

OVERVIEW ON AGILE REQUIREMENT ENGINEERING: PRACTICES, TECHNIQUES AND CHALLENGES

¹MAHDI MOUSAEI, ²TAGHI JAVDANI GANDOMANI

¹Young Researchers and Elite Club, Isfahan (Khorasan) Branch, Islamic Azad University, Isfahan, Iran

²Department of Computer Engineering, Boroujen Branch, Islamic Azad University, Boroujen, Iran

Email: ¹moosayimahdis@gmail.com, ²t_javdani@azad.ac.ir

ABSTRACT

Requirement engineering (RE) is one of the important phases of development life cycle. Many software projects are failed because that was made without enough requirements information. RE is one of a process of customer services which have to expect customer from a system. Correct executions of engineering requirements caused to reduce the cost of software projects. In the agile methods unlike traditional methods, RE agrees with the change of new requirements. In this research, a survey, about practices, techniques, challenges, and activities, is conducted in agile requirement engineering. Moreover, this survey compares traditional RE and agile RE in different aspects.

Keywords: agile requirement engineering; agile development; requirement engineering; RE challenges; RE techniques; RE practices;

1. INTRODUCTION

Requirements are usually divided into two types, functional and non-functional requirements [1]. Requirements are the base of all software products and Requirements Engineering (RE) plays important role in system development [2]. RE is one of the processes to find the customer needs. Agile development methods are used to communication between customers and agile teams[3]. In RE phase care should be taken because adding errors in this phase to remove it is very expensive in another phase of development. In RE is the activity by which the needs of the systems are specified. RE involves the processes to find out the needs of stakeholders, assessment of requirements, modeling, negotiating, validating, documenting, managing and implementing these requirements [4]. According to more identifications of requirements in the projects the success of the projects so increases [5]. The goal of this research aims to understandable challenges, practices and related issues RE in agile software development.

2. REQUIREMENTS ENGINEERING IN AGILE DEVELOPMENT

One of the difference between in traditional and agile methods, RE are done in traditional methods before going to the design phase, while in the agile methods done all over the phases development and agrees to change of new requirements even late in the development lifecycle [3]. Unknown requirements are one the main cause of failure of software projects but there are many practices for managing requirements in agile development. These methods emphasize on a continuous interaction with the customer [6, 7]. In this research describe the activity of agile requirements engineering which involves feasibility study, requirements elicitation, requirements analysis, requirements documentation, requirements validation, requirements management.

a) Feasibility study

For all new systems, the requirements engineering should start with a feasibility study[8]. The Feasibility Study gives the overview the goals of system. In this activity to be answered to some of the important questions in the organization [3].

b) Requirements elicitation

In requirements elicitation, the agile teams work with stakeholders to find out about the services of the system. Techniques used for requirements elicitation in AD consist Interviews, Brainstorming, Ethnography, Use Case analysis [3].

c) Requirements analysis

Requirements analysis define the elicited requirements are unknown, imperfect, ambiguous or inconsistent, and the decision to resolve these issues. Techniques used for requirements analysis in the AD include Joint Application Development (JAD), Modelling, and Prioritization [3].

d) Requirements documentation

The goal of requirements documentation is to communicate requirements between stakeholders and agile teams. For example, these techniques used UML modelling [3].

e) Requirements validation

The purpose of requirements validation is to ensure that requirements really define the system which the customer wants. Methods used for requirements validation in agile include Requirements review, Unit testing, Evolutionary prototyping, and Evolutionary prototyping [3].

f) Requirements management

In the AD, managers should provide and maintain a framework for the interaction between the agile teams and the stakeholders. Agile practices can play a main role in the management of large projects [3].

3. COMPARISON BETWEEN TRADITIONAL RE AND AGILE RE

In traditional RE with using stakeholders and requirement engineer, the requirements are identified with different techniques but, the Agile RE the product owner explains requirements for the development team and requirements put on the product backlog. Product owner confirms the requirements from customer and Guarantees that they are completing and stable. Requirement prioritization and modelling are done in the requirement analysis step. In Traditional RE Requirement document is created by developers for future development but, the Agile RE develops only the short documentation for future. In traditional RE, requirements are examined by developers and customers but, the Agile RE, the customer is involvement in all over of development process and requirements are examined at every cycle. Traditional RE Requirement management changing requirements are done by the Change Control Board. In Agile requirement management changing requirements done at any stage [9]. Table 1 indicates the summary of comparison between traditional and agile RE.

Table 1. Summary of comparison between traditional and agile RE [9]

Traditional RE	Role		Activities		Artefacts	
	Traditional RE	Agile RE	Traditional RE	Agile RE	Traditional RE	Agile RE
A. Requirement Elicitation	System Analyst, Requirement Engineer, Domain expert.	Stakeholder, Product Owner, Team Members, Visionary.	Discover requirements, System Constraint, Prototyping, Brain Storming, Interviewing	Description of all features, wish list item.	List down Requirement	Vision, Product backlog.
B. Requirement Analysis						
1. Requirement Prioritization	Customer, developer, Team leader	Product owner, Scrum Master	Most valuable features, set priority.	Prioritized functions	Prioritized Requirements	Prioritized Product backlog List
2. Modelling	Software Engineer, System Analyst, Graphics Designer	Development Team	Bridge between analysis and design	Preliminary Model	Data Flow Model, Semantic Data Model, OOD	Iterative Functional Model

C. Requirement Documents	Technical Writer Stakeholder, customer,	Product owner, Development Team	Preparation of Software requirement specification	Preparation of System Architecture	Software Requirement specification document	User stories, product backlog
D. Requirement Validation	Development Team, System Analyst	Product owner, Development team	Accepted Requirements as per set standard.	Project status meeting, project demonstration meeting, Retrospective meeting	Validate requirements	Product backlog, Sprint backlog, Sprint
E. Requirement management	Requirement Engineer, Customer, Developer	Product owner, Development Team	To capture information, version control management.	Update Sprint Backlog	Baseline Version control document	Product backlog, Sprint backlog

4. REQUIREMENTS ENGINEERING TECHNIQUES FOR AGILE APPROACHES

There are many techniques for agile methods. In this part indicate to the techniques to improve the efficiency of requirements engineering processes in agile development.

a) Customer involvement

The Agile approach focuses very powerfully on customer interaction. Involvement with Customer can be one of the causes of success or failure of projects. In traditional RE and agile methods Emphasis on the involvement of stakeholder. One of difference between traditional methods and agile methods is that in traditional methods the customer is involved during the Primary phase of the project while agile methods involved the customer is throughout the whole development process [3, 10].

b) Interviews

Interviews provide direct access to the requirements of the project. Speaking to the customer is the best way to obtain information needed for agile approach. Direct interaction helps to relationships between customers and developers [3, 10].

c) Prioritization

A common method is first developing features with higher priority. The prioritization should be repeated during all development process because new requirements are added during development. Requirements should prioritize at the beginning of each development cycle [11].

d) JAD sessions

The JAD sessions include approaches to increase user involvement, speed up development, and improving the quality of specifications [12]. In the agile approach and using JAD can resolve conflicts between stakeholders' requirements. JAD sessions encourage customer involvement and confidence with each other.

e) Documentation

Create documentation complete in agile software development is not cost effective. Also, the lacks documentation in agile development might cause problems for agile teams [10]. The goal of requirements documentation is to communicate requirements (or knowledge sharing) between stakeholders and agile teams [3].

f) Agile Projects Contracts

Initially, the important critical requirements are determined and later on project leaders, can define an initial cost for agile projects and also guess the cost of Later changes [3].

g) Smaller agile teams are flexible

Small agile teams can ongoing communications between team members and stakeholders and they can better control requirement change. Research shows that small agile teams increase the chances of the project success [3].

h) Requirements splitting

If requirements are too complex, splitting helps the customer to cut up it into simpler ones. Also, it helps to agile teams to improve understand the functionalities requested by the customer and helps agile teams working in parallel with frequent communications between them [3].

i) Requirements traceability

The relationship in TDD environment is a fine practice to identifying the Requirements traceability. The traceability links between test cases and code should be identified and evolved to control co-changes. Traceability between requirements, source code and unit tests can be used to drive software development, by identifying requirements for which unit tests or source code has not been implemented yet [3].

5. AGILE REQUIREMENT ENGINEERING PRACTICES

In this research described the methods of RE in agile software development. Requirement engineering practices used in agile methodology classified in below.

a) Direct communication

Communicate team members and client directly with each other, is a characteristic of agile RE. The goal of agile RE is helped to effectively transfer ideas from the customer to the development team, rather than create wide documentation [13]. Most organizations avoid formal documentation of specifications. Instead, they use techniques such as user stories to define requirements.

b) Interaction and participation with customer

Interaction and participation with the customer are introduced as the main reasons for success or failure of projects. In agile development, requirements clearly are not defined; instead, they arise during the development process. Agile RE consistently is done at each development cycle. At each cycle starts the customer meets with the development team to provide detailed information for special features that must be implemented [11].

c) Preparation of user stories

User stories are created as characteristics of the customer requirements [14]. These stories have changed from documentation to discussion [15]. User stories make easy communication and better comprehensive understanding between stakeholders [14].

d) Prioritization of Requirements

Agile development implements early the highest priority features. Prioritization frequently happens in the planning meetings at the beginning of each cycle, but in traditional RE, requirements are usually prioritized once [13].

e) Change management

Change management is a big challenge for traditional approaches but, in agile RE, Because of dynamic nature change management is the greatest benefit for agile RE [11, 16].

f) Cross-functional teams

A cross-functional team is responsible for defining and documenting the requirements for a feature. In the agile approach, developers, testers, designers, and managers work together. That caused to reduce challenges such as to be fulfilled requirements and communication gaps [11, 16].

g) Prototyping

Prototyping is a suitable idea for a system to help determining the requirements. Some projects with using prototyping as a way to communicate with their customers [16].

h) Testing before coding

Many companies use tests to take complete requirements and design documentation are related to production code. The main challenge of Testing is the developers aren't accustomed to writing tests before coding [11].

i) Requirements modelling

Requirement modelling is an essential activity in RE [17]. Requirement modelling is executed in agile software development methods, but it is different from RE in traditional software development methods [16]. Modelling agile requirements is based on planning, which it provides visual and easy to read goal graphs for project managers, and team members [18].

j) Review meetings and acceptance tests

Frequent review meetings with agile methods are used for requirements validation. During the meeting, the features released and customers provide feedback and comments [13].

k) Code refactoring

The concept of code refactoring is modified of code structure to improve on the structure and improve changes [13, 19]. Also, code refactoring is a method to adopting changes in requirements [20].

l) Shared conceptualizations

Conceptualization is one the concepts related to cognition [21]. Shared conceptualization is an acceptable description of how the team is able to RE activities in practice [22]. Agile teams with using shared conceptualizations in development help to solving the problems [16].

m) Pairing for requirements analysis

Agile software development methods suggest that sharing tasks between a pair of developers have very advantages. Adopting the pairing practice for requirement analysis persuade stakeholders to adopt more roles [23]. One of the methods to close communication gaps between agile teams is Pairing practice in requirements analysis [16].

n) Retrospectives

Retrospectives are planned one day after the Sprint Reviews meeting. Analysis of many metrics such as: estimate Hours (difference between available capacity and hours expended on engineering activities), Effort Deviation (difference between estimated and actual effort in percentage), Percentage of stories that were 'Not Completed' as per plan in each iteration, Quality of the software in terms of number of defects, and Overall Productivity of the team during retrospectives meeting help to teams for identifying suitable actions to improve the overall performance [24].

o) Continuous planning

Continuous planning is not generally adopted and applied all over the organizations and that it currently includes only a certain type of planning (e.g., Release planning). The importance of continuous planning will increase dramatically in disordered business environments [25].

6. AGILE METHODS AND RE CHALLENGES

In traditional approach a some of standards to requirements engineering are well defined but, in agile development haven't been Special standards for requirement engineering yet defined [26]. According to the advantages of agile RE methods have received much attention, but these practices have several challenges that continue we will review them [13].

a) Lack of documentation

In agile RE, emphasis on communication instead of documentation. Also, development teams focus on the implementation of the functionalities rather than documentation requirements. In a high-speed environment, may there are not the time for documentation [13]. Lacks documentation might cause a variety of problems [11]. If the team members don't communicate with the customer, the documentation helps us track unexpected changes [27].

b) Customer availability

Availability of customer is a basic principle of agile software development. Availability of customer will cause that both customer and development teams work together and questions of the development team can quickly answer by the customer [28]. Customer availability is one the Effective communication between the customer and team [11]. Also, Change of requirements can be manageable if determined by the customer quickly [27].

c) Project budget and time estimation

The agile methods towards RE caused that estimation of costs and schedules more difficult than with traditional methods. There isn't formal RE phase in organizations for the initial estimation of project size and typically it is based on the known user story. Accurate estimates of budget and time during the early stages of a project in agile software development is difficult Because of the nature of unstable project domain, requirement volatility, and dynamic planning and design phases [11, 13].

d) Inappropriate software architecture

The architecture chosen by the development team during the early cycles may inappropriate. Redesign of the architecture may add significantly to project cost. Rebuilding, as a continuous activity to improve the design, often doesn't completely respond to the problem of inappropriate architecture [11, 13].

e) Ignoring non-functional requirements

A main problem of RE in agile development is inadequate attention to Non-functional requirements. Customers often pay attention to core functionality and ignore issues related to scalability, maintainability, portability, safety or performance [11, 13].

f) Change of requirements and also re-evaluation

Change of requirements and re-evaluation is an important feature in agile methodology. Agile methods provide the flexibility with any changes of requirements, but it might create the issues on changes evaluation. RE-KOMBINE, which supports to shift flexibility. RE-KOMBINE, for managing the inconsistency in requirement problems [29]. There are tools for aid agile requirements engineering. Requirement changes have reduced tremendously and managing requirements become easier with using tools such as Redmine and JIRA[27, 30].

g) Conflicting Viewpoints amongst Team

One the other challenge software requirements are lacks communication between the onshore and offshore site distributed teams. When the agile team is distributed in different locations it is difficult to efficient coordination between team members [31].

7. CONCLUSIONS

Requirement Engineering is one of the important steps in software development. Engineering requirements and agile methods are complementary to each other. Correct execution of engineering requirements in the development phase can be reduced the costs in the life cycle of the software. In this research describe the activity of requirements engineering in the agile method. In this research, we studied differences of requirements engineering between the traditional and agile methods and after it, introduced techniques and practices to improve the efficiency of requirements engineering processes in agile methods. In finally have been studied challenges of requirements engineering in the agile methods.

REFERENCES

1. Ambler, S. *Agile Requirements Modeling*. 2018 2017 [cited 2018; Available from: <http://www.agilemodeling.com/essays/agileRequirements.htm>.
2. Schön, E.-M., J. Thomaschewski, and M.J. Escalona, *Agile Requirements Engineering: A systematic literature review*. Computer Standards & Interfaces, 2017. **49**: p. 79-91.
3. De Lucia, A. and A. Qusef, *Requirements engineering in agile software development*. Journal of Emerging Technologies in Web Intelligence, 2010. **2**(3): p. 212-220.
4. Shah, T. and V. Patel, *A review of requirement engineering issues and challenges in various software development methods*. International Journal of Computer Applications, 2014. **99**(15): p. 36-45.

5. Ambreen, T., et al., *Empirical research in requirements engineering: trends and opportunities*. Requirements Engineering, 2018. **23**(1): p. 63-95.
6. Sillitti, A. and G. Succi, *Requirements engineering for agile methods*, in *Engineering and Managing Software Requirements*. 2005, Springer. p. 309-326.
7. Peeters, J. *Agile security requirements engineering*. in *Symposium on Requirements Engineering for Information Security*. 2005.
8. Rantanen, E., *REQUIREMENTS ENGINEERING IN AGILE SOFTWARE PROJECTS*. 2017.
9. Batool, A., et al. *Comparative study of traditional requirement engineering and agile requirement engineering*. in *Advanced Communication Technology (ICACT), 2013 15th International Conference on*. 2013: IEEE.
10. Paetsch, F., A. Eberlein, and F. Maurer. *Requirements engineering and agile software development*. in *Enabling Technologies: Infrastructure for Collaborative Enterprises, 2003. WET ICE 2003. Proceedings. Twelfth IEEE International Workshops on*. 2003: IEEE.
11. Cao, L. and B. Ramesh, *Agile requirements engineering practices: An empirical study*. IEEE software, 2008. **25**(1).
12. Sawyer, P., I. Sommerville, and S. Viller, *Requirements process improvement through the phased introduction of good practice*. Software Process: Improvement and Practice, 1997. **3**(1): p. 19-34.
13. Ramesh, B., L. Cao, and R. Baskerville, *Agile requirements engineering practices and challenges: an empirical study*. Information Systems Journal, 2010. **20**(5): p. 449-480.
14. Daneva, M., et al., *Agile requirements prioritization in large-scale outsourced system projects: An empirical study*. Journal of systems and software, 2013. **86**(5): p. 1333-1353.
15. Alam, S., et al., *Impact and Challenges of Requirement Engineering in Agile Methodologies: A Systematic Review*. International Journal of Advanced Computer Science and Applications, 2017. **8**(4): p. 411-420.
16. Inayat, I., et al., *A systematic literature review on agile requirements engineering practices and challenges*. Computers in human behavior, 2015. **51**: p. 915-929.
17. Yang, Z., et al. *A systematic literature review of requirements modeling and analysis for self-adaptive systems*. in *International Working Conference on Requirements Engineering: Foundation for Software Quality*. 2014: Springer.
18. Boness, K. and R. Harrison. *Goal sketching: Towards agile requirements engineering*. in *Software Engineering Advances, 2007. ICSEA 2007. International Conference on*. 2007: IEEE.
19. Fowler, M. and K. Beck, *Refactoring: improving the design of existing code*. 1999: Addison-Wesley Professional.
20. Berry, D. *The inevitable pain of software development, including of extreme programming, caused by requirements volatility* University of Waterloo. in *Proceedings of the international workshop on time constrained requirements engineering*. 2002.
21. Barsalou, L.W., *Situated conceptualization*, in *Handbook of categorization in cognitive science*. 2005, Elsevier. p. 619-650.
22. Abdullah, N.N.B., et al. *Communication patterns of agile requirements engineering*. in *Proceedings of the 1st workshop on agile requirements engineering*. 2011: ACM.
23. Yu, Y. and H. Sharp. *Analysing requirements in a case study of pairing*. in *Proceedings of the 1st Workshop on Agile Requirements Engineering*. 2011: ACM.
24. Sureshchandra, K. and J. Shrinivasavadhani. *Adopting agile in distributed development*. in *Global Software Engineering, 2008. ICGSE 2008. IEEE International Conference on*. 2008: IEEE.
25. Suomalainen, T., R. Kuusela, and M. Tihinen, *Continuous planning: an important aspect of agile and lean development*. International Journal of Agile Systems and Management, 2015. **8**(2): p. 132-162.
26. Heck, P. and A. Zaidman, *A systematic literature review on quality criteria for agile requirements specifications*. Software Quality Journal, 2018. **26**(1): p. 127-160.
27. Elghariani, K. and N. Kama. *Review on Agile requirements engineering challenges*. in *Computer and Information Sciences (ICCOINS), 2016 3rd International Conference on*. 2016: IEEE.
28. Pichler, M., H. Rumetshofer, and W. Wahler. *Agile requirements engineering for a social insurance for occupational risks organization: A case study*. in *Requirements Engineering, 14th IEEE International Conference*. 2006: IEEE.

29. Ernst, N.A., et al., *Agile requirements engineering via paraconsistent reasoning*. Information Systems, 2014. **43**: p. 100-116.
30. Sarkan, H.M., T.P.S. Ahmad, and A.A. Bakar. *Using JIRA and Redmine in requirement development for agile methodology*. in *Software Engineering (MySEC), 2011 5th Malaysian Conference in*. 2011: IEEE.
31. Ganesh, N. and S. Thangasamy, *Issues identified in the software process due to barriers found during eliciting requirements on agile software projects: Insights from India*. International Journal of Computer Applications, 2011. **16**(5).

AUTHORS PROFILE



Mahdi Mousaei received the M.Sc. in Software Engineering from University of Isfahan (Khorasgan), Iran in 2018. His research interests are software engineering are Agile software development, software testing, software quality, requirement engineering, and risk management in agile software development.



Taghi Javdani Gandomani is a PhD candidate in Software Engineering at the University Putra Malaysia. His research interests in software engineering are Agile software development, Software Process Improvement, Development Methodologies and Empirical studies. He is also lecturer in Islamic Azad University, Iran and has more than 14 years industry experience in software development.