ENHANCEMENT IN AGILE METHODOLOGIES USING REQUIREMENT ENGINEERING PRACTICES

[1*]Javed Ali Khan, [2]Izaz Ur Rehman, [3]Yawar Hayat Khan, [4]Shahdab Ali Shah , [5]Wasiat Khan
[1,2,3,5].Department of Software Engineering, University of Science and Technology Bannu, Pakistan
[4]Government Polytechnic Institute Karak,Karak Pakistan
*Corresponding E-mail: engr_javed501@yahoo.com

ABSTRACT: *Requirements gathering, Specification, documentation, and reviewing software requirements are important phases in every software development model. These tasks are also performed in Agile development latest software development methodology--. Now a days agile development has become the first option of every software development organization. Agile development techniques are famous for delivering software in time and within budget. This paper recommends requirement engineering practices, introduced in agile methodology to make the agile development process more effective and efficient. This paper also suggests some improvements in larger projects when they are developed through agile methodology. Adding requirements engineering techniques to agile methodology will further enhance the performance of agile development. Comparison of agile and traditional software development techniques is also presented in the paper. This paper also discusses traceability problems in agile development and solution is suggested to the problem.*

Keywords: Agile development, Traditional software development, Scrum, Requirement Engineering.

INTRODUCTION

Software Agile Methodologies have gotten a huge reputation in the last few years for constructing software products [1, 2, 3]. Agile methodology aim is to deliver frequent release of software of high quality and in accordance to the customers needs [4]. Agile methodology enables development teams to deliver projects within budget and time.

Agile development techniques produce better results than traditional development techniques. Now a days agile methodologies are dominant over traditional software methodologies to deliver quality software. Agile methodologies are based on incremental and iterative approaches, for which requirements are gathered through joint application development (JAD) and group interview techniques [5]. Together all these things help in developing a software product which is flexible to accept future changes. In agile methodologies there is limited documentation, heavy customer's involvement (sometimes customer become a part of the development team) and importance is given to people as compared to processes. Agile methodology is composed of different practices which adds same values and fundamental characteristics. The Agile Manifesto lays emphasis on "stakeholders and communications" over methodologies and tools, deliverable software over detail documentation, stakeholder's cooperation over contract concession and fixing quick changes over following a detail plan" [6].

Requirements do not show the implementation instead this technique elaborates what is to be made [6]. For requirements engineering process, many techniques are available to gather requirements and make sure that the requirements gathered are complete, correct, consistent and relevant. Main goal of RE techniques is to ensure that requirments are accurate beofore developing the actual software in order to thwart expansive rework.

This Goal may be achieved by following the two main assumptions:

1. Errors found in later stages of software development will cost more in order to fix them [3].
2. Make an established set of requirements (in form of SRS) before the actual development (design and implementation) of software.

Requirements engineering helps in establishing the functionalities that the users expect from the software, the constraints under which software will operate and developing the software in executable form. Aim of the requirements engineering process is to create a software document which works as a basis for knowledge sharing while agile methodologies emphasis on front to front communication between development team and stakeholders; even sometimes customer become part of development team in agile methodologies. Several authors have discussed the relationship between agile methodologies and requirements engineering [7,8,9,10,11] in these papers. They have discussed some useful RE methodologies in AD methods. They have carried out a comparison of RE practices between conventional software development techniques and agile techniques.

This article put light on the problem that how customer's requirements can be gathered and evaluated in agile development methodologies. It is therefore suggested in this paper that how requirements engineering practices can be mixed with agile methodologies in order to solve the problems in this domain.

AGILE DEVELOPMENT

The Aim of agile methodologies is to make the organization agile, but what actually agile means? According to Jim, Highsmith Agile means that organization must be able of "faster delivery , quick change "[12]. Different organizations follow different agile practices also their emphasis is different, but each agile methodology has same agile manifesto.

- Agile development focuses on peoples rather than processes.
- There are frequent contacts between the development team and customers to discuss the project matters.

- In Agile development there is more focus on coding, which is deliverable to customer rather than documentation.
- Agile development enables software to accommodate the requirement's change quickly.
- Planning in agile development is shortened while focus is laid on what customer wants.
- In agile good practices are chosen in the best interest of the development team and customer to develop software.
- In agile development software is developed through Incremental approach. The duration of each iteration (requirement, design, implementation and testing) is limited to 2 or 3 weeks. Shown in figure 1.
- After the completion of each iteration, the software/product is released to users. The users test that release to give frequent feedback to the development team. This approach provides certain advantages; as
- Due to early release of software increment, customer can use that software increment as working software ,so that if any errors exist in that software ,will be identified in earliest time to overcome the effect on the project schedule.
- Customer is involved through out the development process due to which he feels control over the product as progress is visible to him.
- Agile team is able to deliver a working increment on time which helps in building the confidence between agile team and customer.
- Before working on agile iteration, software requirements are identified with the mutual effort of customer and agile team and priorities are assigned to the requirements in order to be included in the iteration.
- Customer is highly involved in a development process in order to identify problems in a software iteration in earlier stage. Sometimes customer is part of an of the development team and is present with development team through out development.

Due to agile software methodologies it has become possible to deliver high quality software products within estimated time and cost under continuously and quickly changing requirements. Literature shows that agile methodologies have a huge reputation in software development and IT industries. Fig.2 shows that around about 69% of software companies are using one or more agile methods for software development and organizational development [11].

Companies where requirements are not clear, prefer to use agile development practices to solve this problem. Iterative and incremental techniques are used for this purpose and all members of the agile development team, and customer are active contributors and each member is encouraged to give input in the development process. Agile development methodologies are very strong in handling software change request even in the later phases of software development activities. Agile development helps in establishing a quality product with limited team members and resources due to multiple agile iterations. Agile development has less documentation due to which it is difficult for a new team member to continue with the development team. Agile methodologies are criticized for less documentation.

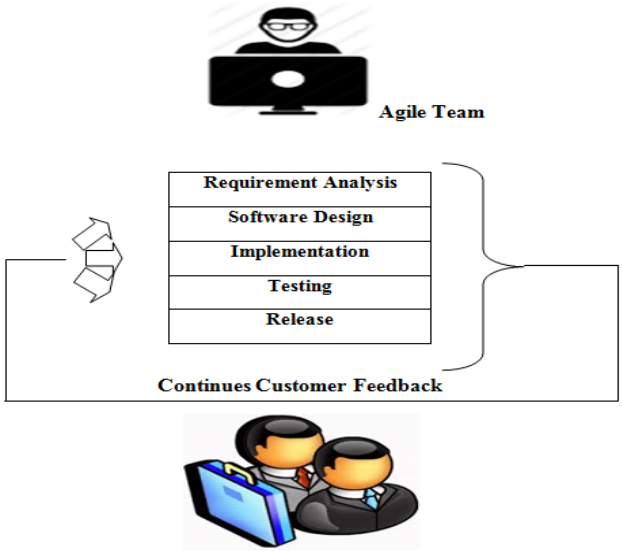

Fig. 1. Agile Life Cycle

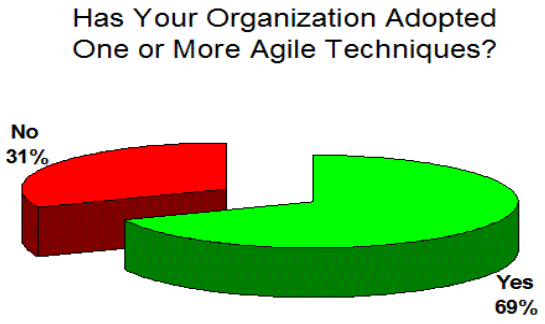

Fig. 2. Agile Implementation[11]

Agile development lacks in basic work flow. Agile software development lacks in Agile methodology performancein case of big projects.In a big project with larger teams,many iterations are bulit and it cuases in difficulty for the developers to remember which iteration is being constructed currently and how many iterations are released.It has also a problem to locate the iteration for the changes, demanded by customers. Besides this agile team requires highly qualified and talented members which are not easily available in market. The less documentation will cause problem in implementation which results in failure. By implementing the conventional development methodologies in agile development the agile performance will be improved. It can be seen from the literature that conventional development methods are non iterative mostly which are vulnerable to later design issues.Agile development has an incremental nature which can easily solve later design problems and encourages requirement change. Below we have discussed some common methodologies of agile development keeping in mind the requirement engineering perspectives.

Extreme Programming

XP is composed of simplicity, communication, customer feedback and guts [13]. It is rather new approach of software development based on agile methodology. In XP requirements gathered from the users are noted down on story boards. A user story is an explanation of a software feature which adds functional value to the customer. Before implementation every story is discussed in details in an open ended way. A developer needs sufficient details in order to be able to calculate total effort necessary to implement the story. Customer and development team then prioritize stories based on these estimates and time available for the next iteration. In XP, Pair programming concept is used in which two programmers work on a single task using a single computer. Where one programmer works on the development of a storyboard the other programmer looks for the code to figure out any error at run time. The customer is present with the development team through out the development process in order to make the development process smooth and error free. In XP, it is necessary that each team will work on story development not more than two weeks.[13]. XP does not clearly define requirements engineering techniques in details. It also does not define Software development life cycle activities.

Scrum

System development process is supervised by Scrum by implementing ideas on elasticity, malleability and productivity from industrial perspectives. Scrum concentrates on how development team must work together to develop a Quality product in a changing environment [14, 15]. The word scrum is inspired by the scrum in rugby, which shows that the game is restarted when some accident, violation or change occurs. The scrum methodology is inspired from rugby where each member of the team places hands over each other and do quick planning that how to place the ball in the opponent side. The same practice is followed in scrum methodology where each member of scrum team is responsible to produce quality software. Scrum methodology composed of the product backlog, Spring Backlog and daily scrum. Product Backlog is very important regarding requirements engineering perspectives. It contains all functional requirements which are prioritized already. Product Backlog is similar to Software requirement specification document which contains all software and non functional requirement based on negotiations with customers. Each sprint is limited to 30 days. The high priority requirements are taken from the Product backlog and moved to the sprint backlog. While working on sprint no change is allowed in the current sprint. After development, sprint is released and customer uses it as a working software. If any errors or requirements mismatch occurs in current sprint, they are notified and are fixed in the other sprint.

Feature Driven Development (FDD)

In FDD main focus is on design and building phase rather than casing whole software development methodology, also FDD is composed of small iterations.[16]. Firstly, complete domain model of software is developed by domain specialists and software developers. The complete model is composed of class diagram which contains classes, relationship amongst classes, methods and attributes. Methods in a class represent functional requirements which are core user requirements for constructing the software. FDD feature is customer based function. Feature list items are prioritized by the development team. Feature list developed is tested and reviewed by domain experts.[17]. FDD suggests that 30-minuets meeting should be conducted every week so that status of features developed is debated and report must be written regarding the meeting.

Dynamic System Development Method (DSDM)

DSDM is agile methodology which resembles rapid software development whose primary goal is to develop and deliver quality software. [18]. When requirements are known then it is best use DSDM. DSDM agile methodology has first two phases similar to conventional software engineering models which are feasibility study and business study. Detailed requirements are indentified during these two phases while rest of the requirements are indentified during software development process. DSDM is quite open ended, there is no restriction on which method you elaborate requirements from customers. Main modules of software are handedover quickly to the customer, while the rest of functionalities are delivered in increments.[19]. Basic steps of DSSM' are regular customer participation, regular increments, team decision, integration testing after completion of each increment, change support and development. The core idea behind DSDM is "test as you go" [18].

Adaptive Software Development

ASD provides guidelines for incremental development of complex and large softwares [20]. ASD support continuous prototyping to overcome any requirements confusion and ambiguity. ASD suggests that the Water fall model performs very well when requirement are known and understandable, but its performance becomes worse when the requirements are changing frequently. So it is essential to introduce change tolerant methodology. The first increment of ASD must be short, to make sure that the customer is fully involved and project is going in accordance to the needs of the customer. Customer conducts a group review at the end of each cycle. Working software is explored during the review meeting. Meeting outcomes are documented as the change is requested.

Requirements engineering with agile development perspectives

Requirements engineering is one of the most important and critical activity in the software engineering domain. Requirements engineering works as a foundation for obtaining quality software. Requirements engineering resides inside the domain of software engineering.

Studies [30] indicate that 70% of the system errors are due to the inadequate system specification and 30% of the system errors are due to design issues.

Many factors are involved in requirements engineering like technical capabilities, organization atmosphere, product domain and market analysis. Therefore, the requirements engineering approaches vary from organization to organization and product to product due to above factors. Requirements engineering is composed of few main activities

which are elaborating requirements, modeling, and communication, agreeing and evolving requirements [31].In conventional software development techniques, requirements engineering processes focus on preparing whole requirement and prepare software specification documents before going to design phase while in agile requirements engineering requirements are evolved with the development of software. Also in agile development the requirements engineering phase vary with respect to projects, customers and development team. Moreover, this paper elaborates Agile requirements engineering activities in details which are discussed as follow.

Requirements Elicitation

Requirements elicitation is very critical and important part of requirements engineering. Requirements elicitation means understanding software product and knowing what functionalities user wants to be performed by the software. In this phase of requirements engineering main functional requirements are elaborated from different stakeholders (domain experts, end users etc) of the developing organization to understand the proposed system. Before going to design phase, the elaborated requirements from stakeholders needs to be analyzed, modeled and validated in order to make sure that all ambiguities, incompetence have been removed. Different techniques are used for gathering requirements from stakeholders some of which are defined in the below section.

Interviews

Interview is a type of requirements elicitation, in which facts about required software are identified from the key representatives of the organization. Ambiguities and misunderstandings can be pointed out in order to fixed them. There are mainly two type of conducting interviews

Close ended Interview: In Close ended interview requirements expert has selected questionnaires to be asked from the key stakeholders of the organization.

Open ended interview: In open ended interview, requirements expert does not have pre-defined questionnaires, he can ask questions in an open ended way from the organization representatives.

There are no strict rules for conducting both types of interviews. In general interviews can be conducted in any open environment like on a coffee table, inviting representatives to lunch/dinner or having a walk with customer. Requirements expert can ask one question, while discussing it new questions can be asked to be answered. [2].Rich information can be gained through interviews which helps the development team a lot. Sometimes interviews become cumbersome when data, gathered from different stakeholders through interviews, becomes conflicting.

Brainstorming

Brainstorming is a requirements elicitation technique used in a challenging environment. Creative solutions can be created through brainstorming for any specific requirements gathering problem. It is a group technique for requirements gathering, firstly problem has to be defined to be work over then problem must be diagnosed in a short time. Project manager has vital role in brainstorming as he defines time for the creative activity and also makes sure that everyone has expressed his/her opinion freely. At the end of creative session, the informations gathered, are evaluated by the team members also with the help of graph.The interconnection and dependences amongst the discuss ideas are presented in order to minimize the conflict and ambiguity in it.

Observation and Social Analysis

Observations technique is also used for requirements gathering; where requirements expert visits the customer organization physically and observes the work the customers perform. Observations techniques gives best result when informations are gathered from such organization where users hesitate to provide key information. To extract such type of information, requirements expert is even employed in the organization so that he observes each task they perform. Observations may also be performed indirectly, where the organizational tasks are viewed by some other sources like recording through video camera.

Prototyping

Prototype is a technique in which initial draft of the system is available in the early stage of the development process. When requirements are ambiguous and unclear to both customers and requirements expert then a prototype is developed which simulates the requirements to make it clear and unambiguous. There are two types of prototyping

Throw-way prototype used to understand the critical and ambiguous requirements and is discarded. Another one is evolutionary prototype,also used to understand the ambiguous and unclear requirements but this prototype becomes the part of working software. Prototype can be paper based (where system mockup is developed on piece of paper) “Wizard of Oz” prototypes(where prototype is based on the user inputs and individual simulates response to understand system requirements) or computerized prototypes (where with help of automated softwares an executable prototype is developed).

Questionnaires

Questionnaire is a research gadget composed of number of questions and other necessities for the aim of collecting requirements from the customers.[23]. It is the cheapest method to gather requirements as it does not require any special efforts. It is one time effort to prepare some generic questions which will help in gathering requirements for the required softwares.

Requirements Analysis

Requirements analysis provides further understanding of the requirements we have already elicited . Requirements analysis ensures that requirements are consistent (requirement will not be ambiguous), complete (no requirement is missing), feasibility (making sure that requirements are implementable in term of budget and schedule). With the help of requirements prioritization and agreement conflicts are resolved between requirements. Doubtful requirements are prioritized to find out important requirements. Requirement analysis contains the following techniques

Requirements Prioritization

Now a days softwares are developed with incremental approach to deliver working software quickly to customers. Also software projects contain too many requirements which

cannot be implemented in single increment. Therefore stakeholders along with requirement engineering team will identify high priority requirements from the list of all available requirements to be implemented in the first iteration. Firesmith says that software product may be composed of hundred and thousands of requirements [27]. Apparently all the requirements identified are not in the interest of customers, so with limited resources requirements are priotrized to deliver high quality software. When a single stakeholder is involved in the development of software, requirements of high significance and low significance can be easily identified. But situation becomes worst when many stakeholders are involved in the development of software, so identifying high and low priority requirements become challenging as every stakeholder have own perception about requirements[32]. Prioritization helps in resolving conflicts amongst different stakeholders and building consensus amongst them. On the basis of priority stakeholders can assign resources to requirements with the help of requirements prioritization [25]. Prioritization helps in identifying harms in requirements like misapprehension of requirement or any ambiguity in requirements.It also suggests how to remove these problems in order to avoid future changes [26]. Requirements prioritization is very essential in software development as it helps in minimizing project failure [24].

Modeling

Requirements model works as a bridge to transform requirements from analysis to software design phase. In requirements modeling phase, requirements are represented in several graphic forms like flow charts, data flow diagrams, use cases, sequence, class etc these help both the development team and stakeholders to understand the project.

Joint Application Development (JAD)

JAD is group meeting used to discus, negotiate and gather functional requirements for a specific software to be developmed. JAD is very beneficial and cost effective if organizational customers and team members are prevented from "running out of course". Such type of meeting includes strategies to increase user's involvement , exploring development and improving specification quality.[29]. The aim of JAD is to explain the software project at different stages in details, make a solution for it and observe the project till its completion. JAD is composed of Project Managers, Top level executives, organizational users, domain experts and external technical experts.[28]. If any conflict occurs between stakeholders on requirements, JAD can be used to resolve the conflict amongst stakeholders and build consenus amongst them.

Use Cases/Scenarios

Uses cases elaborate communications between customers and system under development where main focus is on what customers want to do with the system. Sequence of interaction between system actors (for example: any human, other computer system or any software component) and system is described by use cases. Use cases can be used at early stages of the development process as they represent system functional requirements. Purpose of use case is to validate functional requirements both by customer and requirement analyst. Scenarios are the description of interaction between customer and requirements team. Scenarios must contain entry and exit criteria, objectives, primary actors, normal flow of the events and alternatives of the events.[22].

Requirements Documentation

To communicate requirements between developers and stakeholders, software documentation is very helpful and beneficial. Documentation helps in evaluating software process and subsequent softwares (requirements, software design, verification and validation and software testing). It also works as a baseline for change control. Requirements document must be complete, reliable, easy to understand, crisp, unambiguous and implementable. Requirements specification document is a part of contract based on customer- developer relationship.

Requirements Validation

Requirement validation is the process to make sure that requirements gathered are according to the interest of customer and are valid to be implemented. Validation process accepts inputs in the form of requirements documents and company standards/ knowledge, then produces output in the form of reported problems along with corrective process. General techniques used for validation of requirements are Reviews and requirements testing. Requirements validation finds out problems in the requirements documents either by review technique or by testing and also suggests procedure how to cope with these problems.

Requirements Management

The goal of requirements management is to manage requirements changes occurred either by customer or organization. Requirement tractability is very important for managing requirements. For successful projects, it is necessary that the project support requirements management covering both, version control and requirement change management.[29]. With appropriate requirements management process, by removing inaccurate, incomplete and ambiguous requirements, organization can save up to 20% of the total cost.[29].

REQUIREMENT ENGINEERING RECOMMENDATION TO AGILE DEVELOPMENT

This section introduces some recommendation to improve the performance of agile requirement engineering process and also help in producing quality requirements.

Feasibility Study

In feasibility study main focus paid is to achieve goal of company. In feasibility study, it is analyzed whether the desired software will be implemented practically in terms of performance, contribution of software to business and also whether software is fulfilling the goal of the organization. In agile development, feasibility study must be included in order to give answer to questions like Is your organization ready to work on Agile development? Will software be developed within the given budget and time?. Feasibility study is a short report answering the above questions which will make both customer and developing organization confident to start work on the software.

Customer Involvement

Customer plays key role in agile development. Literature shows that customer's involvement throughout the development process leads the project to success while lack of customer interest and its involvement in the development process results in the project failure. It is always difficult for the development team to take a user perspective while developing the software which usually creates inconsistencies. To overcome such inconsistencies agile methodology recommends having a customer representative which is aware of all the customer needs. Customer must be domain expert in the field in which the software is being developed and also have some authority to take some important decisions in time without waiting for the approval of the competent authority. In order to develop generic products for which any company is not paying directly, development team can assign responsibility of customer to any member with in the development team. He will behave like a customer and will be responsible to answer all questions of development team.

Requirements Prioritization

In software system many requirements exist and it is not feasible to implement all the requirements in one iteration, therefore stakeholder together with requirement team can decide which requirements are of greater importance and which are less important. A vital feature of requirements prioritization is that it adds multiple values to stakeholders.It is essential for the stakeholders to know that in which respect requirements priority is going to be identified, whether the priority is in form of market value, fast implementation, high usability or high quality software. It is essential for the customers that prior to requirements prioritization they must know the prioritization criteria in order to avoid any confusion and misunderstanding. Agile methodology must support requirements prioritization, as products are built in increments so it is necessary to prioritize requirements.

Requirements Reviews

In agile development a quick requirements review must be conducted in order to identify missing and incomplete requirements. According to Standish Group Report [30], most important factor which causes project failure is incomplete requirements which are 13.1%. By applying quick review on requirements, which are indentified by the customers using different requirements gathering techniques, can result in the elimination of incomplete requirements and make the project successful.

Waste in Requirements

Agile methodology should focus on identifying and removing waste in requirements. It is necessary to identify and remove waste in requirements because it will create further waste in later phases of software development.It increases the cost and time which will ultimately leads the project to failure in term of cost and functionality. If waste in requirements occurs than it causes a lot of problems in later phases like more code will be written which causes higher cost, code become complex, final delivery of the working software is delayed and also maintenance becomes difficult. Waste in requirements can be identified by involving customer and also introducing requirement prioritization to select high priority requirements.

Requirement Elicitation using Customers language

To make the project successful, the requirements experts in agile development team should gather requirements from the customers using natural language. To achieve this agile expert must have knowledge of the system which is going to be under developed. RE must gathered requirements using language of customers, making easy to collect clear and understandable requirements.

Decomposing Complex Requirements

When the requirement expert team considers a requirement too complex to be implemented then requirement expert can decompose it into implementable and understandable requirements. Decomposing requirements can give better understanding to agile development team.

Non Functional Requirements

There is no formal procedure in agile methodology to eliciiate non functional requirements. While gathering functional requirements from customers, non functional requirements are elaborated implicitly. Non functional requirements are of great importance in software development, without non functional requirements software fails. Usually customers are unaware of non functional requirements prior to working software. Agile development team should educate the customers regarding the importance of non functional requirements. Agile expert needs to save the time and resources in order to identify non functional requirements before the product is delivered to customer. Projects get successful when non functional requirements are considered and attention is paid to them.

Volatile Requirements

Agile experts are aware of the fact that at the start of software project all requirements cannot be elaborated from the customers at once. It is also known that nature of requirements are volatile, requirements may change due to customer needs, technical and social environment. As changes in requirements are inevitable therefore agile experts need to accommodate changes in development process. Therefore agile methodology suggests frequent increments of software to accommodate requirements changes till the end of the project.

Requirement Traceability and Configuration

In agile development software product is developed in incremental approach, which causes problems when it comes to maintenance. Without any proper requirement traceability and configuration management approach it is difficult to remember that in which increment change has to be accommodate. Situation becomes worst when lager systems are developed on agile methodology.Using agile development a working software in the form of increment is delivered to customer after every 14 days. Customer uses the working increment and if any errors exist in it these are reported to agile team. In that time another increment is delivered to customer and so on. It becomes difficult for agile team to manage the changes in corresponding increments. Also if change is applied in one increment it also affects other requirements. Traceability is required to keep track of these requirements. . Requirements traceability and configuration management tools should be used for this purpose.

CONCLUSION

Agile development is very rich methodology for software development, which is almost adopted in many of the development organizations now a days. This paper provides detailed analysis of requirements engineering techniques and practices in agile development. It is concluded in the paper that agile development can be made more effective and productive with the introduction of some useful requirement engineering practices. This article has also suggested that how performance of agile methodology can be improved in large scale software development. It is recommended in the article that taking care of non functional requirements along with functional requirements can result in success of the project. Traceability and configuration management helps in the management of larger software development using agile development. In future work, practices introduced in this article for agile development will be implemented in software industries to check the performance of agile development.

REFERENCES

[1] Abrahamsson P, Salo O, Ronkainen J, Warsta J (2002) Agile software development methods: Review and analysis. EPSOO 2002, VTT Publications 478.

[2] Beck K, Beedle M, Bennekum A, Cockburn A, Cunningham W, Fowler M, Grenning J, Highsmith J, Hunt A, Jeffries R, Kern J, Marick B, Martin RC, Mellor S, Schwaber K, Sutherland J, Thomas D (2001) Manifesto for Agile software Development. Accessed on 5th December 2004, online at: http://www.agilemanifesto.org/ Cohn M, Ford D (2002) Introducing an Agile process to an organization. Access on 5th Decemberc 2004 ttp://www.mountaingoatsoftware.com/articles/IntroducingAnAgileProcess.pdf

[3] Poppendieck T, Poppendieck M (2003) Lean software development: An agile toolkit for software development managers. Addison-Wesley, London UK.

[4] Frauke Paetsch, Dr. Armin Eberlein, Dr. Frank Maurer "Requirements Engineering and Agile Software Development" in Proceedings of the Twelfth IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE'03).

[5] K. Beck, A. Cockburn, R. Jeffries, and J. Highsmith, (2001). "Agile manifesto". http://www.agilemanifesto.org, 23-02-2010.

[6] S. Bose, M. Kurhekar, J. Ghoshal, "Agile Methodology in Requirements Engineering," *SETLabs Briefings Online*. http://www.infosys.com/research/publications/agilerequirements- engineering.pdf, February, 2010.

[7] A. Eberlein and J. Leite, "Agile Requirements Definition: A View from Requirements Engineering," Proceedings of the International Workshop on Time-Constrained Requirements Engineering (TCRE'02), Germany, 2002.

[8] R. Goetz, "How Agile Processes Can Help in Time-Constrained Requirements Engineering," *International Workshop on Time-Constrained Requirements Engineering*, 2002.

[9] A. Sillitti, G. Succi, "Requirements Engineering for Agile methods," *Engineering and Managing Software Requirements, Springer*, 2005.

[10] B. Ramesh, L. Cao and R. Baskerville, "Agile requirements engineering practices and challenges: an empirical study," *Info Systems J*, "doi:10.1111/j.1365-2575.2007.00259.x", 2007.

[11] J. Highsmith, "Agile Software Development Ecosystems," *Addison–Wesley, Boston*, MA, 2002.

[12] Kent BeckExtreme Programming explained, Addison-Wesley, 1999.

[13] Ken Schwaber, Mike Beedle: *Agile Software Development with Scrum*, Prentice Hall, 2001.

[14] Pekka Abrahamsson, Outi Salo, Jussi Rankainen & Juhani Warsta : *Agile software development methods - Review and analysis*, VTT Electronics, 2002.

[15] Pekka Abrahamsson, Outi Salo, Jussi Rankainen & Juhani Warsta : *Agile software development methods - Review and analysis*, VTT Electronics, 2002.

[16] Peter Coad, Eric Lefebvre, Jeff De Luca : Java Modeling in Color with UML, Prentice Hall PTR, 1999, Chapter 6.

[17] Jennifer Stapleton, "DSDM - Dynamic System Development Method," *Addison-Wesley*, 1995.

[18] Marc Clifton, J. Dunlap. (2009) [Online]. Available: http://www.codeproject.com/Articles/5097/What-Is-DSDM.

[19] James A. Highsmith III: *Adaptive Software Development*, Dorset House Publishing, 1996.

[20] Beichter F. et al, " SLAN-4-A Software Specification and Design Language", IEEE Transaction on Software Engineering, SE- 10,2,1994, pp 155-162.

[21] Gerald Kotonya and Ian Sommerville: *Requirements Engineering*, John Wiley & Sons, 1997.

[22] (2010) The WikiPedia website. [Online]. Available: http://en.wikipedia.org/wiki/Questionnaire.

[23] Hatton, S, Early prioritization of goals. In Advances in conceptual modeling – Foundations and applications ,pp. 235-244.

[24] Karlsson, J., & Ryan, K. A Cost-Value approach for prioritizing requirements. *IEEE Software, 14*(5), 67-74.

[25] Karlsson, J., Wohlin, C., & Regnell, B. An evaluation of methods for prioritizing software requirements . *Information and Software Technology, 39*(14-15), 939-947.

[26] Firesmith, D, Prioritizing requirements. Journal of Object Technology, 3(8), 35-47.

[27] Linda A. Macaulay: *Requirements Engineering*, Springer Verlag, 1996.

[28] The IBM website. [Online]. Available: http:// www-142.ibm.com/software/products/us/en/category/SW740.

[29] Standish Group, CHAOS Report 1994. Accessed 5th December 2004.

http://www.standishgroup.com/sample_research/chaos_1994_1.php

[30] Sommerville, I. Software engineering, Wokingham, England: Addison-Wesley,5th edition.

[31] Nuseibeh, B., & Easterbrook, S. (2000). *Requirements engineering: A roadmap*. Proceedings of the Conference on the Future of Software Engineering, 35 – 46.

[32] Javed, K H, Izaz, U R, Yawar, H K, Iftikhar J K, Salman R,Comparison of requirement prioritization techniques to find best prioritization technique, I.J.Modern Education and Computer Science,2015,11,53-59.