

A novel risk management model in the Scrum and extreme programming hybrid methodology

Mahnaz Afshari¹, Taghi Javdani Gandomani^{1,2}

¹Data Science Research Center, Shahrekord University, Shahrekord, Iran

²Department of Computer Science, Shahrekord University, Shahrekord, Iran

Article Info

Article history:

Received Nov 3, 2021

Revised Jan 4, 2022

Accepted Jan 25, 2022

Keywords:

Agile software development

Hybrid methodology

Risk management

Team performance

ABSTRACT

Risk management in software development has always been one of the necessities of software project management. The logical nature of software projects and products has caused several challenges and risks in these projects. On the other hand, with the emergence of agile methodologies, especially Scrum, and extreme programming (XP) methodologies, in recent years, this issue has become more serious. This is mainly because emphasizing limited documentation in these methodologies has caused these methods to pay little attention to some aspects of project management, particularly risk management. Concentrating on this challenge, the current study has proposed a risk management model in the hybrid methodology, combining Scrum and XP. Using this model in a case study shows this model's success in achieving risk management purposes. The results of this study indicate an appropriate reduction in the number of reworks, change requests, identified risks, and occurred risks. Moreover, the number of eliminated risks and team productivity have increased.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



Corresponding Author:

Taghi Javdani Gandomani

Department of Computer Science, Shahrekord University

Shahrekord, Iran

Email: javdani@ieec.org, javdani@sku.ac.ir

1. INTRODUCTION

Small and big software projects are done with different features and various functions, and most of the sensitive and critical systems use software applications in large businesses widely. Software projects face different challenges and risks due to their differences from other industrial projects [1]. One of the major differences between software projects and the others is the intangibility of software projects, that is the software project's progress, errors, and defects usually remain hidden until the final stages of software development. In other projects like civil projects, however, errors and defects in the product and process can be observed and examined easily [2]. The special nature of software projects leads the development process and intended product to face different challenges and risks.

In every project, considering the project risks is one of the important points, and that is why risk management is specifically considered as a serious activity in project management. Regarding the fact that software systems are usually sensitive and these systems' risk appetite can lead to unpleasant problems and happenings, considering risk management in software projects is more important than in other industries [3]. Software development and its associated activities are seriously affected by the adopted methodology and process employed in this field. The adopted methodology affects all the major activities and umbrella activities such as the risk management process.

Although the existing project management standards and their recommended framework such as the project management body of knowledge (PMBOK) and projects in controlled environments (PRINCE2) can

be efficient in heavyweight software development methodologies, considering the fundamental differences of these methods and agile methodologies cannot be effective in agile risk management [4]. Therefore, having a model and method for risk management, which is peculiar to agile methods and consistent with agile features and process seems necessary [5]. The way of applying risk management in agile methodologies, which has drawn a lot of attention today, is different from one methodology to another.

Besides the prevalence of agile methods, the combination of Scrum and extreme programming (XP) methodologies is known as a popular methodology [6]. The purpose of the present research is to propose a risk management model based on Scrum and XP combination in software development. The rest of this paper has been organized as follows. Section 2 briefly introduces agile methodologies, and section 3 describes risk management in software development. In section 4, the related work is investigated, and section 5 shows the research method employed in this study. The proposed model and its evaluation are given in section 6, and the research conclusion is presented in section 7.

2. AGILE SOFTWARE DEVELOPMENT

The software development process was conducted through traditional models and methods such as waterfall, spiral, and RUP for many years [7]. There were various problems with these methods such as huge documentation that was much time-consuming, resistance against the requirements changes, rigorous process disciplines, less attention to human power, and so on. These issues became more troublesome with the emergence of the Internet and the widespread usage of software applications [8].

Agile software development methods were officially introduced in 2001, where some of the famous software experts decided to launch a revolution in the software industry. They left the traditional methods with all of their intrinsic problems and replaced them with agile methods [9]. Agile methods were introduced by signing a manifesto called agile manifesto in which agile principles and values were defined simply [10]. In this manifesto, values, like accepting changes, considering the customer as a development team member, focusing on the working software, and paying much attention to individuals' interactions, have been emphasized.

Agile methodologies are methods each of which have been created with different viewpoints and features concentrating on solving particular problems. Well-known agile methodologies such as Scrum, XP, crystal, feature-driven development (FDD), and dynamic systems development method (DSDM) are among these methodologies [11]. Each method defines its particular process, practices, roles, artifacts. However, many issues and challenges need to be handled in the transformation to agile methods including methodology adoption, team performance and productivity, risk management, and human-related issues [12]–[16]. In recent years, combinations of these methodologies are often employed in most agile teams. Among the hybrid agile methodologies, the combination of Scrum and XP has been widely used [17].

2.1. Scrum

Scrum is the most popular method of agile software development [18], [19]. It is a method to develop software by emphasizing project management. In this methodology, software development is carried out within constant and specified time intervals (less than a month), known as sprints. The main Scrum focus is on the development of potentially shippable software product increments in each sprint by emphasizing individual interactions.

Scrum emphasizes that many aspects of product development are not predictable and a lot of requirements may be added or changed during software development. Hence, Scrum is suitable for those projects that either have many unclear requirements or some cannot be predicted or analyzed for any reason. Initially, all requirements suggested by the customer are prepared in the form of a list known as the product backlog. Then, in a meeting called sprint planning, some of the requirements with higher priority are selected by the development team. In addition, the approximate time of working on each requirement is determined in this stage. The selected requirements are put in sprint backlog and then the team begins working on the requirements in the list. During each sprint, team members do their tasks and may have daily meetings as well. At the end of each sprint, the team shows the product increment to the customer in a meeting called sprint review. Some requirements may need revision or take more time. In this case, they will be returned to the product backlog to be planned in the next coming sprints. This process will be repeated to finish all the requirements included in the product backlog.

2.2. XP

The XP is another popular agile method, as reported [17]. The XP was first introduced in 1990 [20] and was completed after several years. The concentrating of the XP method is on producing high-quality software and meeting customer satisfaction through responding to their changing demands in every stage of

the development process. XP believes that the final goal of a development project is to produce high-quality codes that can be executed and maintained.

The XP process observes a project in terms of communications, simplicity, feedback, and courage. In the XP process, customers declare what they expect of the product besides the development team. The customers' declarations in terms of their requirements are called user stories. The development team members then design and develop the desired software and after each iteration, they deliver a part of the product to the customer to be checked and verified by him/her and to receive feedback. The changes suggested by customers are then applied to the software product. A customer is like a team member with a continuous presence to specify the requirements and their prioritization. In the XP method, the test-driven development practice is used, i.e. before writing any code, its acceptance criteria and test cases are written.

Scrum and XP provide excellent practices that help software teams in their projects. These two methods are complementary in some way. While Scrum focuses on project management, XP provides helpful practices in software development. As mentioned before, the combination of Scrum and XP is widely used in agile teams. Their combination is illustrated in Figure 1. In this model, the XP practices have been used in each iteration.

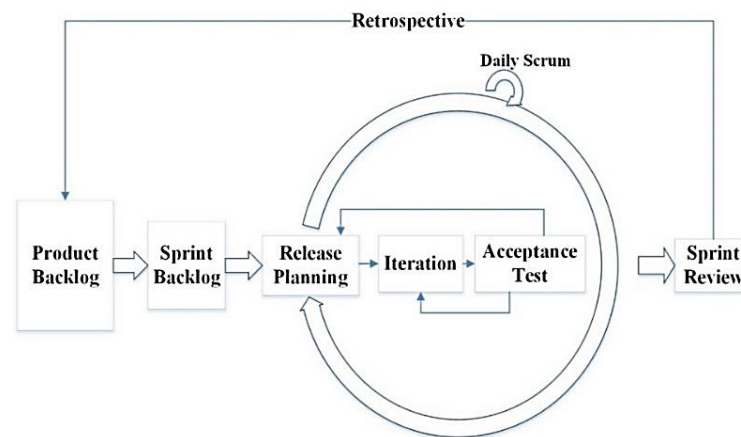


Figure 1. The combination of Scrum and XP

3. RISK MANAGEMENT

There are various definitions of risk. However, risk is an abstract concept that everybody knows and understands but it might not be easily defined. Commonly, the risk is the possibility of loss. Moreover, according to Cambridge Dictionary, the risk is the possibility of something unpleasant happening [21]. Based on this definition, because it has not happened in the present time, some measures can be taken to prevent it from happening or manage it in case it happens. That is why the concept of risk management was raised.

Like all industrial projects, software projects are exposed to various risks. Due to the difference between software projects and other projects, specific attention to risk management seems very necessary. This is mainly because software projects are not tangible and also the possibility of complex and difficult errors to occur in such projects is much more than in other projects [22]. In a very common definition by Boehm, risk management in software projects is a particular discipline that its objectives are to identify, address, and eliminate the potential risks before they threaten the success of software projects or lead to rework in the project [23].

Software risks have different categorizations. In general, whatever poses a threat to the software project is considered a risk. Table 1 shows the most important software risk factors. Given different categorizations for introducing different risks, a specific process needs to be done regardless of the type of determining risk. Figure 2 illustrates this general process. Generally, it is required that the risks be identified and investigated and a solution be presented for them. Next, the solution is supposed to be applied and its results are controlled and examined.

Paying attention to risk management has always been considered in project management standards and frameworks. PMBOK, PRINCE2, principles-based, sustainable project management methodology (PRISM), the organizational project management maturity model (OPM3), ISO1006, and project management competency development framework (PMCDF) are some of the standards that have paid appropriate attention to risk management issues. In these standards, general strategies have been proposed to

manage and handle the potential risks in projects. They perceive the presence of several documents and products to be necessary to identify and control risk. This might be helpful in heavyweight software development methodologies, but it does not have an appropriate function in agile development methodologies in which limited documentation is a must. Moreover, considering a comprehensive risk management model, as proposed in the aforementioned standards, strongly reduces the agility promised by agile methods. For these reasons, considering strategies specifically designed for agile methodologies seems to be necessary.

Table 1. The most important software risk factors

Factor	Description
Efficiency risk	the level of uncertainty about the fact that to what extent the product meets the customers' needs
Cost risk	the level of uncertainty about the fact that whether the budget allocated to the product is sufficient for its costs
Support risk	the level of uncertainty about the easiness of the product maintenance and adaptation
Timing risk	the level of uncertainty about accurate and proper observance of product timing to deliver it to the customer



Figure 2. The general process of risk management

4. RELATED WORK

One of the most important reasons behind failure in software projects is the lack of risk management or its improper usage. Previous studies reveal that agile methodologies implement few risk management methods. The need for risk management and lack of official methods for that have been emphasized by several researchers in agile methods, especially Scrum [24]. One of the challenges of agile methods is the lack of comprehensive up-front planning, which is because of their specific nature. In one study, a combination of XP and PRINCE2 methodologies was proposed to get access to a suitable project management strategy and risk control. The results of using that hybrid model are indicative of an increase in the quality of software products and a decrease in time and cost of software development [25], [26].

In another study, Agile methodologies and a risk management approach were presented through a particular model. Although employing this methodology resulted in risk management improvement, that model could only be used in some specific projects. That model consists of three main phases including product vision planning, product roadmap release planning, and implementation each of which consists of its particular risk management mechanisms [27]. The literature review also shows that a combination of the same methods used for industrial projects can be utilized for risk management in software projects. For example, PMBOK consists of several knowledge areas to ensure project success [28], [29].

Empirical studies suggest that risk management is not considered an official approach in many software companies. Furthermore, in some software methodologies, risks are not officially checked and in some others, informal methods are used to control risk. In some studies, Scrum is also introduced as one of the risk management approaches [30]. In a research study, a risk management mechanism was added to Scrum to achieve capability maturity model integration (CMMI) objectives. The results of that study indicated that the presented strategy had over 70-percent success in identifying risks and controlling them [31].

Using the PRINCE2 methodology, a novel model was proposed for risk management in Scrum in another study. Since both methodologies are process-oriented, combining them is possible. In that model, the level of risk effect was found to be the most important feature of a risk. In addition, the sprint planning meeting has been mentioned as the most important place to discuss the potential risks. The results of this research show that embedding Scrum in PRINCE2 methodology improves product delivery time. However,

no results have been presented regarding the reduction of agility or other parameters [32]. This has been emphasized in another study [33].

Using Scrum in large-scale projects is increasing. But Scrum does not provide any methods for risk management by itself. In another study, risk management in Scrum was investigated and some strategies were proposed while implementing Scrum practices [5]. Moreover, another study put forward a prince2-based model in Scrum, whose results indicated improvement in the development process and reduction of software development challenges in the real world [34].

5. RESEARCH METHOD

The present study has been particularly done as a case study. Figure 3 illustrates the steps taken in this research. As can be seen in Figure 3, the literature review investigated different aspects of risk management in software development as well as Scrum and XP methodologies and their combination. Afterward, a model was proposed for risk management in Scrum and XP combination and that model was employed in a case study and evaluated later on. As a way in which one theory is evaluated in a real case, the case study is one of the best ways that can be employed in empirical studies in software engineering.

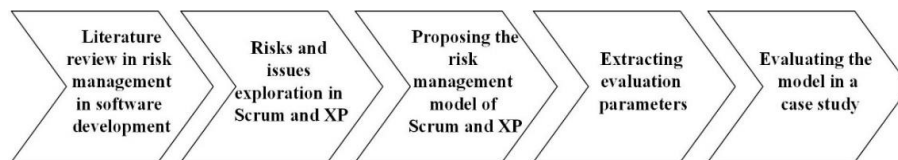


Figure 3. Steps in the research process

5.1. Case study

The place under study is an active company in the field of software development using agile methods. The development team consisted of 9 members whose level of education and experience have been presented in Table 2. On this team, a combination of Scrum and XP was used for software development. Team members were familiar well with both Scrum and XP and had worked together for at least 3 years. They were named M1 to M9 to preserve their confidentiality and privacy.

Table 2. The detail of the team members

No.	Name	Role	Software development experience (Year)	Agile development experience (Year)
1	M1	Scrum master	14	6
2	M2	Developer	9	4
3	M3	Developer	6	3
4	M4	Analyst	8	4
5	M5	Developer	5	5
6	M6	Developer	7	6
7	M7	Developer	4	3
8	M8	Developer	6	4
9	M9	Architecture	9	3

In the team understudy, no systematic and standard method was used to do risk management and most of the risk-related activities were done empirically through an approach similar to PMBOK. In that team, the empirically predictable risks were initially investigated and suitable strategies were adopted. To conduct this case study and consider time and team limitations in the company under study, the team was divided into two groups of 5 and 4 members. The 4-member group began doing relevant processes like before, as the control team (CT), and the second team employed the proposed risk management model for doing the risk management process, as the studied team (ST). The company's project was in the field of developing a payroll software package ordered by a client in a 4 months' project. Eight sprints had been designed for this project, and statistical and analytical details will be presented in the subsequent sections.

5.2. Evaluation criteria

Evaluation of every model's efficiency has to be done based on factors covering the main purpose of the model. The literature review is indicative of the fact that in software teams, the parameters that show improvement in risk management are generally those which either directly indicate improvement in reduction

of the number of risks and their consequences or indirectly show the effectiveness of risk management. Table 3 demonstrates the evaluation criteria considered in the present research.

Table 3. The criteria adopted for model evaluation

No.	Name	Role
RI	Risks identified	Number of risks identified in each sprint
RO	Risks occurred	Number of risks that occurred in each sprint
RH	Risks handled	Number of risks handled in each sprint
RW	Re-works	Number of re-works due to the risks (misunderstanding or unclear requirements) in each sprint
CR	Change requests	Number of risks change requests due to the requirements related risks in each sprint
TP	Team productivity	The ratio of done story points to selected ones

6. THE PROPOSED MODEL AND EVALUATION OF ITS EFFICIENCY

The proposed model in this study is based on the combination of Scrum and XP. Figure 4 shows the proposed model in concise. In this model, all risks have to be identified at first and placed in the risk backlog. Then, they are required to be prioritized based on factors determining priority. When the risks are prioritized, the risks with higher priority enter the Scrum iteration cycle to be considered for risk management.

In this cycle, which is based on the XP method, the way of facing the selected risks in the current iteration is planned through different strategies discussed in the previous sections. Afterward, the strategy associated with that risk is applied and finally, the way of risk elimination is observed and controlled. This cycle is followed for all risks selected to be considered in this sprint. At the end of the sprint, the harmony between strategies and risks is investigated, and in case new risks are created during this process or the existing risks have remained, they will return to the risk backlog so that they can be analyzed and eliminated in subsequent sprints.

No new role has been defined in this model and team members will be in charge of risk management issues simply with the roles defined in Scrum and XP. This is important for two reasons: first, there will be no conflict against the multitasking structure of Scrum; second, the promised agility in these two methodologies will not be at risk by adding the human overhead. In the following, the proposed model will be evaluated in terms of the evaluation criteria introduced in the previous section.

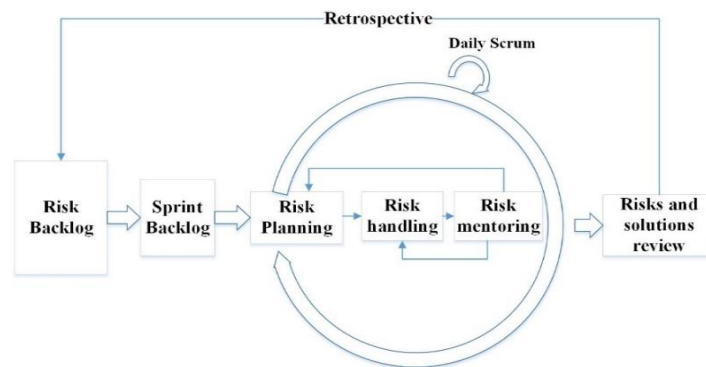


Figure 4. The proposed risk management model

6.1. Number of identified risks

Table 4 shows the number of identified risks (RI) during the project. Also, Figure 5 illustrates this criterion in both teams. It should be noted that SP stands for Sprint in all the figures. As shown in Table 4 and Figure 5, employing the risk management model, leading to a higher rate in risk identification in the team under study.

Table 4. Number of identified risks during the project

	SP1	SP2	SP3	SP4	SP5	SP6	SP7	SP8	SP9	SP10
ST	7	9	7	6	6	5	6	4	4	5
CT	6	7	5	4	5	3	3	3	3	2

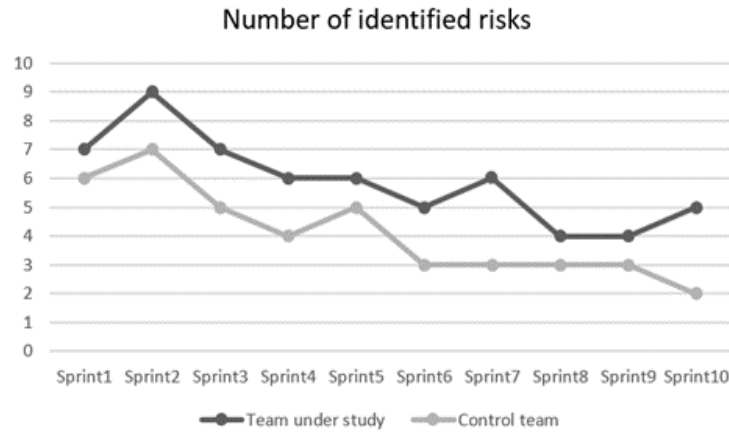


Figure 5. Number of identified risks in this study

6.2. Number of occurred risks

The number of occurred risks (RO) can be seen in Table 5 and Figure 6. In both teams, the number of risks involved decreases as the project progresses. However, the number of the occurred risks in the team understudy is less than control team. This shows the effeteness of the model in avoiding the potential risks during the project.

Table 5. Number of occurred risks during the project

	SP1	SP2	SP3	SP4	SP5	SP6	SP7	SP8	SP9	SP10
ST	4	3	3	4	2	2	3	2	3	2
CT	4	7	4	6	5	5	3	5	4	4

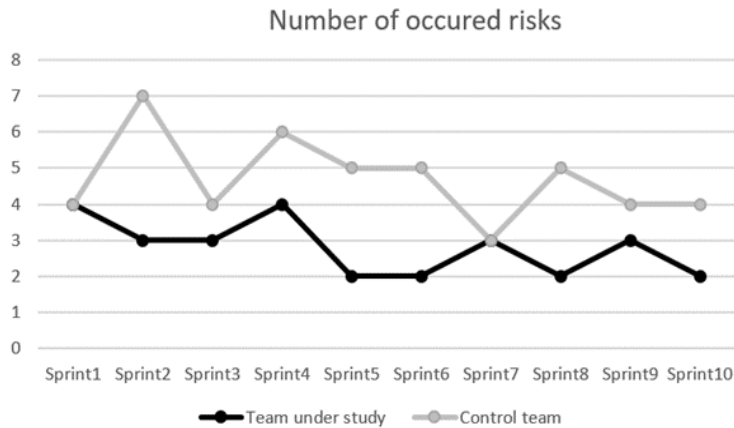


Figure 6. Number of occurred risks in this study

6.3. Number of handled risks

The number of handled risks (RH) is shown in Table 6 and Figure 7. The data obtained show that the number of the handled risks decline by the progress of the project in both teams. However, it seems that the team understudy was more successful than the control team in handling the probable risks. This also shows that employing the risk model has led to this achievement.

Table 6. Number of handled risks during the project

	SP1	SP2	SP3	SP4	SP5	SP6	SP7	SP8	SP9	SP10
ST	3	3	4	2	3	2	2	2	2	1
CT	1	2	2	1	2	3	2	1	1	0

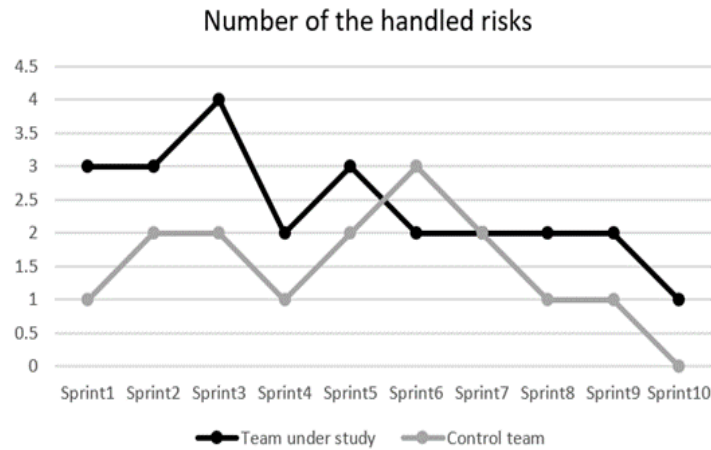


Figure 7. Number of handled risks in this study

6.4. Number of re-works

The number of re-works (RW) has been used to identify whether risk management can reduce re-works caused by uncontrolled risks. Table 7 and Figure 8 show the results of data collection regarding this metric. The table and figure show that the team understudy did fewer re-works compared to the control team.

Table 7. Number of re-works done during the project

	SP1	SP2	SP3	SP4	SP5	SP6	SP7	SP8	SP9	SP10
ST	3	4	3	3	0	2	3	3	2	2
CT	5	5	4	4	4	5	4	5	4	3

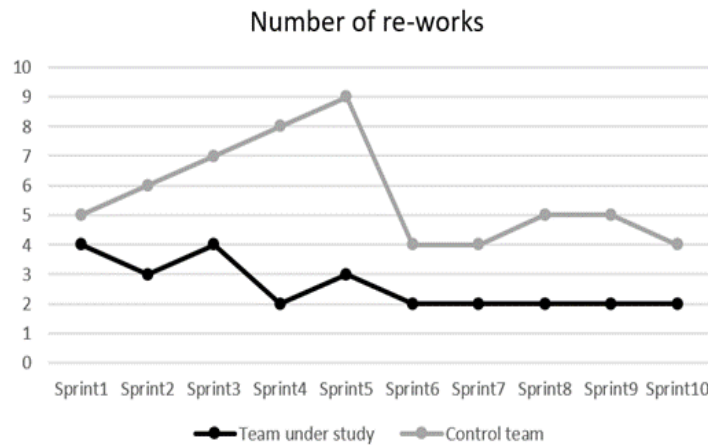


Figure 8. Number of re-works in the sprints

6.5. Number of change requests

The change request (CR) is another metric that can show how well the proposed model can lead to fewer change requests. Table 8 and Figure 9 show the results regarding this factor. It seems that the team understudy faced fewer change requests compared to the control team.

Table 8. Number of change requests during the project

	SP1	SP2	SP3	SP4	SP5	SP6	SP7	SP8	SP9	SP10
ST	4	3	4	2	3	2	2	2	2	2
CT	5	6	7	8	9	4	4	5	5	4

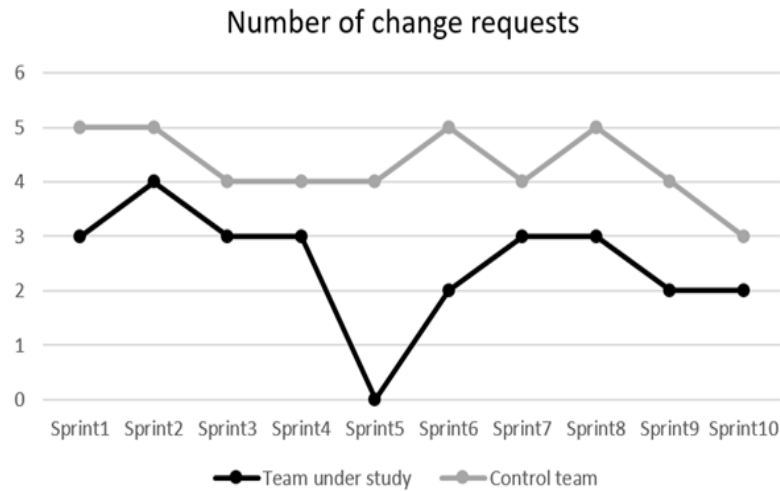


Figure 9. Number of change requests in this study

6.6. Team productivity

Team productivity (TP), as mentioned previously, is calculated by the ratio of done story points and selected ones. Table 9 shows the selected and completed story points. Table 10 and Figure 10 compare team productivity in this study. As Figure 10 shows, team performance was higher in the understudy team compared to the control team. This criterion also shows the effectiveness of the employed risk model again.

Table 9. Selected and completed story points in this study

	SP	SP1	SP2	SP3	SP4	SP5	SP6	SP7	SP8	SP9	SP10
ST	selected	390	430	440	430	390	400	380	410	400	430
	done	360	400	410	390	370	370	360	390	370	410
CT	selected	340	310	310	340	330	320	320	320	330	330
	done	300	270	270	300	270	260	260	270	280	280

Table 10. Team performance in both teams

	SP1	SP2	SP3	SP4	SP5	SP6	SP7	SP8	SP9	SP10
ST	92.3	93.0	93.2	90.7	94.9	92.5	94.7	95.1	92.5	95.3
CT	88.2	6	87.1	88.2	81.8	81.3	81.3	84.4	84.8	84.8

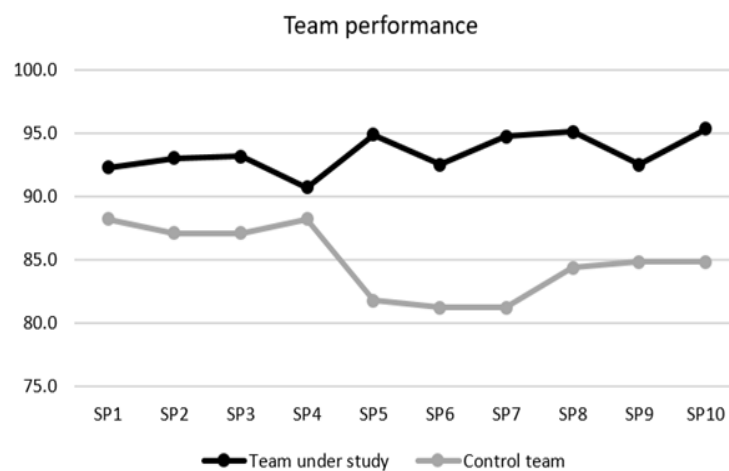


Figure 10. Team performance in all sprints

7. CONCLUSION

Risk management in software development has always been one of the important aspects of software project management. This challenge in agile methodologies is more serious than in regular ones because of their nature. The combination of Scrum and XP is one of the most popular software development methodologies among software companies. In the current study, one risk management model in the combination of Scrum and XP has been proposed, which will have the minimum overhead cost for development. In this model, only one product has been added to compound methodology products as the risk backlog. Moreover, besides the team tasks of Scrum, risk management tasks in each sprint are done in three steps, i.e. planning, applying a strategy to face risk, and supervision (control). The results of using this model in a case study reveal that this model causes suitable improvements such as reduction of the number of duplications, change requests, identified risks, and occurred risks as well as increasing team efficiency and eliminated risks.




REFERENCES

- [1] E. J. Braude and M. E. Bernstein, *Software engineering: modern approaches*, 2nd ed. Waveland Press, 2016.
- [2] L. R. Vijayarathy and C. W. Butler, "Choice of software development methodologies: do organizational, project, and team characteristics matter?," *IEEE Software*, vol. 33, no. 5, pp. 86–94, Sep. 2016, doi: 10.1109/MS.2015.26.
- [3] R. Srivastava, "An analysis on risk management and risk in the software projects," in *Analyzing the Role of Risk Mitigation and Monitoring in Software Development*, IGI Global, 2018, pp. 83–99.
- [4] G. Jamali and M. Oveisi, "A study on project management based on PMBOK and PRINCE2," *Modern Applied Science*, vol. 10, no. 6, Apr. 2016, doi: 10.5539/mas.v10n6p142.
- [5] B. G. Tavares, C. E. S. da Silva, and A. D. de Souza, "Risk management analysis in Scrum software projects," *International Transactions in Operational Research*, vol. 26, no. 5, pp. 1884–1905, Mar. 2019, doi: 10.1111/itor.12401.
- [6] T. Javdani Gandomani and M. Ziaei Nafchi, "An empirically-developed framework for agile transition and adoption: a grounded theory approach," *Journal of Systems and Software*, vol. 107, pp. 204–219, Sep. 2015, doi: 10.1016/j.jss.2015.06.006.
- [7] R. Albuquerque, R. Fernandes, R. M. Fontana, S. Reinhr, and A. Malucelli, "Motivating factors in agile and traditional software development methods: a comparative study," in *Communications in Computer and Information Science*, vol. 680, Springer International Publishing, 2017, pp. 136–141.
- [8] Y. Leau, W. K. Loo, W. Y. Tham, and S. F. Tan, "Software development life cycle agile vs traditional approaches," in *International Conference on Information and Network Technology*, 2012, vol. 37, no. 1, pp. 162–167.
- [9] B. Meyer, "Making sense of agile methods," *IEEE Software*, vol. 35, no. 2, pp. 91–94, Mar. 2018, doi: 10.1109/MS.2018.1661325.
- [10] K. Beck *et al.*, "Manifesto for Agile Software Development," *The Agile Alliance*, 2001. <http://agilemanifesto.org/> (accessed Jun. 01, 2020).
- [11] D. Cohen, M. Lindvall, and P. Costa, "An introduction to agile methods," in *Advances in Computers*, vol. 62, Elsevier, 2004, pp. 1–66.
- [12] F. Tavakoli and T. J. Gandomani, "A novel team productivity model for XP teams," *Journal of Cases on Information Technology*, vol. 20, no. 4, pp. 93–109, Oct. 2018, doi: 10.4018/JCIT.2018100106.
- [13] T. J. Gandomani and M. Z. Nafchi, "The essential prerequisites of agile transition and adoption: a grounded theory approach," *Journal of Internet Computing and Services*, vol. 17, no. 5, pp. 173–184, Oct. 2016, doi: 10.7472/jksii.2016.17.5.173.
- [14] T. J. Gandomani and M. Z. Nafchi, "Agile transition and adoption human-related challenges and issues: a grounded theory approach," *Computers in Human Behavior*, vol. 62, no. 257–266, 2016.
- [15] M. Ziaei, T. Javdani, and Z. Tavakoli, "Knowledge management in software development based on the leavitt s model traditional versus agile methods," in *Fifth International Conference on Advances in Computing, Communication and Information Technology*, Sep. 2017, pp. 1–4, doi: 10.15224/978-1-63248-131-3-09.
- [16] T. J. Gandomani and M. Z. Nafchi, "Agile coaches and champions: two hidden facilitators of agile transition," in *Emerging Innovations in Agile Software Development*, IGI Global, 2016, pp. 24–36.
- [17] VersionOne, "14 annual state of agile report," *stateofagile.com*, 2020. <https://stateofagile.com/#ufh-c-7027494-state-of-agile> (accessed Jun. 01, 2020).
- [18] A. Agarwal, *The basics of SCRUM: a simple handbook to the most popular agile Scrum framework*. Independently Published, 2017.
- [19] K. S. Rubin, *Essential Scrum: a practical guide to the most popular agile process*, 1st ed. Addison-Wesley Professional, 2013.
- [20] W. C. Wake, *Extreme programming explored*, 1st ed. Addison-Wesley Professional, 2001.
- [21] Cambridge, "Cambridge dictionary," *Cambridge University*, 2019. <http://dictionary.cambridge.org/dictionary/english/risk> (accessed Jan. 01, 2020).
- [22] H. Barki, S. Rivard, and J. Talbot, "Toward an assessment of software development risk," *Journal of Management Information Systems*, vol. 10, no. 2, pp. 203–225, Sep. 1993, doi: 10.1080/07421222.1993.11518006.
- [23] B. Boehm, "Software risk management," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 387, Springer Berlin Heidelberg, 1989, pp. 1–19.
- [24] S. Lopes, R. Gratão de Souza, A. Contessoto, A. Luiz de Oliveira, and R. Braga, "A risk management framework for Scrum projects," in *Proceedings of the 23rd International Conference on Enterprise Information Systems*, 2021, pp. 30–40, doi: 10.5220/0010448300300040.
- [25] J. Nyfjord and M. Kajko-Mattsson, "Commonalities in risk management and agile process models," in *International Conference on Software Engineering Advances (ICSEA 2007)*, Aug. 2007, doi: 10.1109/ICSEA.2007.22.
- [26] M. Esteki, T. J. Gandomani, and H. K. Farsani, "A risk management framework for distributed Scrum using prince2 methodology," *Bulletin of Electrical Engineering and Informatics*, vol. 9, no. 3, pp. 1299–1310, Jun. 2020, doi: 10.11591/eei.v9i3.1905.
- [27] J. Nyfjord and M. Kajko-Mattsson, "Outlining a model integrating risk management and agile software development," in *EUROMICRO 2008 - Proceedings of the 34th EUROMICRO Conference on Software Engineering and Advanced Applications*,




- SEAA 2008, Sep. 2008, pp. 476–483, doi: 10.1109/SEAA.2008.77.
- [28] V. Yilmannela, “A model for risk management in agile software development,” *Communications of Cloud Software*, vol. 3, pp. 1–10, 2012.
- [29] M. Godse and B. Rajiv, “Improvement of project management knowledge areas using Scrum technique,” in *Lecture Notes on Multidisciplinary Industrial Engineering*, Springer Singapore, 2021, pp. 133–149.
- [30] A. B. C. Brandão, “Risk management in software projects using Scrum framework,” Stockholm, 2012.
- [31] E. Talal Alharbi and M. R. Jameel Qureshi, “Implementation of risk management with SCRUM to achieve CMMI requirements,” *International Journal of Computer Network and Information Security*, vol. 6, no. 11, pp. 20–25, Oct. 2014, doi: 10.5815/ijcnis.2014.11.03.
- [32] M. Tomanek and J. Juricek, “Project risk management model based on prince2 and Scrum frameworks,” *International Journal of Software Engineering and Applications*, vol. 6, no. 1, pp. 81–88, Jan. 2015, doi: 10.5121/ijsea.2015.6107.
- [33] J. Irizar and M. Wynn, “Centricity in project risk management: new dimensions for improved practice,” *International Journal on Advances in Intelligent Systems*, vol. 8, no. 1 and 2, pp. 209–218, 2015.
- [34] M. Mousaei and T. Javdani, “A new project risk management model based on Scrum framework and prince2 methodology,” *International Journal of Advanced Computer Science and Applications*, vol. 9, no. 4, pp. 442–449, 2018, doi: 10.14569/IJACSA.2018.090461.

BIOGRAPHIES OF AUTHORS



Mahnaz Afshari    received an M.Sc. degree in Software Engineering from Isfahan Branch, Islamic Azad University, Iran in 2019. Now, she serves as a researcher in Data Science Research Center, Shahrekord University, Iran. Her research interests include Agile software development and Agile team management. Email: afshari.mhz@gmail.com.



Taghi Javdani Gandomani    received the Ph.D. degree from UPM, Malaysia in 2014. He is currently an Assistant Professor of Software Engineering at Shahrekord University, Iran. His research interests include Agile software development, software process improvement, software project management, and empirical software engineering. He is also the leader of the Data Science Research Center at Shahrekord University, Iran. Email: javdani@ieee.org, javdani@sku.ac.ir.