"Activities and Practices of Requirements Engineering in Agile Software Environments"

Ahmed Jasem Kahar

Master of Information Systems, University of Mosul, Iraq

ahmed.kahar@uomosul.edu.iq

Abstract:

Requirements are one of the important factors for software success. However, Requirements Engineering (RE) activities, in the Waterfall process model, are done sequentially in the analysis phase, which makes it difficult when RE practitioners think and reason about them in Agile Software Development (ASD) process model. RE practitioners need to find the established RE activities conventions in the ASD process model, especially with an increase of software companies, which transform to the ASD process model, in order to foster their transition to the new model. The objective of this study is to provide RE activities in ASD to allow RE practitioners to utilize the appropriate activity for specific ASD methodology. RE activities in ASD are handled repetitively and on a small scale, which makes them embedded in the development life cycle. This paper focuses on highlighting them in different ASD methodologies and discusses the practices that resolve the traditional Waterfall model issues. Although the ASD model has resolved some traditional RE issues, it introduced other issues, such as a consequence of trying to achieve an adequate balance between agility and stability. In addition, there is a lack of practices that target non-functional requirements.

Keywords: Requirements Engineering, Requirements Elicitation, Requirements Documentation, Requirements Validation, Agile Requirements.

1- Introduction

It is widely acknowledged in the literature that requirements are critical for the success of software projects. Requirements Engineering (RE) is an essential part of the software development process, especially in the system analysis phase. It is concerned with improving knowledge acquisition and knowledge sharing that allows a more complete understanding of the application domain constraints and stakeholder needs [1]. However, RE activities are done differently in Waterfall against Agile Software Development (ASD) process models.

RE activities in the Waterfall model are done sequentially in the analysis phase. Thus, it is difficult when RE practitioners think and reason about them in ASD; as a separated phase, especially with an increase of software companies, which transform to ASD process model [2]. On the other hand, in ASD, RE activities are integrated at the whole development process and done iteratively. We mention RE practitioners instead of system analysts, project managers, or requirements engineers, as in ASD, the multifunctional team, with one person acting in more than one role. The need for agile RE has been raised in the literature as [1] mentioned, in order to better improve RE "software engineering community will likely implement three distinct sub processes as RE is conducted: (1) improved knowledge acquisition and knowledge sharing that allows a more complete understanding of application domain constraints and stakeholder needs. (2) Greater emphasis on iteration as requirements are defined, and (3) more effective communication and coordination tools that enable all stakeholders to collaborate effectively." [1].

ASD practices have resolved some of RE challenges that were faced in the Waterfall model [3]. These challenges are (1) communication issues, (2) overscoping, (3) requirements validation, (4) requirements documentation, and (5) rare customer involvement [3]. Communication issues are solved using practices such as, face-to-

face communication, collocated teams, on-site customer or alternate customer representations, and integrated RE process in ASD. Overscoping is reduced by gradual detailing of requirements and participation of cross-functional teams. Requirements validation is met by prototyping, which helps in providing the customer with a blueprint of the product, and therefore helps in requirements validation. Requirements documentation, which is characterized by long Software Requirements Specification (SRS) documents, is reduced by face-to-face communication and user stories. User stories are precise and provide a to-the-point explanation of user needs. Finally, a rare customer involvement challenge is met by a requirements prioritization practice by the customer for all iterations which ensures that the customer goals will be met [3].

RE practitioners need to find the established RE activities in the ASD process model, in order to foster their transition to the new model. The objective of this study is to provide an RE activities mapping in ASD to allow RE practitioners to utilize the appropriate activity for specific ASD methodology. Section II introduces background and RE activities. Section III gives relevant work. Section IV illustrates the RE activities mapping in ASD methodologies. Section V gives a discussion on the impact of ASD on RE activities. Finally, section VI gives the conclusion.

2- Background

This section introduces a background for our research. In the Waterfall process model, RE activities include a set of ordered processes for capturing, gathering, documenting, and validating requirements regarding the users' needs and demands. Next, we introduce them and the goal of each one. We start with requirements elicitation, analysis, documentation, validation, and finally, management. They are well-established in the traditional Waterfall model, as have been revolutionized in

ASD model in a manner that achieves the manifesto for agile; values and principles [4].

2-1 Requirements Elicitation

It is the first activity in the RE process. Through which the requirements of a system are discovered and elaborated through consultation with stakeholders, from previous documents, and domain knowledge. During this activity, the boundary for the proposed system is defined. Also referred to in the literature as requirements acquisition. Requirements elicitation is the process through which the requirements specification is derived [5].

There are different requirements elicitation techniques, which could be organized into 7 categories. They are traditional, collaborative, prototyping, modeling, cognitive, contextual, and agile techniques [6]. Table 1 illustrates each category with its techniques.

Table 1. Requirements Elicitation techniques categorized.

Technique Category	Technique's Name
Traditional	Interviews, surveys, task analysis, and questionnaires.
Collaborative	Focus groups, workshops, and brainstorming.
Prototyping	Prototyping.
Modelling	Scenarios, goal-based approaches, business process models, and use cases.
Cognitive	Ontology, card sorting, and repertory grid.
Contextual	Ethnography and ethnomethodology.
Agile	Mind mapping, user stories, and group storytelling.

2-2 Requirements Analysis

It is the activity which is concerned with reaching a richer and more precise understanding of each requirement and representing them in multiple ways. It is used to get a better understanding of the whole business and to check if the elicited requirements are consistent, complete, and feasible. Sometimes, during these activities, the requirements can be modeled to make them clearer for the developers. It consists of analyzing the information which elicited from users to identify their task goals and classify them to functional and non-functional requirements [7]. Analysis techniques such as Joint Application Development (JAD), requirements prioritization, and modeling.

2-3 Requirements Documentation

It is the activity that results in producing the requirements specification, which is the output of the RE process. There is a wide variety of ways for expressing a requirements specification, ranging from informal natural language to more formal graphical and mathematical notations [8].

2-4 Requirements Validation

It is the activity through which possible problems in the requirements specification are detected before the specification is used for development. The validation checks if the requirements statements are valid, not contradictory, and if they satisfy the customer's needs. Test cases are used in this phase to discover the ambiguities and vagueness of written requirements. The requirements specification is validated to ensure its accuracy, consistency, and relevance.

2-5 Requirements Management

Requirements management supports all RE activities. It is an activity which is concerned with requirements versioning and control, traceability, and change. The traceability is between requirements, or between requirements and other software elements, such as features. In addition to, tracking requirement status [7].

3- Related Work

This section presents the related studies for agile RE. Schön et al. [9] introduced a Systematic Literature Review (SLR) for agile RE. The study focuses on approaches that target the stakeholder involvement in the process. The authors concluded that there is a lack of building a shared understanding of the user perspective in ASD. They identified four methodologies that were integrated into ASD to increase the understanding of user needs. These methodologies are Human-Centered Design, Design Thinking, Contextual Inquiry, and Participatory Design.

Liskin et al. [10] aim to understand how requirements artifacts are used for daily work in Kanban. They concluded that the communication with stakeholders enhanced as artifacts helps in mitigating the misunderstandings between the participants. In addition, the study mentioned collaboration challenges that arise when artifacts are too detailed. However, it is difficult to generalize their findings as their study was a qualitative study that may reflect subjective opinions.

4- RE activities in agile methodologies

RE activities, described above, have different techniques to achieve their objectives. RE practitioners used to handle them in order in the Waterfall model. However, in the ASD model RE activities have two important characteristics; they are done in a small scale and iteratively. Following is our survey of the RE activities in ASD

methodologies. First, we list the most adapted ASD methodologies in companies, according to the annual State of Agile report [2].

Secondly, we mention RE activities in each one. Figure 1 illustrates a survey results for the most adapted ASD methodologies, as mentioned Scrum is the most adapted methodology, then ScrumBan, which is the combination of Scrum and Kanban (a lean methodology), then Kanban, and eXtreme Programming (XP). XP practices partially involved in the hybrid part of the report, as well as in other methodologies, which is the reason we specify it here.

4-1 Scrum

Scrum is a wide-spread, adapted methodology, due to the fact that it is light weight and could be adapted easily. The Scrum is based on a set of values, which is achieved by a set of principles and practices. These principles and practices supply the foundation to engineering approaches for the Scrum practices implementation [11]. Figure 2. shows the practices. Requirements elicitation in Scrum, which concerned capturing domain knowledge and user needs, has been spread throughout the whole development process.

First in Product backlog preparation, when the Product Owner understands the needs and priorities of the organizational stakeholders, the customers, and the users well enough to act as their voice. Second, during sprint planning and after development during the sprint review/release review with Product Owner or the end users. The used elicitation techniques could be varied or mixed between traditional, collaborative, and agile techniques [6]. Scrum does not restrict specific techniques; however, it establishes a framework to work in.
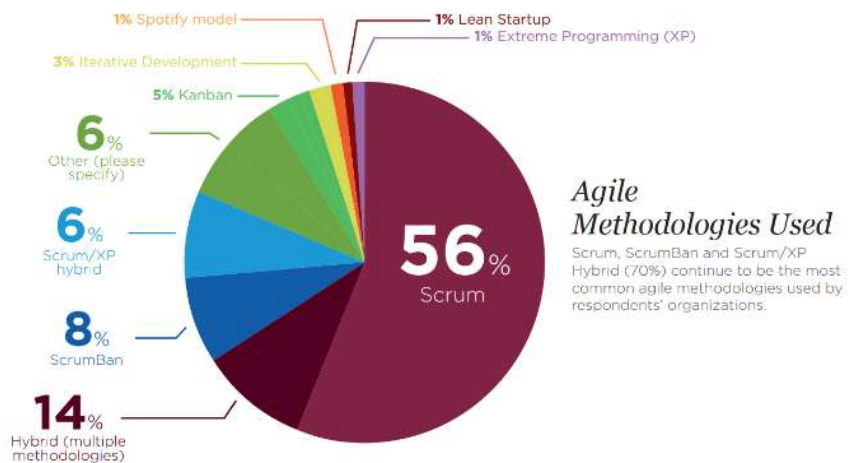
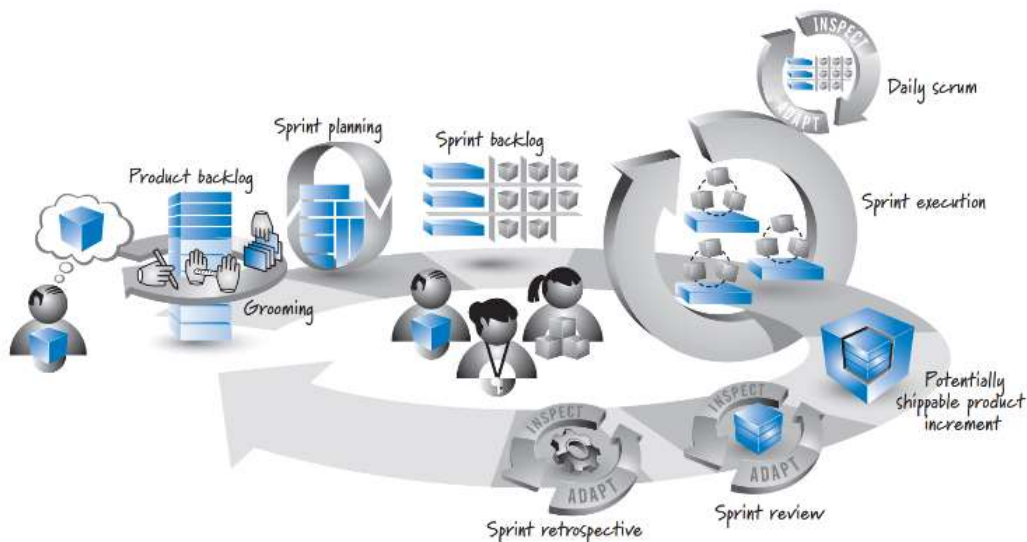Figure 1. Agile methodologies usage survey [2].

Figure 2. Scrum methodology [11].

Requirements analysis activities in Scrum could be found in Product Owner analyses the feasibility of the requirements, Backlog Refinement meeting, and Product Owner prioritizes the Product Backlog. Requirements Documentation is handled by face-to-face communication, as well as by writing user stories. Requirements Validation is done by Review meetings, which is more effective validation by working software.

Requirements Management activities will be found in iterative RE, short releases, and customer-feedback [12]. This could especially be elaborated via Product Backlog tracking (by changing the requirements (added/deleted) to/from Product Backlog), and Sprint Planning meetings. Table 2 summarizes RE implementation in Scrum [13].

4-2 Extreme Programming

XP is one of the earliest agile methodologies, which focuses on technical practices, and well-documented methodologies [14]. It aims to produce higher quality software, as well as quality of life for the development team. XP has twelve rules, namely: Planning Game, Small Releases, Metaphor, Simple Design, Tests, Refactoring, Pair Programming, Continuous Integration, Collective Ownership, Onsite Customer, 40-Hour Weeks, and Open Workspace. In addition to these practices, XP suggests a development life cycle as shown in Figure 3 [15]. XP project [15]. One important rule in XP is On-Site Customer, which recommends that the development team must have someone from the customer side [14].

Table 2. RE Implementation in Scrum.

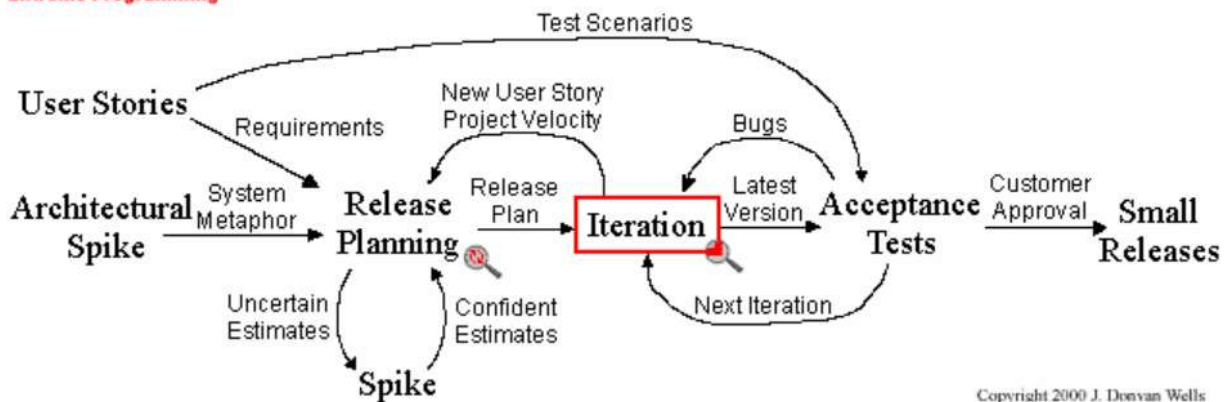| RE Activity	Scrum Implementation
Requirements Elicitation	Product Owner formulates the Product Backlog, stakeholders' participation in the Product Backlog preparing and Sprint Review meeting.
Requirements Analysis	Backlog Refinement meeting (grooming), Product Owner prioritizes the Product Backlog, Product Owner analyses the feasibility of requirements.
Requirements Documentation	Face to face communication.
Requirements Validation	Review meetings.
Requirements Management	Sprint Planning meeting, Product Backlog tracking by change requirements (added/deleted) to/from Product Backlog.

Figure 3.Extreme Programming Project [15].

Requirements elicitation and analysis in XP happen iteratively as there is a customer representative in the team, On-Site Customer. First in the Planning Game rule, where at the beginning of each iteration the User Stories' writing activity and Release Planning (see figure 3) occur. The planning starts to define, estimate and prioritize the User Stories, as a requirement elicitation artifact, for the next release [14]. Requirements Documentation realized through face-to-face communication and User Stories. User Stories in XP is for facilitation rather than documentation purposes. However, requirements management are not specified, as the User Story cards are destroyed after implementation [16].

4-3 Kanban

Kanban is an agile software development methodology based on Lean software development, which aims to minimize waste. Lean software development focuses on seven principles. They are: 1) eliminate waste, 2) amplify learning, 3) decide as late as possible, 4) deliver as fast as possible, 5) empower the team, 6) build integrity in, and 7) see the whole. Kanban emphasizes on "just-in-time" delivery [17]. It prioritizes tasks and defining workflow as well as required time to delivery [18]. The word "Kanban" is a Japanese word means visual work. Kanban has main principles: visualize workflow, limit work in progress, and measure and manage Flow [19].

Kanban board provides a visualization of the requirements' progress through the development workflow, which could be used as a requirement management tool. Figure 4 shows an example of User Stories visualized in Kanban [20]. Requirement management, especially requirement tracing was done through the Kanban board. However, there are no specific artifacts or activities for other RE activities.
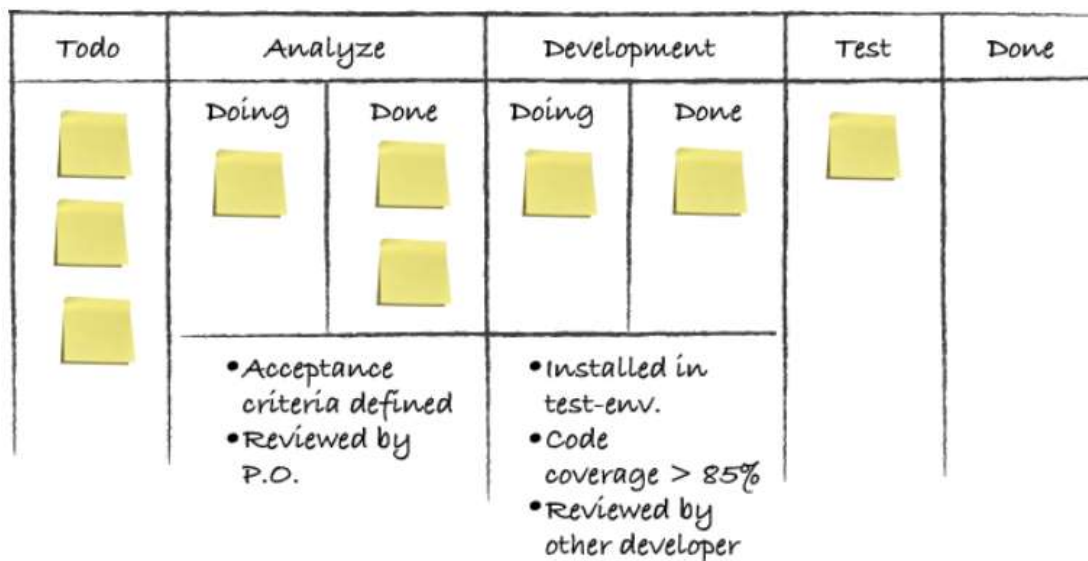
Figure 4. Kanban board visualizes the requirement's progress [20].

In spite of Kanban and Scrum are different ASD methodologies, the development teams may combine activities or techniques from Kanban into Scrum. For instance, Kanban board is used by Scrum team to track and monitor User Stories' progression. In addition, Kanban board helps in minimizing the work in progress, which allows the development team to focus on Sprint items and deliver the working software as planned in the Sprint Planning Meeting. However, the development teams acknowledge the difference between the two methodologies, such as the delivery cadence/rate. Scrum teams commit to deliver working software at the end of each Sprint, while Kanban teams deliver software continuously [21].

5- Discussion

RE activities are crucial for the software requirement development. They formalize and organize the required work to produce good requirements. However, they are

challenging, especially when RE practitioners transformed from Waterfall to ASD model. ASD practices have resolved some of RE challenges in the Waterfall model such as communication issues, and rare customer involvement. The solution of the previous challenges has been done by adapting the list of practices in different ASD methodologies.

These practices were mentioned in [3], they are:

- Face-to-face communication.
- Customer involvement.
- User stories.
- Iterative requirements.
- Requirements prioritization.
- Change management.
- Cross-functional teams.
- Prototyping.
- Testing before coding.
- Requirements modelling.
- Review meetings and acceptance tests.
- Code refactoring.
- Shared conceptualizations.
- Pairing for requirements analysis.
- Retrospectives.
- Continuous planning.

ASD methodologies are varying regards to the implementation of RE activities. Scrum methodology is the one that most covered RE activities. XP emphasis on technical practices, and there is not a clear requirements management established practices. Kanban focuses on visualizing work, as this is a good approach for requirement management and tracking.

Requirements elicitation in Scrum methodology is done during the Product Backlog preparation, or Review meeting. Product Owner collaborates with Business Analyst/Developer, to elicit and elaborate the requirements. As well during the sprint, Developer could request clarification for more details from Product Owner, as onsite customer reprehensive. While in XP, Planning Game practice, where elicitation techniques like interviews, brainstorming and prioritization are used. Kanban, as following Lean principles, postpone elicitation and analysis until the last responsible moment, i.e., before requirement development/realization to working software. Table 3 summarizes and compares requirements elicitation in agile methodologies.

Requirements analysis in Scrum could be applied during prioritizing Backlog items, and Backlog Grooming, Where Product Owner prioritizes the Product Backlog, and analyses the feasibility of requirements. While in XP, the Planning Game practice, where analysis techniques like JAD, brainstorming, and prioritization are used. Together software developers and onsite customers move the user story cards around on a large table to create a set of stories to be implemented as the first (or next) release. Table 4 summarizes and compares requirements analysis in agile methodologies.

Requirements documentation in ASD methodologies has been eliminated. Scrum, XP, and Kanban use User Stories for facilitation rather than documentation. They depend on an onsite customer and emphasis on face-to-face communication. Table 5

summarizes these points.

Table 4. Requirements elicitation in agile methodologies.

Agile Methodology	Requirements Elicitation
Scrum	During the Product Backlog preparation, or Review meeting Product Owner with collaboration with Business Analyst/Developer, they elicit the requirements. As well during the sprint, Developer could ask question to Product Owner, as onsite customer reprehensive.
XP	Planning Game practice, where elicitation techniques like interviews, brainstorming and prioritization are used. Onsite customer.
Kanban	Onsite customer. Postpone elicitation and analysis until the last responsible moment, i.e. before requirement development/realization to working software.

Table 5. Requirements analysis in agile methodologies.

Agile Methodology	Requirements Analysis
Scrum	Prioritized Backlog items, Backlog Grooming, Product Owner prioritizes the Product Backlog, and analyses the feasibility of requirements.
XP	Planning Game practice, where analysis techniques like JAD, brainstorming and prioritization are used. Together software developers and onsite customers move the user story cards around on a large table to create a set of stories to be implemented as the first (or next) release.
Kanban	Onsite customer

Requirements validation in ASD may be handled differently, as a result of applying two agile principle, "Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale" and "Working

software is the primary measure of progress [4]." Scrum validates the requirements through the Review meeting and receives immediate feedback from the Product Owner. In addition to using the acceptance criteria in the User Story to validate the requirement. While in XP, in addition to using a small release iteration, Stories are also translated into acceptance tests during the iteration. These acceptance tests are run during this iteration [15]. Table 6 summarizes requirements validation in agile methodologies.

Table 6. Requirements documentation in agile methodologies.

Agile Methodology	Requirements Documentation
Scrum	User stories, and face to face communication.
XP	User Stories in XP is for facilitation rather than documentation. Onsite customer is an on-time reference for instant elaboration or clarification.
Kanban	User stories, and face to face communication.

Requirements management in Scrum is represented in Sprint Planning meeting, Product Backlog tracking by change requirements (added/deleted) to/from Product Backlog. While in XP, Requirements management are not specified, as the User Story cards are destroyed after implementation. Kanban provides an efficient requirement tracing tool, which is Kanban board. This board visualizes the state of a User Story during the development. Table 7 summarizes requirements management in agile methodologies.

Table 7. Requirements validation in agile methodologies.

Agile Methodology	Requirements Validation
Scrum	Review meetings. Small iterations/sprints.
XP	Test Driven Development, prototyping, working software used by Onsite Customer
Kanban	Onsite Customer

Although the ASD model has resolved some traditional RE issues, it introduces other issues, as a consequence of trying to achieve an adequate balance between agility and stability. These issues are lack of practices that target non-functional requirements such as security and scalability, minimal documentation that raises traceability issues. In addition, the lack of customer availability for requirements clarification and feedback, which increases the rework, and a lack of harmony among customers [3].

Table 8. Requirements management in agile methodologies.

Agile Methodology	Requirements Management
Scrum	Sprint Planning meeting, Product Backlog tracking by change requirements (added/deleted) to/from Product Backlog.
XP	Requirements management are not specified, as the User Story cards are destroyed after implementation
Kanban	Tracing requirements via Kanban board, such figure 4 depicts.

6- Conclusion

RE activities, in ASD, have two main characteristics. First, they are handled on a small scale, which makes them embedded in the development life cycle, rather than isolated in a separate phase. Second, they are repeated frequently, under the idea of iteration. This paper presented a survey for RE activities in ASD methodologies. In order to help RE practitioners, who transform from the Waterfall model, to think and reason about RE activities in ASD, especially with an increase of software companies, which transform to ASD process model. ASD methodologies introduce various practices regards to RE activities implementation, such as face-to-face communication, customer involvement, and user story. However, there is still a need for more practices that deal with the issues that introduced by ASD methodologies. These issues are a result of finding a balance between agility and stability. Agility to deliver fast software products, as well as responding to change, and the minimal documentation that raises traceability issues. Requirement stability is required during an adequate amount of time, during development. In addition, there is a need for practices that target non-functional requirements such as security, usability, and scalability. Our future work will focus on studying how non-functional requirement elicitation is done in ASD. Our ultimate objective is to provide the RE practitioners' community with a complete guide to RE activities in ASD. This guide will include different RE activities, which handle either functional or non-functional requirements.

References

[1] R. S. Pressman, Software engineering : a practitioner's approach, 7th ed., New York: McGraw-Hill, 2010.

[2] "VersionOne 12th Annual State of Agile Report," VersionOne, 2018.

[3] I. Inayat, S. S. Salim, S. Marczak, M. Daneva and S. Shamshirband, "A Systematic Literature Review on Agile Requirements Engineering Practices and Challenges," *Computers in Human Behavior,* vol. 51, pp. 915-929, 2015.

[4] K. Beck, M. Beedle, A. v. Bennekum, A. Cockburn, W. Cunningham, M. Fowler, J. Grenning, J. Highsmith, A. Hunt, R. Jeffries, J. Kern, B. Marick, R. C. Martin, S. Mellor, K. Schwaber, J. Sutherland and D. Thomas, "Manifesto for Agile Software Development," 2001. [Online]. Available: http://agilemanifesto.org/. [Accessed 16 January 2019].

[5] E. Hull, K. Jackson and J. Dick, Requirements Engineering, 3rd ed., London: Springer-Verlag, 2011.

[6] C. Pacheco, I. García and M. Reyes, "Requirements Elicitation Techniques: A Systematic Literature Review Based on The Maturity of The Techniques," *IET Software,* vol. 12, no. 4, pp. 365-378, 2018.

[7] K. Wiegers and J. Beatty, Software Requirements, 3rd ed., Redmond, Washington: Microsoft Press, 2013.

[8] I. Sommerville, Software Engineering, 9th ed., New York: Addison-Wesley, 2011.

[9] E.-M. Schön, J. Thomaschewski and M. J. Escalona, "Agile Requirements Engineering: A Systematic Literature Review," *Computer Standards & Interfaces,* vol. 49, pp. 79-91, 2017.

[10] O. Liskin, , K. Schneider, F. Fagerholm and J. Münch, "Understanding the Role of Requirements Artifacts in Kanban," in *Proceedings Proceedings of the 7th International Workshop on Cooperative and Human Aspects of Software Engineering*, 2014.

[11] K. S. Rubin, Essential Scrum A Practical Guide to The Most Popular Agile Process, Pearson Education, Inc, 2013.

[12] N. R. Darwish and S. Megahed, "Requirements Engineering in Scrum Framework," *International Journal of Computer Applications,* vol. 149, no. 8, pp. 24-29, 2016.

[13] V. N. Vithana, "Scrum Requirements Engineering Practices and Challenges in Offshore Software Development," *International Journal of Computer Applications,* vol. 116, no. 22, pp. 43-49, 2015.

[14] M. Al-Zewairi, M. Biltawi and W. Etaiwi, "Agile Software Development Methodologies: Survey of Surveys," *Journal of Computer and Communications,* vol. 5, no. 5, pp. 74-97, 2017.

[15] D. Wells, "Extreme Programming: A gentle introduction," 8 10 2013. [Online]. Available: http://www.extremeprogramming.org/map/project.html. [Accessed 23 12 2018].

[16] N. Baruah, "Requirement Management in Agile Software Environment," in *The 2015 International Conference on Soft Computing and Software Engineering*, 2015.

[17] M. Poppendieck and T. Poppendieck, Lean Software Development: An Agile Toolkit, Addison Wesley, 2003.

[18] H. Lei, F. Ganjeizadeh, P. K. Jayachandran and P. Ozcan, "A Statistical Analysis of The Effects of Scrum And Kanban on Software Development Projects," *Robotics and Computer-Integrated Manufacturing,* vol. 43, pp. 59-67, 2017.

[19] M. O. Ahmad, D. Dennehy, K. Conboy and M. Oivo, "Kanban in Software Engineering: A Systematic Mapping Study," *The Journal of Systems and Software,* vol. 137, pp. 96-113, 2018.

[20] M. Hammarberg and J. Sundén, Kanban in Action, Shelter Island: Manning Publications Co., 2014.

[21] M. Rehkopf, "Kanban vs. Scrum," [Online]. Available: https://www.atlassian.com/agile/kanban/kanban-vs-scrum. [Accessed 09 02 2019].