Journal of Information Technology Management

Print ISSN: 2008-5893

Online ISSN: 2423-5059

Exploring Story Cards for Evaluating Requirement Understanding in Agile Software Development

Sarika Sharma*

*Corresponding Author, Research Scholar, Amity Institute of Information Technology, Amity University, Sector-125, Noida, 201307, (U.P.) India. E-mail: sarika.s17@gmail.com

Deepak Kumar

Professor, Amity Institute of Information Technology, Amity University, Sector-125, Noida, 201307, (U.P.) India. E-mail: deepakgupta_du@rediffmail.com

Abstract

From the recent literature review, it is evident that existing agile methodology lacks the method to evaluate the requirement understanding of agile team members for a given set of requirement chosen for agile software development. Hence, there is a need to introduce a requirement understanding check to ensure every agile team member follows the given requirement clearly without any ambiguity. To fill this existing gap, this research paper proposes to extend the usage of story cards to evaluate the understanding of the given requirement and to highlight any challenges and risks in the early stage of requirement understanding under agile software development methodology, if any. This paper primarily focuses to introduce a robust requirement understanding evaluation process in agile methodology. The research results were found to be motivating and were analyzed by comparing the data-points using time-series for performing agile query analysis, agile team velocity analysis and agile team involvement analysis for two agile teams where one team delivered the sprint output using agile traditional method while another team opted for proposed approach. A considerable decrease of 33.07% was observed in the number of queried raised and a significant increase of 26.36% in agile velocity was observed for agile sprint under proposed approach when compared to agile traditional approach. Also, a significant shift from 40%-80% team involvement under traditional agile method was uplifted to 80%-90% team involvement under proposed approach.

Keywords: Software Engineering; Agile Methodology; Requirement Understanding; Story Cards.

DOI: https://10.22059/jitm.2022.85646 Manuscript Type: Research Paper University of Tehran, Faculty of Management

Received January 07, 2021 Accepted March 25, 2021

Introduction

Requirement gathering is the key to success for any software development project. Agile methodology has simplified the requirement gathering a lot when compared to waterfall model which has clearly improved the success rate of agile projects. Agile methodology does not encourage heavy documentation for requirement gathering. Hence, the documents like System Requirements Specification (SRS) have almost disappeared. Agile methodology has made the task of requirement gathering much simplified since it has the ability to handle the frequent changes in business requirement with the help of short delivery cycles. Agile suggests capturing the requirement through pre-defined format called user stories. The user stories are written in the format like As a (stakeholder role)…I want (specific requirement) …So that (business benefit). In initial days of agile, the user stories were captured using story cards in physical mode later on the story cards disappeared as software tools took their place. The one drawback associated with user stories is that they are at high level since it is really difficult to capture all the requirements details in a single line of text. Hence, to make user stories more accurate the acceptance criteria are added so that the agile team clearly knows what conditions needs to met against which the working software will be accepted. There is defined format for the acceptances criteria as well such as; Given <I am a stakeholder>; When < condition >; Then <action>. Every user story is further broken down into tasks. The successful completion of all these tasks denotes that the given user story is now ready for review to be conducted by customer.

Although, agile methodology has practices like product grooming sessions of few hours where the given user stories are further refined for better understanding of requirement but there is no way to evaluate if agile team is able to understand the requirement correctly or not.. In the absence of any requirement understanding process, post product backlog grooming session team immediately starts working on the development of the scoped user stories which may cause a risk to the delivery of software product taken under development. Hence, to fulfill this gap this research work proposes to use the story cards for evaluating requirement understanding of agile team .The results from the research shows the sprints delivered with traditional agile approach have more likelihood of inaccurate delivery due to lack of understanding of requirement by agile team when compared to sprints where newly proposed approach was incorporated.

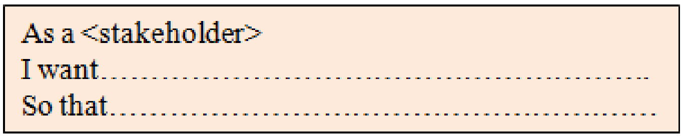
In this paper we will focus on usage of user stories for requirement understanding. This paper is structured as follows. The provided literature review highlights the related work done so far and the existing gaps. The proposed approach details about the research algorithm and evaluation criteria for examining research results. The methods and implementation provides an insight about the method employed to execute the proposed research followed by the detailed discussion on research results. The conclusion and way forward draws the attention towards the fact that the proposed research work successfully filled the existing gap related to requirement understanding in agile software methodology.

Background

Agile project management methods revolutionized the way how software projects are executed and organized (Stettina & Hörz, 2015).The gathering, understanding and managing agile requirements is a key factor to the success of agile software development effort (Kavitha & Thomas, 2011). From the recent literature review, it has been observed that plenty of research has been done on agile requirement gathering (Alam et al., 2017) (Rida et al., 2016) and agile requirement managing (Okesola et al., 2019) (Shim & Lee, 2019). (Lucassen et al., 2015) and (Moreno & Yagüe, 2012) acknowledges user stories as a widely used notation for formulating agile requirements in agile development projects. In (Wake, 2003) INVEST frame-work highlights every good user story has to be independent, negotiable, valuable, estimable, small, and testable. (Beck, 2004) first introduced the concept of user stories to software development which was later developed and extended to other agile methodologies like scrum by (Cohen, 2004). In simple words, (Jeffries, 2001) user stories comprise of three aspects known as 3C: card, conversation and confirmation which are essential for requirement gathering. Here, third 'C' confirmation relates to user story acceptance criteria. However, in this research work, we propose to extend the third 'C' to the agile team members as well by confirming whether they understand the agile requirement or not. More than 30 automated tools are available for agile practitioners for assisting requirement management.(Zhu, 2017) suggests a user friendly visual approach to manage the business value for frequently changing agile requirements but does not provide a plat-form for agile practitioners to reflect their agile requirement understanding. (Trkman et al., 2016) proposes to a business process model to explore the dependencies among user stories in order to enhance requirement understanding. However, identifying the dependent user stories does not assure that agile team clearly understands the given agile requirement in the form of user-story. It is possible that for a group of dependent user stories, agile team members may still have little understating issue. (Dalpiaz & Brinkkemper, 2018) confirms 90% of agile practitioners employs user-story for capturing requirements. It further provides the quality user story framework driven by 13 quality criteria to improve the quality of user-stories under three wide categories syntactic, semantic and pragmatic. But, it does not attempt to extend the usage of user-story cards beyond requirement gathering. Since, (O'hEocha & Conboy, 2010) user story represents a technique of establishing a shared understanding of software requirements using a low-overhead, user centric and flexible approach. Therefore, it gives raise to the need of introducing a requirement understanding check for all agile team members. In (Daneva et al., 2013)highlights that understanding requirements dependencies is of paramount importance for the successful deployment of agile approaches in large outsourced projects but does not provide any approach for evaluating requirement understating of agile team members. In (López-Martínez et al., 2017) provides a Bayesian algorithm based knowledge model to determine complexity and importance of user-story estimation technique. The research work highlights that the team members with low experience have problems in performing an accurate estimation when compared to experienced agile practitioners but does not confirms that the provided requirements were equally clear or not to both the group of individuals which could have influenced their estimation ability in return. In (Al-Garni, 2018) ,the twelve requirements prioritization techniques highlighted were the Analytical Hierarchical Process (AHP), the Numerical Assignment (NA), the Binary Search Tree (BST),the Planning Game (PG), the Cumulative Voting (CV), the Quality Functional Deployment (QFD), the cost-value approach, the Win-Win method, MosCow, the Top10, the bubble sort approach and the Wiegers approach. But none of the listed techniques has been extended to examine the requirement understanding. The research work (Miranda et al., 2009) proposes an efficient user story sizing technique but does provides any details around whether the agile team clearly understands the requirement provided through user-story or not. The recent CHAOS summary reports 75% of software projects fails due to inaccurate estimation (Khan et al., 2019) . Since agile effort estimation techniques like planning poker, T-shirt sizing and many others are mainly based on user-stories therefore, it becomes critical for the agile team members to clearly understand the requirement to avoid any inaccurate effort estimation.(Dimitrijević et al., 2015) emphasis on the usage of the automated user-story management tools namely Planbox, tinyPM, Agilo for track, ScrumDesk, VersionOne supported by their comparative study but unfortunately all these tools are primarily focused on user story management and does not aid in evaluating requirement understating of agile team members. Hence, this research work aims to fill this gap.(Alyahya et al., 2016) evaluates 30 agile planning tools but none of them supports any method to evaluate the requirement understanding of agile team members. (Beck & West, 2004) clarifies user story is distinct from similar techniques (scenarios and use cases) because it is always told from the point of view of the user, not the developer. Therefore, it is of paramount importance that a new should be introduced in agile software development where agile team members including developers can present their point of view about the provided agile requirement. (Baruah, 2015) consider poor requirements and changes to requirements as one of the causes for project overrun and quality issues in the delivered software but does not regard lack of requirement understanding of agile team members as a issue. (Strode, 2012) concludes most coordination is for managing requirement, expertise, historical, and task allocation dependencies; all forms of knowledge dependency. Hence, improved requirement understanding at earlier stage in sprint planning will reduce some of co-ordination efforts. Hence, this paper suggests an effective and efficient way to evaluate requirement understanding by agile team right at the beginning of every sprint resulting into reduced delivery risk. Also, once the agile team goes through the requirement evaluation stage, it clearly understands what needs to be delivered without any ambiguity. It has also been observed the execution of user stories were faster when compared to execution of user stories without requirement understanding procedure.

Proposed Approach

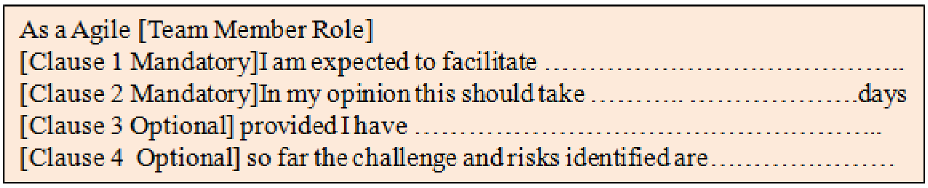
This research paper suggests extending the usage of user-story cards to evaluate the understanding of the given requirement for a software development project under agile methodology. The components of the user-story cards are the stakeholder role, high level requirement and benefit. Here, the stakeholder role corresponds to the individual who provides the requirement to the agile team, further the high level requirement details the requirement in layman's language and the third and the last component of user-story cards captures the value achieved on the delivery of the given business requirement. The figure 1 depicts the usual format of agile user-story card.



As a <stakeholder>
I want……………………………………………………..
So that……………………………………………………..

Figure 1. Agile User-Story Card

The proposed agile user-story card format shown in figure 2 employs four new clauses when compared to traditional agile user-story card format as shown in figure 1.The newly extended user-story cards have two mandatory clauses and two optional clauses.



As a Agile [Team Member Role]
[Clause 1 Mandatory]I am expected to facilitate ………………………………………..
[Clause 2 Mandatory]In my opinion this should take ……….. ………………….days
[Clause 3 Optional] provided I have …………………………………………………..
[Clause 4 Optional] so far the challenge and risks identified are…………………

Figure 2. User-Story Card Extended format for Requirement Understanding

In general, the first and foremost statement "As a" help the agile team member to understand his role within the agile team. The clause 1 helps the team member to state what s/he is going to contribute in the given sprint followed by clause 2 that helps the team member to provide an estimation of efforts in the number of days. The clause 3 lists down the information or support or infrastructure the team member needs to perform the delivery and the clause 4 provides an opportunity to the team member to raise any challenge or risk early in the requirement understating phase. However, based upon the role of the team member the content of the all the newly added four clauses will be updated to cater the need of the specific team member. For example the Scrum Master might use the newly introduced user-story cards for estimating the expected numbers of days and the expected resources needed to deliver given requirement. In addition to this, the scrum master can add the details to the proposed format to capture any challenge foreseen while delivering the said requirement like agile team recruitment. Similarly, the newly proposed format of user story for requirement validation can be extended to other team members playing the role of developer, testers, DevOps and others in the agile team to capture the expected numbers of days and expected resources required to deliver the given requirement and to highlight any challenges that could hamper the project delivery as shown in figure 3, 4 and 5 respectively.

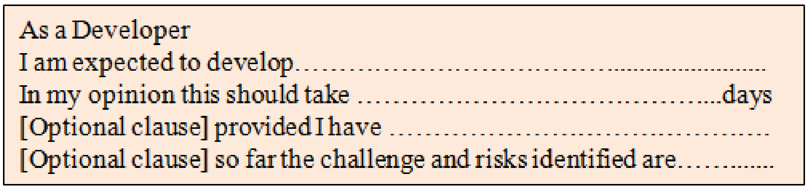

As a Developer
I am expected to develop…………………………………..........................
In my opinion this should take ……………………………………...days
[Optional clause] provided I have ………………………………………
[Optional clause] so far the challenge and risks identified are………........

Figure 3. Proposed User-Story Card for Agile Developer

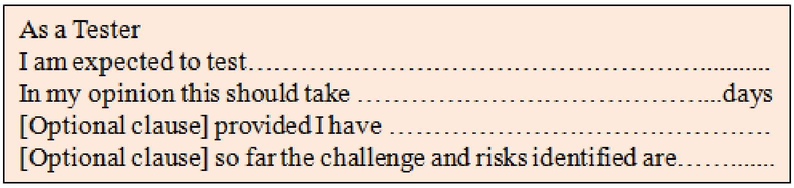

As a Tester
I am expected to test…………………………………………………..........
In my opinion this should take ……………………………………...days
[Optional clause] provided I have ………………………………………
[Optional clause] so far the challenge and risks identified are………........

Figure 4. Proposed User-Story Card for Agile Developer

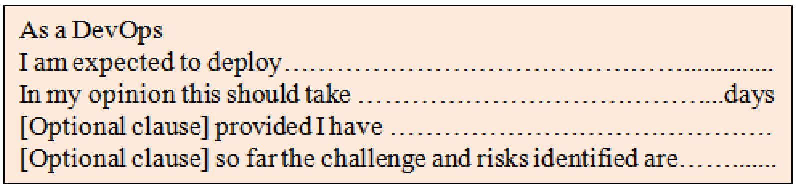
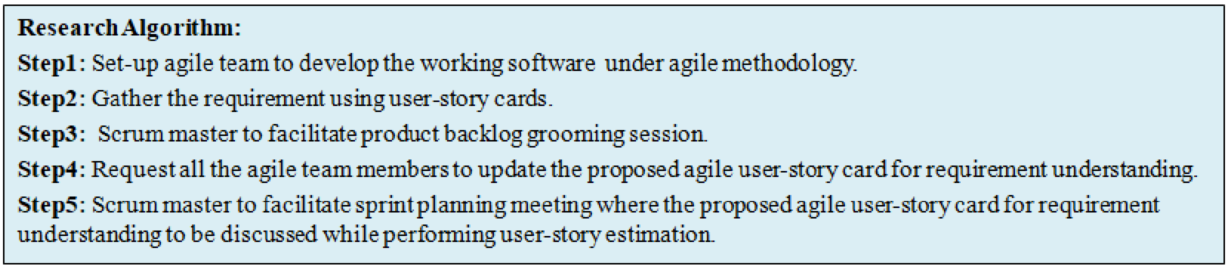

As a DevOps
I am expected to deploy…………………………………………..............
In my opinion this should take ……………………………………...days
[Optional clause] provided I have ………………………………………
[Optional clause] so far the challenge and risks identified are………........

Figure 5. Proposed User-Story Card for DevOps

Research Algorithm

The proposed research algorithm as shown in figure 6 is simple to follow where agile team member were asked to update the newly proposed user-story cards for each requirement provided through user-stories from agile release backlog.


Research Algorithm:

Step1: Set-up agile team to develop the working software under agile methodology.

Step2: Gather the requirement using user-story cards.

Step3: Scrum master to facilitate product backlog grooming session.

Step4: Request all the agile team members to update the proposed agile user-story card for requirement understanding.

Step5: Scrum master to facilitate sprint planning meeting where the proposed agile user-story card for requirement understanding to be discussed while performing user-story estimation.

Figure 6. Research Algorithm

Evaluation Criteria

The research results were analyzed by comparing time-series for two agile teams (Team A, Team B) where for Team A all the sprints were executed using agile traditional method while for Team B for all the sprint the proposed approach was taken into consideration. Time series analysis comprises methods for analyzing time series data in order to extract meaningful statistics and other characteristics of the data. Hence, it was employed for performing the result analysis for the three evaluation criteria chosen for performance analysis. The three evaluation criteria considered to determine the effective of proposed approach over agile traditional approach were agile query analysis, agile team velocity analysis and agile team involvement analysis Experiment Data Record

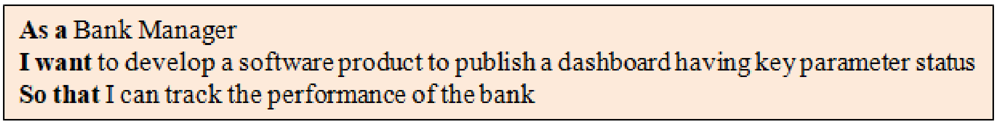
Materials and Methods

Agile Team Set-up

The two agile teams of 15 team members each were set-up to deliver the sprints as per the agile traditional versus the proposed approach. The profiles of all agile team members from both the agile teams has similarity index of 0.95. Both the agile teams were given same set of requirement to develop working software for a small banking module having 150 user-stories spread over 10 sprints each. Each sprint was of 10 days in length based upon the agile team consensus.

Agile Requirement

The given agile requirement was share with both the agile teams as shown in figure 7. As a Bank Customer I want to transfer money from one country to another in 10 seconds So That I can quickly send money to my relatives living in abroad. Although, the requirement seem to be simple at first glance but had a lot of salient feature that need to address at the requirement understating stage. For example the first observation is the given requirement involves an international transaction therefore currency conversion comes into picture. The security and performance are other two critical requirements followed by customer account details. Therefore, to ensure agile team understand all the aspects of given requirement there is need to support the current product backlog grooming session with the proposed user-story cards to ensure all the minute details of the provided requirement are discussed at the very beginning of the sprint.

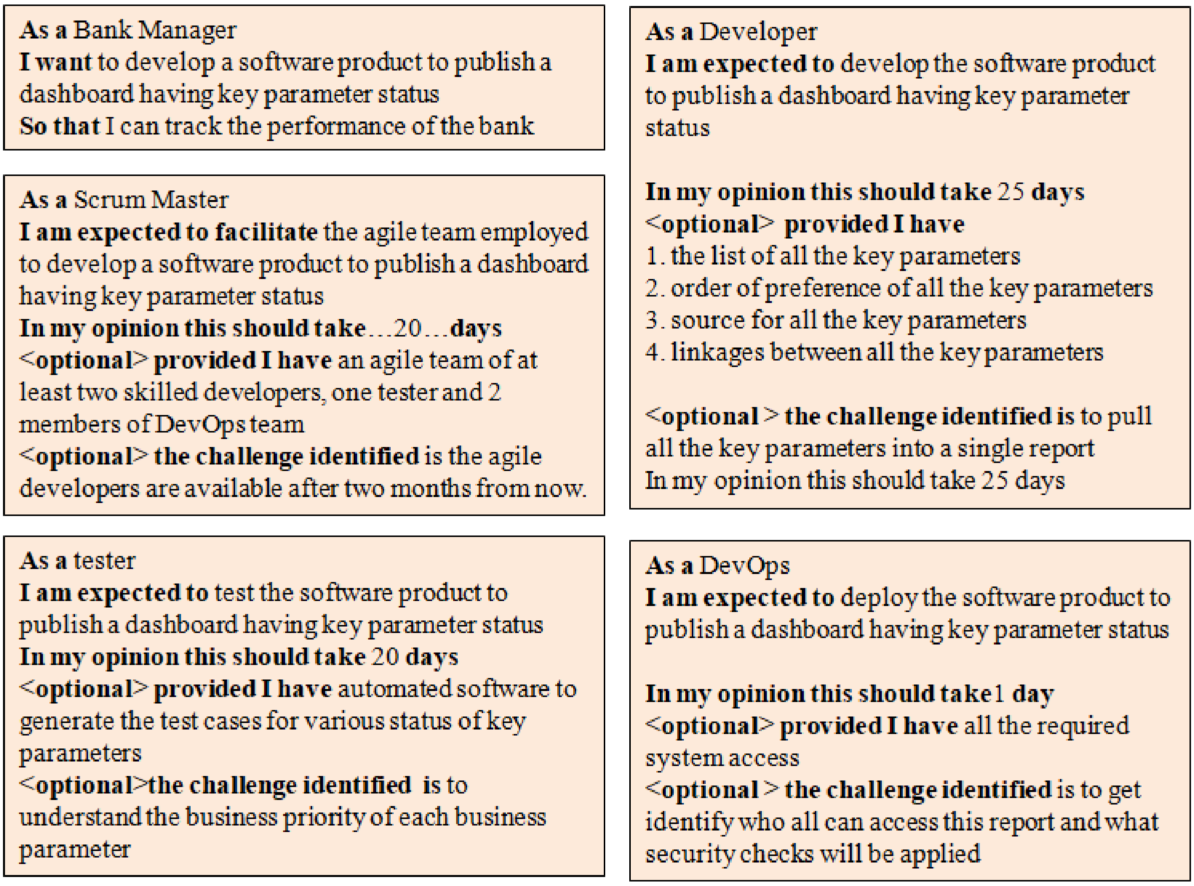


As a Bank Manager
I want to develop a software product to publish a dashboard having key parameter status
So that I can track the performance of the bank

Figure 7. Sample User-Story Card for Requirement Gathering

Agile Process

As per the standard agile practice, a common product grooming session was conducted by product owner for both the teams where all the user stories were explained by the product owner and the agile team members asked questions for clarity wherever required. Both the agile teams were then asked to initiate the sprint planning meetings separately using planning poker. All the agile ceremonies remained similar for the agile teams except the Team B was asked to update the newly proposed requirement understanding user-story cards for each user story mentioned in the release backlog (as shown in figure 8.) which was discussed during the sprint planning meeting of Team B. Also, both the agile teams were asked to keep a track of the length of their agile meetings like daily stand-up, sprint review and sprint retrospective along with the number query raised by team members on daily basis. The tracking of number of query and number of query rose were not seen as an over-head but were required to gather the data points for research findings. As shown in figure 8, all the team members of agile team B included scrum master, developer, tester and devops to update the newly proposed requirement understanding user-story cards to confirm their understanding.



As a Bank Manager
I want to develop a software product to publish a dashboard having key parameter status
So that I can track the performance of the bank

As a Scrum Master
I am expected to facilitate the agile team employed to develop a software product to publish a dashboard having key parameter status
In my opinion this should take…20…**days**
<**optional**> **provided I have** an agile team of at least two skilled developers, one tester and 2 members of DevOps team
<**optional**> **the challenge identified** is the agile developers are available after two months from now.

As a tester
I am expected to test the software product to publish a dashboard having key parameter status
In my opinion this should take 20 **days**
<**optional**> **provided I have** automated software to generate the test cases for various status of key parameters
<**optional**>**the challenge identified** is to understand the business priority of each business parameter

As a Developer
I am expected to develop the software product to publish a dashboard having key parameter status

In my opinion this should take 25 **days**
<**optional**> **provided I have**
1. the list of all the key parameters
2. order of preference of all the key parameters
3. source for all the key parameters
4. linkages between all the key parameters

<**optional** > **the challenge identified is** to pull all the key parameters into a single report
In my opinion this should take 25 days

As a DevOps
I am expected to deploy the software product to publish a dashboard having key parameter status

In my opinion this should take 1 **day**
<**optional**> **provided I have** all the required system access
<**optional** > **the challenge identified** is to get identify who all can access this report and what security checks will be applied

Figure 8. Sample User-Story Card for Requirement Understanding

Results

The three evaluation criteria namely query analysis, agile team velocity and agile team involvement analysis were chosen to analyze the efficiency of the proposed approach versus traditional agile approach are discussed below.

Query Analysis

Query analysis acted as a key metric to demonstrate the degree of requirement understanding of both the agile teams. It can be interpreted as more the number of queries during sprint execution lesser is the requirement understanding. From table 1, it is clearly visible that the agile team following the proposed approach got the plat-form with help of proposed extended story-cards to raise and clarify the requirement understanding queries during the sprint planning session which was not the case for agile team following the traditional approach. Hence, it was observed the number of queries raised were consistent throughout the sprint in traditional agile method while the number of queries raised were higher in the beginning of the sprint but then were decreased to single digit towards the end of sprint under proposed approach. A considerable decrease of 33.07% was observed in the number of queried raised during entire agile sprint under proposed approach when compared to agile traditional approach since a number of repeated queries were found to reduced drastically under proposed method. The figure 9 shows the time-series representation of query analysis for traditional approach versus proposed approach.

Table 1. Data for Time Series Query Analysis

Sprint Details	Sprint Days	Average Number of Queries per Day (Traditional)	Average Number of Queries per Day (Proposed)
Sprint Planning	1	35	120
Sprint Execution	2	28	25
Sprint Execution	3	26	12
Sprint Execution	4	27	5
Sprint Execution	5	30	3
Sprint Execution	6	26	3
Sprint Execution	7	24	4
Sprint Execution	8	26	3
Sprint Execution	9	26	1
Sprint Review	10	15	0
Total Queries		263	176

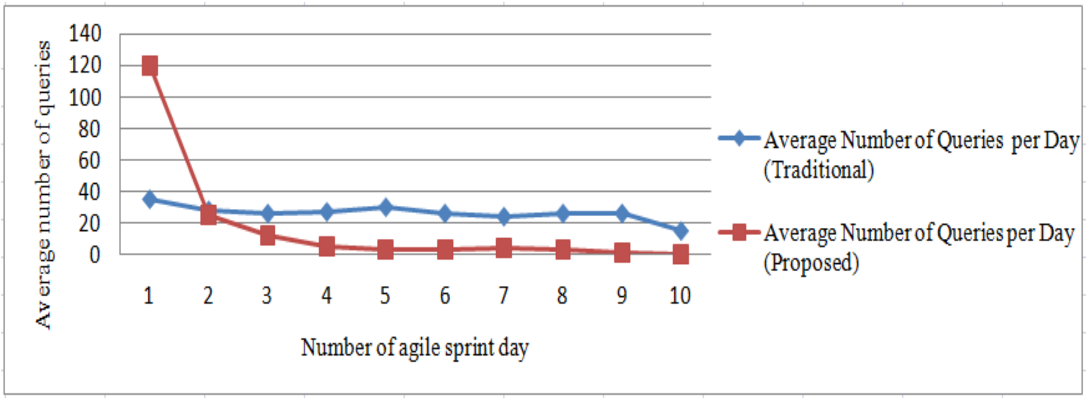


Figure 9. Time Series for Query Analysis

Agile Team Velocity

In agile software development, the velocity is a measure of story points delivered by agile team during a sprint. A significant increase of 26.36% in agile velocity was observed under proposed approach when compared to agile traditional approach. The increase in agile team velocity under proposed approach is driven from the fact that agile team saved large amount of time by getting all the requirement understanding queries resolved during sprint planning as discussed above. Hence, under proposed approach the team worked seamlessly on the assigned user-stories and delivered 26.36% more story-points when compared to team velocity under traditional approach. The table 2 provides the comparative study of story-point delivered under proposed and traditional approach and figure 10 provides a time series for agile team velocity analysis.

Table 2. Data for Time Series Agile Team Velocity Analysis

Sprint ID	Number of Story Points Delivered Per Sprint (Traditional)	Number of Story Points Delivered Per Sprint (Proposed)
1	110	140
2	120	155
3	90	120
4	100	110
5	95	120
6	100	135
7	100	125
8	120	145
9	130	160
10	125	165
11	115	140
12	120	145
13	95	130
14	105	145
15	125	150
Total	**1650**	**2085**

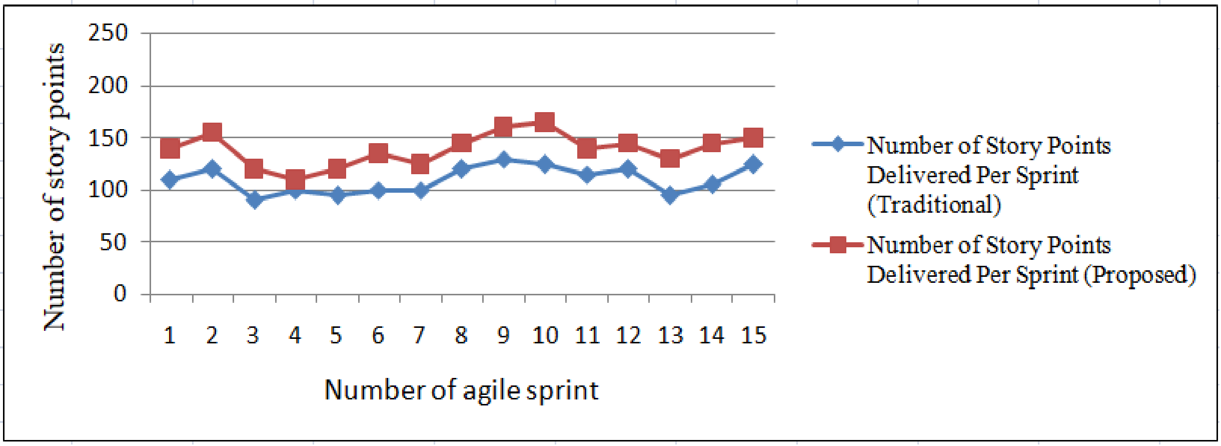


Figure 10. Time Series Agile Team Velocity Analysis

Agile Team Involvement Analysis

A short survey was conducted for both agile teams where agile team members from both the agile teams were asked to rate their team involvement experience on the given scale of 0-10%,10-20%,20-30%,30-40%,40-50%,50-60%,60-70%,80-90% and 90-100%. From table 3, it is clearly evident that a higher team involvement emerged under proposed approach when compared to traditional approach which was an added advantage. A significant agile team involvement of 80-90% under proposed approach was observed when compared to agile team involvement largely between 40%-80% under traditional approach. Figure 11 depicts the time-series data for agile team involvement under proposed and traditional approach.

Table 3. Data for Time Series Agile Team Involvement Analysis

Team Involvement Scale (0-100%)	**Agile Team Member Involvement Using Likert (Traditional)**	**Agile Team Member Involvement Using Likert (Proposed)**
10-20%	0	0
20-30%	0	0
30-40%	0	0
40-50%	3	0
50-60%	2	0
60-70%	6	0
70-80%	4	0
80-90%	0	9
90-100%	0	6
Total	**15**	**15**

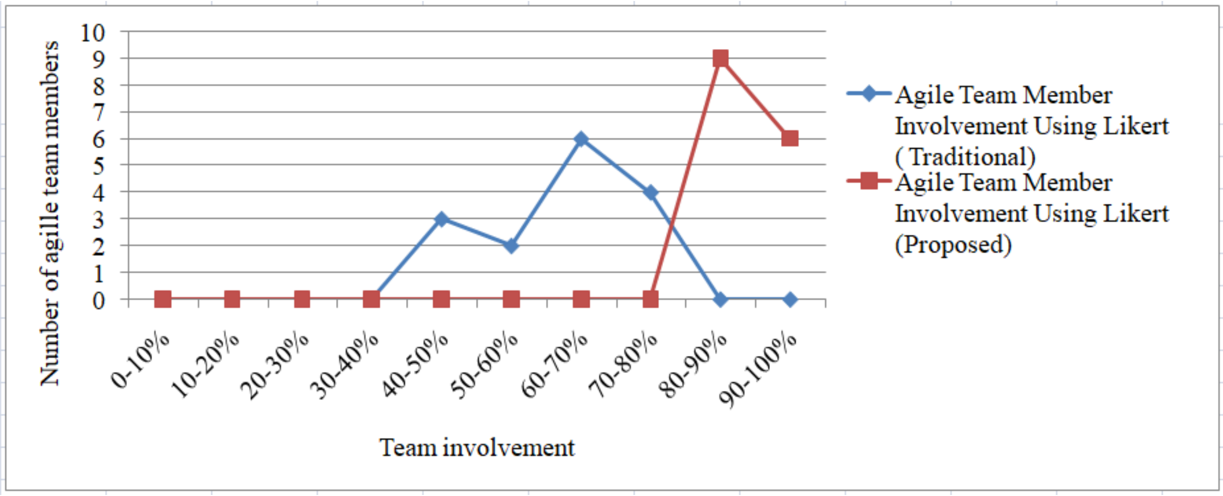


Figure 11. Time Series for Agile Team Involvement Analysis

Conclusion

With the help of this research work, it has been observed that user-story cards can be successfully used to evaluate the requirement understanding of agile team members. The extended use of user-story cards proved to be useful for handling majority of the requirement related queries in the beginning of the agile software development project. The proposed usage of story-cards with new structure also increased team involvement and improved the team velocity. Hence, this research paper recommends the usage of proposed agile story cards for the evaluating the requirement understanding of agile team.

Way Forward

The results from the provided research work were found to be encouraging and meaningful. Therefore, the researcher further plans to extend the usage of story-cards for requirement prioritization, test-case development and test-case prioritization in agile software development with help of machine learning algorithms.

Conflict of interest

The authors declare no potential conflict of interest regarding the publication of this work. In addition, the ethical issues including plagiarism, informed consent, misconduct, data fabrication and, or falsification, double publication and, or submission, and redundancy have been completely witnessed by the authors.

Funding

The author(s) received no financial support for the research, authorship, and/or publication of this article

References

Al-Garni, B. M. (2018). Analysis And Review Of Prescribing Clinical Decision Support System Within The Context Of Nhs Secondary Sector. *Life: International Journal of Health and Life-Sciences*. https://doi.org/10.20319/lijhls.2018.43.6071

Alam, S., Nazir, S., Asim, S., & Amr, D. (2017). Impact and Challenges of Requirement Engineering in Agile Methodologies: A Systematic Review. *International Journal of Advanced Computer Science and Applications*. https://doi.org/10.14569/ijacsa.2017.080455

Alyahya, S., Alqahtani, M., & Maddeh, M. (2016). Evaluation and improvements for agile planning tools. *2016 IEEE/ACIS 14th International Conference on Software Engineering Research, Management and Applications, SERA 2016*. https://doi.org/10.1109/SERA.2016.7516149

Baruah, N. (2015). Requirement management in agile software environment. *Procedia Computer Science*. https://doi.org/10.1016/j.procs.2015.08.414

Beck, K. (2004). Extreme Programming explained: Embrace change. Reading, Mass. In *Addison-Wesley*.

Beck, K., & West, D. (2004). User Stories in Agile Software Development. *Scenarios, Stories, Use Cases: Through the Systems Development Life-Cycle*.

Cohen, M. (2004). User Stories Applied. *Engineering*.

Dalpiaz, F., & Brinkkemper, S. (2018). Agile requirements engineering with user stories. *Proceedings - 2018 IEEE 26th International Requirements Engineering Conference, RE 2018*. https://doi.org/10.1109/RE.2018.00075

Daneva, M., Van Der Veen, E., Amrit, C., Ghaisas, S., Sikkel, K., Kumar, R., Ajmeri, N., Ramteerthkar, U., & Wieringa, R. (2013). Agile requirements prioritization in large-scale outsourced system projects: An empirical study. *Journal of Systems and Software*. https://doi.org/10.1016/j.jss.2012.12.046

Dimitrijević, S., Jovanovic, J., & Devedžić, V. (2015). A comparative study of software tools for user story management. *Information and Software Technology*. https://doi.org/10.1016/j.infsof.2014.05.012

Jeffries, R. (2001). Essential XP: Card, conversation, confirmation. *Ronjeffries.Com*.

Kavitha, C. ., & Thomas, S. M. (2011). Requirement Gathering for small Projects using Agile Methods. *IJCA Special Issue on Computational Science - New Dimensions & Perspectives*.

Khan, M. I., Din, Z. U., Abid, M. A., & Naeem, T. (2019). User Story Characteristics Affecting Software Cost in Agile Software Development: A Systematic Literature Review. *International Journal of Computer Science and Network Security*.

López-Martínez, J., Ramírez-Noriega, A., Juárez-Ramírez, R., Licea, G., & Jiménez, S. (2017). User stories complexity estimation using bayesian networks for inexperienced developers. *Cluster Computing*. https://doi.org/10.1007/s10586-017-0996-z

Lucassen, G., Dalpiaz, F., Van Der Werf, J. M. E. M., & Brinkkemper, S. (2015). Forging high-quality User Stories: Towards a discipline for Agile Requirements. *2015 IEEE 23rd International Requirements Engineering Conference, RE 2015 - Proceedings*. https://doi.org/10.1109/RE.2015.7320415

Miranda, E., Bourque, P., & Abran, A. (2009). Sizing user stories using paired comparisons. *Information and Software Technology*. https://doi.org/10.1016/j.infsof.2009.04.003

Moreno, A. M., & Yagüe, A. (2012). Agile user stories enriched with usability. *Lecture Notes in Business Information Processing*. https://doi.org/10.1007/978-3-642-30350-0_12

O'hEocha, C., & Conboy, K. (2010). The role of the user story agile practice in innovation. *Lecture Notes in Business Information Processing*. https://doi.org/10.1007/978-3-642-16416-3_3

Okesola, J., Adebiyi, M., Okokpujie, K., Odepitan, D., Goddy-Worlu, R., Iheanetu, O., Omogbadegun, Z., & Adebiyi, A. (2019). A systematic review of requirement engineering practices in agile model. *International Journal of Mechanical Engineering and Technology*.

Rida, A., Nazir, S., Tabassum, A., Sultan, Z., & Abbas, R. (2016). Role of Requirements Elicitation & Prioritization to Optimize Quality in Scrum Agile Development. *International Journal of Advanced Computer Science and Applications*. https://doi.org/10.14569/ijacsa.2016.071239

Shim, W., & Lee, S. W. (2019). An agile approach for managing requirements change to improve learning and adaptability. *Journal of Industrial Information Integration*. https://doi.org/10.1016/j.jii.2018.07.005

Stettina, C. J., & Hörz, J. (2015). Agile portfolio management: An empirical perspective on the practice in use. *International Journal of Project Management*. https://doi.org/10.1016/j.ijproman.2014.03.008

Strode, D. (2012). a Theory of Coordination in Agile Software Development Projects. *Researcharchive.Vuw.Ac.Nz*.

Trkman, M., Mendling, J., & Krisper, M. (2016). Using business process models to better understand the dependencies among user stories. *Information and Software Technology*. https://doi.org/10.1016/j.infsof.2015.10.006

Wake, B. (2003). *INVEST in Good Stories, and SMART Tasks*. XP123: Exploring Extreme Programming.

Zhu, X. (2017). Agile mining: a novel data mining process for industry practice based on Agile Methods *and visualization. Opus.Lib.Uts.Edu.Au.*

Khan, Tayyab, Karan Singh, Mohamed Abdel-Basset, Hoang Viet Long, Satya P. Singh, and Manisha Manjul. "A novel and comprehensive trust estimation clustering based approach for large scale wireless sensor networks." IEEE Access 7 (2019): 58221-58240.

Bibliographic information of this paper for citing:

Sharma, Sarika & Kumar, Deepak (2022). Exploring Story Cards for Evaluating Requirement Understanding In Agile Software Development. *Journal of Information Technology Management,* Special Issue, 9-22.