Open Access

Full Length Article

ORIGINAL ARTICLE

Extended Scrum Process Model Using Software Reliability Engineering Concerns

Muhammad Hamid¹, Rizwan Yousaf², Saima Munawar³

¹ Ph.D. Scholar, School of Computer Science, National College of Business Administration & Economics, Lahore ²Director, Eminent Business Intelligence Cooperation, Lahore ³ Ph.D. Scholar, School of Computer Science, National College of Business Administration & Economics, Lahore

ABSTRACT

Agile software methodologies have been established and evolving since the early 1990s. Due to their short development cycle, agile software processes emphasize on rapid delivery of software. Consequently, inadequate time is spent on design and other aspects of the software reliability concerns. In this research paper, one of the most widely used agile methodology Scrum is further explored; to overcome the software reliability challenges. The primary objective of this work is to propose a new model for Scrum, by adding reliability engineering practices to achieve a reliable software system. Moreover, data collected from the software industry, Scrum experts have helped us evolve the proposed integration; it has enhanced the overall validity of this integration, as the feedback from industry experts have also been incorporated. The overall result indicated after integration that, it also has a positive impact on cost and time of developing software. This research has opened new doors to extend our work in the agile community and reliability engineering practices that can ultimately succeed in building high-quality, reliable applications.

Keywords: Agile Software, Scrum Process Model, Reliable applications, software reliability

Author`s Contribution	Address of Correspondence	Article info.
¹ Data analysis, interpretation and	Saima Munawar	Received: July 6, 2017
manuscript writing, Active participation in	Email: Saima.munawar@vu.edu.pk	Accepted: November 9, 2017
data collection		
² Conception, synthesis, planning of		
research		
^{3.} Interpretation and discussion		
Cite this article: Hamid M, Yousaf R, Munawa	r S. Extended Scrum Process Model Using	Funding Source: Nil

Cite this article: Hamid M, Yousaf R, Munawar S. Extended Scrum Process Model Using Software Reliability Concerns Engineering. J. inf. commun. technol. robot. appl.2017; 8(2):1-10. Funding Source: Nil Conflict of Interest: Nil

INTRODUCTION

The origins of agile methodologies have connected with the concepts of iterative and incremental development. Traditional development methodologies have surpassed through the partition of planning, execution and revision phases into smaller iterative cycles, and endorsing practical and measurable outputs at the end of each iteration cycle ¹. Scrum model is an agile methodology that to manage development projects through an iterative and incremental method ^{2, 3, 9}. Scrum system development has several technical and environmental variables like the variation in technology, period, and changes in requirements in each process⁴. This enables the development process towards more complicated, and changeable for system development ⁷. Because of such variations, Scrum can deliver a highly acceptable system to its users ⁸. In the meanwhile, a

elSSN: 2523-5739

major problem for Scrum developers in a changing environment is that seemingly small changes can ripple throughout the system to cause major unintended impacts elsewhere ^{3, 4}. Software developers need mechanisms to understand how a change to a software system will affect the rest of the system ^{5, 6}. For successful operational software, software reliability plays an important role 7. Software reliability is defined as the probability that a system or a capability of a system is continued to function without failure for specific periods in a specified environment ¹³. Scrum process model, has lacked support for software reliability concerns in detail such as, (i) system definition for reliability and defining its failures, (ii) calculating the user stories probabilities and their occurrence rate, (iii) reliability strategies in a system, (iv) identifying critical user stories in a system and (v) tracking reliability growth. It suggests the applicability of software reliability in the Scrum process model through a proposed model to resolve reliability related issues.

The remainder of the paper is organized as follows. In Section 2, background and related work are described. In section 3, research methodology is presented. Section 4 gives the details of the proposed model, which integrates Scrum development model with reliability engineering phases. Section 5 presents a data collection and analysis. Section 6 presents a discussion of the results of surveys and interviews conducted with Scrum practitioners in industry. Section 7 concludes the paper and gives pointers to future work.

BACKGROUND

In this section, we introduce the existing Scrum model and explain the need for reliability engineering practice.

Existing Scrum process model

Scrum introduces the concept of sprint that represents an iteration in a time-boxed development cycle with a duration of two weeks to one month. The core of Scrum is comprised of a set of sprints that delivers a working software at the end of each sprint. Sprint usually extends from one to four weeks. Also, it defines three roles (Scrum master, product owner, and Scrum team), two artifacts (product backlog, sprint backlog). Moreover, Scrum process holds a couple of meetings to enhance communication. Backlog refinement meeting defines the initial items of the product backlog. Sprint planning meeting defines the goals for the next upcoming sprint. A daily Scrum meeting is also carried out during a sprint, for issues discussion and collaboration ⁶.

In the meanwhile, the main issue for Scrum developers in a very dynamical atmosphere is that appearing little changes will ripple effect throughout the system to cause significant unplanned impacts elsewhere. Scrum developers need mechanisms to understand how a change to a software system will affect the rest of the system ^{5, 6}. The software reliability concerns play a vital role in handling these changes7. The Scrum process model lacks the support to manage software reliability concerns in detail such as; system definition for reliability and defining its failures, calculating the user stories probabilities and their occurrence rate and tracking reliability growth. Lack of this software reliability concerns further maximizes the maintenance issues in the Scrum process model. Over the life of a software system, the software maintenance effort has been estimated to be more than 50% of the total life-cycle cost ^{3, 5, 13}. Software maintenance is an expensive process where existing software has modified for a variety of reasons, including correcting errors, adapting to different data or processing environments, enhancing to add functionality, and altering to improve efficiency ¹⁰. As software evolves, the task of maintaining it becomes more complicated and more expensive^{11, 13.} The software maintenance process can only be optimized if the precise and unambiguous information is available about the potential ripple effects of change on an existing system ¹².

The need for Reliability Engineering

The process of Software Reliability Engineering (SRE) comprises of six phases. These are described as, (i) define the product, (ii) applying operational profiles, (iii) preparing for the test, (iv) executing the test,(v) engineering the just right reliability and (vi) guide test¹³. Using SRE, these phases throughout in an iterative Scrum development process helps software development teams to deliver highly reliable software applications.

The Scrum process model lacks the supports of Software reliability concerns. These reliability concerns are (i) system definition for reliability and defining its failures in an early phase of the Scrum process model. The definition of these features in the early phase of Scrum process model increase system reliability, because

Scrum developers can easily judge the reliability of the system during testing. (ii) Calculating the user stories probabilities and their occurrence rate. The scrum process model provides backlogs to list project features, but it does not provide any ways to calculate the user stories probabilities and their occurrence rate in a system. If the probability of the occurrence rate of these user stories is included with the product and sprint backlogs phase, this can help the Scrum team during the planning and development of a system. (iii) The scrum process model lacks the ability to make Reliability Strategies for a project. SRE practices establish Reliability Strategies to deal with FIO (Failure Intensity Objectives)¹³. FIO, having a quantifiable definition of reliability. It is the key to being able to measure and track reliability during testing as a means of helping decide if you have reached that sweet spot in testing not too early or not too late. When it is time to release your product. Integration of software reliability strategies in a sprint-planning meeting, increase the overall reliability of a system. (iv) The most critical user stories in a sprint execution are not handled in detail with respect to reliability concerns in a current Scrum process model. That may cause system failure. The software reliability engineering is identified most critical user stories and system failure operations. The integration of these features can also classify proper actions to decrease the outcome of these failures in a Scrum process model ¹⁴. (v) Tracking reliability growth graph misses in the scrum process model. SRE offers to track reliability growth graph, in the form of Guide test. FIO is plotted on the

graph against time to evaluate failure rates and perform according to actions to improve software reliability of the current release in a Scrum process model ^{13, 15}.

The literature analysis elaborated confront problems in Scrum and benefits offered by Software Reliability Engineering builds a case for integration for both. The main idea is to integrate reliability-engineering practices in a lightweight manner to Scrum process model with the expectation to improve reliability in a software system.

RESEARCH METHODOLOGY

The research methodology consisted of the six major steps; (1) Literature review, (2) Integration of reliability concerns in Scrum, (3) Questionnaire design, (4) Data collection and analysis, (5) Revision in the proposed Integration. The flow diagram of the methodology is given in figure1.



Figure. 1. Flow Diagram of Research Methodology

In the first phase, literature review, we explored research articles, books, and reports dealing with Scrum and software reliability concerns. The search strategy included electronic databases and manual searches of journal and conference proceedings. The following electronic databases were used; (IEEE, ACM, Elsevier, Science, Google Scholar and Springer). We searched the databases using the search terms listed in Table I. Category 1 has more keywords and showed many variations of the same term "Use of Scrum Practices." All these search items were combined by using the Boolean "AND" operator, which entails that an article that focuses on both Scrum and Software Reliability Concerns is retrieved. That is, we searched every possible combination of one item from Category Type 1 AND Category Type 2. The search excluded articles that address editorials, prefaces, discussion comments, news, and summaries of tutorials, workshops, panels and poster sessions. We only searched for papers that are written in English and available online.

We could not able to find any publication that is directly dealing with reliability engineering concerns in the context of the Scrum process model.

3

TABLE 1. Search terms used in the review				
Туре	Category	Keywords		
1	Use of Scrum	Agile		
	Practices	Agile Practices		
		Agile Methods		
		Scrum		
		Scrum Practice		
		Scrum Method		
		Scrum Methodology		
		Scrum Process		
2	Use of Software	Software Reliability		
	Reliability	Engineering		
	Engineering	Software Reliability		
	practice	Engineering practice		
		Reliability Concerns		
		Reliability Practices		
		Software Reliability		
		concerns		

The second phase, integration of reliability concerns in Scrum, consists of two steps, including (1) Analysis and (2) integration. In the analysed step, we examined in depth the Scrum process model and software reliability concerns. Our goal was to integrate the reliability concerns in the Scrum process model. For this purpose, firstly we identified the Scrum practices that impact the reliability of the application. We found six Scrum practices. Secondly, we analyzed SRE model and identified the suitable SRE phases for integrating with Scrum practices to mitigate reliability concerns. In the second step, we integrated the SRE concerns into the Scrum process model.

In the third phase, industrial survey, we developed a semi-structured questionnaire. Our goal was to obtain feedback from IT industry regarding extended Scrum model developed during the second phase. The purpose of the questionnaire was to check the practical applicability of SRE and Scrum process model integration in the industry. By conducting this survey, the proposed integration was being evaluated by the industry experts.

In the fourth phase, result evaluation collected data from industry analysis through interviews and questionnaires is listed for analyzing reliability concerning Scrum. For evaluating the result, the chi-square test has implemented in the questionnaire responses. The chisquare test for responses has been analyzed by using the IBM SPSS 19. The combined mean, standard deviation, and variance have been analyzed. The frequency of response is also discussed in detail with valuable suggestions. In the fifth phase, revision of proposed integration was done. The change was based on the feedback by the industry experts, identification of gaps in the existing model and analysis of the overall feedback given by the people working on real-life industry problems.

Proposed Extended Scrum Model using SRE:

In this section, we present the Extended Scrum Model using reliability concerns. We developed this model by integrating SRE concerns into the Scrum process model. For integrating software reliability concerns in the current Scrum process model, we identified suitable areas for incorporating them. These areas have been carefully determined to keep in view activities performed by Scrum teams and will add value with respect to software reliability concerns. The resultant extended model is presented in Figure 2. The model consists of six phases with integrated SRE concerns. Activities highlighted with a green color (Scrum Reliability Integration [SRI-1, SRI-2, SRI-3 SRI-4 SRI-5 and SRI-6]) are those activities that we integrated with the Scrum process model.



Figure. 2. Integration of Reliability Concern in Scrum Process Model

Below we explain reliability concerns in Scrum process model in detail.

SRI-1.

SRI-1 integrates 'Define Product' phase of reliability engineering with the planning phase of Scrum. In Scrum Planning phase, system reliability and the acceptance level of system failure is not explicitly defined. Therefore, it is difficult to judge the reliability of the system during testing^{13, 15}. We have tackled this problem by integrating 'Define Product' phase of SRE with Scrum Planning phase. The 'Define Product' phase, in the Scrum process model, uses system definition for reliability and defining its failures.

SRI-2 and SRI-4.

SRI-2 integrates Operational Profiles with a Product Backlog, and SRI-4 integrates Operational Profiles with Sprint Backlog. During the planning and development time in the Scrum process model, product and sprint backlogs are generated that contains the product features lists. The Scrum process model lacks support to calculate failure probabilities of user stories in product backlog and sprint backlogs. Integration of 'Implementing operational profile' phase, of SRE ^{13, 14} in product and sprint backlog phase resolves this issue. Using the concept of operational profile helps in developing user stories along with their probabilities of occurrence rate. Implementing operational profiles in the product and sprint backlog phase is also helpful in optimizing development and testing resources.

SRI-3.

SRI-3 integrates Engineering Just Right Reliability with Sprint Planning Meeting. During the sprint planning meeting, selective functions are picked for development with a development strategy on how to build the sprint.

The Scrum process model lacks the ability to properly identify reliability strategies during the sprint meeting. SRE evaluates reliability measures in 'Just Right Reliability' phase ¹³. In 'Just Right Reliability' phase of SRE establishes Reliability Strategies to analyze and calculate FIO (Failure Intensity Objectives). As a project proceeds in Scrum, the obtained operational profile list can help to set failure intensity objectives. The obtained FIO can assist in defining reliability strategies, which can reduce overall system time and cost of a project.

SRI-5

SRI-5 is the integration of Prepare for Test and Execute Test with Sprint Development. Each sprint in

Scrum follows Develop, Wrap, Review and Adjust cycle. Scrum creates user stories, test cases and executes test cases. During the Develop phase of this cycle, user stores are tested. To address critical user stories regarding system failure, Scrum team can incorporate 'Prepare for Test' and 'Execute Test' from SRE¹³. Test cases can be prepared and executed for most critical user stories listed under the operational profile in a product backlog and sprint backlog. During the review phase of Scrum development teams and management, people meet and review work completed. New test cases can be added to meet failure intensity objectives ^{6, 13}.

SRI-6.

SRI-6 integrates Guide Tests with Sprint Review and Shipment. If the Scrum process model applies reliability measures, it needs visibility on tracking reliability increase or decrease. This feature is offered in the form of Guide Test. During the closure phase, a software increment is released is to the customer. Here SRE 'Guide Test' can play a vital role in helping in tracking reliability growth, rehearsing the acceptance test and acceptance or rejection of the release. Failure intensity and failure intensity objectives ratios (FI/FIO) are plotted on a graph against time to evaluate failure rates and comments to perform improvement actions are made during guide test ^{13, 15.}

Data Collection and Analysis:

A comprehensive survey of industry professionals was conducted during this study to strengthen our research. The purpose of the survey was to discuss the proposed integration in Scrum process and get the valuable feedback from the industry experts. The facts collected from the survey helped us to incorporate the software reliability with the current Scrum process model and the overall quality of the proposed model was improved. The focus of most of the questions asked in the survey was to judge the impact of integration on the Scrum process model using reliability engineering concerns on overall system cost and time.

Survey Participants

We selected professionals from 20 different organizations who are currently using the Scrum process model in their projects. The participants were asked to share their experiences in implementing Scrum process model using software reliability concerns. The 13-survey

5

participant belongs to web domain; five belong to ERP and two belong to the desktop application. The participants of this study included Scrum masters, Product owners. The role of each person is shown in the Figure. 3. Names of companies are not disclosed to keep their privacy.

RESULT AND DISCUSSION

In this section, we will discuss the findings of the survey, analyze them and identify the gaps in our proposed addition to the Scrum process. For ease of analysis and interpretation the questions were classified into two categories, namely, time (T) and cost (C), description of each category is given below.

T: Purpose of this category was to identify the impact of software reliability concerns integrated into Scrum on time to market.

C: Purpose of this category was to identify the impact of adding in the Scrum process on the overall cost of the project.

Questions asked responses received, findings and analysis are given in the following sections (6.1 to 6.2) of the paper.



Figure 3. Extended Scrum Model Survey Participants Analysis of Time to Market Questions

T1. What impacts on the delivery of the product, when requirements are decomposed into business values and implement Operational Profile in product & sprint backlog phase?

Through this question, it had investigated the overall impact on project's time, when the scrum team further sub-divided features into business value and implementing the operational profile in a product and sprint backlog phase. The analysis shows that if we decomposed the features into the operational profile and business value, the overall project time to market would decrease. The respondents indicated that implementing the operational profile adds more value to increase productivity. Because implementation of the operational profile has a positive impact on the performance of a developer because, if a proper guidance is given to the developers at the start; it will have a positive effect on their performance and thus consequently reducing the overall development time. According to the feedback of (30 %) respondents, the total time to market the product will increase. This is due to the inclusion of the operational profile because Scrum is a lightweight process, adding operational profile will make it heavier and more time consuming, and thus the essence of using the agile process model will be lost. 10 % respondents indicated that implementing the operational profile will not affect on time to market because according to their experience, the focus is the execution of user stories and they use different tools that automatically check the dependencies among the user stories. Major feedback showed that implementing the operational profile in product backlog and sprint backlog phase has a positive impact to decrease the overall time to market of developing new software through the proposed model.

T2. What is the impact of implementing the operational profile on overall project time, if later change in code or major bugs fixed?

This question was designed to check the overall impact of the change in the code and fixation of major bugs on the project time. According to 70 % respondents, the time to market will decrease, if implementing operational profile is developed and followed, and a major bug fix is required. According to the feedback, there are many tools available that can handle the code level dependency, but an implementing operational profile can better manage the change in requirement, especially in a case where the development team is changed, i.e., old team members leave the organization, and new members join the development team. In this scenario, implementing operational profile will serve as a better tool for developers to fix the problem or enhance the functionality of the system within minimum possible time. According to feedback from some respondents, they feel that they can better judge the critical user stories through implementing operational profile. According to one respondent the fixation of defect which is detected late and the project is

elSSN: 2523-5739

a large-scale project may cost hundreds or thousands of dollars, so he agreed to develop the implementing operational profile at the beginning and appreciated the inclusion of this list in Scrum. Similar was a response of another expert that creation of implementing operational profile at the beginning saves a lot of time and maintenance cost, according to him this was because implementing the operational profile is a pattern to carry out functions that are mostly used or critical for successful completion of the project. According to two respondents, implementing the operational profile helps the developers to handle later change in code or when a major bug fix is required. 30% responded indicated that they are aware of the critical user stories and they have the knowledge and expertise to handle the changes that occur late in the development cycle or major bugs that are detected late. Hence, from the above feedback, it can be inferred that inclusion of implementing operational profile reduces the development time and consequently time to market.

T3. What is the effect of following the software reliability strategy in the early phase of defining product and sprint planning meeting phase?

The objective of this question was to check the impact of software reliability by following the reliability strategies in early phase and sprint planning meeting. The analysis showed that by developing the reliability strategy in the initial phase and sprint planning, meeting decreases time to market. Because in second or next iteration this strategy helps the developers to handle the critical features better. Some Scrum experts share that, if critical features are identified in an early phase, it allows the Scrum developers to emphasize on these features. 20 % respondents said that time to market the product would increase. Because Scrum is a lightweight process and adding these steps in the early phase of the Scrum process, maximize the efforts of the developers that will increase the time to market. 20 % Scrum experts also share their experience that Scrum is a lightweight process and Scrum experts have cross-functional skills, so they need a new role that handles these tasks. Hence, it can be concluded that addition of software reliability strategy in the early phase of defining product and the sprintplanning meeting will decrease the time to market.

T4. What is the impact on total duration of the project, if we properly follow the testing strategies at the

beginning of the product backlog and Sprint backlog phase through implementing the operational profile?

Through this question, the objective was to investigate the effect on the overall project time, if the testing strategies are followed adequately at the beginning of the product backlog and sprint backlog phase using implementing operational profile. The results show that after proper implementation of testing strategies at an early phase of product backlog and sprint backlog phase, time to market will decrease. This answer was given by 50% of the respondents. According to the industry experts, this is because when the testing strategy is added at the beginning, the developers are able to execute the most frequent operations; this consequently reduces the overall project time. This also increases the customer satisfaction as most of the critical functions are tested at an early stage of development. According to three respondents, implementing operational profile helps the developers to plan test cases for most vital operations. 30% respondents share their experience that different testing tools are used to check the dependencies among the features or operations so this change may have no impact. An important point to note is that tools do not elaborate and identify the critical or most frequently executed operations. One respondent indicated that they have an independent quality assurance team that handles testing related issues at the beginning. According to 20%, respondent's time to market will increase if testing strategies are executed at the beginning with the help of implementing operational profile.

From the above results, it can be inferred that implementing the operational profile has a positive impact on the project's time and the overall time of the project will be reduced, and the reliability will be improved. Additionally, it was seen that by following the testing strategies at the beginning of the product backlog phase and sprint backlog phase, customer satisfaction is increased and the overall time to market is decreased. The combined frequencies have been presented as shown in Figure 4.



Figure. 4. Overall Time to Market

ANALYSIS OF COST TO MARKET QUESTIONS

C1. What effects on project cost, when following proper engineering just right reliability at the define product phase and in sprint planning meeting phase?

Through this question, we wanted to investigate the impact on the overall project Cost, if engineering just right reliability, are followed at the beginning of the product backlog phase and in sprint planning meeting phase. The results show that proper implementation of the engineering just right reliability at early phases of defining product phase and sprint planning meeting phase will decrease overall project cost because Scrum developers emphasize more on critical or most frequently used features. By adding it in Scrum process, maximum test scenarios are checked on critical or most frequently used functions, and in the second iteration, it is checked that objective set regarding reliability are achieved in the first phase or not. Additionally, this helps in better planning for coming releases. The average cost to market is 10%, and higher cost to market is 10% because its take more developer's time of development to the engineering concerns regarding reliability in the early phase of development. Hence, by following the "engineering just right reliability" will decrease the overall cost of the project and reducing the chance of failure.

C2. What is the impact of implementing the operational profile on overall project cost, if later change in code or major bugs fixed?

This question was designed to check the overall impact of changes in the code and fixation of major bugs on the project cost. In this question, the objective was to see the effect of decomposed requirements in to implement the operational profile on overall project cost, if later change in code or major bugs fixed. The results show that if the requirements are decomposed into an operational profile that better handle the later change in code. Because in the current Scrum development process, there is no producer to handle the later change in code. According to 30% respondents, the developing application cost will increase because implementing the operational profile is overhead to Scrum experts.

C3. What effects on the overall cost of the project that includes maintaining cost after implementation of an operational profile in product backlog phase and sprint backlog phase?

In this question, we wanted to investigate the impact on overall project cost, if executing operational profile in product backlog and sprint backlog phase. After analysis of collected data, it is found that after implementing the operational profile in product backlog and sprint backlog phase, customer satisfaction will be increased, and overall project cost will be decreased. Because in later development iteration the Scrum team gets help from that operational profile to maintain software in a better manner. The operational profile gave a better view of the developer to maintain the coding relevant issue through that operational profile. 20% respondents give their belief that the operational profile increase cost of the project because the operational profile adds more effort and resources to the project.

C4. Which type of result occurs, when we decomposed the features into the operational profile and adding reliability features to those operations that are most critical or most frequently execute operations?

The purpose of this question was to check the impact on overall project cost if implemented operational profile that handles the dependencies and complexities among the requirements and handles which operations are more critical or more frequently executed. The analysis showed that after creating the operational profile that manages the dependencies and complexities among the requirements, the overall project cost would be decreased. According to respondents, it is easy to judge the critical and more frequently executed operations through implementing the operational profile, consequently having a high impact on business value. If these critical operations contain errors that arise due to miscommunication by the client, then it can severely affect the project's time and increase the maintenance cost of the project. Implementing the operational profile at the beginning provides test scenarios to be executed at the start, which helps us to avoid the problems at the later stage of development. It is

important to mention that implementing operational profile also indicates which operations are more critical or most frequently executed; this maximizes the customer satisfaction. The combined frequencies have been presented as shown in Figure 5.



Figure. 5. Overall Conclusion of Cost

The combined mean, standard deviation, and variance have been performed on the left column of ordinal questions of cost and time as given in Table2 and Table 3.The chi-square test has implemented in the interview responses and tested the hypothesis that the means of four variables have equal or significant. The chi-square test for responses has been analyzed by using the IBM SPSS 19 as given in Tables 4 and Table 5.The results showed that the model has significant output.

Table 2: The mean, standard deviation and variance on C1, C2, C3, C4					
C1 C2 C3 C4					C4
N	Valid	20	20	20	20
	Missing	0	0	0	0
Mean		2.70	2.00	3.20	2.70
Std. Deviation		.657	1.124	.410	.470
Variance		.432	1.263	.168	.221

 Table 3: The mean, standard deviation and variance on T1, T2, T3, T4

		T1	T2	T3	T4
Ν	Valid	20	20	20	20
	Missing	0	0	0	0
M	ean	1.50	1.60	1.40	1.80
Sto	d. Deviation	.688	.940	1.046	.894
Va	ariance	.474	.884	1.095	.800

Table 4: Chi-Square Tests on C1,C2,C3,C4				
	C1	C2	C3	C4
Chi-Square	19.600	8.800	7.200	3.200
df	2	3	1	1
Asymp. Sig.	.000	.032	.007	.074

Table 5: Chi-Square Tests on T1, T2, T3, T4				
	T1	T2	Т3	T4
Chi-Square	7.600	3.200	2.400	2.800
df	2	1	3	2
Asymp. Sig.	.022	.074	.494	.247

From the above results and discussion, it is found that if features are decomposed to be operationally focused and business valued, the overall project cost decreases. In Scrum practices, changing requirement is a baseline assumption. The implementation of operational profile explains what the customer wants. There are different tools available that describe the dependencies among the user stories, but these tools do not identify the most critical and more frequently executed user stories. In Scrum practices, change in code or major bugs fixed will result in a ripple impact on applications that have a high impact on cost using operational profile, by using our integration this effect will be minimized. The analysis shows that after creating the operational profile, that handles the dependencies and complexities among the requirements, will decrease overall project cost and time. Implementing operational profile also indicates which operations are more critical or most frequently executed, which maximizes the customer satisfaction.

When the requirements are decomposed into an operational profile that handles the dependencies between the requirements and later changes in code, maximum test cases are developed for critical and most frequently executed operations. The overall results show that decomposition of features into operational profile, proper implementing the software reliability concerns at an early phase of defining product phase, and sprint planning meeting phase will decrease the overall cost of the developing new applications.

CONCLUSION AND FUTURE RECOMMENDATIONS

Scrum process models lack support for the reliability engineering process. The main finding of this research has integrated Scrum process model using software reliability engineering practices. A Survey has been conducted through a questionnaire, which shows that overall performance of the Scrum team increases by adding the reliability concerns in the Scrum process. The

elSSN: 2523-5739

reliability concerns maximize the reliability of the product and fewer chances of the occurrence of errors. Therefore, it also decreases the overall cost of the product, but it increases the time of first few releases. In a later version, it reduces the total time to market.

The survey, conducted by industrial software experts showed a positive attitude towards our work, and it has been an excellent source of admiration and motivation for us. The survey results are almost positive with respect to time, cost, and efforts in applying reliability concerns in agile based organizations. From industry, collected facts researchers concluded that by adding reliability concerns in Scrum reduce the overall cost of the project and decrease time to market. Experts also shared new dimensions and thoughts to extend this work that will continue the evolution of Scrum based software development improvement and extensibility. Though Scrum team's survey was quite helpful to conclude the extended scrum process, the empirical study still remains to be carried out in order to measure extended Scrum process validity. In this regard, we need to gather empirical data on a few agile products built using extended Scrum process. Moreover, this arises further many research directions, i.e., which measurements need to be done to support this model, how does the model perform in a real-world industry setting, how much empirical findings match with the industry expert opinions found in this research?

REFERENCES

- López-Martínez, J., Juárez-Ramírez, R., Huertas, C., Jiménez, S., & Guerra-García, C. (2016, April). Problems in the adoption of Agile-Scrum methodologies: A systematic literature review. In Software Engineering Research and Innovation (CONISOFT), 2016 4th International Conference in (pp. 141-148). IEEE.
- Beck, K., Beedle, M., Van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., ... & Kern, J. (2001). Manifesto for agile software development.
- Hernández, L., Vite, N., Alvarez, F., & García, A. R. (2016). Situational Factors Which Have an Impact on the Successful Usage of an Agile Methodology for Software Maintenance: An Empirical Study. In Trends and Applications in Software Engineering (pp. 99-109). Springer, Cham.
- Harvie, D. P., & Agah, A. (2016). Targeted scrum: Applying mission command to agile software development. IEEE Transactions on Software Engineering, 42(5), 476-489.
- Schneidewind, N. F. (1987). The state of software maintenance. IEEE Transactions on Software Engineering, (3), 303-310.
- 6. Sillitti, A., Ceschi, M., Russo, B., & Succi, G. (2005, September). Managing uncertainty in requirements: a survey in

documentation-driven and agile companies. In Software Metrics, 2005. 11th IEEE International Symposium (pp. 10-pp). IEEE.

- Day, I. I., & Jesse, H. (1996). An Investigation of Software Metrics Effect on Cobol Program Reliability (Doctoral dissertation, Virginia Tech).
- Schwaber, K. (1997). Scrum development process. In Business object design and implementation (pp. 117-134). Springer, London.
- Larman, C. (2004). Agile and iterative development: a manager's guide. Addison-Wesley Professional.
- Harrold, M. J., & Malloy, B. (1993). A unified interprocedural program representation for a maintenance environment. IEEE Transactions on Software Engineering, 19(6), 584-593.
- Pfleeger, S. L., & Bohner, S. A. (1990, November). A framework for software maintenance metrics. In Software Maintenance, 1990, Proceedings. Conference on (pp. 320-327). IEEE.
- Moreton, R. (1990, June). A process model for software maintenance. In Journal of information technology (Vol. 5, No. 2, pp. 100-104). Palgrave Macmillan UK.
- Musa, J. D. (2004). Software reliability engineering: more reliable software, faster and cheaper. Tata McGraw-Hill Education.
- 14. Yang, G. (2007). Life cycle reliability engineering. John Wiley & Sons.
- Dhillon, B. S. (2007). Applied reliability and quality: fundamentals, methods, and procedures. Springer Science & Business Media.