A Scrum Model as a Tool for Track Changing Requirements

Fatma H. Fazzani – Dept. of Computer Science – Faculty of Education– University of Omar El–Mokhtar

Abstract:

Over the years, the importance of requirements engineering in software development and its important role in collecting and documenting all customer requirements has become clear and evident. However, with the development of requirements engineering methodologies from traditional to modern ones, such as the agile methodology, it has become difficult to ascertain the validity of the important requirements for stakeholders and whether they must be implemented during the development process without the slightest doubt that they may be modified later or re–maintained. In this paper, focus has been placed on the importance of stakeholder vision and their view of the desired system through the application of the most important models of the agile methodology for the development and technology of Scrum to reach the largest coverage of their requirements with the least dependencies in the future.

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

I. INTRODUCTION

Development software systems have an increasingly critical role in our daily life. The efficient construction and evolution of requirements engineering systems to satisfy the functionality and quality that stakeholders expect is the key to success [1]. Traditional requirement engineering consisting of some of the steps such as identification, analysis,

documentation, validity and management of requirements for the system to be developed [2]. Requirement elicitation phase, an analyst gathers requirements system with any other information from the stakeholders, and organizes them to the table of software requirement specification document [3]. After documentation of system requirements from last phase, then will be needed to verify all those requirements in the verification phase. Any changes in requirements are been by requirement management phase. Each expected change needs reviewing and estimation before it is accepted by the change control board [4]. otherwise, many challenges has been arisen by using the traditional requirements engineering methods, such as the poor communication, low level of control, cultural differences, etc., and these challenges push software companies or organizations to the transition from traditional software development methods to agile methods [5]. Agile methods is one of be the software development processes. It became popular during the last few years [4]. Agile methodologies emphasize to combine all system stakeholders into make decision about what system to be, fast coding, follow up on all changes that may occur to the collected requirements, interaction between individuals, and delivering the software within time and budget constraints. Agile requirements engineering method is an approach different from the traditional requirements engineering methods due to Agile method focuses on iterative approach as requirements engineering activities like elicitation; negotiation, documentation are carried out in small development cycles [6]. Both the methods have similar objectives but the main difference is that the traditional methods focuses more on documentation but Agile method has reduced concentricity on documentation and keeps focus on face to face communication with customers decreases the communication gaps between customers and developers. In this paper we have proposed

scrum tool one of the most useable in agile techniques by taking the regular feedback from the customer through review meetings. Additionally, the various definitions of Scrum from different papers, but according to what supports our theory in this research the scrum is a thin practice, produces potentiality of team to focus on complex challenges according to changing requirements in developing and deploying of high range projects by upgrading association, inventiveness and fertility[4,7]. Scrum is a good technique for the large as well as the small systems with the frequently changing requirements.

Many of researches presented to combine the customer in the development cycle of the system to follow up and verify the iterative changes of requirements during the development process. The authors in [8] proposed a conceptual model-driven DDRE infrastructure to increase the effectiveness of RE, improve software quality, and eventually will help to increase the trust of users in software applications. Also, the authors in [5] proposed a conceptual framework to make the requirement engineering process more effective and add to the suppleness of it by focusing on abilities, communication, roles and relationships between customers and developers. Moreover, the authors in [9] attempted of answering how scrum framework can benefit from RE techniques is introduced and showed how scrum can be ability to deal with the changing environments .

As a result, it is important to develop a mechanism that helps avoid changing requirements, poor communication, feedback, and responsiveness between the customer and the development team. This would advise the requirements engineer to propose a simple and developed agile scrum mechanism to reduce communication gaps between customers and developers and to keep track of the changing requirements

and ambiguous desires of customers. Therefore, an iterative process is required to communicate with customers at any stage of system development.

II. METHOD OVERVIEW

Requirements engineering is considered the most important stage of the development of any system because of its great role in dealing with the most important critical point, which is the change of requirements during the development process, which causes the loss of cost and effort [10]. Therefore, scrum is a collaborative agile development method has the ability to deal with changing requirements and environments that describes a group of meetings, set of tools, and roles that supports teams work together to manage system development.

In this research, we present the work of the scrum tool in a simple, developed and specific manner in terms of what are the steps used to track the most important requirements and their variables during development. The proposed method is based on the concept of scrum, which consists of activities such as roles, artifacts and activities. In our proposed scrum process, which is consists of four sprints:

- Sprint preparing
- Sprint mapping
- Daily scrum
- Sprint documenting

After the end of each sprint, we will have a set of backlogs that collect an abstract of the specifications of each sprint, and these backlogs will be reviewed and analyzed by the developer's team. Figure 1 illustrates a proposed scrum model.

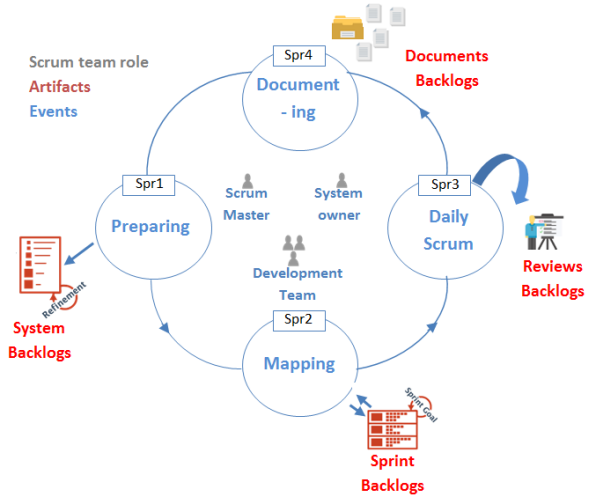


Figure 1: A Proposed Scrum Model

A. Sprint preparing

At this sprint, we will define the scrum team, the role and responsibility of each of them within the team, and we will create a simplified way to communicate with each other. The scrum team will work collaboratively throughout system development. A scrum team consists of three roles :

- Scrum master who responsible for ensuring a Scrum team is operating as effectively as possible with Scrum values. The Scrum Master's responsibilities are as following [11]:
- Facilitate daily Scrum meetings (also called "daily standups").
- Lead sprint planning meetings.
- Conduct "retrospective" reviews to see what went well and what can be improved for the following sprint.
- Keep a pulse on team members, through individual meetings or other means of communication.
- Manage obstacles that arise for the team by communicating with stakeholders outside of the team.

- System owner who ensures the Scrum team aligns with overall system goals. They understand the functionality needs of the system, system owner such as Stake holder or customer.
- Development team is composed of professionals who do the hands-on work of completing the tasks in a Scrum sprint. Maybe, consists of 7 people.

Considering, that the importance of our research resides in tracking the change of requirements during the development of the system, this sprint represents the major sprint in displaying the most important requirements that may cause a lot of confusion or lack of clarity and that can change during the system processing. In this sprint, we will use a customer journey map to ensure the validity of the needs and desires of customers and stakeholders and to know the extent of their experience with the system to be developed, It is a tool that helps the customer describe his needs and his vision of a system according to the mechanism of the system under development [12]. At the sprint we have used customer journey, it is a tool that helps scrum team understand the series of connected experiences that system owner desire and needs. Therefore, we have improved the tool to meet the main objective of this research in focusing on changing requirements, by asking a set of questions about what the system should be like in the future by classifying the questions from the highest degree gradually to the lowest, with teaching the answer of each member of the system owner and challenging his role within the team and trying to put answers to these questions in the form of actions while continuing to teach them to follow them in after and follow-up what is the emotional response to each action and its modification in the event of his dissatisfaction in order to finally reach a group of actions in the list of

system backlogs for what the system owner wants for his system in the future. Figure 2 illustrates a proposed customer journey map diagram.

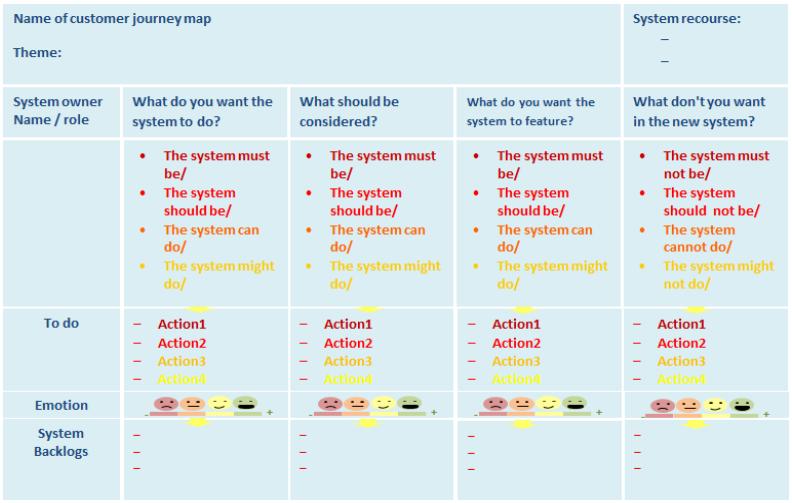


Figure 2 Proposed customer journey map diagram

B. Sprint mapping

In the sprint mapping, we will analyze the list of system backlogs that we extracted during the previous sprint which was the last product of the proposed scheme, while following the same desired content for this research which is to focus on conclusions that contain ambiguity or lack of clarity and who are the most demanding and ambiguous decision makers. In this sprint, we will use user-story map to help us determine the features and functionality required for the system, and they also include presence and confirmation of decision-makers to support all their decisions. User story mapping is a valuable tool for software development, once you understand why and how to use it [13]. A proposed customer-journey map can easily evolve into a user-story map by adding the activities, steps, and details. User-story maps are from the perspective of the system and communicate system owner' granular interactions therein. They're system

owner for alignment and tactical planning for developing and releasing features and system iterations that aim to solve the problems uncovered in the journey map. Here we will collect all the outputs of the system into a single epic for the set of activities that it wants in the system, noting that during unpacking, the actions that we have defined by lack of clarity will be misled within the epic in order to be followed up and clarified successively, each activity is prioritize based on its importance in the work of the system Figure 3 illustrates a proposed epic diagram.

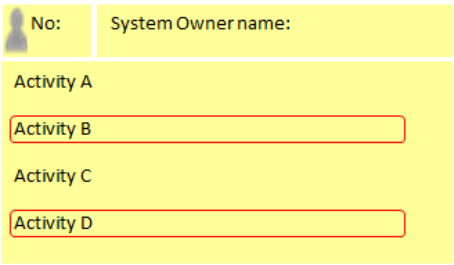


Figure 3: a proposed epic diagram

After extracting all the epics, it became easy for them to create user stories map with easier images. In user stories map, we will divide each activity into a set of steps and each step into a set of tasks. In addition, we will follow up on all the activities that were taught in the previous clip and work on them more clearly, as shown in Figure 4.

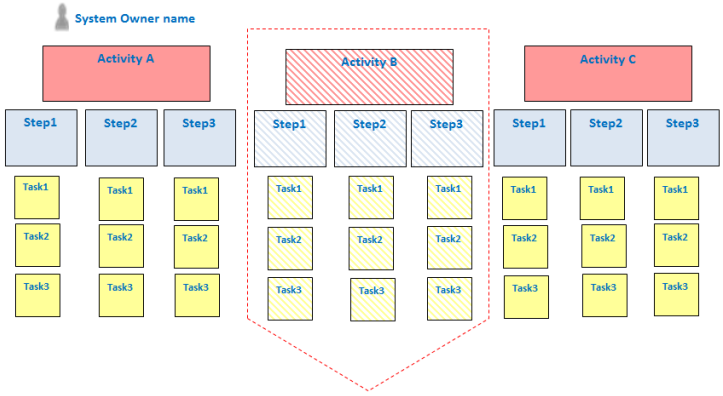


Figure 4: a proposed user stories map

C. Daily Scrum

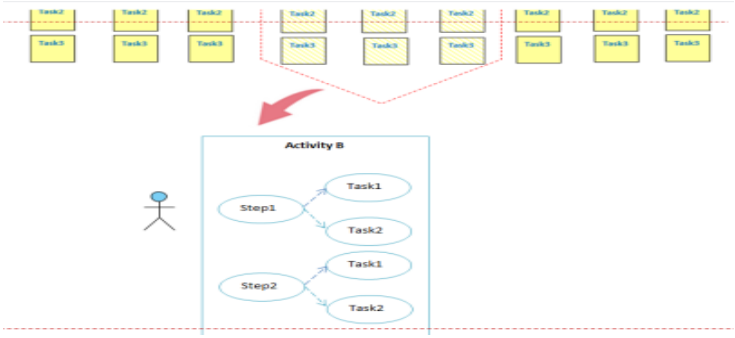
At the traditional scrum process the sprint begins its cycle and every day, the scrum master leads the team in the daily scrum meeting. This is a 15 minute meeting designed to clarify the state of the scrum. Unfortunately, all this stopping every time in the process of developing the system may cause a lot of time wasted and boredom for the owners of the system every time to review their ideas and suggestions. In this paper, we have suggested that we put the daily scrum as a fixed stage of the scrum sprints' in which all the opinions and suggestions of the product owners are represented and presented to them in a functional way, and make sure that their ambiguous and unclear needs and requirements are valid, and confirm their validity and conformity to what they want in the future. At this sprint we will use the use case diagram to ensure the required functionality of the system; use cases ate narrative description of interactions between users and a software system [14]. Daily scrum allowed everyone on the scrum team to see the status of all aspects of the system in real time. We will use the use case diagram as a prototype to present it to the owners of the system, and as we previously presented the importance of this work in its focus on clarifying the requirements that are ambiguous and subject to change, we will derive the use case from every use story map for the owners of the system, and stress on the owner who may cause some confusion or present his needs in an unclear way, as shown in Figure 5.

Figure 5: Refined use case diagram

D. Sprint Documenting

After we collected, monitored, and verified all the needs and desires of the owners of the system, and included clarification of all the obscure requirements, we now only have to document all that we extracted during the previous sprints clearly through the use of use case specification table. Such as the following table 1[15].A use case specification captures the requirements, typically of a system, in the form of a use case that contains the descriptive requirements steps in a logical sequence so that document specification can be understood by system owner to obtain sign-off of their requirements and for development team to understand what is needed by system to test and build the system functionality detailed the system use the use case.

Table1. Use case specification



Use Case Name:	
Actor(s):	
Summary Description:	
Priority:	
Post-Condition(s):	
Basic Path:	
Alternative Paths:	
Business Rules:	
Non-Functional Requirements:	

III. CONCLUSION

In this paper, we have presented a clear and simplified mechanism for the concept of scrim, according to clear steps and specific tools to highlight the vague requirements and remove misunderstandings of what the owners of the system want. We have also developed the tools that were used to suit the proposed business idea, such customer journey map diagram to extract and understand the primary needs of the system owners. And then,

to sequence the ease of focusing on clarifying the ambiguity, a new user stories map was proposed to fit the use case diagram derivation to continue verifying and ensuring clarity of requirements in their discussion with the owners of the system at the end of the meeting. As a future work, this approach needs more evaluation to improve its purpose. After that an automated tool can be built to simplify its usability.

ACKNOWLEDGMENT

We are grateful to Dr. Mohamed A. Hagal for his encourage, feedback and comments .

REFERENCES

[1] Lucia, A.D. and A. Qusef, "Requirements engineering in agile software development", Journal of emerging technologies in web intelligence, vol.2, no.3, pp.212-220, 2010.

[2] A. Sillitti, M. Ceschi, B. Russo, and G. Succi, "Managing uncertainty in requirements: a survey in documentation-driven and agile companies", Proc. 11 Th IEEE Int. Symposium on Software Metrics, pp. 10-17, 2005.

[3] Carrizo, D., Dieste, O., & Juristo,N., "Systematizing requirements elicitation technique selection". Information and Software Technology, no.56, pp.644-669,2014.

[4] Chakraborty, S., Sarker, S., & Sarker, S, "An exploration into the process of requirements elicitation: A grounded approach", Journal of the association for information systems, vol. 11, no.4, 2010.

[5] Asma Batool, Yaser Hafeez *, Sohail Asghar , Muhammad A. Abbas and Muhammad S. Hassan, "A Scrum Framework for Requirement Engineering Practices", Pakistan Academy of Sciences, vol.50, no.4, pp. 263-270,2013.

[6] Abrahamsson, P., O. Salo, J. Ronkainen & J. Warsta, “Agile Software Development Methods: Review and Analysis”, VTT Electronics, VTT Publications, p. 107-478 ,2002.

[7] K. Schwaber, “Agile project management with Scrum Redmond”, Wash.: Microsoft Press, 2004.

[8] Xavier Franch1 , Norbert Seyff2 ,Marc Oriol1 ,Samuel Fricker 2 , Iris Groher3, Michael Vierhauser3 and Manuel Wimmer3, “Towards Integrating Data-Driven Requirements Engineering into the Software Development Process”, International Working Conference on Requirements Engineering: Foundation for Software Quality, 26th International Working Conference, p.135-142, March 24–27, 2020 .

[10] Nagy Ramadan Darwish, Salwa Megahed, “Requirements Engineering in Scrum Framework”, International Journal of Computer Applications, vol.149,no.8, September 2016.

[11] Valpadasu Hema, Sravanthi Thota , S.Naresh Kumar , Ch Padmaja , C.Bala Rama Krishna, K Mahender, “Scrum: An Effective Software Development Agile Tool”. IOP Conference Series: Materials Science and Engineering, 2020.

[12] Mark S. Rosenbaum, Mauricio Losada Otalora, Germa´n Contreras Ramı´rez, “How to create a realistic customer journey map”, Business Horizons, no.8, 2016.

[13] Martin Fowler, Alan Cooper, and Marty Cagan, “user story mapping discover the whole story build the right product”, United States of America, O’Reilly Media,2014.

[14] Dusan Savic, Alberto Rodrgues Da Siliva,Sinisa Vlajic, Sasad.Lazarevic,”Use Case Specification at Different Levels of

Abstraction", 18th International Conference on The Quality of Information and Communications Technology, 2012

[15] "What is Use Case Diagram?", Visual Paradigm,2022.[online]. Available: https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-use-case-diagram/[accessed:Dec.12.2022].