Proceedings of the Pakistan Academy of Sciences:
A. Physical and Computational Sciences 53 (4): 417–429 (2016)
Copyright © Pakistan Academy of Sciences
ISSN: 2518-4245 (print), 2518-4253 (online)

Pakistan Academy of Sciences

Research Article

An Empirical Study and a Framework for Effective Risk Management in Scrum

Muneeb Ali*, Yasir Hafeez, and Bushra Hamid

University Institute of Information Technology,
PMAS-Arid Agriculture University, Rawalpindi, Pakistan

Abstract: Now a day, agile methods are broadly used for software development. The agile methods are expected to provide virtuous outcomes and producing better quality software products that achieves the customer requirements. In view of the contemporary scenario, it is clear that secure and better quality software products are foremost apprehension. This research study deals with risk management within the scrum framework. The purpose of this research was to propose and validated a framework developed that produces quality product. The continuous approach of scrum overlooks the risk issues which can result in changes and cost expansion. To mitigate this risk, a free scrum model is proposed. This model is produced by combining the activities of risk management and in scrum methodology. A case study has been employed to evaluate proposed framework for mitigating risks effectively in scrum process. We used a qualitative approach with structured interviews, to validate the proposed work. We have explored both the existing principle theory for risk management and the results of different empirical studies to build the framework. On the said base, we have drawn up vindicated proposals for the framework. Results of case study has shown that the proposed framework is suitable for the developing a quality software product. By employing the risk management activities into scrum methodology, as per proposed framework, there is a promising scrum model to control risk. This also ensures software quality along with benefits of cost reduction, experience gained and customer satisfaction. This framework has implications with the effective risk management in scrum way of development and provide valuable insights for risk management scrum. The case study provides direction for future research and lesson learned. It will also provide assistance to apply effective principles of risk management in scrum to develop high quality software product. Our future research will be directed toward the generalization of this framework. The proposed framework activities will be applied on different agile methodologies and other case studies will be conducted, so the results can be generalized.

Keywords: Agile, scrum, software quality, risk management, risk register

1. INTRODUCTION

Mainstream organizations are improving the process of software development in order to achieve better quality products. Scrum is considered the best agile methodology for agile application development [1]. The scrum focuses on continuously delivery of the software product. It is an iterative and incremental agile software development model for managing development of software products. Scrum is more concerned with deliver software in short term. This result in compromising the quality of product by ignoring the risk and security issues identification and mitigation.

In traditional software development, the risk management is done by applying different techniques and tools. These tools and techniques restrict the decision making regarding the risks [2]. The main objective of risk management is to determine the potential issues before it occurs, so its avoidance or mitigation can be planned accordingly. The objectives can be achieved by handling the problems as per required across the life of the software development. Risk management is manifold and complex in traditional software development [3].

Agile way of software development may process

Received, April 2016; Accepted, November 2016

*Corresponding author: Muneeb Ali, Email: muneebalikiyani@gmail.com software faster but how they cater our quality requirements. It provides the ability to respond quickly to change, frequent deliveries of working software and close customer collaboration On the other hand, this agility can cause overlooking the potential threats. We know from traditional software development that risk registers is an approach to maintain quality in the software product. The technique is effective with the traditional software development, because the gathered in the beginning of project. In agile, due to close collaboration the requirement changing become a major loop hole for risks.

The recent trend shows that scrum is the most employed agile method of software development. The purpose behind using scrum model is to deliver the required software to the customer by making teams that work in short cycles, iteration by iteration. Scrum is more concerned with the project management and expects that the self-organizing team pulls any needed practices into the process via the mechanism of variation.

In Scrum software development, the security and risk management are not considered essential from the start of development process. Security of product has become an essential part of the end product. It can be concluded that risk management from the first phases of development should be introduced. The process of risk management should continue throughout the development cycle. Scrum development framework emphasis on providing the maximum benefits with in less the time. Effective risk management can be achieved by employing these activities:

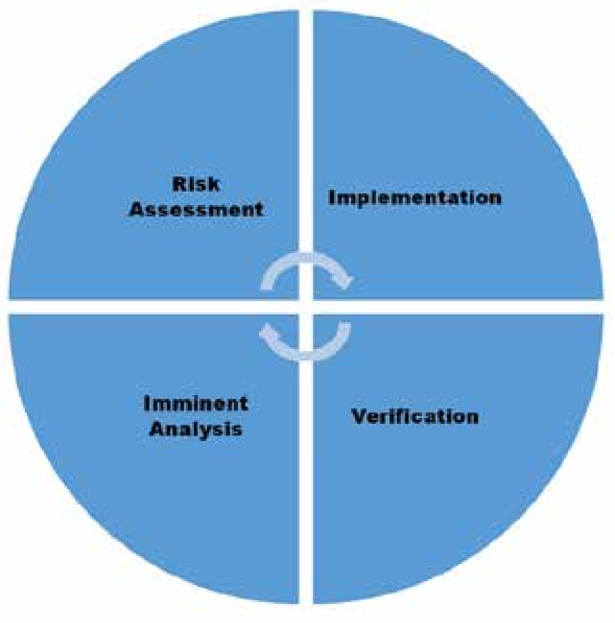

Fig. 1. Effective risk management.

1. Identification of risk, risk analysis and prioritizing the identified risks on severity basis.
2. Planning and Implementation.
3. Monitoring and Review to verify risk are treated and removed.
4. Imminent Analysis of risk.

1.1. Research Objectives

Software development projects are exposed to risk like incomplete requirements, non-traceable requirements, time limitations, unrealistic schedule, communication and technology change [4]. These issues of project development can be addressed by applying effective risk management.

It could be argued that the technical issues are of less important than managerial issues in software development projects. The reasons for the failure of projects are usually management foibles rather than technical mistakes [5].

Risk can be identified in scrum but the cause of risk, factors to evaluate risk and practices to handle the risk, cannot be identified [6]. These arises a need to develop a framework for scrum method that incorporates effective risk management process to handle the above mentioned issues.

2. LITERATURE REVIEW

Risk is usually considered as possibility of loss. Shapira (March, 1987) presented the view of risk as variation in possible outcomes. Risk is considered for negative outcomes of the project by 80 percent of the manager [1]. The probability of risk occurrence and impact of risk are the factors that enhance the priority for risk handling.

According to Janus et al. [7] there seems to be no traditional Quality Assurance in Agile Software Development, even though Agility promises to deliver high Quality Software. The lack of quality assurance in agile methodology enhances the probability of risk occurrences. Alharbi et al. [8]

explored that risk register should be used with in scrum in order to improve quality of the product. They claimed that use of risk register doesn't affect the agility of a project. The risk assessments of every sprint will be maintained in the risk register. The register will be used to monitor the occurrence of risk during every sprint.

Panday et al. [9] described a model for the risk management in the software development process. The presented risk management model controls the known and unknown issues or risks during the development process of software. Pohl et al. [10] presented a secure model for scrum. The security issues should be considering during the course of entire process of development. Their model emphasis on incorporating security issues into practice without affecting the principles of scrum method.

Wanderley et al. [11] conducted a study which support the fact that risk management contributes toward the success of the IT project. The study was conducted on publications from 1999 to 2007 and selected 29 publications with empirical data. After analysis of those research studies it was concluded that risk management has impact on the IT projects.

Offshore software development has become very popular because of its cost effectiveness. It benefits by pooling labor from countries having low wages. Islam et. al. [12] has addressed to the threats associated with this trend. There are certain challenges to overcome such as geographical, communicational and cultural differences. The author has proposed tailored risk management framework to overcome these risks. The researcher proposed that risk should be assessed and managed at earlier stages, so its size of loss will be nominal. For the said purpose, goals are linked with the risks in a relational model. For goals KAOS extension has been employed. For validation, case study has been conducted on framework (GSRM). The results showed better management of risk after integration of framework.

According to Ylimannela et al. [13], Agile development is based on short iteration cycles, which allow and respond to changes in business environment. Using agile development is itself risk management at project level. He has created a model to manage risks in agile development environment. The suggested model has been proposed in order to address the problem arose during the interviews. The suggested model is based on existing models and interviews.

For managing risk in a formal way the team, the product owner or Scrum master can:

1. Use burn down chart for risk [13].
2. Prioritize the outrageous risk requirements first in the upcoming sprint [14].
3. Risk board can be used with two colors of notes [15].
    i. Red notes: Describes the risks
    ii. Yellow ones: Describes risk responses
4. Employee a risk registers [16].

2.1. Critical Factors

There are different dimensions in which risk can be classified such as organizational, people, process, and technical.

1. Organizational Factor
    i. Cultural differences
        a. Too traditional
        b. Too political
    ii. Large organizational size
    iii. Lack of commitment or proper management
    iv. Lack of logistical support
    v. Intellectual property rights
2. People
    i. Lack of expertise
    ii. Lack of project management competency
    iii. Lack of team coordination
    iv. Conflict of individuals / groups
    v. Depraved customer relationship
3. Process
    i. Undefined scope
    ii. Undefined requirements
    iii. Lack of frozen/ agreed requirements
    iv. Improper planning
    v. Lack of progress tracking mechanism
    vi. Lack of customer involvement
4. Technical
    i. Lack of ample set of correct scrum practices
    ii. Inappropriate use of technology and tools

3. MATERIALS AND METHODS

Risk management hold worth, if it is not only identified and communicated in start, but also, a proper sequence of activities should be involved throughout the development cycle. Our finding from the literature review conducted on the base of below mentioned research questions have indicated the need of consolidated conceptual framework.

3.1 Research Questions

RQ1: How the Risk management is integrated in scrum methodology?

RQ2: What are the critical factors for effective risk management procedures in scrum methodology?

RQ3: What standards and practices are employed in scrum methodology for risk management?

By addressing these research question, we are able to identify the essential activities of risk management that can amalgamate with scrum methodology. The identified critical factors has enlighten the existing challenges to be addressed. The outcomes of RQ3 are the current industrial practices of risk management in scrum.

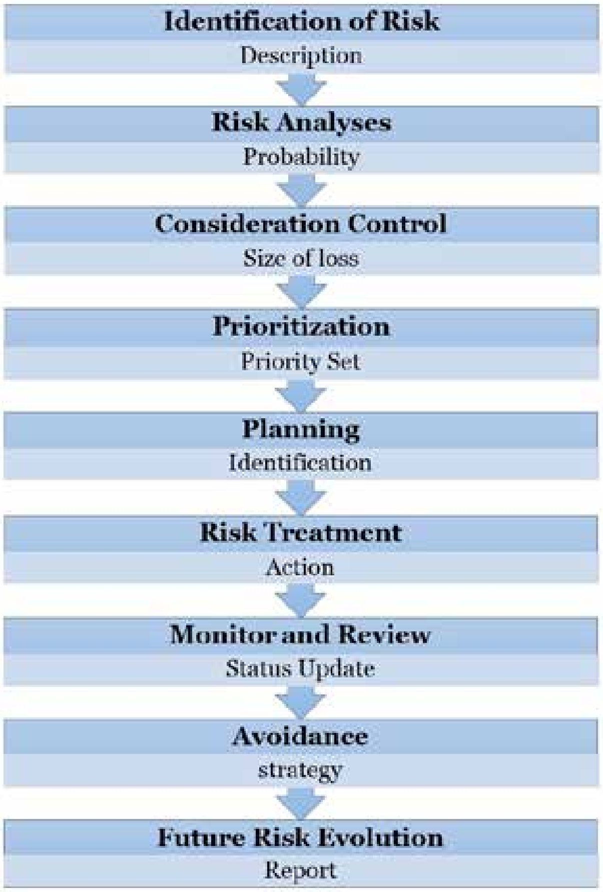

Fig. 2. Risk management activities.

In this research study we have proposed a framework that helps the software engineers to develop a quality product by managing risk. To achieve this, we have determined a complete process of risk management mapped on the scrum. Our proposed model involved of four major activities for risk management.

Given below is a brief description of the said above risk management activities.

3.2. Risk Assessment

It is one separate thing to identify and outline list of potential issues, whereas it is an entirely different matter to address them. This is where assessment comes to play its role.

The identification activity in product backlog stage of scrum framework is used to detect the security and risk issues. The identified issues are listed and tagged as well.

The vision of customer is converted into manageable chunks and prioritize in the product backlog. A short description regarding the identified issue is sorted in the register. During the refining of the product backlog these issues (identified during the product backlog) are analyzed for the validation of potential risk. The potential issues are then priorities according to their impact. The issues related to security and risks are marked by the tags.

3.2.1 Identification of Risk

The base step of risk assessment is risk identification. The team of stakeholders review all items of backlog within the scope of project. The review is formulated from different perspective of various categories of risk. These results in a list of identified potential risk that could have a significant negative impact on the success of project.

The risk that can effect project goals, are identified, classified and report these risks. The outcome of identification process is a list of risks [14]. The resultant risk list depends on project and the environment. For small, noncomplex projects (low-budget projects), there are few risks with little ambiguity. For large, complex projects (high-budget projects), there are effect by uncertain environment. The risks can provide for the risk assessment and risk control process. The mitigation of these risks can be performed by listing and flagging each item with color. These colors are assigned according to the priority of the risk.

3.2.2 Risk Analysis

When the risks have been documented and all items are analyzed. The cause of the risk analysis is to evaluate the loss possibility and magnitude of each software risk item. The contribution is the software risk statement and situation developed in the appreciation phase.

After the completion of risk assessment, risk analysis is conducted to identify the chances of occurrence and, if so, when the risk is likely to occur in the overall time-line of project. There are several conventional methods that can be employed for the risk analysis, such as cost risk analysis, reliability analysis, decision analysis, and schedule analysis.

3.2.3 Consideration Control

Consideration Control is calculation of losses. Possible action to reduce or remove such threats are considered. Possible risk factors are identified in each item, for instance business, technical and nontechnical features of product and any other areas that effect the goals of product development.

3.3 Implementation

During every Sprint of scrum, Sprint planning results into two artifacts; sprint goal and sprint backlog. During the sprint planning meeting the solutions for tagged issues are identified. The identified solution is listed according to the tag no assigned to the issue. The consideration control on marked issues is done to evaluate the size of loss. The size of loss expected in result of that issue entered in the register. Finally the issues are treated accordingly the solution identified in the planning meeting.

3.3.1 Prioritization

The integrities of identified risk can be identified by the prioritization of risk. Exposure is the product of the possibility of incurring a loss due to the risk and the potential magnitude of these losses. On complex and large projects i.e. high-cost projects that are usually environment uncertain.

It would be much difficult, to provide a plan or strategy for catering the effects of potential risk, in every phase of the project. By assigning each risk, with a risk priority value, the stakeholders now have a road-map for catering threats. Risk effects are catered by contriving contingency plans for the task with highest to the lowest risk priority factor.

3.3.2 Response Planning

The approaches to deal with risk are identified in this step.

Three strategies for risk planning are introduced:

1. Avoidance: Attempt to minimize the possibilities of risk [17].
2. Minimization: Attempts to decrease the impact of risk.
3. Control: Actions are implemented to reduce the impact of the risk.

3.3.3 Risk Treatment

The strategy is used to mitigate consequence of risk acceptance and transfer. Risk treatment is related to tendency for taking risks. Behavior towards taking risks may change over time through training, education and experience. The threshold of taking risk by organization depends upon the stable risk treatment. As a consequence, competitive control of organization may increase.

3.4 Verification

During the daily scrum meeting the rectified issues are monitored and reviewed. Status of the issues is updated during the meeting.

3.4.1 Monitor and Review

In response of every risk item monitoring and review take place. This tracking helps to achieve the goals of risk management processes. Execution of risk management is ensured by evaluating the risk treatment activity is performed throughout the project development.

3.5 Imminent Analysis

3.5.1 Future Analysis

During the sprint review this is ensured that the issues identified at start of the scrum are catered. If the treated issues are no longer threat for future, a brief report is prepared consisting of the issues that occurred in each sprint. If the same issue has any

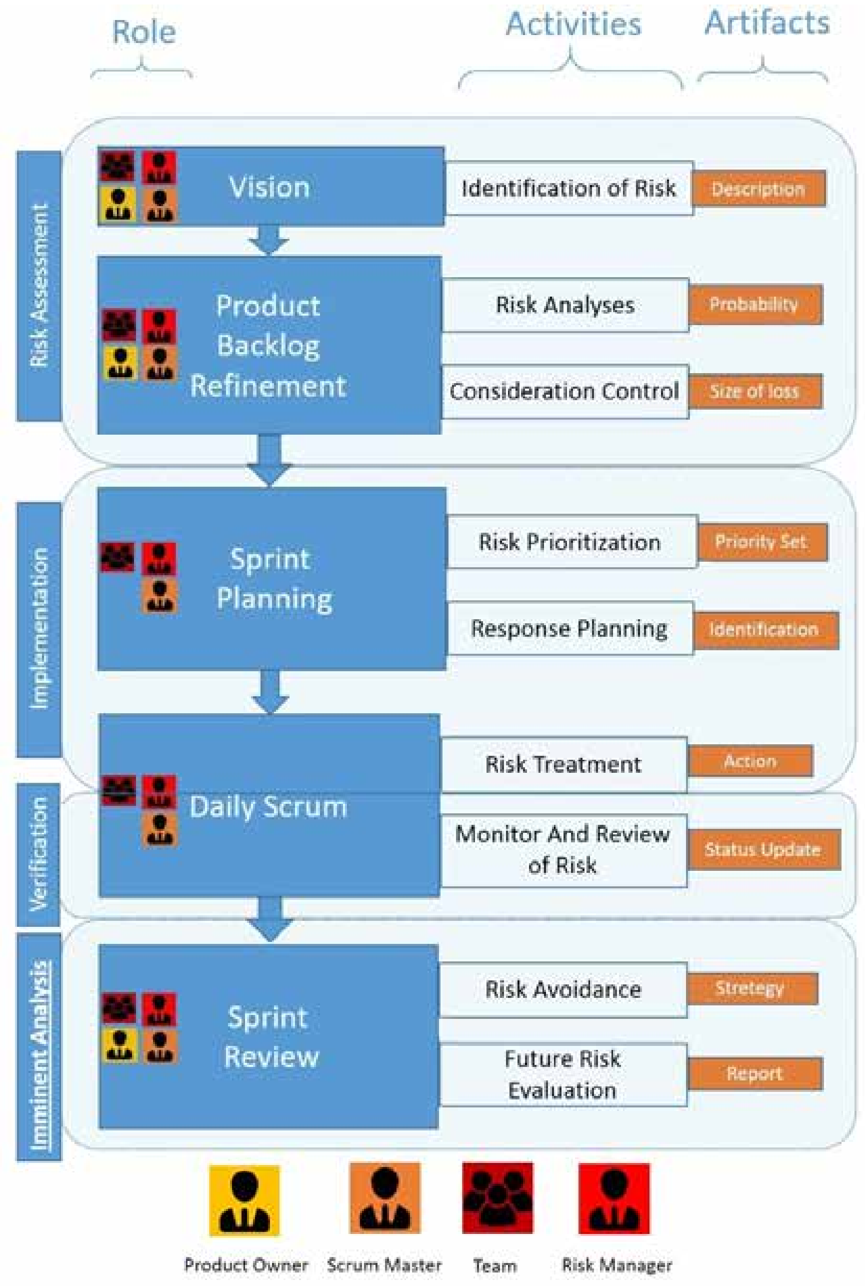

Fig. 3. Risk management in scrum.

chance of reoccurrence in sprint, the team refers the issue to next sprint1 and so on.

3.5.2 Avoidance strategy

It is known that the most efficacious risk avoidance plan is to establish effective communication throughout the life cycle of project. Oftentimes, scrum master fail to keep the entire stakeholder in the loop about the project.

Formal risk avoidance methods depend upon understanding the user requirements, obtaining domain information and effective communication [18]. These factors ensure that the plan will achieve project objectives.

3.6 Future Risk Evaluation

Directions for future occurrence of risk will be discussed and gap of the risk treatment are identified. All the information gathered during the development process is analyzed to predict the chances of occurrence of each risk item. At the time of sprint review and retrospective of scrum, a report is prepared then all the learned instructions are acknowledged. This information stored in the information repository for making decision in future projects.

In Scrum software development the risk management is not considered essential from the start of development process. The proposed framework mitigates the risk after performing proper assessment, its implementation, verification and analysis for future aspects. Fig. 2 represents the proposed model.

Every activity performed in scrum contributes toward the management of risk. The risks will be identified in the vision activity of scrum. Followed by, analyses of the identified risk items. During analysis the risk occurrence, the size of loss and its priority is determined. These activities come under the risk assessment phase and are performed during the creation and refinement of product backlog. The prioritized risks are then treated according to a proper plan framed during the daily scrum. The treated risk are then review to confirm its removal in daily scrum meetings. The review is conducted on daily basis so its effects can be monitored, and in-case plans for mitigation can be altered accordingly. This will minimize the cost of treating the risk. The sprint review is open for more risk to be identified and listed accordingly with the retrospective. The progress of the project is then shared with stakeholders.

3.7 Validation

Case study has been used as a research method [19]. While designing the study, we emphasized on human sense making and how the mechanisms of risk management were understood by the participants involved. The research method is selected to collect experiences which could be used to improve performance of the corresponding projects. The project selected under this umbrella is a new development of company's internal web application. The web application is collection of several different level of tracking phases.

The data from stakeholders and project participants was collected by conducting semi-structured interviews. During interview, notes were created in such a way that attendants were able to comment and make corrections as and if required. This case study is performed on web development project in global Soft tech Company. The company's head office is in Europe, but it has offsite activities in several locations in Asia like Pakistan. The agile methods are use in company from past 10 years and it is common to have project of different nature i.e. large scale small scale, distributed etc.

The interviews conducted were from 15 participants including all stakeholders. The interviews comprised questions of 9 in total, out of which 6 questions were related to the experience and background. In other questions, issues and suggestion were gathered. While interviews project participants and stakeholders were directed to concentrate on all possible aspects of the theme.

The data gathered from interviews was analyzed qualitatively. While performing the qualitative analysis, issues identified by the interviewee was counted. A meeting with the participants was organized, in the first half of meeting participants were able to comment on each issues. In the second half of the meeting, improvement regarding the raised issues, were voted for further actions and study. However, here, we have reported the core findings and leaving the rest analysis for further studies.

3.8 Proposed Model in Action

The case study was completed in 5 months. The

Table 1. Scrum in action.

Event	Sprint 1	Sprint 2	Sprint 3	Sprint 4	Sprint 5
Sprint Planning	4 hour/week	6 hour/week	8 hour/week	10 hour/week	12 hour/week
Daily Scrum	15 min/day	15 min/day	15 min/day	15 min/day	15 min/day
Sprint Review	2 hour/week	3 hour/week	4 hour/week	5 hour/week	6 hour/week
Sprint Retrospective	1 hour/week	2 hour/week	3 hour/week	4 hour/week	5 hour/week

Table 2. Risk identification.

Identification by	Risk
Customer	Resistance to change by End Users
	Delay in delivery
	Conflicts between End Users
Product owner	Frequently changing requirements
	Effective identification of System requirement
	Vague and Incorrect system requirements
Team	Project complexity
	Use of new and immature technology
	Less experience and technical complexities
Risk Manager	Project planning and control
	Inexperience team
	Failure to manage end user expectation
	Failure to gain user commitment

team has to go through five sprint to complete the project. Following are the schedules of the conducted case study.

The stakeholders have to conduct session of sprint planning of 40 hours in total, the daily scrum meeting timing in total of 27.5 hours in total, the sprint review was conducted in total of 20 hours and the sprint retrospective 15 hours in total project.

The stakeholder of the said case project gathered during the activities of vision and in review. A few of risk identified by the stakeholders are mentioned in Table 2.

The evaluation of these items, results in the priority set (Table 4), size of expected loss ranked in 1-10 (Table 5) and planning (Table 6) how to mitigate the risk items.

For every risk item a proper mitigation plan has been formulated. The description of plan for a few selected risk item are shown in Table 7.

Each risk item is considered a separate entity and treated (Table 8) according to the priority set of the risk. The process of evaluation of the risk item is performed in the daily scrum meeting.

These item are treated during the daily scrum and afterwards, the mitigation of each item is monitored and verified (Table 9).

These items are reviewed (Table 10) in final so their future occurrence can be avoided. The future benefits (Table 11) are also considered at the end of the project, so the artifacts can be used for future planning.

Table 3. Risk description.

Risk	Description
Resistance to change by End Users	End User may be hesitant towards the change
Delay in delivery	Project complexities
Conflicts between End Users	End user may have different vision about the same requirement.
Frequently changing requirements	Customer with foggy vision.
Effective identification of System requirement	Requirement gathering in agile manner.
Vague and Incorrect system requirements	May acquire ambiguous requirements.
Project complexity	Interdependencies or Interconnections
Use of new and immature technology	Technology is introduced in the same year, less support available
Less experience and technical complexities	Lack of expertise.
Project planning and control	Un-experienced Scum Master
Inexperience team	Team have no hands on practices for the technology.
Failure to manage end user expectation	User requirement, budget and timeline clashes
Failure to gain user commitment	Un-interested users and with less awareness of technology.

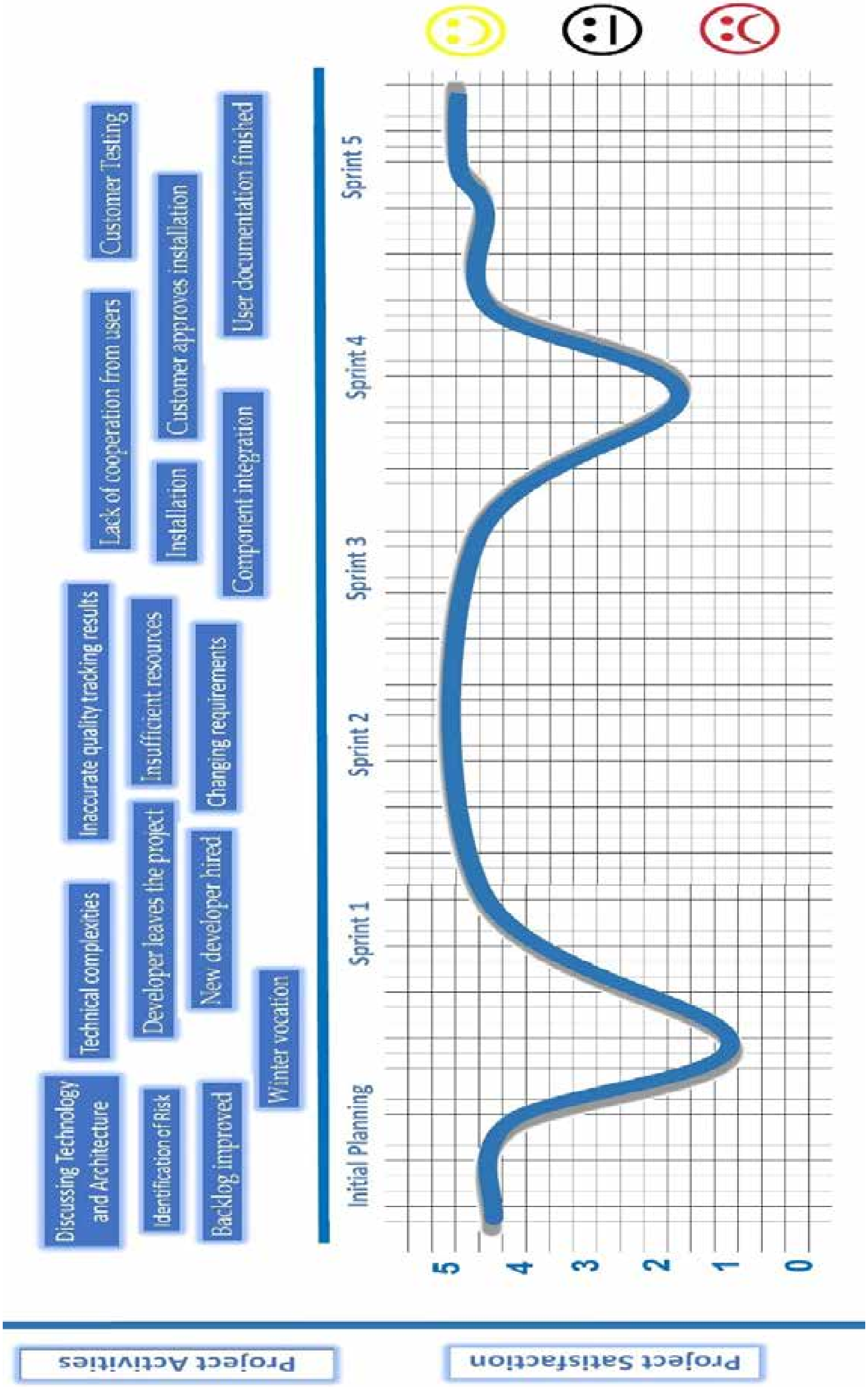

Fig. 4. Case study results.

Table 4. Risk analysis.

Risk	**Probability**
Resistance to change by End Users	70%
Delay in delivery	40%
Conflicts between End Users	20%
Frequently changing requirements	50%
Flaws in identification of System requirement	40%
Vague and Incorrect system requirements	20%
Project complexity	40%
Use of new and immature technology	80%
Less experience and technical complexities	30%
Project planning and control	50%
Inexperience team	30%
Failure to manage end user expectation	30%

Table 5. Consideration control.

Risk	**Size of loss**
Frequently changing requirements	8
Failure to manage end user expectation	8
Use of new and immature technology	6
Project planning and control	6
Flaws in identification of System requirement	6
Resistance to change by End Users	5
Project complexity	5
Less experience and technical complexities	5
Vague and Incorrect system requirements	4
Conflicts between End Users	4
Inexperience team	3
Delay in delivery	3
Failure to gain user commitment	2

Table 6. Risk prioritization.

Risk	**Prioritization**
Use of new and immature technology	Red
Resistance to change by End Users	Red
Failure to gain user commitment	Orange
Frequently changing requirements	Orange
Project planning and control	Orange
Project complexity	Orange
Flaws in identification of System requirement	Orange
Delay in delivery	Orange
Less experience and technical complexities	Purple
Inexperience team	Purple
Failure to manage end user expectation	Purple
Conflicts between End Users	Purple
Vague and Incorrect system requirements	Purple

Table 7. Risk planning.

Risk	Planning
Use of new and immature technology	Look for alternatives and Training
Resistance to change by End Users	Development of interest
Failure to gain user commitment	Development of interest
Frequently changing requirements	Freezing requirements
Project planning and control	Experience Scrum Master
Project complexity	Experience Team
Flaws in identification of System requirement	Use of Requirement Technique
Delay in delivery	Proper scheduling
Less experience and technical complexities	Employing experts
Inexperience team	Employing experts
Failure to manage end user expectation	Use of Prototyping
Conflicts between End Users	Rectify conflicting requirements
Vague and Incorrect system requirements	Remove ambiguous requirement

Table 8. Risk treatment.

Risk	Treatment
Use of new and immature technology	Used mature technology
Resistance to change by End Users	Conducted session to brief the ease of use.
Failure to gain user commitment	Engage in project for interest development.
Frequently changing requirements	Developed requirement documents
Project planning and control	Employing experience Scrum Master
Project complexity	Conducted training
Flaws in identification of System requirement	Effective requirement engineering
Delay in delivery	Scheduled Events
Less experience and technical complexities	Employed experts
Inexperience team	Employed experts
Failure to manage end user expectation	Used Model to verify the design
Conflicts between End Users	Improved requirement analysis
Vague and Incorrect system requirements	Improved requirement analysis

Table 9. Monitor and review of risk.

Risk	Monitor	Review
Use of new and immature technology	√	√
Resistance to change by End Users	√	√
Failure to gain user commitment	√	√
Frequently changing requirements	√	√
Project planning and control	√	√
Project complexity		√
Flaws in identification of System requirement	√	√
Delay in delivery		√
Less experience and technical complexities	√	√
Inexperience team		√
Failure to manage end user expectation	√	√
Conflicts between End Users	√	√
Vague and Incorrect system requirements	√	

Table 10. Future risk evaluation.

Risk	Future Evaluation
Use of new and immature technology	Foresee the technological changes.
Resistance to change by End Users	Training session planning.
Failure to gain user commitment	Engage in project for interest development.
Frequently changing requirements	Consider developed requirement documents a baseline
Project planning and control	Planning and management of future activities
Project complexity	Utilizing gained Expertise
Flaws in identification of System requirement	Effective employing of requirement engineering
Delay in delivery	Improved Scheduled experience
Less experience and technical complexities	Keeping employees upto-date with new technologies
Inexperience team	Conducting in house tanning and workshops
Failure to manage end user expectation	Managing a design Repository
Conflicts between End Users	Employing experience in requirement communication
Vague and Incorrect system requirements	Employing experience in requirement analysis

Table 11. Benefits of risk management

Risk	Future Evaluation
Use of new and immature technology	Knowledge to choose the technology.
Resistance to change by End Users	Experience to conducted session.
Failure to gain user commitment	Designs for user interest programs
Frequently changing requirements	Requirement documents Repository
Project planning and control	Assets for future
Project complexity	Expertise
Flaws in identification of System requirement	Effective employing of requirement engineering
Delay in delivery	Improved Scheduled experience
Less experience and technical complexities	Assets for future
Inexperience team	Assets for future
Failure to manage end user expectation	Design Repository
Conflicts between End Users	Experience in requirement analysis
Vague and Incorrect system requirements	Experience in requirement analysis

4. RESULTS AND DISCUSSION

On the subject of our research questions, we were able to get confirmation that risk management in scrum is likely improved using the proposed framework. The web development project is completely functional and no risk issues in any sprint of scrum were reported. While performing the study more improvement needs were highlighted rather than working issues. None of the identified issues was so severe that it would have suffered the working of project.

Conclusion drawn on the base of working is that framework has worth of use. The process management and stakeholders' involvement issues were mainly identified. This gives a contrary impact, but while observing in detail it became vibrant that some skills are required for adjusting Scrum to risk free environment.

5. CONCLUSIONS

The success of project can be increased by effective risk management practices. The uncertainties are considered as the risk of the project that result in cost expansion and reduce the quality of product. Timely decision is vital for controlling the risk issues. The roles involved in the development of project should be able to identify these issues without any problem. In our research study we have employed risk management activities in a Scrum model. This enables to identify the risk by marking the issues and maintaining repositories for future reference. Thus risk issues can be identified and tracked at any stage of sprint. We have conducted a case study and results have shown that by adopting secure and risk free scrum model a quality software product can be produced. Our future research will be directed toward the generalization of this framework. The proposed framework has been developed and evaluated for collocated environment. However, in future we will extend our work in distributed environment, so the results can be more significant.

6. ACKNOWLEDGEMENTS

We thank every person at our institution who contributed in the study or offered views for improvement of the study.

7. REFERENCES

1. Malik, M.U., H. Nasir, & A. Javed. An efficient objective quality model for agile application development. *International Journal of Computer Applications* 85(8): 19-24 (2014).
2. Cerpa, N., & J.M. Verner. Why did your project fail?. *Communications of the ACM* 52(12): 130-134 (2009).
3. McManus, J. Risk management in software development projects. *Routledge,* London (June 25, 2012).
4. Bali, V. & S. Bali. *Software Engineering*. S.K. Kataria and Sons Publishers, New Delhi, India, p. 269-278 (2008).
5. Reddaiah, B., S.P. Ravi, & L.S. Movva. Risk management board for effective risk management in scrum. *International Journal of Computer Applications*65(12): 16-23 (2013).
6. Garvey, P.R. *Analytical Methods for Risk Management: A System Engineering Perspective*. Chapman-Hall/CRC Press, Taylor & Francis Group, Boca Raton (2008).
7. Janus, A., R. Dumke, A. Schmietendorf, & J. Jäger. The 3c approach for agile quality assurance. In: *3rd International Workshop on Emerging Trends in Software Metrics (WETSoM)*, Zurich, Switzerland, p. 9-13 (2012).
8. Alharbi, E.T., & M.R.J. Qureshi. Implementation of risk management with SCRUM to achieve CMMI requirements. *International Journal of Computer Network and Information Security* 6(11): 20-25 (2014).
9. Pandey, P.K.D. Development of risk management model for secure software product. *South Asia Journal of Multidisciplinary Studies* 1(3): http://gjms.co.in/index.php/SAJMS/article/view/840 (2015).
10. Pohl, C., & H.J. Hof. Secure Scrum: Development of secure software with SCRUM. *Ninth International Conference on Emerging Security Information, Systems and Technologies,* Venice, Italy https://www.researchgate.net/publication/277307837_Secure_Scrum_Development_of_Secure_Software_with_Scrum (2015).
11. Wanderley, M., J. Menezes, C. Gusmão, & F. Lima. Proposal of risk management metrics for multiple project software development. *Procedia Computer Science* 64: doi:10.1016/j.procs.2015.08.619, 1001-1009 (2015).
12. Islam, S., & Houmb, S. H. Towards a framework for offshore outsource software development risk management model. *Journal of Software,* San Bernardino, California 6(1): 38-47 (2011).
13. Ylimannela, V. A Model for risk management In Agile software development. March 2012 http://www.cloudsw.org/under-review/a6f468c9-4857-4206-96ee-f67df0583d41 (Accessed: 22 April, 2016).
14. Layton, M.C. How to manage risk within Agile management - for dummies. Agile Project Management for Dummies, May-2012. http://www.dummies.com/how-to/content/how-to-manage-risk-within-agile-management.html (Accessed: 17 Jan, 2016).
15. Veethil. S.T. Risk management in Agile. Scrum alliance. https://www.scrumalliance.org/community/articles/2013/2013-may/risk-management-in-agile (Accessed: 17-Jan-2016).
16. Odzaly, E. E., & Des Greer, D. S. Lightweight risk management in Agile projects. In: *Conference: 26th Software Engineering Knowledge Engineering Conference (SEKE),* Vancouver, Canada*:* doi: 10.13140/2.1.4681.0882 (2014).
17. De Bakker, K., A. Boonstra, & H. Wortmann. Does risk management contribute to IT project success? A meta-analysis of empirical evidence. *International Journal of Project Management* 28(5): 493-503 (2010).
18. Hijazi, H., Alqrainy, S., Muaidi, H., & Khdour, T. Risk factors in software development phases. *European Scientific Journal* 10(3): http://www.eujournal.org/index.php/esj/article/view/2624 (2014).
19. Runeson, P., & Höst, M. Guidelines for conducting and reporting case study research in software engineering. *Empirical Software Engineering* 14(2): 131-164 (2009). doi:10.1007/s10664-008-9102-8.