SindhUniv. Res. Jour. (Sci. Ser.) Vol. 51 (04) 677-680 (2019)

http://doi.org/10.26692/sujo/2019.12.107

SINDHUNIVERSITYRESEARCHJOURNAL(SCIENCE SERIES)

Process of Requirement Mining Using Agile Technique

M. NASEEM, S. S. ZIA, T. J. A. MUGHAL*, U. YOUSUF, M. KHAN

Department of Software Engineering, Sir Syed University of Engineering & Technology

Received 11th March 2019 and Revised 17th September 2019

Abstract: Requirement engineering is a process in pursuit to develop a software system that can recognize, evaluate, manuscript and legalized its necessitates. Recent studies have promoted the concept of software development using agile software approaches. Several techniques have been adopted to introduce a software that is proficient enough to meet the customer requirements. This quick agile alteration requires an organization to be improved on continuous basis in their approaches to develop Requirement Engineering software. These agile alterations are beneficial due to its minimum cost strategy of change through software process. In case the project is poorly managed and the cost ratio over it is high it could mess up the entire project and the system showed its failure and problems stays remain. This work introduces a strengthen approach that concentrates on requirements and related information so as to limit and mitigate the effects of change in an agile process based on the identified challenges associated to agile development.

Keywords: Requirement Engineering, Agile development, Process

1. INTRODUCTION

Recent corporate community admires the concept of Agile software development (ASD) because executing this software in business domain enables the corporations to make instant changes in the route of product expansion decisions. Supple methodologies in requirement engineering are required to furnish short term goals of corporation's track variations. Further more agile approaches such as Scrum, Kanban or Extreme Programming (Matharu *et al.* 2015) are usually merged Human Centered Design HCD actions to highlight a value driven methodology in product development (Schön *et al.* 2017).

In order to pursue this aim, the domain of agile Requirement Engineering arose significantly in previous decade. This platform of agile encourages plan environs of revision, collaboration, booster club, quick provisions, focus on client and visibly ranked which one should be set aside for every assigned task in progression of agile methodologies (Dingsøyr *et al.* 2012). Requirement engineering in agile approaches responsible in exploring, assessment, interviews and teamwork (Elghariani and Kama 2016). Likewise agile approaches are advised to implement requirement engineering undertakings involved user participation, supplies listing, requirements demonstration and its credentials(De Lucia and Qusef 2010). Tailing agile methodology the user exerts meticulously along with development crew to suitably respond to variations and to persistently endorsed the delivery of product (Poppendieck and Cusumano 2012). Areas that requires modifications at any instance can benefit from its dynamic and obtainable features. Agile approach presented as low risk approach in global software development (GSD) and reduces the requirement of synchronization efforts that consequently enhance the productivity (Inayat *et al.* 2015). The aim of this article is to identify the requirement mining process that will incorporate the agile development cycle that will improve the results.

The next segment concisely provides an outline on agile process development. Section 3 discusses the proposed requirement mining process that will incorporate with agile development process. Section 4 shows the benefit of the requirement mining while incorporate it in the agile development process. In section 5, some limitation is to be highlighted and Section 6 shows the conclusion and future direction.

2. AGILE PROCESS DEVELOPMENT

In agile methodology, the client proceed thoroughly with the development group in order to give quick responses to changes required and continually validate the delivery of product. It is also evidenced in the literature that projects that adopt agile methods exhibiting greater productivity (Shrivastava and others 2010), a smaller amount amendment needed, and more proficient charges for fixing shortcoming (Glaiel, Moulton, and Madnick 2014). When comparing with other software development methods that are traditionally document oriented, Agile development is found to be more code base. Though it is not the main point but rather a sign to differentiate between them (Rakshith and Patil 2013).

++ Correspondence author: Syed Saood Zia, saood_zia@hotmail.com

*Faculty of Engineering, Sciences and Technology, Hamdard University, Karachi

Traditional analytical procedures are not required in agile development it rather involve adaptive strategy. Considering the need for amendment agile methodologies strive to give quick fix responses through adaptive method.

Agile methodologies work with real manforce than involving complex software analysis. Highly competitive field experts to make direct collaboration are the key resource of agile development .this what makes the agile method more preferable than traditional data centric laborious software development methods.

In order to execute the proper functioning of agile a very well defined and systematic approach is followed that comprises of a route map that thrives to use adaptive strategy in considering the requirement and their delivered solutions (Abrahamsson *et al.* 2017). Following 12 guiding line principles are the base of agile development (Soundararajan, Arthur, and Balci 2012):

1. To satisfy customers by delivering early and continuously valued software products.
2. Welcome altering requests, homogenously in late expansion procedure.
3. Delivering workable software yields regularly basis (weekly rather than monthly)
4. Collaborative working among business people and agile work force is mandatory throughout the project.
5. Motivated and enthusiastic individuals are deemed to be assigned for project development, who should be trusted and supporting team members.
6. One to one meetings are best suited during the tenure within the team members.
7. Progression of the agile is predominantly assessed with the conveyance working software.
8. Agile development need to follows a sustainable progression to maintain persistent pace.
9. Vigilant supervision over methodological excellence and good scheme of adaptation appreciated.
10. Work methodology needs to be simplified.
11. Unsurpassed structural designs, necessities, and enterprises surface from self-organized teams
12. Systematic intervals, the team imitates on in what way to develop more effective and adjustments made consequently.

In light of the above principles and this being a highlight of the agile manifesto, it is important to note that RE practices such as observations, interviews, customer involvement, requirements prioritization, modeling, and documentation, workshops and strong team collaboration in an iteration based agile development, have also been suggested to be used like in traditional methods.

3. PROCESS OF REQUIREMENT MINING

As a remedy to the risk of failing in some of the principals proposed by the agilists in the agile manifesto and to remedy risk of project management it is propose that the requirement mining process as an intervention in the original agile process as follows;

Step 1: Domain expert are included as the intermediate body to work for the client in place of the proxy who may not understand software development, on assumption that the domain expert understands software development and the clients business.

Step 2: Domain expert works with the development team and generates user requirements and system requirements. System requirements are characteristics that components of a merchandise requisites hold in product to make their consumer requirement fulfil. Domain experts are deemed to simplify and identifying the user requirements and make them accessible into the proposed solutions. These provided elucidations from agile are the gradient of possessions that a project stakeholder are willing to have, such as what would be its responsibility, how much it cost, how much would it weigh, and what are its boundaries and interfaces will be.

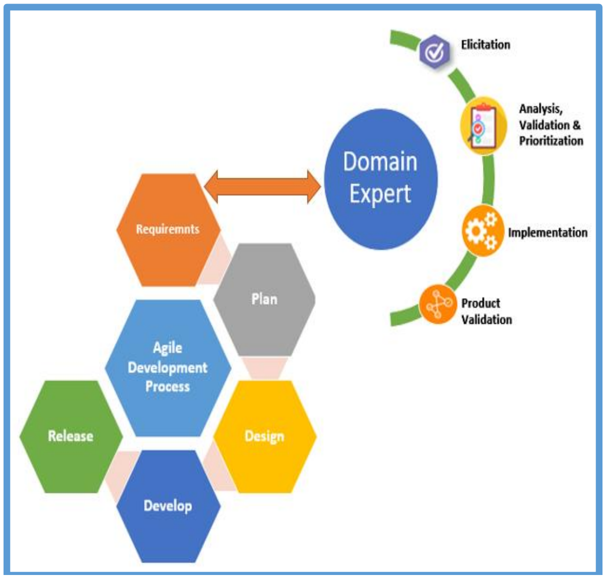

Fig. 1: Proposed Requirement Mining Process in Agile Software Development Process

Step 3: Information generated through requirements mining process gives us precise and substantial information for test case generation, anomaly detection and traceability though the alignment of the user requirement to the system requirement.

Once the system requirements being according to user requirements this shows that change has brought and we can be able to track the cause and point of change. This proposed requirements mining is cognizant of the principals proposed by the proponents of the agilists, in the same faith we consider the fact that a knowledgeable person as a domain expert will save the entire process substantial time and extra costs by mining the requirements and related information which will be used for traceability, test case generation and anomaly detection. It is important to note that most of the problems experienced in agile development can be settled by the three activities and the solutions coined to settle these defects can fit in traceability, test case generation and anomaly detection.

4. BENEFITS OF REQUIREMENT MINING

There is deliberate focus on settling the gaps in agile development, although and have come up with different solutions, the solutions are spread although the agile software development process. This proposal consolidates all these activities as responsibility of the domain expert who is introduced in place of the surrogate customer. Among other challenges this proposal mitigates the problems associated with conflict and disagreement. That has drawn attention to the effect towards requirements that are well aligned and reduce the possibility of refactoring and struggling with the trouble of variation in management at any stage of the software development.

Although there is considerable solutions that have been coined to solve the challenges associated to agile methods (Sekitoleko *et al.* 2014), a lot more emphasis requires to be put into working around making requirements to make them extremely clear and most especially rich enough to avoid rework particularly those that arise out the changes of the client’s needs. Even if in the agile manifesto continuous change of requirements is tolerated (Batra *et al.* 2010; Niu *et al.* 2018), it is paramount to note that at a certain point, time and costs may change due to change of requirements and hence conflict which will affect the ultimate project. This same information collected for requirements is very important for traceability which is often looked at as a burdensome, time wasting and unnecessary process by most agilists in practice yet, in some situations during agile practice, traceability is ever more adopted especially while building safety-critical, larger, and distributed software development projects (Abrahamsson *et al.* 2017), it is often in these environments that the benefits of traceability outweigh.

5. CONCLUSION

This paper evaluates the performances in the development of agile methodology and figures out the challenges that are deemed to be considered and fix them the way that their core principals remains the unchanged. The proposed solutions according to organization requirements are the interventions that hosts in the agile methodology shown in figure1. This intervention attempt to minimize the shortcomings arose in identifying the challenges and exert to accomplish the trademark. This intermediate intervention in agile software is being proposed to enhance the proficiency of the system that thrives to limit and mitigate the effects of change. Future direction could be the infusion of domain experts in every phase of agile development.

REFERENCES:

Abrahamsson, P., O. Salo, J. Ronkainen, and J. Warsta. (2017). “Agile Software Development Methods: Review and Analysis.” ArXiv Preprint ArXiv:1709.08439.

Batra, D., W. Xia, Debra E. V. Meer, and K. Dutta. (2010). “Balancing Agile and Structured Development Approaches to Successfully Manage Large Distributed Software Projects: A Case Study from the Cruise Line Industry.” CAIS 27: 21Pp.

Dingsøyr, T., S. Nerur, V.G. Balijepally, and N. B. Moe. (2012). “A Decade of Agile Methodologies: Towards Explaining Agile Software Development.” Elsevier.

Elghariani, K and N. Kama. 2016) “Review on Agile Requirements Engineering Challenges.” In 2016 3rd International Conference on Computer and Information Sciences (ICCOINS), 507–12.

Glaiel, F. S., A. Moulton, and S. E. Madnick. (2014). “Agile Project Dynamics: A System Dynamics Investigation of Agile Software Development Methods.”

Inayat, I., S. S. Salim, S. Marczak, M. Daneva, and S. Shamshirband. (2015). “A Systematic Literature Review on Agile Requirements Engineering Practices and Challenges.” Computers in Human Behavior 51: 915–29.

Lucia, A. De, and A. Qusef. (2010). “Requirements Engineering in Agile Software Development.” Journal of Emerging Technologies in Web Intelligence 2 (3): 212–20.

Matharu, G. S., A. Mishra, H. Singh, and P. Upadhyay. (2015). “Empirical Study of Agile Software Development Methodologies: A Comparative Analysis.” ACM SIGSOFT Software Engineering Notes 40 (1): 1–6.

Niu, N., S. Brinkkemper, X. Franch, J. Partanen, and J. Savolainen. (2018). “Requirements Engineering and Continuous Deployment.” IEEE Software 35 (2): 86–90.

Poppendieck, M and M. A. Cusumano. (2012). “Lean Software Development: A Tutorial.” IEEE Software 29 (5): 26–32.

Rakshith, H. P. and A. P. Patil. (2013). “An Improvised Prespective to Agile Methodology.” In 2013 IEEE 3rd International Conference on System Engineering and Technology, 381–84.

Schön, Eva-Maria, D. Winter, M. J. Escalona, and J. Thomaschewski. (2017). “Key Challenges in Agile Requirements Engineering.” In International Conference on Agile Software Development, 37–51.

Sekitoleko, N., F. Evbota, E. Knauss, A. Sandberg, M. Chaudron, and H. H. Olsson. (2014). “Technical Dependency Challenges in Large-Scale Agile Software Development.” In International Conference on Agile Software Development, 46–61.

Shrivastava, S. V and others. (2010). “Distributed Agile Software Development: A Review.” ArXiv Preprint ArXiv:1006.1955.

Soundararajan, S., J. Arthur, and O. Balci. (2012). “A Methodology for Assessing Agile Software Development Methods.” In Agile Conference, 51–54.