MS
005.1
NOS

- SCRUM (Computer Software development)
- Agile Software development.
- Backlog problems.

SCRUM Adoption: A Solution for Backlog Problems

Author

NOSHEEN KHAN

357-FBAS/MSSE/F12

Supervised by

Dr. NAVEED IKRSM

Department of Computer Science

Faculty of Basic and Applied Sciences

International Islamic University Islamabad

2015

Dedication

I dedicate my research work to the

Creator and most merciful

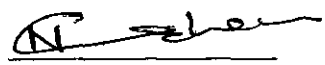
ALMIGHTY ALLAH

HOLIEST man Ever Born on Earth,

PROPHET MUHAMMAD (Peace Be Upon Him)

And to

My family and friends.



NOSHEEN KHAN

357-FBAS/MSSE/F12

A dissertation Submitted To

Department of Computer Science,

Faculty of Basic and Applied Sciences,

International Islamic University, Islamabad

As a Partial Fulfillment of the Requirement for the Award of the

Degree of

MS Software Engineering (MSSE)

Department of Computer Science & Software Engineering

International Islamic University Islamabad

Date:_____

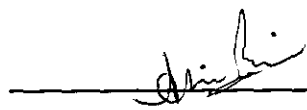
FINAL APPROVAL

This is to certify that we have read the thesis submitted by Nosheen Khan, Registration no. 357-FBAS/MSSE/F12. It is our judgment that this thesis is of sufficient standard to warrant its acceptance by International Islamic University, Islamabad for the degree of MSSE.

COMMITTEE:

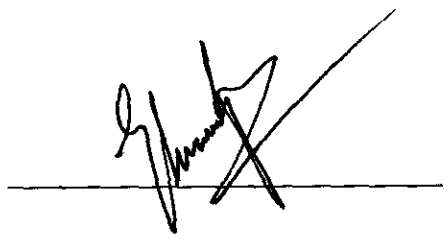
INTERNAL EXAMINER:

Muhammad Nasir
Lecturer
m.nasir@iiu.edu.pk



EXTERNAL EXAMINER:

Dr. Sheheryar Mailik
Assistant Professor
Sheheryar.malik@riu.edu.pk



INTERNAL SUPERVISOR:

Salma Imtiaz
Assistant Professor
salma.imtiaz@iiu.edu.pk

SUPERVISOR:

Dr. NaveedIlkram
Associate Dean
naveed.ikram@riu.edu.pk

Declaration

We hereby declare that this Thesis "**SCRUM Adoption: A Solution for Backlog Problems**" neither as a whole nor as a part has been copied out from any source. It is further declared that we have done this research with the accompanied report entirely on the basis of our personal efforts, under the proficient guidance of our teachers especially our supervisor Dr. **NAVEED IKRAM**. If any part of the system is proved to be copied out from any source or found to be reproduction of any project from any of the training institute or educational institutions, we shall stand by the consequences.

NOSHEEN KHAN

357-FBAS/MSSE/F12

Acknowledgements

All praises to Allah the most merciful whose blessings made it possible for me to complete this research work.

First and foremost, I offer my deepest gratitude to my supervisor, Dr. NAVEED IKRAM, the person behind this idea of the work, without his guidance and support it would not be possible to complete this research work. I am grateful to product owner and team members of the software house with whom I performed the pilot study of my questionnaires. I also like to thank all the team members who worked on the project and cooperate with me during and after the case study.

I would also like to show my gratitude to my all family members for their continuous support for the completion of my work and to my parents for their unconditional support throughout my degree. Besides, a special thanks to my friends for their support.

ABSTRACT

SCRUM is one of the most popular methods from the agile family that has gained extensive fame in the recent years. However, a company may face many problems during adoption of SCRUM framework. This research aims at identifying and collecting different reported problems about SCRUM adoption by software development firms. The focus is primarily on the most discussed one: backlog related problems, and to compile the backlog related problems. Based on suggestions and recommendations from earlier studies a novel based solution is proposed, which addresses all backlog related problems in this study. We applied the proposed solution in a small company to evaluate the effectiveness. Several tools were assessed against our evaluation criteria and TRELLO was selected to manage the project in our case study. Furthermore, training was administered to the participants to give them an idea about SCRUM methodology. Upon the completion of the project, interviews were conducted with all team members who worked on the project to find out if they found it helpful. The results from the case study indicate that solution did help the company in addressing the most of the backlog related problems. Some additions and improvements were also suggested to the proposed solution by participants.

Keywords – SCRUM adoption, agile, backlog problems, user stories

1. Table of Contents

List of Figures

List of Tables

CHAPTER 1: INTRODUCTION

1. Introduction

The working of development teams is complicated due to volatile enterprise environments and rapidly changing requirements. These factors have played a major role in the failure of several software projects [31]. To cater this issue, agile methodologies have been proposed which are based on iterative and incremental development. Agile methods promote frequent delivery of product features prioritized as per the customer needs with an aim to deliver business value in each of the iteration [15].

1.1. Agile

Agile is an approach to plan and manage projects, which depends on people, their mindset and creativity rather than process or method. It differs from other approaches in a way that it places more emphasis on change management than extensive planning [9]. Rashina et al. [15] argued that agile methods are meant to deal with dedicated, collaborative and co-located teams. According to [19], agile is gaining popularity among practitioners as compared to other development methodologies. Agile provides much simple and easier steps than other traditional approaches. It offers fast delivery, dynamic requirements, more involvement of customer and earlier testing.

1.2. SCRUM

The two most widely used agile methods are SCRUM and XP, where SCRUM deals with project management while XP's focus is on developmental practices [15]. SCRUM scores high on the popularity chart of agile development and is used by companies since the early 1990s to manage complicated product development [17]. SCRUM is a method made for small teams which includes a series of short development sprints that remain from one to four weeks [26].

1.3. SCRUM process

There are different types of roles, artifacts and meetings in SCRUM. Product owner (PO) is one of the most important roles in SCRUM, whose job is to create and prioritize product backlog items and to take all key discussion about it. SCRUM master is another role who facilitates the SCRUM process and guides the team to understand SCRUM concepts. Requirements are written in the form of user stories (US) by product owner. All user stories place in product backlog. Product owner and team decide, select and move the desired product backlog item (PBIs), based on the priority, from product backlog to sprint backlog in sprint planning meeting. In sprint planning meeting, detail estimations and discussions on upcoming sprint backlog items are done by PO and team. Sprint backlog contains items which have to be completed in next sprint. Each sprint comprises the time span of one to four weeks. Daily SCRUM meeting of fifteen minutes is conducted during the sprint phase on daily basis, where each team member briefly describes what he/she is working on. SCRUM master is responsible to conduct, supervise and to keep this meeting on track. Being self-organizing and cross-functional in nature, SCRUM team knows how to perform or implement backlog item without controlling, directing or depending on anyone [16]. Team performs development and testing in parallel throughout the sprint and a complete and workable increment came out at the end that can be used by the customer. Review meeting is held with the customer at the end of every sprint to demonstrate the product increment for the acceptance. Retrospective meeting is conducted at the end of each sprint right after the review meeting. In this meeting product owner, SCRUM master and team members discuss, what they learn during this sprint, what went well so they can apply it again in the similar situation and what went wrong so that the similar situation may be avoided in the future [18]. Effort is estimated on the daily basis by checking how much work is remaining through sprint backlog. It is also a common practice in SCRUM to add the effort for the team as a whole and update the sprint burn down chart [16].

1.4. SCRUM Adoption

As a company grows, it faces numerous challenges to develop the organization for future. It may face problems like tracking progress, meeting deadlines, recruiting talent, lack of product ownership and inability to assess much work has done [6]. SCRUM is capable to deal with these

problems. Most of the times, organizations skip important aspects while adopting SCRUM which leads them to have many issues at later stages.

SCRUM can't just apply. It implements in many ways depend on the needs of specific company or project. The experience differs on how, who and for which purpose SCRUM is used [25]. When company sets up for the adoption of SCRUM, it has to make several adjustments and is required to introduce changes at all levels of the company. These changes include resources allocation, alignment of backlogs with strategic decisions, organizational support to the teams, power shifts in decision-making and implementation of a shared decision- making process at the operational level [3]. However, sometimes a company may fail to adopt SCRUM because of problems that may cause it to give up and go back to the old approach.

The purpose of this paper is to identify problems faced during the process of SCRUM adoption in a software firm, to choose the most frequent technical problem one among identified problems, and to propose a solution for it.

1.5. Research Question

RQ1: What are the problems faced by software development companies during the SCRUM adoption?

RQ2: How most of these problems can be resolved with minimum intervention?

1.6. Motivation

SCRUM is a successful agile approach [26], hence number of software companies are adopting this approach. However, sometimes a company may fail to adopt SCRUM because of problems [1-14], [22-32], that may cause it to give up and go back to the old approach. Although, researchers [33, 38, 51] have suggested ways to address some of these problems, we were unable to find any minimalist approach to address these problems.

1.7. Research - Process

Following is the research process of this work:

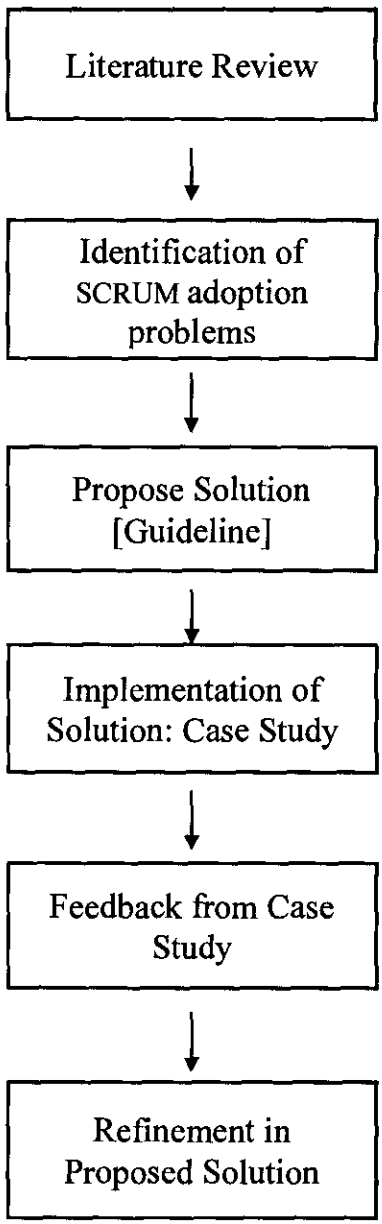


Figure 1-1: Our Research Process

1.8. Research Method

We applied the proposed solution in a small company to evaluate the effectiveness. A case study in a small company adopting the SCRUM framework was conducted. The case study was conducted from start to the end of first software development project in the company. Different types of case studies are designed by researchers, the selection of which depends on problem or work e.g. the *exploratory* type is used when research is on the way to find what will happen or to generate ideas for new research, *descriptive* to show the current status of situation, *explanatory* to see for explanation for a problem and improving to improve some aspects of the studied experience.

- Single case study was enough as it is the typical case, i.e. informs about common situations/experiences.
- We chose typical web based application with small team from industry, because agile usually is for smaller team and small projects.

1.8.1. Data Collection

The data was gathered from multiple sources which include:

- Semi-structured interviews with the developers, SM, and PO of project.
- Tool log, product backlogs, sprint backlogs and SCRUM artifacts (attachments).
- Personal notes made during the case study and observations.

- **Questionnaire for the interview**

We designed separate questionnaires, one for the product owner and the other for the team members to retrieve the information about their experience with scrum. Questionnaire designed for team members, consisted of twenty six questions and follow-ups, including questions and follow-ups, where most of the questions were about the product backlog, and sprint backlog. The questions were both open and close ended. Questionnaire of PO was comprised of twenty nine open and closed-ended questions with follow-ups and were mostly about the product backlog and prioritization. The questionnaires are given in appendix C.

- **Pilot Study**

We conducted pilot interviews with professional product owner and SCRUM developers to find out the applicability of the questionnaire in real-life settings. The product owner was from Malaysia, with three years of experience in SCRUM while the SCRUM developer was from Pakistan with seven months experience in SCRUM. The pilot interviews were conducted on Skype and were recorded audibly. We then transcribed the audio recordings into the textual form to analyze, if the questionnaires are effective enough to use it in the case study. Improvements were made in the light of answers of professionals to finalize the questionnaires.

1.8.2. Data Analysis Methods

In this research, we used qualitative data analysis method to analyze the transcribed data. The qualitative data analysis offers a deeper and high degree explanation. We used the four-stage process, with embedded thematic network analysis activities on collected data [21].

1.8.3. Research OBJECTIVE

This research aims at identifying and collecting different reported problems about SCRUM adoption by software development firms. The focus is primarily on the most discussed one: backlog related problems, and to compile the backlog related issues. Based on suggestions and recommendations from earlier studies a novel based solution is proposed, which addresses all backlog related problems in this study. Our proposed solution provides a guideline for the companies, want to get rid of their current approach in order to adopt SCRUM smoothly.

1.8.4. Research Question for the case study

RQ: How most of these problems can be resolved with minimum intervention?

1.8.5. Unit of Analysis

A project developed by a software development company adopting SCRUM methodology.

1.9. Major Contribution

We identified and collected different reported problems about SCRUM adoption by software development firms. Based on suggestions and recommendations from earlier studies a novel based solution is proposed, which addresses all backlog related problems in this study in a way that with a little intervention, the maximum numbers of problems can be solved. The results from the case study indicate that solution did help the company in addressing the most of the backlog related problems.

1.10. Thesis Outline

Chapter#2: This chapter briefly introduces SCRUM and discusses why companies opt for SCRUM at first place. For that, we went through papers from literature (best to our effort) that discuss their journey from traditional approach towards the SCRUM and highlight problems, issues and challenges that occurred during that process along with the suggestions, improvements and solutions discussed in respective papers. All the identified problems are discussed in detail to give the idea to reader about the first part of research i.e. problems in SCRUM adoption.

Chapter#3: This chapter gives details about the proposed solution that may reduce backlog related problems.

Chapter#4: This chapter presents our research methodology. It also highlights the process of implementation and gives details about case study process that was conducted to check and find if the proposed solution is feasible. The basic purpose of the case study is to show how effective the proposed solution could be for companies that are shifting towards SCRUM. We have also given a brief overview of selected "SCRUM tool" i.e. TRELLO.

Chapter#5: This chapter is about analyzing the collected data from interviews and making interpretations after conducting case study and interviews. It also contains words on how effective the proposed solution is for the companies that are planning to move towards SCRUM.

Chapter#6: This chapter is aimed at the conclusion that has been drawn from analyzing and crediting all who have made any contribution to this research work. In the end, directions for the future research are provided.

Chapter 2: RELATED WORK

2. Problems in SCRUM Adoption

SCRUM is one of the most popular agile methodologies [8], [26]. Large numbers of software companies are shifting from traditional methods to SCRUM. Literature shows that SCRUM has been adopted by even large and successful companies such as Yahoo, Microsoft, Intel and Nokia [8]. However, sometimes a company may fail to adopt SCRUM because of problems that may cause it to give up and go back to the old approach.

We reviewed different research papers that have discussed problems faced during the SCRUM adoption. We present the identified problems in the following sub-sections.

2.1 Organizational Support

The way SCRUM team works, is a challenge for SCRUM adopters because not only it requires changing in the way of working but also affects the organizational structure [10], [25], [30]. The team not getting sufficient resources and support from the organization side would a negative impact on the self-management of a team [3]. Problems related to organizational aspects that SCRUM adopters face, e.g. co-location of team members, availability of knowledge to the organization and management support are mentioned in the literature [23].

2.1.1 Lack of Human Resource

The roles of product owner (PO) and SCRUM master (SM) are critical for success in the projects using SCRUM. Consequences of not having these roles can have a serious impact on requirements understanding and acceptance of released product. It has been indicated in several studies that due to lack of resources, the individuals were assigned multiple roles [6], [23] which resulted in a conflict of interests. It is also reported that due to lack of resources, the SM was not hired [14], [24]. The loss of human resources during iterations [3] is another problem faced by SCRUM adopters.

2.1.2 Lack of SCRUM Training

SCRUM needs proper understanding and is very hard to learn by itself. Training to understand and exercise SCRUM in a right way is mainly intended for the practitioners. Several practitioners agree with the fact that they did not use SCRUM in the right way till they went through a proper SCRUM training session [28]. Lack of training is a problem highlighted in many studies [1], [2], [4], [12], [13], [22], [23], [24], [25], [28], and [30]. According to [13], around 50% team members did not know about SCRUM method and did not get any formal training.

Everyone may not be able to get the chance to attend the SCRUM training mainly because a new team member is added up to the team in the middle of a project [22]. In one such study, it was stated by practitioners that they just adopted SCRUM blindly [4], and later they realized that some aspects were missing, and the process was not being followed in the right way [4], [28]. Some of the studies also mentioned that in the absence of formal training sessions, team members felt unmotivated and were unable to meet the expectations [24]. Sometimes, they did not understand the reason behind the SCRUM concepts [22]. The team members were of the opinion that it is not easy to learn the SCRUM by themselves or with the help of other team members in a short time [28].

- **Understanding of the SCRUM ceremonies, roles and artifacts**

A formal training on SCRUM for everyone involved in the SCRUM adoption is required. Several problems were further triggered due to lack of training e.g. team members did not know the importance of SCRUM ceremonies and considered the daily SCRUM meeting as a disturbance in their work [13]. They called daily stand up meeting unnecessary [2], [11], [13], [26], and wastage of time [14], [24]. The importance of the daily SCRUM meeting is highlighted in [24] by stating that skipping it may affect the learning process among team members. In some cases, team members kept referring to SM instead of other team members during the daily SCRUM meeting that turned this meeting to a typical status meeting [8], [10]. Another problem related to SCRUM ceremonies was that there was no learning from retrospective meeting [1] or team members used to discuss and resolve the problem instead of

discussing the improvements in this meeting [8]. The planning meetings were sometimes way long to be effective [7] and much of the time was spent on PBIs specification [4]. Similarly sometimes team ended up by working on the tasks that were not discussed in the sprint planning meeting [10], [12]. It is the duty of the SM to make team realized that daily SCRUM meeting is a meeting where they have to discuss their work instead of showing their progress.

There may not be a good understanding of SCRUM roles due to lack of training. Different problems have been reported such as interference from PO or SM by asking status [13]. SM role not being taken seriously [6] and was involved in tasks such as filing reports or getting coffee for the team members [1], or not leading the planning meeting properly [8], [26]. Several studies highlighted the point that because the PO was involved in multiple projects or engaged in other duties [2, 5], team had complaint that a dedicated product owner was not available [2-3], [5-6]. In some studies, SM complained about the team that they were not taking the responsibility. They always come up with the questions like what they should do next after completing the previously assigned task. The SM wanted them to see the tasks as a pool from where they can choose their task independently [3]. The team should be taught the importance on SCRUM ceremonies, roles, and concept properly so they can follow them accordingly. One of the core concepts of SCRUM is having self-managing and self-organizing team but interference from PO and SM side or team asking PO about what next task they should do, show that they did not realize the importance of these concepts.

Another problem is the conflict between team members about updation of time spent vs. time remaining [8]. This problem is the result of another problem i.e. "not realizing the importance of SCRUM artifacts". Another problem related to SRUM artifacts is, that PB (product backlog) is not managed by PO exclusively, or the team members may not update sprint backlog and other related content time to time [2]. The burndown charts were considered less beneficial [8], [13] or not regularly updated [26]. The traceability of requirements was highlighted in several studies as a problem e.g. it was not possible to establish exact tracing among requirement and user stories [2], the requirements were getting changed through emails and were not included in the PB [13], or not having dynamic links of backlog items thus making information inconsistent [32].

- **Lack of knowledge**

  It is essential to gain enough knowledge about SCRUM to know how it is different from other traditional methods, to understand its working, roles, and concepts for both customers and practitioners. Lack of knowledge [3], [23], [28], [30] about agile methods is mainly found with fresh graduates [23] because they were taught only traditional methods during course work. One of the team members in stated that they learnt it in hard way; the team had minimal references because their SCRUM master was new so they had to learn it by themselves. Limited knowledge about agile methods may create problems like giving feedback or advice to others on how to perform a task [3] or if one has to deal with customers [23].

From the literature discussed above, we can infer that it is very important for the companies adopting SCRUM to have the support of executive management because they are the one who are sponsoring the team. The lack of support from organization can occur due to several reasons e.g. conflicting priorities within the company or because the management doesn't understand the agile methodologies. All of these factors may cause problems like availability of co-location, human Resources, proper training for team members and knowledge to the organization. Lack of human resources causes problem such as unavailability of full time dedicated PO whereas lack of training of SCRUM makes things difficult for the team because they may don't understand reason behind some concepts, leading to low motivation and failure to meet expectations.

Lack or no training in this regard further create other problems like team member not realizing the importance of SCRUM roles, ceremonies and artifacts or lack of knowledge. Without realizing the importance of SCRUM roles, ceremonies and artifacts, team fail to practice them accordingly e.g. without knowing the importance of SCRUM daily meeting they feel it as burden or interface in their work or using retrospective meeting to solve the problems. Equally important is to understand the significance of SCRUM roles such as PO or SM. SCRUM master is the one who plays very important role because he/ she assists the SCRUM process and guides team to understand and adopt the SCRUM methodology, but this role is sometimes not taken seriously and is considered as facilitator. Product owner is a person who is responsible to create and prioritize PBIs and take all key discussion about it, but several studies have indicated that that this role was not the top

priority for the person who was filling it. PO was not leading the planning meeting properly, which caused a lot of silence in those meetings. As most of the team members were not too much talkative, hence, most of the time, they experienced troubles in understanding some of the product owner requests. On the other hand lack of knowledge [23], [30] about agile methods which is a common problem with fresh graduates can be addressed by either providing proper training or putting juniors and seniors in same team so that senior practitioners can transfer their knowledge to juniors.

2.2 Human Behavior

Another important problem is the transition to self-manage teams which is related to the human behavior. It is not easy to change the culture or mindsets of people making agile adoption more formidable for organizations [10]. The reported problems were the mindset of people to learn and to change the way of working [7], [10], [12], [23], [28] self-centeredness [8], [10], and over-specialism [3], [8], [26] in the behavior of team members. We classified these problems into two categories: change resistance and non-cooperative behavior of team members.

2.2.1 Change Resistance

One of the many problems in SCRUM adoption process lies in the mindset of people who are practicing it. It is important that the mindset of the people must be willing to adopt it in the right way. Moreover, it is challenging to adopt agile practices when people are not willing to change or to learn new practices [23], [28]. Not just the mindset but also the culture cannot be changed effortlessly, which make the agile adoption more difficult for most of the organizations [10], [25], [30].

The transition to self-managing teams is also challenging because it is difficult to change suddenly the way of working for some people so everyone may not make to shift towards it [12]. According to the [7], a sudden shift from old traditional methods to entirely new technology like SCRUM resulted in making a greater impact on the development process. Such improvement in process can affect the working of the organization, making employees resist it.

2.2.2 Non Cooperative Behavior

Team members with special skills and creative mind usually think that they do not need to collaborate with other team members. According to them, it is not important to attend these SCRUM meetings because they do not need to discuss their work or problems with others, to help others or to ask for help [3], [8], [10], [26] hence causing problems in the process of the self-management of teams and having the common reference for the estimation and planning. The other stated problem with the behavior of some team members who do not bother what other member was saying in daily stand up meeting [26]. It is important to work like a team in SCRUM because without having the knowledge about each other work, team would be unable to advice or help other team member about how to implement tasks, for feedback purpose or to contribute in process of shared decision [3].

From above discussion about change resistance and behavior, we conclude that the behavior of SCRUM team members is also a bigger problem in SCRUM adoption process. SCRUM teams are self-managing and cross-functional in nature but transition to "self-managing teams" has been emerged as a problem because it is difficult to suddenly change the way of working for some people. Another problem in this regard is the people with creative mind and special skills who make it difficult for others to work in collaborative manner. Such people think they don't need to interact with other members, to attend SCRUM meetings (where they will have to discuss or share their work), to help others or to ask for help.

2.3 Others

2.3.1 Documentation

The agile process puts more emphasis on software working than the documentation as it believes in a minimum possible level of documents. It is a major problem for the practitioners from government sectors [23], upper management of company [14], 24] and quality assurance department [25] as all of these demand detailed documentation.

In Malaysia, SCRUM adopters are dealing with very high pressure for working on extensive documentation if they are making projects for the government sectors [23]. The demand of

documentation also comes from top management and bosses of the company to fulfill the requirements of ISO certificate or sometimes because they are used to with those traditional approaches which come with some level of documentation [24]. External development department responsible for the quality assurance of the projects also demands some level of documentation to either check the maturity level of project or for tracing purposes [25]. To balance these demands, SCRUM teams somehow try to provide documentation which is extra workload and burden on them.

2.3.2 User Story Estimation

To break the work into tasks and then make estimation for each task is also considered as difficult and challenging for SCRUM adopters [27]. The estimation during sprint planning meeting is done by SCRUM team. There are various types of estimations that have been reported in studies, but we are discussing here those, highlighted by most of the studies e.g. time estimations [8], [22], [27], and resources estimations [3] for the tasks and user stories.

Time estimation is the total time allocated to a particular task or user story. According to different studies, it is a challenging task for the teams to estimate accurate time for the user story as teams often make a wrong estimation for how long the story will take to implement. According to [3], only one out of four projects was benefited by SCRUM in terms of protecting resources while other were ended up with unrealistic sprint backlogs having fewer resources with too many tasks.

2.4 Backlog Related Problems

The objective of this research is to focus on the most frequent technical problems i.e. backlog related problems, which we have categorized as:

1. Right level of abstraction in the user story specification
    a) Product backlog's story specification
    b) Sprint backlog's story specification
2. Prioritization of PBIs (product backlog items)

2.4.1 Right level of abstraction in a user story specification

Many researchers have highlighted problems related to the PB and sprint backlog user story specifications. The reported problems about PB user story are that the presented features were too abstract [1], [3], generic [2], [9], ambiguous [5], not properly defined [2, 4, 13], or detailed requirements were not provided before sprint [7].

Problems related to the sprint backlog's story specification are also raised in some of the studies. The SCRUM teams in these studies were having problems such as missing business value part in the stories [2, 10-12], poor estimation of sprint backlog items resulting into incomplete tasks at the end of sprint [3, 9].

As a result of minimal definition of stories in PB, most of the planning meeting/s time was spent on defining stories [4] features identification, project scope, needs of the client, [3] making estimations and other planning because of what these meetings became too long to be effective [7].

2.4.2 Prioritization of PBIs

Problems with the prioritization of PBIs were also discussed by several researchers. The listed problems include "product owner" did not find out what was most important [3], items were not prioritized [6], [8], and priority was rarely reviewed [2].

The selection of important requirements for iterations is a vital part of the agile methodology that can be achieved by continuously prioritizing the requirements [55]. It has been identified through literature that either PO or customer has the ultimate power to prioritize the requirements as s/he is the one who can judge the value of requirements [52]. In SCRUM, the requirements are prioritized on the basis of business value which can be changed in different situations, scenarios, projects and according to the requirements of customer [59].

We have observed that requirement engineering in general and agile literature, in particular, do not provide appropriate information to identify the most suitable technique for prioritizing requirements. According to [59], agile literature provides very rough information about how to

sequence the features of the project. In [53] prioritization is tagged as vague concept via literature and stated that currently companies use informal practices for it. Some of the studies also claimed that to the best of their knowledge there is no systematic empirical research that could help to prioritize the user stories against business value [56, 58]. Also, there is a little information available about prioritization practices in agile development [54]. Zornitza et al. [59] argue that different prioritization methods are used in different situations. They conclude that mainly prioritization criteria is dependent on the context of project.

Paper Reference	Product/Sprint Backlog										
	PBI Specification	SBI Specification	Prioritization	Lack Of Human Resources	Lack Of Training	Understanding events/roles/ artifacts	Lack Of Knowledge	Change Resistance	Non Cooperative People	Documentation	Estimations
[1]	✓				✓	✓					
[2]	✓	✓	✓		✓	✓					
[3]	✓	✓	✓	✓		✓	✓		✓		✓
[4]	✓				✓						
[5]	✓					✓					
[6]			✓	✓		✓					
[7]	✓							✓			
[8]			✓			✓			✓		✓
[9]	✓	✓									
[10]		✓				✓		✓	✓		
[11]		✓				✓					
[12]	✓	✓			✓			✓			
[13]					✓	✓					
[14]				✓		✓				✓	
[22]					✓						✓
[23]					✓	✓	✓	✓		✓	
[24]				✓	✓	✓				✓	
[25]					✓			✓		✓	
[26]						✓			✓		
[27]											✓
[28]					✓		✓	✓			
[30]					✓		✓	✓			
[31]											
[32]						✓					

***Table 2-1:** SCRUM Adoption Problems*

The top row of table 2-1 which is dark gray in color is the category of the major problems and the other with light gray color, category of the secondary problems that occur due to the influence of major problems. As columns of those problems which is the center of focus of this study is filled with light blue color to make it prominent.

2.5 Literature Analysis

The review of the literature (as given in table 2-1) shows that many researchers have identified backlog related problems with SCRUM adoption. Some of researchers have presented solutions for reducing or addressing these problems. We found that some of the problems were the root problems while others were the secondary problems, caused by the root problems. The root or major problems are related to the product backlog, human behavior, organizational support and remaining are placed under other category. Secondary problems are related to the ceremonies, roles, concepts, estimations, user stories, and documentation, etc.

The first major problem is related to the organizational support or upper management not being supportive enough to provide the required resources or training to adopt and implement SCRUM efficiently. If there is a lack of support from organizational side, teams may suffer from lack of human resources (e.g. not having SCRUM master). Lack of training may cause the practitioners; fail to understand the reason behind some concepts or importance of the SCRUM roles, artifacts, and ceremonies. It makes them unable to exercise the SCRUM methodology in a correct way. Ignorance of SCRUM roles and ceremonies will affect the development of a stable product, sprint backlog, and the prioritization process.

The human behavior problem also absurdly affects backlog problems, as a sudden change in the "way of working" causes team members to resist it. This change leads them not to follow all rules and concept of SCRUM, which may end up in producing not well organized sprint backlog. Lack of documentation causes problems like tracing or inconsistency in PBIs. Estimations also affect PBIs as correct estimations help PBIs in the prioritization process. All these problems can somehow be addressed by providing proper training. Also, training of a SCRUM adopters will help them to maintain the right level of documentation and correct estimation.

Although there are a lot of problems, the focus of this research is on backlog related problems. It is the only technical problem that affects all other major problems. The inherited problems of product backlog are "understanding of user story", and "prioritizing user stories". All other problems of the major category are non-technical for which no formal solution can be devised or the solution is obvious. Backlog problems about SCRUM adoption process are reported by many researchers [1-13] and some of them [2], [4-8], [13] have presented suggestions, recommendations for addressing these problems. However, no effort has been made to address these problems as a whole. Therefore, we need a solution that would address these backlog problems as a whole and help the software companies to reduce or overcome same problems during the SCRUM adoption.

Chapter 3: PROPOSED SOLUTION

3. Proposed Solution for Backlog Related Problems

3.1 Introduction

In previous chapter, we discussed problems related to backlog [1- 5] that include "right level of abstraction in user story specification, and prioritization of user stories". We compiled these technical problems, based on which we proposed a single solution that hopefully will solve these backlog problems or the sub-categories of backlog problems as a whole. Furthermore, our proposed solution is applied in a project that was developed in a software house which recently adopted SCRUM. This chapter deals with the proposed solution.

3.2 Proposed Solution

A solution in the form of guideline has been proposed by composing respective reported solutions and improvements from literature in to a whole. The guideline consists of two parts and is detailed in the following sub section.

3.2.1 Guideline 1: Ensuring Right Level of Abstraction in User Story Specification

In SCRUM, requirements are written in the form of user stories. Firstly PO captures requirements from customers and writes them in pre-defined format along with their rough estimations, and acceptance criteria in the product backlog. Then, in sprint planning meeting team moves user stories having highest priority from PB to SB and further elaboration of selected user stories is done with time.

A right level of detail for the US (user stories) is important to minimize the chance of skipping any important information. A story should be independent, negotiable, valuable, estimate-able, small, and testable [34].

- **Product Backlog's Story Specification**

The user story specification is reported as a problem in the literature [1-5, 7, 9, 13]. The user story is a simple method to elicit requirements from customer or users. User stories are

created using the pre-agreed user story template/format [35]. The widely used and standard template for writing user story in PB is shown below:

As a <role>, I want to <action><object>, so that <business value> [33-41].

Where:

"**Role**" – signifies the person or system that will perform the action or one who will be benefited from the activity.

"**Activity**" – the action that will be performed by product or system

"**Business Value**"– shows the importance or market value and the reason [38].

An **acceptance criterion** is the primary item needed for the testing so it should be considered important. If a user story passes the test, it is considered as complete and is taken out of the list. It is the job of the product owner to approve the acceptance criteria and then tester decides whether the story passes the acceptance criteria or not [33].

Story Points /size: Complexity is estimated and shown via story point of the user story [44], [49].

- **Sprint Backlog's Story Specification**

Problems related to the sprint backlog's story specification are also raised in some studies. These problems are that the team was not being able, to define a stable sprint backlog [10], converting the features from sprint backlog to actual work [2] or unrealistic sprint backlog [3].
To gain better understanding and to minimize the level of abstraction, one needs to have proper and complete information about the US. For this purpose, we went through the literature and collected all possible (compulsory and optional) information that could help the user in US elaboration. All the compulsory and optional attributes or information are placed in Table 3-1.

Sr. no.	Attributes	Explanation
1.	Project name [46]	Title of the project
2.	Card no./ story no. or ID [43-45], [47], [49]	Unique identifier or number for the US.
3.	Story name or title [44-46]	Name of the story should tell the content requirements.
4.	Date [41], [44], [45-46], [49]	The date on which the requirement was created.
5.	Due date [41]	It is necessary to know that the requirements are not skipped, ignored or overlooked.
6.	Description[42-43], [45-47]	The requirements should not be less than 5 sentences. They should be comprehensive enough to explain the central meaning of it.
7.	Tasks /subtasks [41]	Different tasks against each user story related to database, interface and development.
8.	State/Status [41], [43], [47], [49]	The status of US or requirement reveals current state of it e.g. Drafted, rejected, incompletely specified, planned, refined, started, done etc. If US got rejected then the reason for rejection should be provided.
9.	Priority [41], [47], [49]	Set the priority as important, medium etc.
11.	Time Estimation [44-47]	Time required to complete the story.
13.	Requirement Owner/Concerned person/ Assignee [41], [44-45]	Link of the owner of the requirement, the person who is answerable for it.
14.	Release Identifier[43], [47]	To assign the US to some specific release.
15.	Sprint Identifier[43], [47], [49]	To assign the US to some specific sprint.
16.	Dependency/ Relation [48]	Link to other requirement which has same scale of abstraction.
17.	Test Case ID[42], [45], [46]	The unique identifier for test case.

Table 3-1: Attributes For the Specification of Sprint Backlog User Story

3.2.2 Guideline 2: Prioritization of PBIs

The listed problems are that "product owner" did not find out what was most important [3], items were not in the priority order [6], [8], and priority was rarely reviewed [22].

Prioritization Techniques	Early Stages	MoSCoW
	Mid Stages	One Hundred Dollar Simple ranking

Table 3-2: Prioritization Techniques

A number of techniques for the prioritization in agile are discussed in the literature; however, we needed to have the context dependent prioritization technique for product backlog. Two studies were found helpful in this regard, one study [51] debated at what stage, which technique should be more appropriate and helpful, and other [50] discussed prioritization techniques for small and large scale of requirements.

Requirements have been classified in three stages in [51], but we will discuss only two stages as per our requirement. The early stage of "requirements and analysis" is the point where requirements are not available in depth, and the customer has only a generic idea about the project. At this stage, requirements are relatively large in number so, MoSCoW is recommended as a suitable prioritization technique at this stage. MoSCoW is an appropriate technique for both small as well as large number of requirements.

At the medium stages, requirements are clearer and client, as well as the team, gain more understanding of the system. At this stage, clients stop adding further requirements and requirements already elicited get more specific. The recommended agile prioritization techniques for this stage are "simple ranking" and "one hundred dollars".

Thomas et al. [50] divided agile prioritization techniques into two categories based on the review work done by Racheva et al. [57]. Small scale agile prioritization techniques such as the $100 allocation technique, the quality functional deployment approach, the weighted criteria analysis, the dot voting, technique, the multi-voting system, the round-the-group prioritization, and the pair-

wise analysis. These techniques are suitable for small number of requirements and are easy, simple in use, and can be used without the support of any tool.

Medium or large scale agile prioritization techniques include MoSCoW technique, wiegers's matrix approach, the planning game, the binary priority list. These techniques deal with relatively larger scale of requirements making them more appropriate for large number of requirements. The algorithms used in these techniques are complex because of its capability of supporting high numbers of requirements. These tools use with the help of a tool.

3.3 Overview of the solution

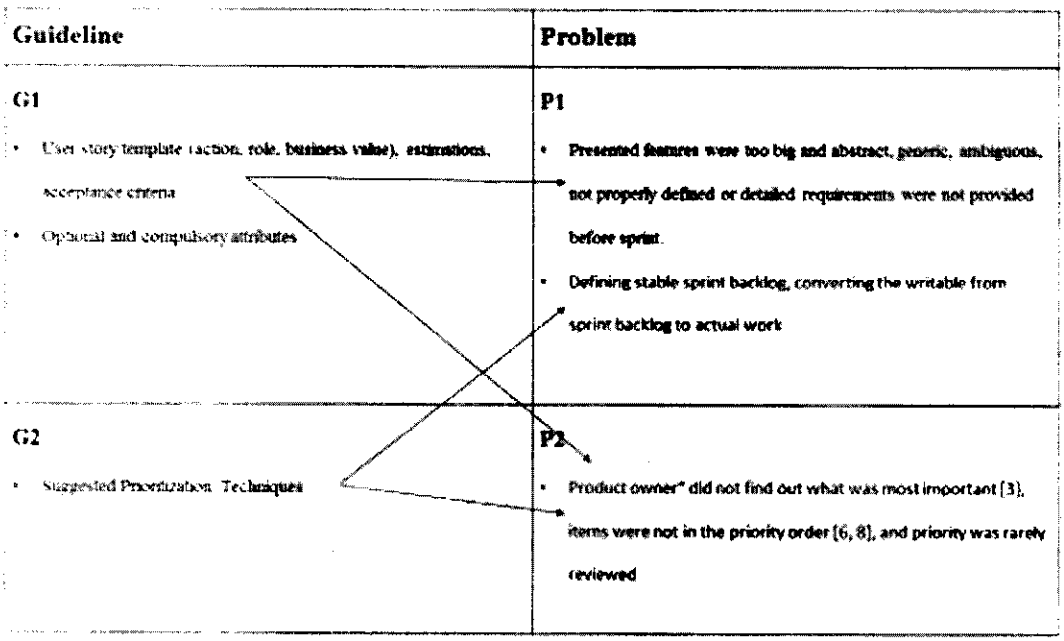


Table 3-3: Mapping Between Problems and Guidelines

We present the mapping between guidelines and problems in the above table 3-3, to give the overview of the solution. The first guideline which is a template for PB user story and attributes for sprint backlog user story. It would not only reduce the abstractness of user story but also help PO in prioritization process as the third part of PB user story template requires to set the business value. Also one of the attributes of sprint backlog user story needs to have the value for estimations

of the user story. Second guideline the prioritization techniques, will be helpful for writing value added user story. These guidelines can be beneficial not only for backlog related problems but also for other problems such as documentation, and estimation.

Chapter 4: IMPLEMENTATION

4. Implementation of solution

The problems related to the SCRUM adoption have discussed, and recommendations have been proposed by several researchers but there was a need to make a solution which will hopefully solve maximum problems with minimum intervention. This chapter provides the implementation details of our proposed solution through an industrial case study [60]. The details of the case study are discussed in the following sub sections.

4.1 Selection of the Case/Company

A case study in a small company adopting the SCRUM framework was conducted. The case study was conducted from start to the end of first software development project in the company using SCRUM. The project started in the first week of July 2014 and completed in the last week of August 2014.

The first activity was the selection of appropriate case. We chose a typical web based application with a small team from industry as agile is suitable for small teams. It is a web based application, being developed in a local software house located in Islamabad. It provides services in the area of medical and health related software products. It also offers customized training in a range of areas including monitoring and evaluation. The company under study was introducing SCRUM methodology; hence it was best and most appropriate for our case study.

4.2 Introduction to the Project

LCMS is Legal Case Management System, it works as a bridge between lawyers and their client. The lawyers don't need to contact their client about every little detail or the clients need to call their lawyers about update or questions, the lawyers can simply update the case information on LCMS and the clients can view it there. It also provides a bridge between lawyers, lawyers can assign other lawyers tasks, inform them about a meeting by leaving a note for them, include them in a meeting and a notification will be sent to them and also they can include them in a matter

(case) or a legal team among other things. LCMS doesn't only help lawyers connect with their client or other lawyers it also help them manage their work more efficiently. The lawyers can manage their matters (cases), agreements, notes, meetings, tasks, clients, court orders, legal teams, opinions, contacts, knowledge and records. A lawyer or client receive notifications if any change is made in a matter or if someone leaves a note for them or invite them for a meeting. Lawyers and clients can manage their own account change names, passwords, profile picture, email etc.

4.3 Team Composition

The criterion for the sample section was that the team should be new to SCRUM methodology. There were five developers, a product owner, and a SCRUM master to develop software project.

4.4 SCRUM Tool Evaluation

The company wanted us to suggest a SCRUM tool that could accommodate at least seven team members, cover most of the SCRUM activities and should be free of cost. The list of the evaluated tools was provided to the company to facilitate them so they can select the best one. Three tools were referred to the company as best and finally they went for TRELLO due to the limitations of other tools. Appendix B, contains a table having comparison of different SCRUM tools.

4.4.1 TRELLO

TRELLO is a free SCRUM tool that allows at least 10 free users to use it. It allows users to perform SCRUM activities i.e. create, plan and prioritize user stories. It offers numbers of boards for different projects where the users can create PB, SP, and other lists.

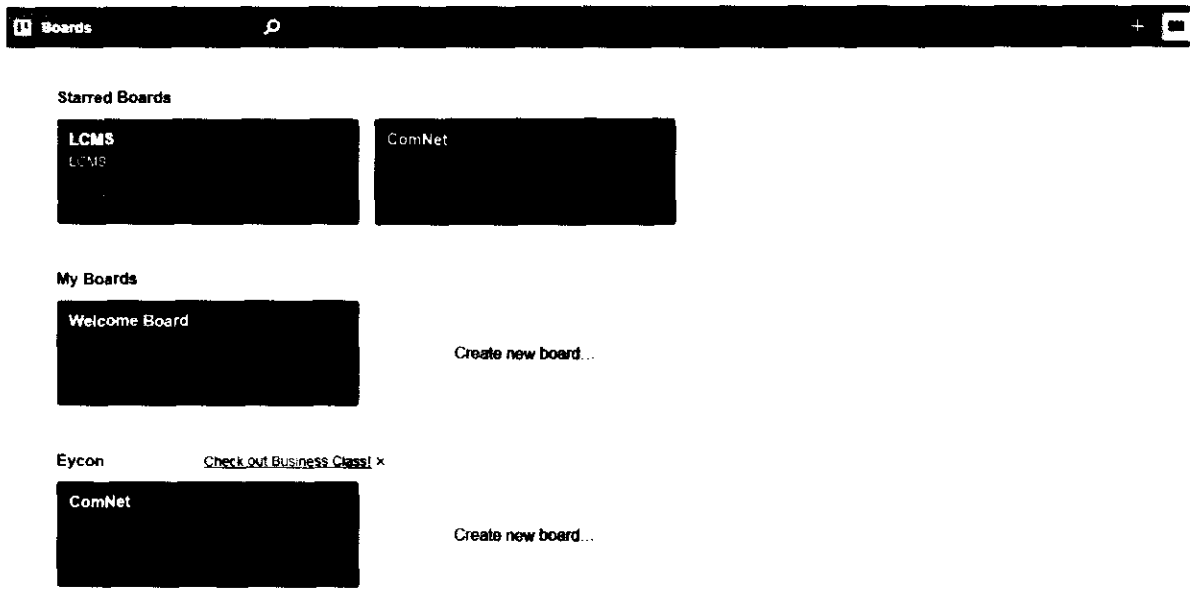


Figure 4-1: TRELLO User Interface

These lists can be named anything. Users can keep their daily sprint items in these lists which they have to implement on that particular day as per their convenience. Items in PB have options like "give business value" and then items get sorted accordingly. The Users can drag and drop the item from PB to SP or they can move it to done or whichever list they want.

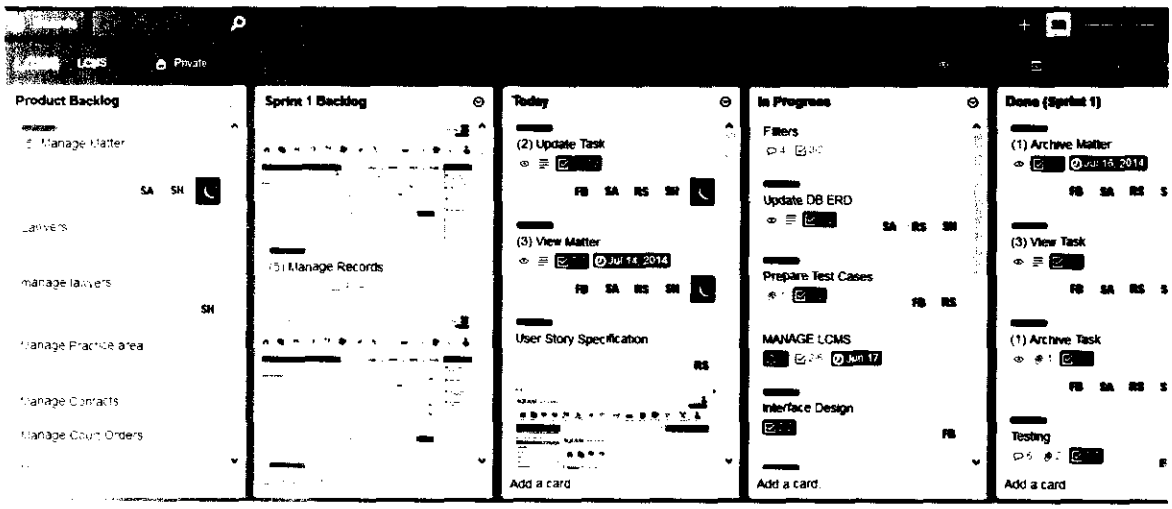


Figure 4-2: Different Lists Created Using TRELLO

For user story, users can enter the description of the user story; assign a member to it, mention due date, and can create checklist and labels for it. There are also other options like, create tasks for US, write acceptance criteria for US and user can attach related artifact to the user story too. Team members can vote the US as "important or "less important". Each activity performed by user e.g. "moving US card from the sprint backlog to done list" is noted in the activity area and is shown to all team members. The limitation of this tool is that everything is manually done from assigning business value, priority to creating PB and other lists.

4.5 Training

The seven member SCRUM team was introduced to SCRUM by providing them a training of three days, to give the idea of rules and concepts of SCRUM methodology. There were five developers, a product owner, and a SCRUM master to develop software project. The training helped them to understand how to conduct the activities of SCRUM e.g. SCRUM sprint planning meeting, daily SCRUM meeting, SCRUM sprint review meeting, SCRUM retrospective meeting and how to create and manage the product and sprint backlog. There were three parts of each session: the presentation, group activity, and quiz. In the presentation session, the activity e.g. "SCRUM sprint planning meeting" was presented in detail with examples, in the activity part all members were asked to perform that activity and in quiz a part a test was conducted.

4.6 Data Sources

We collected the data from multiple sources which include interviews with multiple team members and PO, participant's observation, artifacts in different point of time, tool log and personal noted made during case study. We had the access of tool and backlogs were observed thoroughly time to time. We attended all of the SCRUM planning, daily, review and retrospective meetings. At the end of the project, we conducted interviews session with all the participants of the project.

The data was collected from:

4.6.1 Observations/Meetings

As stated earlier, we attended all of the SCRUM meeting to observe the behavior of participants during SCRUM planning meeting, daily SCRUM meeting and review meeting. We used to note down the meeting minutes, observations if the meeting is going on track and if all the required rules were being followed.

4.6.2 Artifacts/ Log of the Tool

We thoroughly studied and analyzed all the artifacts created or uploaded on the tool e.g. product backlogs, sprint backlog, use case specifications and activities of participants from the tool log during and after the case study.

4.6.3 Interviews

We conducted interview session with all of the participants to know their experience and their take on proposed solution. The interviews were semi-structured in nature that included both open and close ended questions.

4.7 Activities of project

We conducted a case study [60] which covers all core activities of SCRUM methodology. The project was developed for a law firm, and the case management system handled different activities related to the lawyers such as cases, and court appointments.

There were three sprints in this project. We attended all the meetings to observe the work, and listen to the discussions. We had complete access to the tool, as a result, we had a close tab on, what was going on. We were monitoring the movements of cards from PB to SB, work assignment to team member and the attached documents. The data thus came from the interviews; personal notes made during the observation, tool log, and artifacts.

We covered all of the SCRUM activities and reviewed artifacts. We checked if the proposed solution was assisting the process and was helpful in addressing the backlog related problems.

During the project, the SM made sure that the solution was applied properly. Further detail about the activities of the project is discussed below.

4.7.1 Product Backlog Creation

PO came with all requirements hence product backlog was created and PO started adding items in it using TRELLO. The added PBIs were "Manage Matter, Manage Task, Agreements, Notes, Meetings, Tasks, Clients, Court Orders, Legal Teams, Opinions, Contacts, Knowledge, Records, Main Menu System, Interface Design for Login Screen, System Theme and Database Design of Entire System".

4.7.2 Product Backlog Refinement Meeting

The 1st meeting was refinement meeting (estimation meeting) where PO and the team along with SM discussed the PBIs with the highest priority. The PBIs like "Manage Matter, Manage Task" were split in small items such as "Create Matter, Delete Matter, Update Matter, View Matter, Archived Matter, Create Task, Delete Task, Update Task, View Task and Archived Task" in this meeting.

PO then wrote user story in the form of provided format i.e. As <Role> I want <Feature> so that <Business value>. The acceptance criteria in the form of checklist, and story points were given for each of these PBIs.

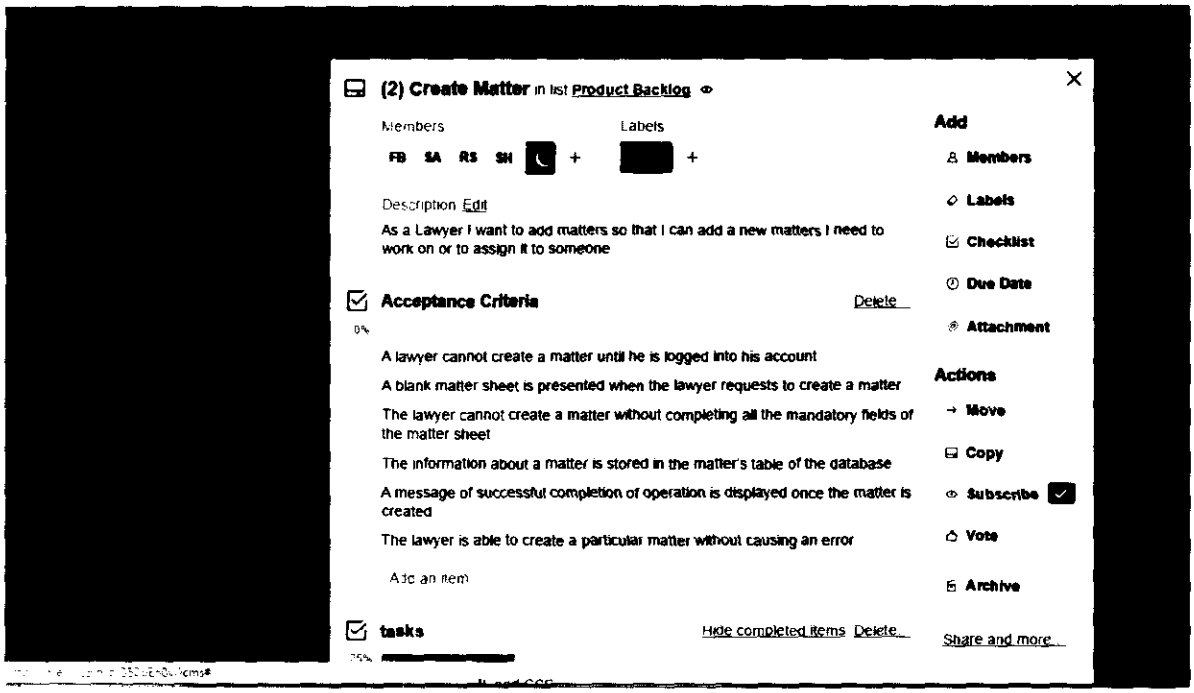


Figure 4-3: Acceptance Criteria for User Story

SPRINT

The duration of first sprint was two weeks.

4.7.3 Sprint Planning Meeting 1

Total three "sprint planning meetings" were conducted. At the start of every sprint, the team used to conduct "sprint planning meeting" where the entire team along PO have to decide which items from PB should move to SB for next sprint.

In the first sprint planning meeting, user stories with the highest priority were moved from PB to SB. After moving these user stories to SB, each user story was given the details, tasks, and due date. "Manage Matter, Manage Task, Main Menu System, Interface Design for Login Screen, System Theme and Database Design of Entire System" were chosen to move from PB to SB.

At regular intervals, the document of use case specification, screenshots of database design, and interface design were added to each user story. In the end, test cases were also written and performed on user stories. The test cases were attached to the tool in the form of MS Word document.

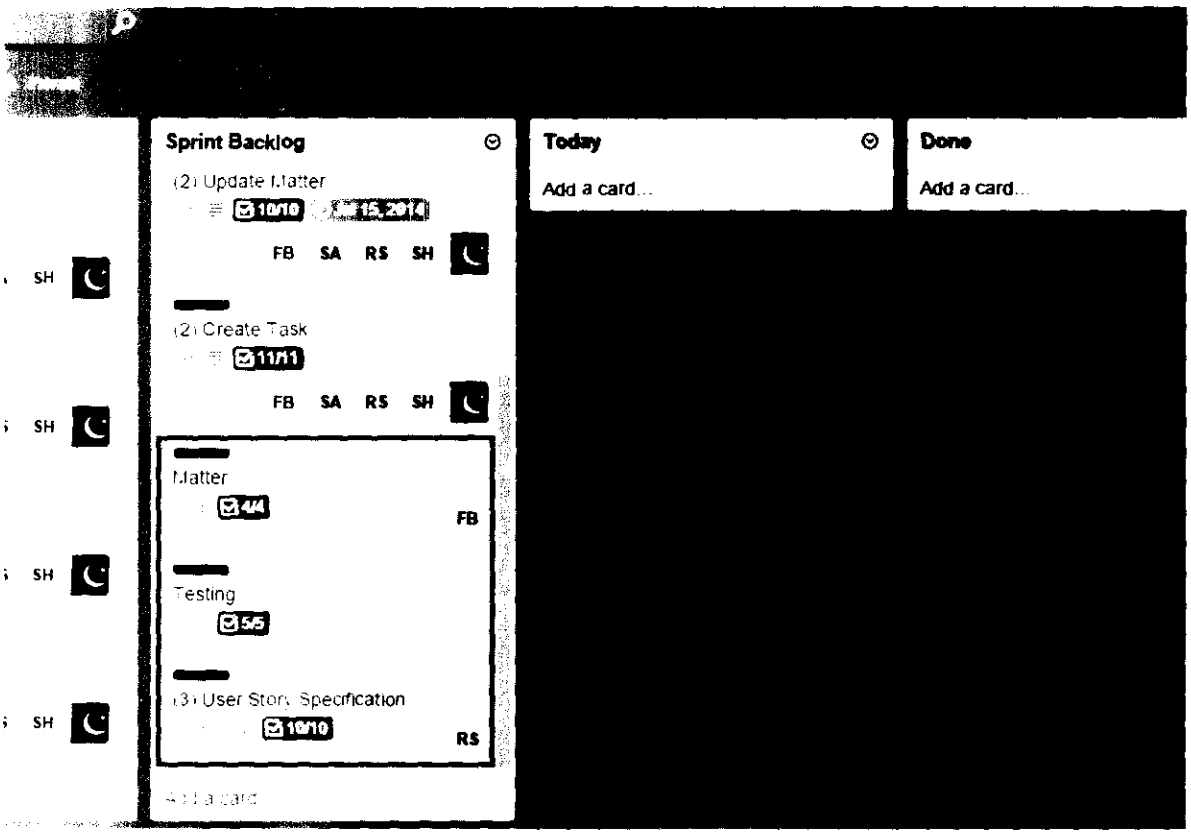


Figure 4-4: Interface, Test Cases and User Story Specification For User Story

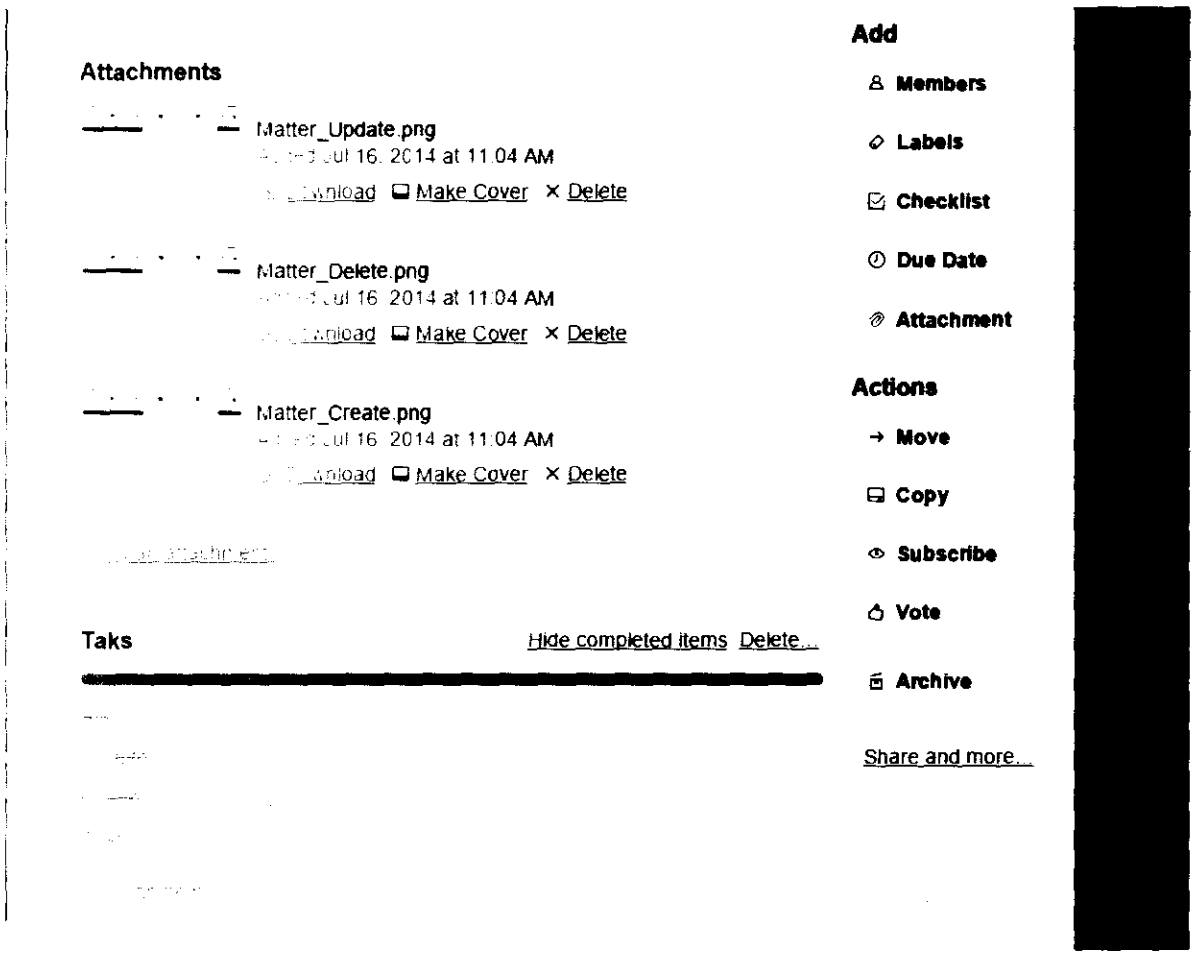


Figure 4-5: Attached Screen Shots of the Interface

4.7.4 Daily SCRUM Meetings

Every day, PO conducted and supervised this meeting where each team member announced what tasks s/he has to perform on that day. Two lists were created namely "today", and "done" in TRELLO. After briefly describing what item/s they will work on that particular day they had to move that item from the sprint backlog to "today" list.

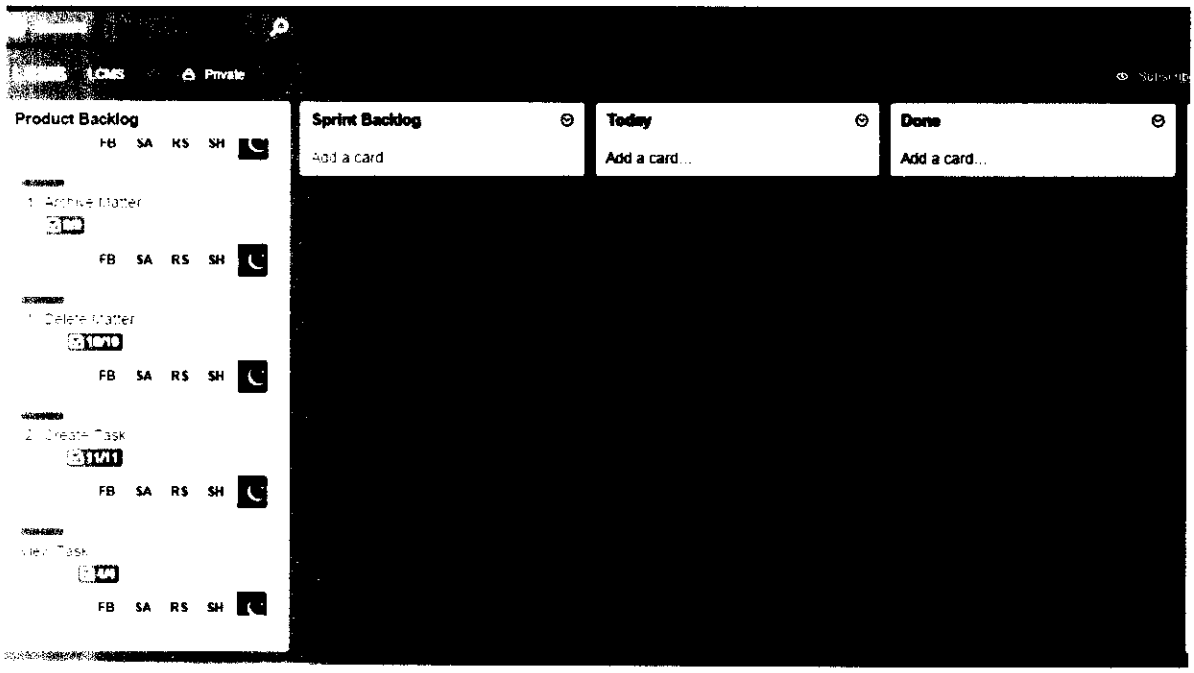


Figure 4-6: Product Backlog, Sprint Backlog, Today and Done

During the two weeks sprint, team created a database, run interfaces, and coding was done as planned. After the completion of each task, the team was instructed to mark that task to track the progress of that item. After the completion of a particular item, the team moved it to "done".

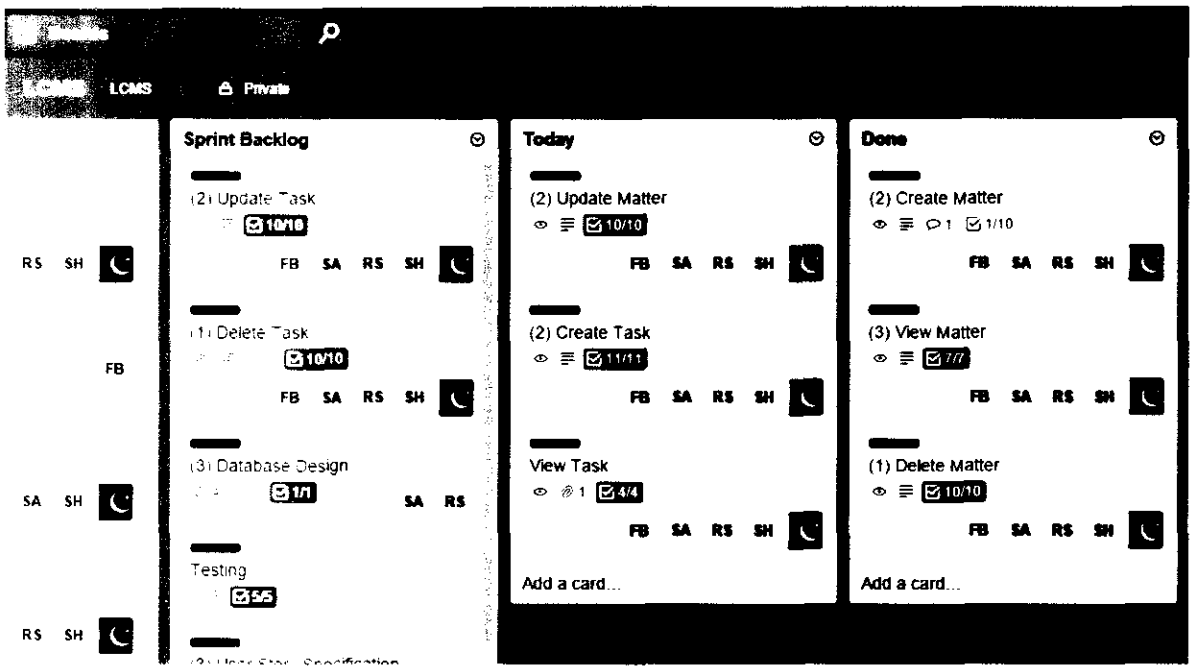


Figure 4-7: Cycle of the User Story

The selected tool did not provide the facility of mentioning all the attributes in solution, so some attributes were explicitly mentioned in the tool and for the rest, a separate file was created.

4.7.5 Sprint Review Meeting 1

After the completion of 1st sprint, the PO reviewed the implemented work presented by the team in a sprint review meeting.

4.7.6 Retrospective Meeting 1

This meeting was conduct at the end of each sprint and only once was attended by customers who suggested some improvements. The important suggestion that was given by SM was that it would be very helpful if all the team members make the point to ask the following two questions to PO in every sprint planning meeting.

Q1: Did the PBIs were prioritized before meeting? (If not)

Q2: When were the PBIs prioritized last time?

"Not accepting the work that does not match the pre-requisite or fulfilled the criteria" will push the PO to perform prioritization. This suggestion was added to the proposed solution after data analysis phase.

The second sprint was three weeks long and third sprint was of two weeks. Everything went the same way as of the first sprint except development of different user stories. The selected items for sprint backlog were **Manage Practice, Manage Contacts, Manage Lawyers and Manage Clients**. There was some problem regarding "Manage Matter", which was brought in the notice of PO, hence was sent back to the PB and later was considered for upcoming sprint that is sprint two. The selected items for this sprint were **"Opinions, Agreement, Notes, Meetings, Record, and Court Order"**.

4.8 Data collection

The data were collected from the transcribed text from recorded interviews, personal notes made during the observation, artifacts, and attachments available on SCRUM tool. Interview session was conducted; each of the participants was interviewed to judge his/ her experience of proposed solution. The interviews were semi-structured in nature that included both open and close ended questions. We recorded all the interviews and discussions in an audio form that were typically 30-35 minutes long. Later on, we transcribed the audio recordings into the textual form for the process of data analysis. After interviews, we checked all user stories from the PB to find out if the given format is followed, and whether the attributes were mentioned as said by interviewees. The attachments, activity log and use cases from the tool were also evaluated to cross check and determine the credibility of the answers of interviewees. The major findings are shortened at the end.

4.8.1 Data from notes

I used to note down (best to my effort) all the important happenings including the discussions, issues, improvements, suggestions and the meeting minutes of each meeting to see if the SCRUM is being followed the way it should be. During one such occasion team was discussing how helpful the interface designs are during the implementation, so it was decided to create GUI for each and

every user story and attach it to the tool for everyone to see. It was made compulsory after the first sprint and later became the part of the solution.

4.8.2 Activity log of the tool

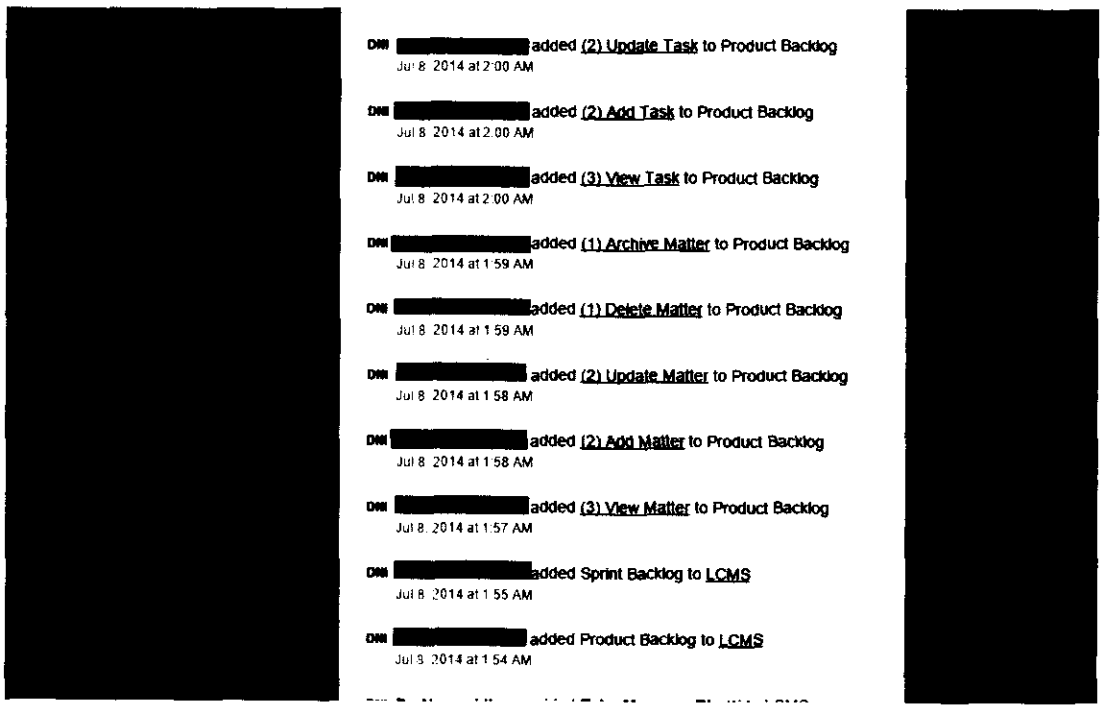


Figure 4-8: Activity Log of PO on TRELLO

4.8.3 Comment about user story specification

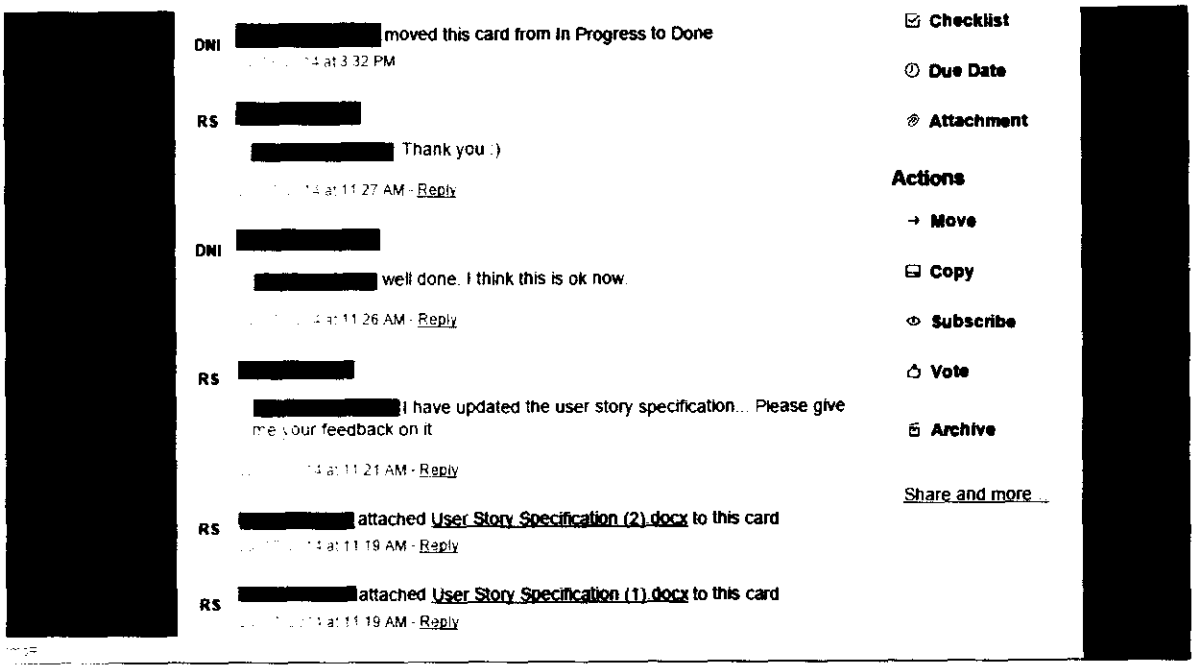


Figure 4-9: Comment about User Story Specification

4.8.4 Data from artifacts for create matter

	Name	Explanation
1.	Project name	Legal case management system
2.	Card no/ story no or ID	CM02
3.	Story name or title	Create matter
4.	Date	8 July
5.	Due date	15 July

6.	Description	As a Lawyer I want to manage matters so that I can add new matter I need to work on or to assign it to someone.
7.	State/Status	
8.	Time estimation	2 hrs
9.	Priority	Medium
10.	Requirement owner	Team member
11.	Release identifier	Nil
12.	Sprint identifier	S-01
13.	Dependency/ relation	-
14.	Acceptance criteria	1) A lawyer cannot create a matter until he is logged into his account 2) A blank matter sheet is presented when the lawyer requests to create a matter 3) The lawyer cannot create a matter without completing all the mandatory fields of the matter sheet 4) The information about a matter is stored in the matter's table of the database 5) A message of successful completion of operation is displayed once the matter is created 6) The lawyer is able to create a particular matter without getting an error 7) While creating a matter, if the user wishes to

	cancel the operation a new blank matter sheet is presented 8) While creating a matter, if the user chooses to logout, the system should prompt the user if there is any unsaved activity
15. Test case ID	LCMS-MM-07

Table 4-1: All Proposed Attributes for Create Matter

Use case Name: Manage Matters	
Brief Description: The Lawyer can create/add, update, retrieve (details of the client) and delete Matters.	
Primary actors: Lawyer	
Preconditions: Lawyer should be logged in.	
Main Flow: Retrieve Matters **User**	**System**
1. Chooses the option to manage list of Matters	2. Presents a list of Matters
3. Selects a particular Matter 4. Chooses to see the details of the Matters	5. Displays the details of the selected Matters
System: Legal Case Management System	

Post condition: Matters are listed, updated and saved.	
Alternative Flow: • **Create new Matters**	
User	**System**
1. After step 2 in the main flow chooses to create Matters.	2. Presents blank sheet to fill in Matter's details
3. Enters data in the Matter sheet 4. Selects option to save Matter data	5. Shows message of successful completion of operation

Table 4-2: Use case for Create Matter

Test case for create matter

ID: LCMS-MM-07

Description:

This test case is to ensure that Manage Matters use case (LCMS-MM-01) is functioning as it was originally established to.

Use case ID: LCMS-MM-01

Use case Name: Manage Matters

Preconditions: Lawyer must be logged in.

Post condition:

Matters are created, archived, listed, updated and saved. In the extension part, matters management is complete.

Main Flow:

User chooses the option to manage a list of matters

System presents list of matters

User selects a particular matter

User chooses to see the details of the matter

System displays the details of the selected matter

Alternative flow:

A1. Create new matter

1. User after step 2 in the main flow chooses to create matter
2. System presents blank matter sheet
3. User enters data in the matter sheet
4. User selects option to save data

   System show message for successful operation

4.8.5 Data from Interviews

Interviews were conducted at the end of the project so it was last activity of the data collection after observation, tool and artifacts emulation. Two separate questionnaires [20] were designed for Product owner and team. The questionnaire for team consisted of twenty six questions and follow-ups in total, where most of the questions were about the product backlog and sprint backlog. While the questionnaire, designed for PO consisted of twenty nine questions and follow-ups, where most of the questions were about the product backlog and prioritization.

Interviews with team members, artifacts and TRELLO (PB and SB) showed that the provided format was followed throughout for user stories and out of five members one said, that this format is useful and not so useful at the same time. According to him though it was useful because they were able to know the business goal, but at the same time US just told us about the role and briefly describe the feature. Another member however appreciated it by saying that the format was important to validate the output of US. About acceptance criteria one member added "it was acceptance criteria through which we understand the user story. Another member said that the US

written in the provided format was very to-the-point, he remarked "In this format user story was written to the point." On the other hand the "*Acceptance criteria*" was one thing which was appreciated by the entire team and found very useful.

We inquire them that if they faced any problem with product backlog user stories, the response was mix. Some team members said that stories were short but most of them were agree with the fact that once or twice they were in situation when the story was vague. In such situation they had to ask the PO for discussion. Once when they were having problem in understanding the US, PO guided them to make separate use case in that particular situation.

Upon inquiring PO about this, he said it was not his responsibility to define user stories as they were coming from customer and that they found the US format useful and were using it. "It was not my responsibility to define (user stories), actually they were coming from customers....As far as format for the user story is concerned we were using it.

According to the team members, some attributes were useful and other were not where TRELLO and artifacts ratified that some attributes i.e. (*project name, card no/ story no/id, story name/title, date, due date, description, state/status, priority, time estimation, sprint identifier, dependency/ relation, test case*) were explicitly mentioned and tasks were created during the project. It is observed that *release identifier*s is the attribute that was ignored merely because of the nature of the project.

The question raised here was why "release identifier" was skipped. When we asked PO, he reasoned that actually this project was not multi released project. He further elaborated that they had done three sprint, hence no release management was done because project was small "we had performed "n" iterations which made up three sprints so obviously we could not afford/manage to make shippable on every sprint and hence no release management was done because project was small".

Also it was observed that the team made some additions to make SCRUM work out for them e.g. for the elaboration of US they added use case specification and the screen shots of GUI interface. The one line US was used to further elaborate in use case specification and on the basis of use case

specification, the design and interface were created in parallel. The interface has nothing to do with requirements, but they added it to their work card as this was their design part. The reason behind this was that though "use case" is detailed enough but they created interfaces from the use cases which were something that instantly convey what they have to do. Some members found "database design" very useful during the time of implementation because it further makes things clear to them. As one member added "through database we came to know about relations & dependencies and other details were present in use cases".

The prioritization was performed in each planning meeting via. a technical point of view as said by the PO of the project. He further explicated that he had to check and fulfill prerequisites and dependencies between different items: "I had to check and fulfill prerequisites and dependencies between different items e.g. to start or perform some tasks which other tasks must have done". No defined or any specific prioritization technique was used for prioritizing the backlog items because according to the PO: "Customers were already clear about their priorities and there was no sudden change happing within product, so there was not as such difficulty that I have to deal with".

The entire team was agreed with the fact that PO never failed to provide prioritized items before or during sprint planning meeting except one member. According to that member only once there was a situation when PO said that he has to confirm the requirement from customer once again, after which the team went with another item with almost same priority.

Chapter 5: DATA ANALYSIS

5. Data Analysis and Interpretation

In this chapter, we discuss how we analyzed the collected data. The finding and results will be also discussed in the light on analyzed data.

5.1 Thematic network analysis

We applied the four-stage process, with embedded thematic network analysis activities on collected data [21]. At the first stage, the audio recordings of interviews were transcribed to perform data analysis and to find out the satisfaction level of the team with the proposed solution. At the second stage, we analyzed interviews to extract concepts (from it) related to the specification of the user stories of PB and SB, and related to the prioritization of US. Four concepts were identified related to the specification and prioritization of US. Then at third stage, we extracted themes and patterns from the concepts which were related to research questions of this study e.g. "proposed attributes". The detail of analysis is given below.

Concepts	**Underlying concepts or knowledge**	**Basic themes**
1. Utility	i. Developer/Coder didn't follow the format. ii. Designer/writer followed the format. a. Designer/writer saw the format. b. Designer/writer read the format. iii. Both developer and writer followed the attributes.	i. PB User Story Format ii. SB US Attributes iii. Prioritization
	iv. PO didn't follow the suggested prioritization techniques.	

2. Consultation	i. about user story ii. about design iii. abut acceptance criteria iv. about interfaces v. about business goals	i. Consult PO about PBIs ii. Consult PO about other problems
3. Sprint Backlog elaboration	i. Proposed attributes: yes ii. Additional attributes were • database design • (screen shots of) GUI interface • checklist for tasks	i. Proposed attributes ii. Additional attributes details
4. Valuable vs. effort and time	i. More understandable ii. Minimizes extra effort.	If adding the details were i. worth adding ii. If it reduced the problems. iii. If it increased the efficiency.

Table 5-1: Basic Themes

Sr. no.	Basic themes	Global theme
1	i. PB US Format ii. SB US Attributes iii. Prioritization	Backlog management

	i. Consult PO about PBIs ii. Consult PO about other problems	
	i. Additional details ii. Proposed attributes	
	i. If adding the details were worth adding ii. If it reduced problems iii. Increased efficiency	
	i. Tool unsupportiveness	

Table 5-2: Global Theme

5.1.1 User Story specification

US at first, was written in PB using one-line standard format where further details e.g. acceptance criteria and complexity/effort estimations get to add up in it, during product backlog grooming meeting. During the SCRUM planning meeting, the desired items from product backlog were picked up and moved it to the sprint backlog for upcoming sprint, where further details along time estimations get to add up.

- **Product Backlog**

The user stories were made available in a prioritized form at the time of sprint planning meetings. This was confirmed by the some team members and by the record of TRELLO. However, the team did not use it during the coding phase as one member commented "I mostly did coding from use cases, I even didn't see user stories when I was doing development".

The format was useful and important because all team members, as well as PO of the project, were valuing it. It is important to validate the output, and in this format US was written to the point. On

the other hand "**Acceptance Criteria**" was something which benefited them as through acceptance criteria, they used to understand the user story.

- **Sprint Backlog**

TRELLO was thoroughly analyzed and the data was studied to check how they use the suggested attributes which were used in the project. It was interesting to note that some attributes were available in attractive way e.g. "*due date*" was added for the card to notify the user about US deadline; ensuring users to develop it within time and after due date, the card was shown with red label indicating that allocated time is over for this user story. *Status* of the user story was changing along with the tasks like green for done, blue for in progress etc. The backlog items were stored in the form of ordered list.

With the help of TRELLO and through interviews, it has been found that team members did some additions to describe sprint backlog user story in order to gain more assistance, as one of the team members commented: "We used to mention different tasks that we have to achieve for user story, GUI interface and use case specification etc."

GUI interface was something about which the entire team was strongly supporting in sprint backlog whereas *use case specification* was valued by some members who were responsible for sprint backlog management. The suggestion of including GUI interface as optional element was found in literature in various studies [33,42,45] but was not included in our solution because we thought that it would not be helpful, but it in this particular case, team found it important.

When asked if they had to deal with any problem related to sprint user stories, one member responded that once the story was not too descriptive because information was not enough, while other member said that he had once found conflicts between interface design and the use cases.

Finally, we concluded that out of suggested attributes, some were useful and other were not, where TRELLO and artifacts ratified that these attributes (*project name, card no/ story no/id, story name/title, date, due date, description, state/status, time estimation, time estimation, sprint*

identifier, dependency/ relation, test case) were explicitly mentioned during the project. It is observed that *release identifier*" was ignored merely because of the nature of the project.

Through database design, they came to know about relations & dependencies and other details were presented in use cases. Among other additions that were made, GUI *interface* was supported by the entire team as the core, whereas *use case specification* was valued by some members who were responsible for sprint backlog management.

5.1.2 Prioritization

Any specific or suggested prioritization technique was not used explicitly or in formalizes way by typically assigning weight to user stories but they were prioritized with the judgment by intuitively keeping both technical and customer perspective in mind.

5.2 Summary of Findings

The template provided for user stories to the SCRUM team was followed throughout and it was valued by most of the team members. According to them, it was useful because they were able to know the business goal (to validate) the output of the user story from it and because the user story was written to the point. The entire team appreciated "acceptance criteria" and found it very useful. Most of the attributes e.g. story name or title, date, due date, description, state, and time estimation were explicitly mentioned during the project while release identifier was skipped merely because of the nature of the project. When asked PO, he reasoned that because this project was not multi-released project.

The whole team was strongly in favor of "acceptance criteria" and the solution has worked for them but, further details were required. As programmers need a high level of elaborations, so some modifications/additions were made by adding use case specification and database design to reduce the problems such as vagueness and abstraction.

We used to visit the company and had attended most of the SCRUM meetings, during which we observed that PO solely performed the prioritization process as suggested in solution. But he did not use any of the suggested techniques, during interview, when inquired about it, he explained

that he used to prioritize item by keeping its "technical dependency" in mind. He further proceeded by saying that he would call it judgment: "I would rather call it judgment rather than technique." The reason for not using the suggested technique was that although MASCOW technique was not used in formalized way by typically assigning weight to user stories, but they were prioritized with the judgment in way intuitively keeping both technical and customer perspective in mind.

With the help of TRELLO, each sprint was planned separately, and one could see product backlog to check how much work is left, and also each sprint backlog could be seen. On daily bases, team used to create bag which contained "what we will do today" (in contrast to the sprint backlog where we see, what one will do in an entire week or two).

5.3 Refinement in proposed solution after feedback

5.3.1 GUIDELINE 1: User story speciation

Product backlog user story: The user story is a simple method to elicit requirements from customer or users. User stories are created using the pre-agreed user story template/format [35]. The widely used and standard template for writing user story in PB is shown below:

As a <role>, I want to <action><object>, so that <business value> [33-41].

Where:

"**Role**" – signifies the person or system that will perform the action or one who will be benefited from the activity.

"**Activity**" – the action that will be performed by product or system

"Business Value"– shows the importance or market value and the reason [38].

An **acceptance criterion** is the primary item needed for the testing so it should be considered important. If a user story passes the test, it is considered as complete and is taken out of the list. It is the job of the product owner to approve the acceptance criteria and then tester decides whether the story passes the acceptance criteria or not [33], [46].

Story Points /size: Complexity is estimated and shown via story point of the user story [44], [49].

Sprint Backlog: All possible (compulsory and optional) information that could help the user in US elaboration. All the compulsory and optional attributes and information with additions after feedback from the team are placed in Table 5-3.

Sr. no.	Attributes	Explanation
1.	Project name [46]	Title of the project
2.	Card no./ story no. or ID [43,44,45,47,49]	Unique identifier or number for the US.
3.	Story name or title [44,45,46]	Name of the story should tell the content requirements.
4.	Date [41,44,45,46, 49]	The date on which the requirement was created.
5.	Due date [41]	It is necessary to know that the requirements are not skipped, ignored or overlooked.
6.	Description[42,43,45,46,47]	The requirements should not be less than 5 sentences. They should be comprehensive enough to explain the central meaning of it.
7.	Tasks /subtasks [41]	Different tasks against each user story related to database, interface and development.
8.	State/Status [41,43,47, 49]	The status of US or requirement reveals current state of it e.g. Drafted, rejected, incompletely specified, planned, refined, started, done etc. If US got rejected then the reason for rejection should be provided.
9.	Priority [41,47,49]	Set the priority as important, medium etc.
11.	Time Estimation [44,45,46,47]	Time required to complete the story.
13.	Requirement Owner/Concerned person/ Assignee [41,44,45]	For tracing purpose Link of the owner of the requirement, the person who is answerable for it.

14.	Release Identifier[43,47]	To assign the US to some specific release.
15.	Sprint Identifier[43,47,49]	To assign the US to some specific sprint.
16.	Dependency/ Relation [48]	Link to other requirement which has same scale of abstraction.
17.	Test Case ID[42,45,46]	The unique identifier for test case.
	Additions	
18	**Use Case [36]**	**The details could be delivered in use case form.**
19	**User Interface design [33, 42, 45,46]**	**Wireframes, Graphical user interfaces, logos or of the designs for system.**
20.	**Database Design**	**ERD diagram or any other design for DB**

Table 5-3: Final Attributes after Feedback

1.10.1. GUIDELINE 2: Prioritization of PBIs

Prioritization Techniques	Early Stages	Moscow
	Mid Stages	One Hundred Dollar Simple ranking

Table 5-4: Prioritization Techniques after Feedback

It would be very beneficial, if along the suggested techniques, team members ask the following two questions to PO in every sprint planning meeting.

Q1: Did the PBIs were prioritized before meeting?

Q2: When were the PBIs prioritized last time?

"Do not accept the work that does not match the pre requisite or fulfilled the criteria", it will drive the PO to perform prioritization.

Chapter 6: CONCLUSION

6. Conclusion

Several types of problems may arise during the process of SCRUM adoption. One of the most frequent and major problems with SCRUM adoption is lack of training, which causes other problems to occur. But most of these problems revolve around PB and the PO. This highlights the significance of the product owner role, so for obvious reasons our proposed solution facilitates PO and PB. Things can get better with guidelines, but everything can't be settled by just providing guidelines. There is a need to add further steps in guidelines and in implementation process, this is the first step towards it. The proposed solution would gradually get refined if applied on multiple cases with different scenarios after which hopefully it will be able to solve these problems.

6.1 Contribution

The contribution of this work is:

- Identification and categorization of different reported problems about SCRUM adoption by software development firms.
- To compile the backlog related issues, as the focus is primarily on the most discussed, broader and technical one.
- Based on suggestions and recommendations from earlier studies a novel based solution is proposed, which addresses all backlog related problems in this study in a way that with a little intervention, the maximum numbers of problems can be solved.
- SCRUM Tools evolution and recommendation.
- To implement the proposed solution in a software house to check if it is workable.
- Refinement of guidelines by adding some suggested attribute, on the base of data analysis and feedback from the participant of case study.

Our proposed solution provides a guideline for the companies, want to get rid of their current approach in order to adopt SCRUM smoothly.

6.2 Limitation and Recommendations and Future Work

As the focus of this research was only on backlog problems, in future the other problems of SCRUM adoption can be taken to focus area. A solution would be proposed for the SCRUM adoption problems other than backlog one.

It was indeed a single case, and there is a need to apply the proposed solution in multiple cases with different scenarios, with a relatively larger team and complex project. Those parts of proposed solution that were skipped by this company are needed to implement. In the future, a project with a large amount of requirements and an extended period of time should be planned where PO should use the prioritization techniques. Our targeted company was a small software house that implemented "our proposed solution" with a trial version of SCRUM tool. If an advanced version of the tool is purchased and used with the functionality of automatic prioritization, the results could be different and more versatile. Although the current tool (trial version) supported prioritization, but it works manually where the user could skip providing business value. In case of advanced version of a tool where PO is enforced to assign the business value to PB user stories, the results will be more accurate with better sorting.

There are some lesson learned from the perspective of case study as the author was a single researcher, it was not possible to be present everywhere and hence may have skipped certain important aspects. It has been also learned that usually people don't follow guidelines or will follow a part of it which they found valuable. It is the responsibility of the SCRUM master to make sure that team members are following the guidelines properly. He/she should keep a close eye on all activities meetings and especially on tool log (if team is using any tool) to make sure that every activity has performed on time.

6.3 Summary of work

SCRUM adoption needs many adjustments, learnings and changes. Several studies highlighted the problems occurred during the process of SCRUM adoption e.g. problems related to the organizational support which is considered as one of the major problems. This problem has categorized as: lack of human resources and lack of training. Lack of training further influence another problem to occur that is "not realizing the importance of SCRUM ceremonies roles and

artifacts". The behavior of people towards SCRUM adoption is also considered as a major problem which is further divided in two sub-categories: change resistance and non-cooperative people. Other problems are the demand of documentation from concerned parties and estimation about user stories. However, the focus of the study is the problem that was technical and broader one among all i.e. backlog related problems, which we have categorized as:

1. Right level of abstraction in the user story specification
   i. Product Backlog's user story specification
   ii. Sprint Backlog's user story specification
2. Prioritization of PBIs

We have complied these backlog related problems, and proposed a novel solution for it which will hopefully solve these backlog problems or the sub-categories of backlog problems as a whole.

- First part i.e. "right level of abstraction in user story specification" consists of two parts; product backlog user story specification and sprint backlog user story specification. A standard format for US is provided to deal with first part along with some suggestions. For the second part, all compulsory and optional information are identified which could help the team to gain more understanding about the user story.
- For the second part, we went through the literature and synthesized what prioritization techniques are being discussed about ordering SCRUM user stories. Unfortunately not many of the studies deal with this problem except a few to guide the level of technique and stages of requirement. There are certain techniques that could be helpful for small scale requirements and some other techniques for larger scale requirements.

The next step was to conduct a case study in a software house with a team (new to SCRUM) along a product owner and SCRUM master. It took two months to complete the project after which we conducted semi-structured interviews with the participants. After collecting the data from interviews the next stage was to perform data analysis on collected data (interviews, tool log, notes, and artifacts) to discover the results and find out if the proposed solution was valuable.

The results show that the provided format was followed throughout the project to write the user stories. According to the most of team members, the format was useful because through it, they

were able to know the business goal (to validate) the output of user story from it and because the user story (in the provided format) was written to the point. Also the team was strongly in favor of acceptance criteria. In the sprint backlog user story specification, most of the attributes were explicitly mentioned during the project while some were skipped merely because of the nature of the project. The solution had worked for them, but further details were required. As programmers need high level of elaborations so some additions were made by team i.e. "use case specification" and "database design". As far as prioritization is concerned instead of using any suggested technique, the prioritization was done via. technical dependency because this project was small and the requirements were less. In other words, MASCOW technique was not used in formalized way by typically assigning weight to user stories but USs were prioritized with the judgment in a way intuitively keeping both technical and customer perspective in mind.

References

[1]. Srinivasan, Jayakanth, and Kristina Lundqvist. "Using agile methods in software product development: A case study." Information Technology: New Generations, 2009. ITNG'09. Sixth International Conference on. IEEE, 2009.

[2]. Santos, Rui, et al. "Agile Technical Management of Industrial Contracts: SCRUM Development of Ground Segment Software at the European Space Agency." Agile Processes in Software Engineering and Extreme Programming. Springer Berlin Heidelberg, 2011. 290-305.

[3]. Moe, Nils Brede, Aybüke Aurum, and Tore Dybå. "Challenges of shared decision-making: A multiple case study of agile software development." Information and Software Technology 54.8 (2012): 853-865.

[4]. Moore, Richard, et al. "SCRUM at a fortune 500 manufacturing company." Agile Conference (AGILE), 2007. IEEE, 2007.

[5]. Read, Aaron, and Robert O. Briggs. "The Many Lives of an Agile Story: Design Processes, Design Products, and Understandings in a Large-Scale Agile Development Project." System Science (HICSS), 2012 45th Hawaii International Conference on. IEEE, 2012

[6]. Block, Matt. "Evolving to Agile: A Story of Agile Adoption at a Small SaaS Company." Agile Conference (AGILE), 2011. IEEE, 2011.

[7]. Vlaanderen, Kevin, et al. "Growing into agility: process implementation paths for SCRUM." Product-Focused Software Process Improvement. Springer Berlin Heidelberg, 2012. 116-130.

[8]. Marchenko, Artem, and PekkaAbrahamsson. "SCRUM in a multiproject environment: An ethnographically-inspired case study on the adoption challenges." Agile, 2008. AGILE'08. Conference. IEEE, 2008.

[9]. Moe, Nils Brede, T. Dingsyr, and O. Kvangardsnes. "Understanding shared leadership in Agile development: A case study." System Sciences, 2009. HICSS'09. 42nd Hawaii International Conference on. IEEE, 2009

[10]. Moe, Nils Brede, TorgeirDingsøyr, and Tore Dybå. "A teamwork model for understanding an agile team: A case study of a SCRUM project." Information and Software Technology 52.5 (2010): 480-491.

[11]. Lorber, Alfred, Kyran D. Mish, and Pat Notz. How We Successfully Adapted Agile for a Research-Heavy Engineering Software Team. No. SAND2013-2962C. Sandia National Laboratories, 2013.

[12]. Long, Ken, and David Starr. "Agile supports improved culture and quality for healthwise." Agile, 2008. AGILE'08. Conference. IEEE, 2008.

[13]. Akif, R., and H. Majeed. "Issues and Challenges in SCRUM Implementation.", in International Journal of Scientific & Engineering Research, August-2012 , Vol. 3, Issue 8

[14]. Hajjdiab, Hassan, and Al ShaimaTaleb. "Adopting agile software development: Issues and challenges." International Journal of Managing Value and Supply Chains (IJMVSC) 2.3 (2011): 1-10.

[15]. Hoda, Rashina, et al. "Agility in context." ACM Sigplan Notices. Vol. 45. No. 10. ACM, 2010.

[16]. Deemer, Pete, et al. "The SCRUM primer." SCRUM Primer is an in-depth introduction to the theory and practice of SCRUM, albeit primarily from a software development perspective, (2010):

[17]. Sutherland, Jeff, and Ken Schwaber. "The SCRUM guide." The Definitive Guide to SCRUM: The Rules of the Game (2011).

[18]. Uikey, Nitin, and UgrasenSuman. "An empirical study to design an effective agile project management framework." Proceedings of the CUBE International Information Technology Conference. ACM, 2012.

[19]. Mansor, Zulkefli, SaadiahYahya, and Noor Habibah Arshad. "Towards the Development of Success Determinants Charter for Agile Development Methodology." International Journal of Information Technology and Engineering2.1 (2011): 1-7.

[20]. Jim spickard. " how to construct an interview protocol." (2005).

[21]. Gill, Asif, Deborah Bunker, and Philip Seltsikas. "Moving Forward: Emerging Themes in Financial Services Technologies' Adoption." Communications of the Association for Information Systems 36.1 (2015): 12.

[22].C. Lynn Cowan, "when the vp is a SCRUM master, you hit the ground running", 2011 Agile Conference

[23].Asnawi, Ani Liza, Andrew M. Gravell, and Gary B. Wills. "Emergence of Agile Methods: Perceptions from Software Practitioners in Malaysia." AGILE India (AGILE INDIA), 2012. IEEE, 2012.

[24].Hajjdiab, H., and A. S. Taleb. "Agile adoption experience: A case study in the UAE." Software Engineering and Service Science (ICSESS), 2011 IEEE 2nd International Conference on. IEEE, 2011.

[25].Hosbond, Jens Henrik, and Peter Axel Nielsen. "Misfit or misuse? Lessons from implementation of SCRUM in radical product innovation." Agile Processes in Software Engineering and Extreme Programming. Springer Berlin Heidelberg, 2008. 21-31.

[26].Moe, Nils Brede, and TorgeirDingsøyr. "SCRUM and team effectiveness: Theory and practice." Agile Processes in Software Engineering and Extreme Programming. Springer Berlin Heidelberg, 2008. 11-20.

[27]. Dingsøyr, Torgeir, et al. "Developing software with SCRUM in a small cross-organizational project." Software Process Improvement. Springer Berlin Heidelberg, 2006. 5-15

[28].Asnawi, Ani Liza, Andrew M. Gravell, and Gary B. Wills. "Empirical investigation on agile methods usage: issues identified from early adopters in Malaysia." Agile Processes in Software Engineering and Extreme Programming. Springer Berlin Heidelberg, 2011. 192-207.

[29]. Ktata, Oualid, and Ghislain Lévesque. "Designing and Implementing a Measurement Program for SCRUM Teams: What do agile developers really need and want?." Proceedings of the Third C Conference on Computer Science and Software Engineering. ACM, 2010.*

[30].Kim, Eunha, and SeokmoonRyoo. "Agile Adoption Story from NHN." Computer Software and Applications Conference (COMPSAC), 2012 IEEE 36th Annual. IEEE, 2012.

[31].Overhage, Sven, and Sebastian Schlauderer. "Investigating the Long-Term Acceptance of Agile Methodologies: An Empirical Study of Developer Perceptions in SCRUM Projects." System Science (HICSS), 2012 45th Hawaii International Conference on. IEEE, 2012.

[32]. Lehto, Ilkka, and Kristian Rautiainen. "Software development governance challenges of a middle-sized company in agile transition." Proceedings of the 2009 ICSE Workshop on Software Development Governance. IEEE Computer Society, 2009.

[33]. Gregorio, Donna D. "How the Business Analyst supports and encourages collaboration on agile projects." Systems Conference (SysCon), 2012 IEEE International. IEEE, 2012.

[34]. Bang, Tom J. "An agile approach to requirement specification." Agile Processes in Software Engineering and Extreme Programming. Springer Berlin Heidelberg, 2007. 193-197.

[35]. Gaur, Vibha, AnujaSoni, and PunamBedi. "An application of multi-person decision-making model for negotiating and prioritizing requirements in agent-oriented paradigm." Data Storage and Data Engineering (DSDE), 2010 International Conference on. IEEE, 2010.

[36]. Gallardo-Valencia, Rosalva E., Vivian Olivera, and Susan Elliott Sim. "Are Use Cases Beneficial for Developers Using Agile Requirements?." Comparative Evaluation in Requirements Engineering, 2007. CERE'07. Fifth International Workshop on. IEEE, 2007.

[37]. Landhäußer, Mathias, and Adrian Genaid. "Connecting user stories and code for test development." Proceedings of the Third International Workshop on Recommendation Systems for Software Engineering. IEEE Press, 2012.

[38]. Zeaaraoui, Adil, et al. "User stories template for object-oriented applications."Innovative Computing Technology (INTECH), 2013 Third International Conference on. IEEE, 2013.

[39]. Sharp, Helen, Hugh Robinson, and Marian Petre. "The role of physical artefacts in agile software development: Two complementary perspectives." Interacting with Computers 21.1-2 (2009): 108-116.

[40]. O'hEocha, Colm, and Kieran Conboy. "The role of the user story agile practice in innovation." Lean enterprise software and systems. Springer Berlin Heidelberg, 2010. 20-30.

[41]. Schetinger, Victor, Cesar Souza, Lisandra Manzoni Fontoura, and Cesar TadeuPozzer. "User stories as actives for game development." (2011).

[42]. Williams, Heather, and Andrew Ferguson. "The UCD perspective: Before and after agile." Agile Conference (AGILE), 2007. IEEE, 2007.

[43]. Kääriäinen, Jukka, et al. "Supporting requirements engineering in extreme programming: managing user stories." Proc. 16th International Conference on Software & Systems Engineering and their Applications (ICSSEA 2003). 2003.

[44]. Gaur, Vibha, AnujaSoni, and PunamBedi. "An agent-oriented approach to requirements engineering." Advance Computing Conference (IACC), 2010 IEEE 2nd International. IEEE, 2010.

[45]. Rees, Michael J. "A feasible user story tool for agile software development?."Software Engineering Conference, 2002. Ninth Asia-Pacific. IEEE, 2002.

[46]. Patel, Chetankumar, and Muthu Ramachandran. "Story card based agile software development." International Journal of Hybrid Information Technology2.2 (2009): 125-140.

[47]. Kaariainen, Jukka, et al. "Improving requirements management in extreme programming with tool support-an improvement attempt that failed." Euromicro Conference, 2004. Proceedings. 30th. IEEE, 2004.

[48]. Gorschek, Tony, et al. "Industry evaluation of the requirements abstraction model." Requirements engineering 12.3 (2007): 163-190.

[49]. Ludlow, Lawrence. "The application of user stories for strategic planning." Agile Processes in Software Engineering and Extreme Programming. Springer Berlin Heidelberg, 2007. 198-202.

[50]. Bebensee, Thomas, Inge van de Weerd, and SjaakBrinkkemper. "Binary priority list for prioritizing software requirements." Requirements Engineering: Foundation for Software Quality. Springer Berlin Heidelberg, 2010. 67-78.

[51]. Hatton, Sarah. "Choosing the right prioritization method." Software Engineering, 2008. ASWEC 2008. 19th Australian Conference on. IEEE, 2008.

[52]. Harris, R. Scott, and Mike Cohn. "Incorporating learning and expected cost of change in prioritizing features on agile projects." Extreme Programming and Agile Processes in Software Engineering. Springer Berlin Heidelberg, 2006. 175-180.

[53]..Lehtola, Laura, MarjoKauppinen, and Sari Kujala. "Requirements prioritization challenges in practice." Product focused software process improvement. Springer Berlin Heidelberg, 2004. 497-508.

[54].Racheva, Zornitza, et al. "A conceptual model and process for client-driven agile requirements prioritization." Research Challenges in Information Science (RCIS), 2010 Fourth International Conference on. IEEE, 2010.

[55]..AL-Ta'ani, Rami Hasan, and RozilawatiRazali. "Prioritizing Requirements in Agile Development: A Conceptual Framework." Procedia Technology 11 (2013): 733-739.

[56]. Racheva, Zornitza, et al. "Do We Know Enough about Requirements Prioritization in Agile Projects: Insights from a Case Study." RE. 2010.

[57]. Racheva, Z., Daneva, M., Buglione, L.: Supporting the Dynamic Reprioritization of Requirements in Agile Development of Software Products. In: Proceedings of the Second International Workshop on Software Product Management 2008, Barcelona, pp. 49–58 (2008)

[58]. Popli, Rashmi, Naresh Chauhan, and Hemant Sharma. "Prioritising user stories in agile environment." Issues and Challenges in Intelligent Computing Techniques (ICICT), 2014 International Conference on. IEEE, 2014.

[59].Bakalova, Zornitza, et al. "Agile requirements prioritization: What happens in practice and what is described in literature." Requirements engineering: Foundation for software quality. Springer Berlin Heidelberg, 2011. 181-195.

[60]. Runeson, Per, and Martin Höst. "Guidelines for conducting and reporting case study research in software engineering." Empirical software engineering 14.2 (2009): 131-164.

AGILEFANT	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
ICESCRUM (OMEGA)	Y	Y	Y	P	N	Y	Y	Y	Y	Y
SCRUM DESK	Y	Y	Y	N	N	Y	Y	Y	Y	Y
PANGO SCRUM	P	Y	Y	Y	Y	N	Y	Y	Y	Y
SCRUM FACTORY	Y	Y	Y	N	Y	P	N	Y	Y	Y
SCRUMWISE	Y	Y	Y	P	N	N	N	Y	Y	Y
AGILEWRAP	Y	Y	Y	N	N	Y	N	Y	Y	Y
AGILO	P	Y	Y	N	Y	N	Y	Y	Y	Y
HANSOFT	Y	Y	Y	N	N	Y	N	Y	Y	Y
KUNAGI	P	Y	Y	P	Y		Y		Y	Y
SCRINCH	P	Y	Y	Y	Y	N	N	N	Y	N

Appendix C: Interview Guide

Questionnaire: to collect data from a product owner about the level of abstraction, and prioritization of the user story.

GENERAL QUESTIONS

1. What is your organization name?

---

2. How long you have been working in this organization?

User Story Definition

Appendices

Appendix A

Acronyms and Abbreviations	
Term	**Description**
PO	Product owner
SM	SCRUM master
PBI	Product backlog item
PB	Product backlog
US	User story

Appendix B

The list of evaluated of SCRUM tools according to the criteria defined in Section 4.5.1. These tools are basic SCRUM tools that are available on the net and the selection of the tools are made by keeping close tab on their popularity after reading reviews and recommendations from most of the SCRUM users. 'Y' stands for 'yes', 'N' stands for 'NO', 'P' stands for 'partially'.

SCRUM Tool										
[illegible]	Y	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	Y	Y	Y
VERSION ONE	Y	Y	Y	Y	P	Y	Y	Y	Y	Y

1. Did you face any difficulty while defining user story?

- Yes
- No
- Other

Follow up: If Yes, Why it was difficult to define user story?

Follow up: How often have you encountered with such situation?

Follow up: How did you deal with that?

2. Did you follow the provided format to describe user story

- Yes
- No
- Partially

Follow up: If Yes, Was it easy to use?

- Yes
- No

Follow up: If No, what were the problems in using format?

Follow up: If Partially, What part did u skip? Why?

Prioritization

3. How often you used to perform the prioritization of PBIs?

4. On which bases you assigned value to each PB item?

5. How you specify/represent that value to PB items?

6. Did you face any difficulty while prioritizing the product backlog items?

- Yes
- No

Follow up: If Yes, What problems?

_____

_____

_____

Follow up: How often did you face such difficulties while prioritizing PB items?

_____

_____

_____

Follow up: What was the reason behind it (difficulties while prioritizing PB items)?

_____

_____

_____

Follow up: How did you deal with that?

_____

_____

_____

7. How many techniques you use in your company for prioritization purpose?

- Single
- Multiple

Follow up: If single, Name the technique.

Follow up: If Multiple, which technique you use for which stage?

8. Did you use MoSCoW at early stage of R&A?

- Yes
- No

Follow up: If Yes, to which extend it overcame the problem?

Follow up: If No, Why you didn't use it.

Follow up: How did you deal with that?

9. Did you use One Hundred Dollar and Simple Ranking at mid stage of R&A?

- Yes
- No

Follow up: If yes, did it minimize your problem?

- Yes
- No

Follow up: If Yes, to which extend it overcame the problem?

_____

_____

_____

Follow up: If No, how did you deal with that?

_____

_____

_____

10. Have you faced with situation where two items were having equal priority?

- Yes
- No

Follow up: If yes, how did you deal with that situation?

_____

_____

_____

Follow up. If yes, did this format help you in any way and how?

4. Did you face any problem/issue related to user story in product backlog?

Follow up: If Yes, What problems usually you faced with user story in product backlog?

Follow up: Why it was difficult to understand?

Follow up: How did you deal with that?

Follow up: How often you encountered with such situation?

Sprint Backlog.

5. How did you describe user story in *sprint backlog*?

6. Did you face any problem/issue related to user story in sprint backlog?

Follow up: If Yes, What type of problems?

Follow up: Why did it occur?

Follow up: How did you deal with that?

Follow up: How often you encountered with such situation?

7. Did you use to mention any attributes for user stories in sprint backlog e.g. [name, priority, duration completed by etc.?

Follow up: Name all the attribute you used to mention while writing sprint backlog user story. e.g. [name, priority, duration completed by etc.]?

8. Did you mention the attributes mentioned in the solution?

Follow up: If No, then why? What was the reason?

Follow up: If partially, then what attributes did you skip mostly? What was the reason?

Questionnaire: to collect data from a SCRUM team members about the level of abstraction, and prioritization of the user story.

GENERAL QUESTIONS

1. What is your organization name?

---

2. How long you have been working in this organization?

User story definition

Product Backlog

1. When did *user story* was made available to team?

---

2. How user story was described in *product backlog*?

---

3. If the provided format to describe user story was followed by product owner?

---

9. Did mentioning these attributes helped you anyway? Explain how?

10. Had the product owner ever failed to provide you a well Prioritized items before or during sprint planning meeting?

- Yes
- No

Follow up: How did you deal with that?

Follow up: How often you encountered with such situation?

Follow up: Why did such situation occur?

11. In which form product backlog was provided to you by product owner? (e.g. order list)