Adapting Scrum Requirements Engineering Practices and studying challenges in Offshore Software Development

[1]Soumitra S Pande
[1]Release Manager, Ericsson Global India Services Pvt Ltd
BWTC, Mahadevpura, Banglore, Karnataka, India

Abstract:
Requirements Engineering is one of the hardest tasks in developing a software system. Reports state that incomplete requirements are the third most important factor for project failure. Today more and more companies are embracing agile methods to develop software. As light weight methods, agile methods have their own requirements engineering practices which are in contrast to traditional requirements engineering practices. Scrum is a widely used agile method in India, a country where a lot of offshore software development is being carried out. This study investigated the level of scrum requirements engineering practices adopted in India and the challenges introduced by these practices.
This research was conducted using qualitative methods. Respondents who play different job roles from nine organizations were interviewed in order to collect data.
Majority of the respondents successfully practices the scrum RE practices, face to face communication, requirements prioritization, iterative requirements engineering and managing requirements change. Some organizations were not comfortable with the test driven development.
The main issue with offshore development is achieving continuous customer involvement and interaction throughout the project. As agile methods rely on customer involvement more than any other development methodology, this might have a bad influence on the project.

Index Terms - Requirements engineering, scrum, challenges, offshore development.

I. INTRODUCTION

Establishing the correct set of requirements is the hardest task in building a software system. As Brooks (1987) states, resolving incorrect requirements later in the system development life cycle, is the most difficult task in software engineering [1]. Based on chaos reports on project failure, incorrect and incomplete requirements are one of the three major reasons that cause software project failure [2].
Requirements Engineering (RE) is the process of identifying, analyzing, documenting and validating the requirements of a software system [3]. In traditional software development such as in waterfall methodology, all the functional and nonfunctional requirements are identified during the RE phase. RE phase concludes by establishing the software requirements specification (SRS) which includes complete detailed descriptions of all the system requirements. In subsequent phases of the development cycle, SRS will be used as a reference and an agreement between the development team and the customer [3]. Beyond the RE phase, requirement
change requests would not be entertained in this methodology. Customer would not see the product till the end of the development life cycle.
However, in order to stay competitive in today's dynamic business environment, organizations need to be more flexible to satisfy the demanding customer requirements. Thus, software development organizations are increasingly embracing agile methodologies as they provide great flexibility to address the evolving customer requirements and rapidly market products.
Agile philosophy constitutes of a set of principles where there is more value on Individuals and interactions over processes and tools; Working software over comprehensive documentation; Customer collaboration over contract negotiation; Responding to change over following a plan [4].
Agile methods are iterative and incremental methods where the development team works in very close collaboration with the customer. Each iteration involves several meetings with the customer to acquire and validate the requirements. Work is carried out through direct communication between the team members and the customer. Less attention is given to detailed documentation and processes [4].
As agile methods concentrate on working software over comprehensive documentation, traditional RE practices are not followed in the same way [5], [6], and [7]. Thus, while some traditional RE issues are resolved by agile RE practices, they might introduce some new challenges as well.
Few empirical researches conducted in the Unites States and Europe have identified agile RE practices and the challenges introduced by them [6], [7], [8], [9]. However there is a scarce of researches conducted in the Asian countries in this area [7]. Today, most of the software development is carried out offshore, in the Asian developing countries such as India and China.

Thus, customer whose involvement is vital to the project success [10], [11], is not available onsite at all required times. This might arise unforeseen challenges as much of the wok is based on direct interactions and collaborations with the customer rather than over a written agreement.

Under the umbrella 'Agile' there is a variety of methods such as Scrum, Extreme Programming, and Kanban to name a few. Scrum is the mostly widely used agile method in India [11]. As Inayat et al (2014) states agile RE practices had only been discussed in the literature from the overall perspective of agile methods and not in the context of any particular methods such as Scrum [7]. It was suggested that more empirical research are required in agile methods, in particular Scrum and XP, in order to discuss the details from the practitioner's point of view. Thus, this study aims to address the following research questions:

1. What requirements engineering practices are used in
Scrum methodology in India?
2. What challenges are caused by the above requirement
engineering practices?

As Scrum is the most widely used agile method in India, it is important to extend the understanding of the RE practices and the issues they cause. The rest of the paper is organized as follows: The Literature Review discusses the agile RE practices and the challenges imposed by these practices based on the previous literature. Methodology discusses the research methodology that was adopted to conduct the research while an analysis of the data collected is provided under the Results and Discussion.

II LITERATURE REVIEW

II.1 Scrum Methodology

Scrum is an iterative and incremental methodology that structures the development into work cycles known as Sprints. A Sprint is a couple of weeks which is less than a month. A Sprint should start and end on the scheduled dates and will never be extended. A Scrum team involves three roles, product owner, development team and the scrum master. Product owner is responsible for expressing the functional requirements and prioritizing them in the product backlog. The team is committed to complete the items in the order of the product backlog within the Sprint. At the end of the Sprint, the developed system is demonstrated to the customer and feedback is incorporated into the next Sprint.

The development team is known as 'cross-functional' as it includes all the required expertise to deliver a potentially shippable system in a Sprint. Development team is known as 'self-organizing' as it possess high degree of autonomy and accountability.

The scrum master is more than a project manager who would support the development team and the product owner to achieve their objectives [12]. Table 1 summarizes scrum terms.

Term	Description
Sprint	Time-boxed' event of 30 days, or less, that serves as a container for the other Scrum events and activities. Sprints are done consecutively, without intermediate gaps.
Product Owner	Responsible for identifying the features to be implemented in the Sprint according got the business value. At the beginning of a Sprint, he will prioritize the features in product backlog and will continuously refine them.
Scrum Master	The role within a Scrum Team accountable for guiding, coaching, teaching and assisting a Scrum Team and its environments in a proper understanding and use of Scrum.
Product Backlog	An ordered list of the work to be done in order to create, maintain and sustain a product. Managed by the Product Owner.
Sprint Review	Time-boxed event of 4 hours, or less, to conclude the development work of a Sprint. It serves for the Scrum Team and the stakeholders to inspect the Increment of product resulting from the Sprint, assess the impact of the work performed on overall progress and update the Product backlog in order to maximize the value of the next period.

Table 1: Generic Scrum Terms [13]

II.2 Agile Requirement Engineering Practices

This section discusses the agile requirements engineering practices and challenges as identified and stated in the literature.

1. Face to face communication is the primary source of identifying requirements in agile methods. At the beginning of a Sprint, Customer or customer representative (known as product owner) sits with the development team and the main requirements that should be developed during the Sprint are identified. These requirements are recorded in the Product Backlog in the form of user stories and in story boards, index cards and paper prototypes like uses cases and dataflow diagrams. Heavy documentation of requirements is discouraged as requirements are expected to evolve and detailed planning for a long period is considered unnecessary. Minimal documentation is maintained and thus product backlog consists of the user stories defined at a high level. Frequent meetings with customer helps to identify the requirement more clearly and accurately as usually customers are unaware of the exact requirement at the beginning of the project. Customer possesses the advantage of driving the project in the direction of his business interest [6], [7].

Challenges:

The effectiveness of this practice is determined by the quality of the customer and development team interaction. In order to have a good quality communication, the customer needs to be available at the required times and a well-established trust should exist between the two parties. As Cao and Ramesh (2008) states, it is very difficult to achieve on site customer representation at all require times. Some projects have to continue with only part time access to product owner acting as the surrogate customer.

When several customer groups are present who are concerned with different aspects of the system, it is difficult to achieve consensus among the stake holders within the short Sprint. Extra effort from the team is required to negotiate and come to an agreement.

Well established customer trust is essential in agile RE.

If the customer is unfamiliar or does not believe in Agile RE, it would be difficult to get a quality feedback [6], [7], [9].

2. Customer Involvement and Interaction is known to be one of the most important reasons in the success of the project [10]. Each Sprint consists of several customer meetings such as initially to gather requirements and at the end to show the demo and acquire feedback. Thus continuous customer commitment and collaboration is essential throughout the life cycle of the system.

Challenges

Have to select the proper customer representatives to collect accurate requirements and to prioritize them. For accuracy you need to have quality interaction with customer. As requirements are highly volatile at the beginning and customer is not clear about it until he see it. Thus development is commenced without spending much time on detailing them.

One or a small number of customer involvement may not be adequate to gather requirements from different perspectives. This leads to incomplete requirements [7].

3. Iterative Requirements Engineering

In Scrum, requirements are not final and predefined but evolves in each Sprint. In each Sprint, customer meets the development team to provide with the next set of requirements to be implemented during that Sprint. An advantage of this practice is that since the customers are not clear about the exact requirements at an initial stage, when they see the evolving system they can be more clear and specific about the requirements. The development team also has the benefit of immediate access to the customer in order to understand the requirements better. Though this point is well covered in SAFe.

Challenges

The initial cost and budget calculation is carried out based on the known limited set of features. During project development, the scope can change drastically making it difficult to get approval from customer as well as the company management [6]

Product backlog maintains minimal documentation. Thus, might introduce several challenges such as inability to scale the software, induct new members and thus evolve the software over time.

Customer often focuses on the key business functionalities and tends to ignore the non-functional requirements such as performance, safety and maintainability. Non-functional requirements when identified late might be very difficult to merge into the system [8], [9].

4. Requirements prioritization.

In traditional RE, requirements are typically prioritized once. In contrast, in scrum, requirements are prioritized in every Sprint in order to meet the customer's business value. Requirements are prioritized with other tasks such as bug fixes, refactoring and incorporating changes to existing code. In traditional RE, requirements are prioritized based on two factors, business value for the customer and technical risks, cost or implementation difficulties for the developer. However, in Scrum, requirements are prioritized only based on business value for the customer [1,6]. SAFe covers this point partially in Product Developmet.

Challenges

When the system is designed with major focus on the business value, it might lead to systems which are not scalable and or stable [9].

5. Managing Requirement change

Agile methods support requirement change throughout the development cycle in order to better satisfy the customer. Agile RE practices support two types of requirement changes: adding or dropping features and changing already implemented features. At customer meetings, customer can request a major requirement change, if their expectations are not met. Frequent communication between the customer and development team avoids the need for major changes.

Challenges
Sometimes the architecture selected at the initial sprint might be inadequate as requirements change.
Thus, redesign of the architecture would add significantly to the project cost [7], [8], [9].

6. Prototyping
Prototyping is an efficient way of reviewing requirements and gain fast feedback from customer. Well understood, high prority requirements are implemented as prototypes [14].
Challenges
Sometimes the rush to market encourages the tendency to deploy these verified prototypes rather
than waiting for a robust final system. Prototypes might not be based on a proper architecture and thus introduce issues later.

7. Test-driven Development (TDD)
In this approach the test is written before the actual code. This is considered as part of requirements/design activity in which the test specifies the code's behavior [7].
Challenges
Typically, the developers are not accustomed to write tests before coding.

8. Review meetings and acceptance test for validation
At the end of every sprint, a review meeting is held with all the team members of the project where developers demonstrate the implemented features and get customer feedback. Acceptance tests are developed by the customer with the help of the QA engineers are another approach to validation and verification [7], [8].
Challenges
Since formal modeling of detailed requirements is lacking, verification of requirements might not be
addressed.

RE Activity	Scrum Implementation
Requirement Selection	Product owner formulates the Product Backlog Any stakeholders can participate in the Product Backlog
Requirements Analysis	Backlog Refinement Meeting 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 Items in Product Backlog for tracking Change requirements are added/ deleted to/from Product Backlog

Table 2 - RE Implementation in Scrum

III. RESEARCH METHODOLOGY

This study was carried out using qualitative research approach. Qualitative methods let you delve into a problem's complexity and develop rich informative conclusions [15].
This research investigates the extent to which agile Re practices are adopted in the industry and the challenges introduced by these practices. Data were collected from nine organizations that employ Scrum methodology. These organization are situated across Pune, Mumbai and Gurgaon the main business capital of India. for several years. The data were collected through interviews. The group of respondents included a variety of organizational roles such as project managers, tech leaders, senior software engineers, quality assurance engineers and business analysts. Most of the interviewees expressed their perceptions based on their years of experience engaged in multiple projects using Scrum methodology. Thus, the unit of data analysis was the individual organization. Table 3 depicts the study respondents' characteristics. Pseudonyms are used to ensure the respondents identity.
To gain insights on the extent of RE practices adopted, the collected data were compared for different organizations. The identified Scrum RE practices and their challenges are based on the respondent's perceptions and belief.

ORGANIZATION	WORK PROFILE/ROLE	CLIENT BASE	JOB POSITION	YEARS OF EXPERIENCE
ORG 1	SOFTWARE DEVELOPMENT	ONSHORE	PROJECT MANAGER	12 +
ORG 2	SOFTWARE DEVELOPMENT	ONSHORE/OFFSHORE	TECH LEAD	10+
ORG 3	SOFTWARE DEVELOPMENT	ONSHORE	SENIOR SOFTWARE ENGG.	7+
ORG 4	SOFTWARE DEVELOPMENT	ONSHORE	QA MANAGER	10+
ORG 5	SOFTWARE DEVELOPMENT	ONSHORE/OFFSHORE	BUSINESS ANALYST	4+

TABLE 3 RESPONSE ANALYSIS

Level	Face to Face Communication	Customer Involvement & Interaction	Iterative RE	Requirements and Prioritization	Change Management	Proto-typing	TDD	Reviews and Results
High	6	3	5	5	4	1	2	3
Medium	3	2	2	3	2	2	3	3
Low	0	4	2	3	1	2	2	3
None	0	0	0	3	2	3	1	0

TABLE 4 RE RESPONSE CHART

IV RESULTS AND DISCUSSIONS

Based on the data analysis, all organization are practicing face to face communication in order to collect requirements. Thus, it is observed that majority of the Indian software companies are practicing this comfortably. Iterative requirements engineering practice is also adopted by 4 organizations at a high level. Hence, requirements are evolved at each sprint. Requirements are prioritized based on the customer's business value in 5 organizations at a high level. Thus, it is observed that many organizations allow the customer to refine and prioritize the requirements in each sprint. 2 out of 4 organizations are managing the requirement change at a high level. Hence, change requests to add, drop or refining requirements are entertained throughout the system lifecycle. Reviewing the product with the stakeholders at the end of a sprint is only practiced by 3 organizations at a high level. Thus, it is observed that the customer might not be available in all cases to get the feedback and validation.

Customer involvement and interaction is not at a high level for many respondents. Many Indian organizations are servicing the foreign client base. These organizations are unable to have the customer onsite on a regular basis. Thus, many respondents reported that achieving continuous customer interaction and commitment is a challenge. This causes several practical issues as stated by a senior practitioner with over 7 years of experience in Scrum. Sometimes few team members are sent to offshore customer to gather the requirements. They discuss with the customer and convey the requirements back to the development team acting as surrogate customers. Thus, real customer requirement might not be conveyed to the development team. This condition becomes worse when technical people such as architects are sent instead of business analysts as they might fail to address the business perspective of the client requirements.

Majority of the Indian organizations are not practicing the test driven development. Only three out of nine interviewed companies stated they practice it successfully. Some of the Indian developers are not comfortable with developing tests before writing the program code. Acceptance tests are written by the customer as stated in the literature [6], [9]. However, many respondents stated that customer involvement in writing acceptance tests are challenging as they do not have the technical background.

Many respondents stated that requirements are documented in the product backlog using a tool such as JIRA. The requirements are written in a high level and hence the details are lacking. As one senior practitioner stated customer might have stated that he wants to implement a payment gateway. But the details such as how he expect it to be developed as in whether he wants to pay by credit card or through payment gateway is not expressed in the product backlog. This might create issues later in the development

when this feature needs to be traced for maintenance. As one senior respondent stated, in the Indian settings, employees tend to leave the company for various reasons. Hence regularly there will be new members joining the project. Previously estimated

schedules and workloads of an experience member would not suit these new members with less experience. Thus new members would find it challenging to complete the work assignments on time and hence or will work for long hours over night. Another issue is, with minimal documentation used, it would be difficult for the new members to trace the requirements.

Another important fact is that the customer should be well aware of agile methods and trust them. As otherwise it is difficult to achieve commitment and quality communication from the customer. The involved customer should possess the required authority in order to take necessary decisions such as in requirements prioritization.

Cost and schedule is initially decided based on the primary known requirements. If the project significantly evolves with many features, then the previous estimations would significantly change. Hence, nowadays some customers tend to have an upper limit for cost. But this is a risk for the software organization. Thus many respondents stated that it is a challenge to plan the cost and schedule for an agile project.

With requirements prioritization practice, there is a tendency to overlook the nonfunctional requirements. Therefore, few companies mentioned that they explicitly gain the nonfunctional requirements at Sprint meetings. Fail to do this, might lead to systems that have difficulties to incorporate important nonfunctional requirements such as security and performance.

When managing requirement change, developer should be able to trace the requirements. Three companies stated that they explicitly ask the developers to maintain a Requirements traceability matrix (RTM) in Excel. If the developers were too lazy or did not have time to update the RTM, it would cause issues later, when addressing the change requests.

In addressing the change requests refactoring is conducted in order to improve the design. Respondents stated that occasionally, they have faced situations where a considerable amount of code needed to be rewritten to refactor the code. Thus developer's time and effort would be wasted.

Four organizations stated that they do not implement a prototype but a releasable product at the end of each sprint. They do not encourage prototypes after the first sprint. Prototyping is an efficient way of reviewing requirements and gain fast feedback from customer. However, when developing a prototype, if the underlining design is not a good one, then products developed by scaling this prototype might be a failure. It would result in unstable, unscaled, systems. Thus, four responding organizations stated that they do not encourage the practice of prototyping. However, four organizations are adopting this practice at a medium and above level.

Table 5 describes the summarized results obtained from interviews on RE practice and the challenges they introduce. It is observed that these results are similar to the literature view [1], [6], [7], [8], [9].

Fig1: Challenges in Scrum RE Practices

V. CONCLUSIONS

This research used qualitative methods to study the level of adoption of scrum RE practices and the challenges introduced by them. Nine organizations that are situated across India, capital of India, was selected for the interviews. The respondents included

professionals with different job roles in the organization such as project manager, tech leads, senior software engineers and quality assurance engineers. All these respondents have over 4 years of experience in scrum.
Majority of the respondents successfully practices the scrum RE practices, face to face communication, requirements prioritization, iterative requirements engineering and managing requirements change.
The least practiced RE practices are Customer Involvement and Interaction, prototyping and test driven development. Compared to western countries, in the Indian context, it is somewhat difficult to achieve continuous customer interaction and commitment throughout the project. This finding is justified as for many software companies the client is an offshore client.
Further research can be conducted to gain further insights on handling traceability of requirements, nonfunctional requirements and handling offshore software development.

VI. ACKNOWLEDGEMENTS

This study was guided and followed up by Mr. Amit M Manchanda, Line Manager EGIL, Gurgaon India. I thanks him for his focused mentorship and important tips on writing this paper

VII. REFRENCES

[1] V.N Vithana, et. Al *International Journal of Computer Applications (0975 – 8887) Volume 116 – No. 22, April 2015*
[2] T. S. Group, "CHAOS Report," July 2014. [Online]. Available: http://www.projectsmart.co.uk/docs/chaos-report.pdf. [Accessed 20 January 2015].
[3] I. Sommerville, Software Engineering, vol. Ninth edition, United States of America: Pearson Education, 2011
[4] W. Cunningham, 2001. [Online]. Available: http://www.agilemanifesto.org/.
[5] E. Bjarnason, K. Wnuk and B. Regn, "A case study on benefits and side effects of agile practices in large scale requirements engineering," Agile RE Conference, 2011.
[6] Lan Cao, & Ramesh, B. (2008). Agile Requirements Engineering Practices: An Empirical Study. IEEE Software, 25(1), 60 - 67. doi: 10.1109/MS.2008.1
[7] Irum Inayat, Siti Salwah Salim, Sabrina Marczak, Maya Daneva, & Shamshirband, S. (2014). A systematic review on agile requirement engineering practices and challenges. Computers in Human Behavior. doi: http://dx.doi.org/10.1016/j.chb.2014.10.046
[8] E. Bjarnason, K. Wnuk and B. Regn, "Requirements Are Slipping Through the Gaps -A Case Study on Causes & Effects of Communication Gaps in Large-Scale Software Development," IEEE 19th International Requirements Engineering Conference, pp. 37-46, 2011.
[9] B. Ramesh, R. Baskerville and L. Cao, "Agile requirements engineering practices and challenges: An empirical study,," Information Systems Journal, vol. 20, no. 5, pp. 449-480, 2010
[10] Misra, S. C., Kumar, V., & Kumar, U. (2009). Identifying some important success factors in adopting agile software development practices. The journal of Systems and Software.
[11] Vithana, V. N., Fernando S. G. S., Kapurubandara M., Success factors for agile software development- A case study from India', International Journal of Computer Applications, Vol.113(17), March 2015
[12] Pete Deemer, Gabrielle Benefield, Craig Larman, & Vodde, B. (2010). The Scrum Primer Retrieved from http://www.goodagile.com/scrumprimer/scrumprimer.pdf
[13] ScrumGuide. (2004) Retrieved December 12th 2014, from http://www.scrumguides.org/docs/scrumguide/v1/Scrum- Guide-US.pdf#zoom=100
[14] Andrea De Lucia, & Qusef, A. (2010). Requirements Engineering in Agile Software Development. Journal of Emerging Technologies in Web Intelligence, 2(3)
[15] Uma Sekaran, & Bougie, R. (2010). Research Methods for Business -A Skill Building Approach (5th ed.). India: John Wiley and Sons, Ltd.
[16] Duncan R.(2001), "The Quality of Requirements in Extreme Programming", The Journal of Defence Software Engineering.