table 1; it contains the expert opinions about existing Traditional RE, proposed Scrum RE and their difference on the basis of 8 requirement engineering success factors. The results of this survey are shown in Fig 3. It is clear that the proposed Scrum RE framework performs better than existing Traditional RE process. Difference in claiming benefits of existing and proposed approach shows a reasonable improvement in

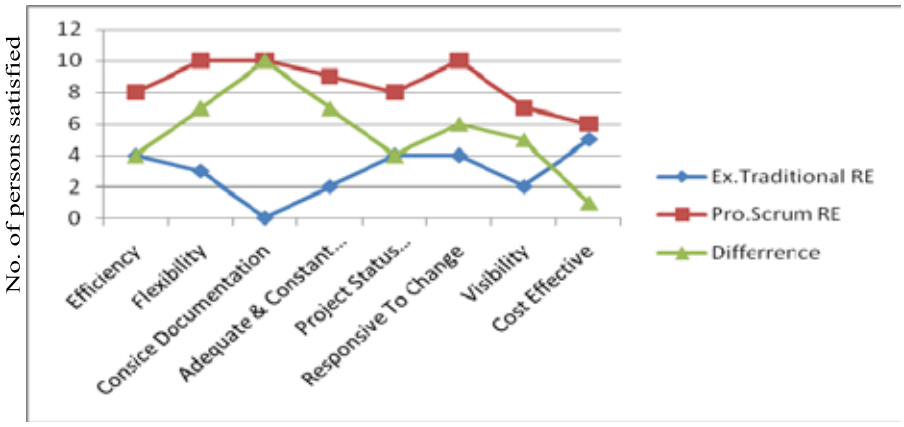

Fig. 3. Graphical representation of cumulative results.

Table 1. Survey expert opinions.

Apparent Benefits		Efficient	Flexible	Concise Documentation	Adequate & Constant Communication	Project Status Demonstration	Responsive to change	Visibility	Cost Effective
Ex. Traditional RE	Yes	4	3	0	2	4	4	2	5
	No	6	7	10	8	6	6	8	5
Pro. Scrum RE	Yes	8	10	10	9	8	10	7	6
	No	2	0	0	1	2	0	3	4
Difference		4	7	10	7	4	6	5	1

development Lifecycle. These mentioned success factors are necessary for the effective requirement engineering process. From the expert opinions it is clear that scrum RE is the most effective method of the requirement engineering process.

5. CONCLUSIONS

Requirement engineering is an important and difficult phase in a sense that different stakeholders have different attitude and expectations about the system and problem occurs when all requirements are elicited at an early stage because at very initial stage customer did not have an idea about the system so changes are must throughout the process. Our main emphasis behind this research is to build and evaluate a framework that can help the requirement engineers in the requirement engineering process. This framework will minimize the complexities and barriers faced during the traditional requirement engineering process. We have evaluated the proposed framework through the expert judgment opinions on the basis of eight requirement engineering success factors. This framework has established the standard guidelines for the effective requirement engineering process. In future work, we will evaluate the value and validation of our framework through more case studies to make it more effective and applicable in any kind of project.

6. REFERENCES

1. Westfall, L. Software Requirements Engineering: What, Why, Who, When, and How? *Software Quality Professional* 7(4): 14 (2005).
2. Nuseibeh, B. & S. Easterbrook. *Requirements Engineering: A Roadmap. Department of Computing*. Imperial College, London (2000).
3. Sen, A.M. & K. Hemachandran. Elicitation of Goals in Requirements Engineering using Agile Methods. *Computer Software and Applications Conference Workshops(COMPSACW)*, 34th Annual IEEE on 19-23 July 2010, Seoul, Koera, p. 263-268 (2010).
4. Wiegers, K.E. *Requirements Management: The Proven Way to Accelerate Development*. Principal Consultant at Process Impact and Sandra McKinsey, Senior Product Marketing Manager at Serena Software, Review, www.serena.com (2005).
5. Agile Alliance URL http://www.agilealliance.com/, last accessed 11 Dec, 2011.
6. Ramesh, B., L. Cao & R. Baskerville. Agile requirements engineering practices and challenges: an empirical study. *Information Systems Journal* 2010 - Wiley Online Library 20(5) 449–480 (17th Nov, 2007). DOI: 10.1111/j.1365-2575.2007.00259.x
7. Paetsch, F., A. Eberlein & F. Maurer. Requirements Engineering and Agile Software Development. *Enabling Technologies: Infrastructure for Collaborative Enterprises,* WET ICE 2003. Proceedings. 12th IEEE International Workshops on 9-11 June, 2003, p. 308-313 (2003).
8. Lucia, A.D. and A. Qusef. Requirements Engineering in Agile Software Development. *Journal of Emerging Technologies in Web Intelligence* 2(3): 212-220 (2010).
9. Savolainen, J., J. Kuusela & A. Vilavaara. Transition to Agile Development Rediscovery of Important Requirements Engineering Practices. *Requirements Engineering Conference (RE)*, 18th IEEE International Workshops, Sept. 27 2010-Oct. 1 2010, Sydney, NSW, Australia, p. 289-294 (2010).
10. Kavitha, C.R., & S.M. Thomas. Requirement gathering for small projects using agile methods. IJCA Special Issue on "*Computational Science - New Dimensions & Perspectives*" NCCSE, p. 122-128 (2011).
11. Abrahamsson, P., O. Salo, J. Ronkainen & J. Warsta. *Agile Software Development Methods: Review and Analysis*. VTT Electronics, VTT Publications, p. 107-478 (2002).
12. Schwaber, K. & J. Sutherland. Scrum. Internet:http://www.scrum.org/storage/scrumguides/Scrum%20Guide.pdf, Mar. 30 2011.

13. Sutherland, J., & K. Schwaber. *The Scrum Papers: Nuts, Bolts, and Origin of an Agile Process*. http://scrumtraininginstitute.com/home/stream_download/scrumpa pers, last accessed 11 Dec 2011.
14. Hossain, E., M.A. Babar, H. Paik & J. Verner. *Risk Identification and Mitigation Processes for Using Scrum in Global Software Development: A Conceptual Framework*. 16th Asia-Pacific Software Engineering Conference, 1-3 Dec. 2009, Penang, p. 457-464 (2009).
15. Patel, C. & M. Ramachandran. Story card maturity model (smm): A process improvement framework for agile requirements engineering practices. *Journal of Software* 4(5): 422-435 (2009).
16. Lucia, A.D., & A. Qusef. Requirements Engineering in Agile Software Development. *Journal of Emerging Technologies in Web Intelligence* 2(3): 212-220 (2010).
17. Coram, M. & S. Bohner. The Impact of Agile Methods on Software Project Management. Engineering of Computer-Based System. ECBS '05. *12th IEEE International Conference and Workshops*, 4-7 April 2005, p. 363-370 (2005).
18. Jun, L., W. Qiuzhen & G. Lin. Application of agile requirement engineering in modest-sized information systems development requirements engineering and agile software development. In: *Proceedings Second WRI World Congress on Software Engineering. Zhejiang University, City College, Hangzhou, China* (2010).
19. Jyothi, V.E. & K.N. Rao. Effective Implementation of Agile Practices: Ingenious and Organized Theoretical Framework. *International Journal of Advanced Computer Science and Applications* 2 (3): 41-48 (2011).
20. Hossain, E., M.A. Babar, & H. Paik. Using Scrum in Global Software Development: A Systematic Literature Review. Global Software Engineering, 2009. ICGSE 2009. Fourth IEEE International Conference on 13-16 July 2009, Limerick, p. 175-184 (2009).
21. Overhage, S., S. Schlauderer, & D. Birkmeier. What Makes IT Personnel Adopt Scrum? A Framework of Drivers and Inhibitors to Developer Acceptance. System Sciences (HICSS), In: *Proceedings 44th Hawaii International Conference,* 4-7 Jan. 2011, Kauai, HI, USA, p. 1-10 (2011).
22. Ramanujam, R. & I. Lee. Collaborative and Competitive Strategies for Agile Scrum Development. Network computing and advanced information management (NCM). In: *7th International Conference*, 21-23 June 2011, Gyeongju, 123-127 (2011).
23. Soundararajan, S. *Agile Requirements Generation Model: A Soft-structured Approach to Agile Requirements Engineering*. Master's thesis, Dept. of Computer Science, Virginia Tech, http://scholar.lib.vt.edu/theses/available/etd- 08132008-193105/ (2008).