International Journal of Electrical and Computer Engineering (IJECE)
Vol. 13, No. 4, August 2023, pp. 4379~4387
ISSN: 2088-8708, DOI: 10.11591/ijece.v13i4.pp4379-4387

Risk management framework in Agile software development methodology

Mohammad Hadi Zahedi[1], Alireza Rabiei Kashanaki[2], Elham Farahani[3]
[1]Department of Information Technology, K. N. Toosi University of Technology, Tehran, Iran
[2]Department of Computer Engineering, Iranian University, Tehran, Iran
[3]Department of Computer Engineering, Sharif University of Technology, Tehran, Iran

Article Info

Article history:

Received Jul 8, 2022
Revised Oct 23, 2022
Accepted Nov 6, 2022

Keywords:

Agile development
Risk management
Software assessment process
Software development project
Software project management
Software quality

ABSTRACT

In software projects that use the Agile methodology, the focus is on development in small iterations to allow both frequent changes and client involvement. This methodology affects the risks that may happen in Agile software projects. Hence, these projects need a clear risk management process to reduce risks and address the problems before they arise. Most software production methodologies must use a framework for risk management, but currently, there is no such framework for the Agile methodology. Therefore, we present a risk management framework for projects that use the Agile methodology to help the software development process and increase the likelihood of the project's success. The proposed framework states the necessary measures for risk management according to the ISO31000 standard at each stage of the Agile methodology. We evaluated the proposed framework in two running software projects with an Agile methodology by a number of expert experts. The results show that using our proposed framework increases the average positive risk reaction score by 49%.

Corresponding Author:

Elham Farahani
Department of Computer Engineering, Sharif University of Technology
Azadi Street, Tehran, Iran
Email: elham.farahani@sharif.edu

1. INTRODUCTION

The topic of risk management was introduced for the first time in 1989 as a new and independent research topic, and the risk-oriented spiral life cycle model was introduced as the first life cycle model of risk management [1]. The complete process of software development includes all stages from requirements engineering (recognition and determination of requirements) to the stage of testing and delivery and maintenance of the software, which is carried out based on one of the methodologies in a certain period of time, and finally this process leads to the production of a soft product. Each software project has a series of limitations and deadlines in the schedule and the use of different resources (financial resources, human resources, and hardware resources.

Risk is an inseparable element of all stages of the project development process, and therefore risk management is an important and necessary part of the decision-making process at each stage of the project. Risks can affect productivity, quality of the final product, timely completion of the project and other resources [2]. Today, as software projects become more complex, more effective risk management is needed for the success of the project [3].

A complete risk management process includes all the activities needed to predict, identify, and eliminate risks. Risk management includes methods, processes, and artifacts that continuously identify, analyze, control, and monitor risks to reduce the risk of project failure [4]. The software development risk

management process includes 4 sub-processes [5]: i) risk identification (identification of all potential events that can have an adverse effect on the software development process and even the final product itself); ii) risk analysis (assessing the effect and adverse outcomes of the identified risk); iii) risk planning (strategic planning to reduce risk happening and determine measures to reduce the adverse effects of risks in the wake of the event); and iv) risk monitoring and control (tracking the risks according to the plan).

Since 2000, the software industry has shifted towards adopting Agile methods that are lightweight and prone to change as opposed to traditional methods [6]. The Agile methodology is an iterative approach to project management where a large project is divided into smaller tasks, which are completed in relatively short iterations throughout the lifespan of the project. It is well suited to handle changes to the project requirements and issues arising in each iteration. Furthermore, constant collaboration between project team members and the project stakeholders is maintained throughout the project lifespan. According to market research conducted by the project management institute the performance metrics of organizations adopting the Agile methodology have been much better, for example: i) Agile organizations have achieved 75% of their goals, while non-agile organizations have achieved only 56%; ii) Agile organizations have completed 65% of their projects on time but for non-agile ones this figure is 40%; iii) Agile organizations have completed 67% of their projects within the planned budget, compared to 45% of non-agile organizations; iv) The revenue of agile organizations has grown 37% faster; and v) Agile organizations have gained 30% more profit.

It is clear there are advantages to using the agile methods and especially technical teams can benefit from this project management approach. Even agile methods have the potential to be integrated with other specific management methods or approaches such as Six Sigma in order to achieve agile goals [7]. Despite its advantages, software projects that use the agile process still face risks specific to the Agile methodology. For example, in these projects, the focus is on development in small iterations, where changes happen frequently, and the client is expected to be heavily involved in the process. This naturally affects the risks that agile projects may face. Furthermore, collaborations within the development team and involvement of the clients are of crucial importance. Hence, the risk of not sharing the right level of knowledge with other stakeholders can significantly hinder effective collaboration and/or involvement. Another issue is that most software developers perceive risks in different ways. This can contribute to instability, inefficiency, and project failure, which is why risk management in software projects should take place on a regular basis. Therefore, understanding what the risks associated with agile projects are, and how they are managed, is important for these projects today.

Currently, most software project management methodologies provide a framework for risk management. However, this is not the case for the Agile methodology. Hence, we intend to provide a risk management framework to facilitate the Agile software development process and to increase the chances of projects' success. Our proposed framework is in line with ISO31000 [8], which is a global standard for risk management. Also, we considered a comprehensive reference book and documents which informs practitioners about methodologies, tools and techniques [9]–[16].

In this paper, we first review the past work in this area and examine their shortcomings. Then we present the proposed risk management framework in section 3. In section 4 we evaluate this framework, and finally in the last section we state the results.

2. RELATED WORKS

In this section, the previous studies conducted in the form of books, papers, and theses in risk management area in Agile software development method and also the upcoming challenges are discussed. In [17], a model is proposed to optimize forecasting and risk management in software development projects. In the proposed model, it is presented using computational methods based on risk analysis and according to a real and practical case study on software development methods. However, this model is not designed for Agile software development. In [18], the difference between project management and risk management in traditional and Agile methods has been investigated. Its purpose is to present a new solution and propose a risk registration component to improve risk management in projects based on the Scrum framework. The proposed risk registration component identifies risks before each sprint to reduce the projected risks. This by its nature will reduce the time and also the expense of software development.

Moran [19] explicitly examines the topic of risk management in Agile software development, where risk management is of importance at the organizational and not just the project level. After introducing and comparing a number of Agile methodologies, this book examines the traditional model of risk management, studies the various aspects of risk management in the introduced methodologies, and then discusses the integrated model. Moran also explains the implementation and tools needed for it in different stages and in terms of roles, rules, project content and risk environment, targeting and determining the risk range, risk mixing as well as risk management. On Agile XP, Scrum and dynamic systems development method (DSDM) methodologies.

Uikey and Suman [20] presented a risk management framework to improve the software development process using Scrum methodology. The proposed framework includes five steps: risk identification, risk transparency and quantification, risk response planning, risk monitoring and control, and risk assessment. Also, in a similar study [21], a risk management experiment in the Scrum development process is presented. In this experiment, a case study including 6 teams has been conducted.

A risk identification and mitigation process for Scrum software development methodology is proposed in [22]. The purpose of this proposed process is to identify and reduce the effects of the main risks associated with the main factors of the project when using Scrum. In [23], a risk management method is presented in a model format. This model is a quick tool to support risk identification, assessment, and monitoring. In this article, the effectiveness of the risk management method presented in agile information systems development projects using DSDM has been investigated, and for this purpose, a case study has been used to analyze the levels of risk identification, estimation and evaluation in the DSDM method.

Coyle and Conboy [24] described the study of risks and their management in agile projects that are distributed on a global scale, and hence, pose different risks due to their global distribution. Finally, all of these cases, along with their problems or shortcomings, are listed. In [25], an overview of risks in agile development environments and strategies used to reduce the effects of risks has been made. For this reason, a survey was conducted among several agile developers. In this paper, it has been concluded that project deadlines and different requirements are two important risks that developers usually face.

In [26], it has been discussed how to act requirements risk management in agile projects and the effect of choosing an agile development method on requirements risks. The results of this study show how effective and challenging the requirements risk assessment is in agile projects. Because it depends directly on the people who do this work and they must have a wide range of knowledge of the project as well as the organization in which it exists. A risk management tool is designed and developed using spreadsheets in [27]. In this proposed tool the important information about project such as budget, budget of risk management can be imported into the tool and the tool analyze the priority of practices in and decide which practice must be done at first.

In [28], a risk management framework for Scrum is proposed. In this framework the project management body of knowledge (PMBOK) is used as a project risk management pattern. A risk management framework for XP development project was proposed to increase the XP project success rate, while using the ISO31000 as a project risk management pattern [29]. A risk management model in the hybrid methodology, combining Scrum and XP was proposed in [30]. Using this model showed this model's success in achieving risk management purposes.

Distributed Agile software development is associated with new risks due to the many differences in the nature of work compared to a non-distributed process, which in the study [31] presents a new framework based on artificial intelligence for risk management in distributed Agile software development has been presented. In [32], an agile development simulator is presented, which based on some important risk factors and their modeling, predicts the expected time and effort to complete the project and the probability of risk happening. This simulator uses the Jira tool, used for project management, to receive information such as project duration, number of implemented issues, and key statistics of issue completion time [32].

In summary as we can see in Table 1 as shown in appendix, in this section, a variety of risk management methods, models and frameworks were studied for Agile software development methodologies. Each of the proposed approaches has drawbacks and shortcomings, and in addition, none of them support all aspects of risk management for Agile methods [33]. As a result, in this study, our goal is to present a risk management framework in Agile methodology, considering all the stages of the risk management process.

3. PROPOSED FRAMEWORK

In the Agile framework, there is no specific method and method for risk management. As explained above, a strategy to reduce risk can be defined in which traditional risk management is customized based on Agile methodology. Below, we describe the risk management process from an agile perspective according to the steps defined in ISO31000 [14].

3.1. Agile risk identification-communication and consultation

Risk identification involves thinking about what might happen and why it might happen, and identifying the list of risks that threaten the completion of the project. There are several obstacles to the completion of a project that include both risks and issues. These can be highlighted during agile daily sessions, many of which are resolved by the team immediately during the session. Initially, due to the nature of the agile project process, there is a greater likelihood of identifying risk elements. Planned threats are highlighted by both the empirical nature of agile project planning and agile project control, which ensures development speed and is constantly re-calibrated.

3.2. Agile risk analysis-risk assessment

Risk analysis is the process of evaluating the probability and effect of each risk. The project manager can do the rating himself or use an expert to do it. Both can be rated on a simple scale from 1 (low) to 3 (high).

3.3. Priority of agile risk, scope, and criteria

In risk prioritization, the project manager identifies significant risks and calculates risk exposure by multiplying the probability (scale 1 to 3) by the effect (also on scale 1 to 3). The numerical value obtained is between 1 to 9. Any risk in the range of 6 to 9 is a significant risk that must be managed. Risks in the range of 1 to 5 are not worth managing. The goal of risk prioritization in the Agile framework is to ensure that the task with the highest risk is completed in the first attempt. This is a continuous and dynamic process throughout the project that starts with the ranking of features and allows you to change priorities and add new information.

3.4. Agile risk management planning-monitoring and review

In planning risk management, the project manager decides based on the nature of each risk what method to adopt to deal with that risk in general. There are different methods to manage risk. There are four methods to risk management planning:

a. Risk protection: this means accepting the loss if it happens. This approach is adopted if the cost of handling the risk outweighs the loss caused by the risk.
b. Avoid risk: in this approach, the risk is avoided by not undertaking the activity that carries the risk.
c. Risk reduction: this includes any method that effectively reduces the likelihood of risk, hence, in turn, exposure to risk.
d. Risk transfer: this involves asking someone else to take the risk.

Agile risk management also suggests some technical methods: i) agile teams are collaborative in nature, so in an agile team, the responsibility for handling a specific risk can be shared; and ii) the agile approach also promotes the idea of examining risk requirements. This means, whether as a feasibility study (DSDM) prototype or as an acceleration (XP) prototype, spending some time reviewing technical requirements and technical issues and the related solutions.

3.5. Agile risk resolution-risk treatment

Managing risks requires implementing a risk management program to deal with any significant risk. The agile team has to undertake an action to mitigate or eliminate the risk. It must be included in the planning of risk management.

3.6. Risk monitoring-recording and announcing

The risk management program must be constantly monitored by the project manager to be able to deal with the risks even if the risks are dealt with outside the agile team. The publication schedule should include any agile teamwork. Finally, we return to the beginning of the risk management cycle (discovering, identifying, tracking, monitoring, and mitigating,) because the project manager must continue identifying the risks.

It can be said that agile risk management should be done at two levels, namely, at the level of the project and also at the level of iteration or sprint. The risk management process at the project level is undertaken by considering the whole project and its requirements at a broader level. The iteration level of the risk management process is performed by taking into account the details of the iteration. These two risk management processes may seem to be separate but go hand in hand throughout the project.

Discovering, identifying, tracking, monitoring, and mitigating risks are required at both levels. The project-level risk management process provides inputs to the iterative level risk management process. It should be noted that not all project-level risks are part of each individual iteration. Some risks may vary for different iterations, while others may be unique to a particular iteration. Risk monitoring happens during iterations and risk assessment sessions are held between iterations. It is imperative to know where the project and iterations are in terms of risk management [34]. We can see risk management framework in Agile methodology based on ISO31000 standard in Figure 1.

3.6.1. Risk management process at the project level

The project-level risk management process involves a number of activities. These activities include identifying, planning, and monitoring the risks, and finally the closing activities at the end of the project. Each of these activities is important as described below.

- Risk identification: in this stage, which is always at the beginning of the project, project risks are identified and included in a list of risks.
- Risk planning: in the planning stage, for each of the risks identified in the identification stage, plans to reduce the effects of the risks are presented. Project risk registration is done at the beginning of the project

by the scrum master/project manager. Project risks are also recorded during the project and the list of risks is continuously updated.

– Risk monitoring: all previously identified risks are reviewed and monitored between project iterations. Also, due to closed risks and any new risks identified during the project, the list of risks is updated. This information is then used to discuss the level of replication.

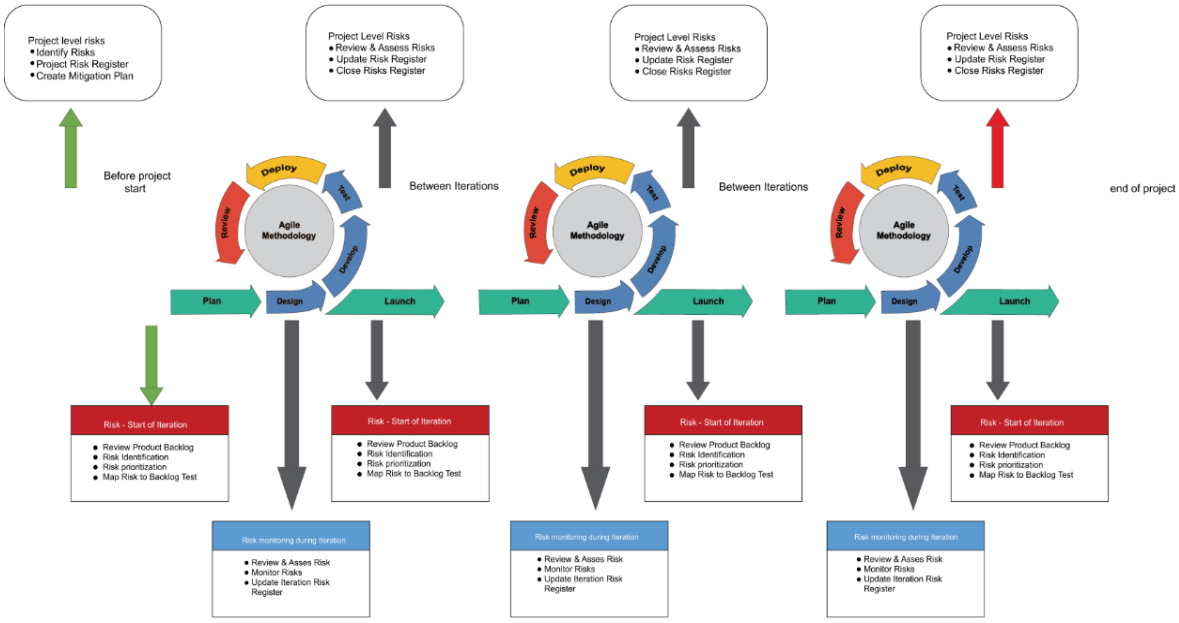

Figure 1. Risk management framework in Agile methodology based on ISO31000 standard

3.6.2. Risk management process at the end of the project

The risk history is updated based on the details of the tasks exposed to risk during the project. This record can be a useful reference for future projects. It is very important to maintain and update it.

3.6.3. Risk management process at iteration level

The risk management process at the iteration/sprint level begins right after the risk identification and initial planning process at the project level. There are generally two risk management tasks for an iteration. First is identification and planning which are performed at the beginning of the iteration, and second is recording and announcing which is undertaken at the time of iteration.

3.6.4. Begin of iteration

At the start of the iteration, a risk management session is held. This includes several steps such as discovery, identifying, analyzing, prioritizing, and mapping the risks. This brainstorming discussion may take 2 to 3 hours to engage all team members. The inputs to the project-level discussion may be considered as inputs for this discussion. Each step is described below.

– Discovery: the team must clearly discover the needs of the user, whether functional or non-functional. Discovering the requirements in the risk identification process will be very helpful.
– Identification: once we have an initial draft of the product backlog, with a good discovery of the requirements, the team can continue the identification process. During the session, each team member must identify potential risks. There are many ways to do this. One simple and effective way is to use sticky notes, for identifying stories on the backlog. As a best practice, no questions are asked or discussed in this session. This is because this should be a time box session which must not overrun the allocated time.
– Analysis: in this step, the identified risks are analyzed and grouped into logical categories or areas: e.g., infrastructure, process, and third parties. During this step, each risk is also ranked (actually the scale does not matter, but it should be kept simple). After completing these activities, the scoring/rankings of the grouped risks are counted. Instead of scoring or ranking, another possibility is to assign probability and impact positive marks.

- Prioritization: after collecting points/rankings for each group, they are ranked in descending order which means the risk group with the highest risk rating is at the top of the list. Now you select three risk group types and leave the remaining risk groups for future discussions. When a new iteration is performed, the previous version of the risk record (updated throughout the previous iteration) is checked to see if any risks still exist. The same process will be repeated until the end of the project.
- Map: mapping is a quick exercise that is done just before the start of the run. The five main identified risks are mapped to backlog/need. This is necessary for close monitoring when implementing this requirement. If there is not any backlog/need, is created. Now a recurring risk record is created for each of the identified risks by reducing the risk reaction (risk response) plan. This will be a reference point for the team during the iteration period. This document is updated during the run execution cycle.
- Activities in an iteration: monitoring is key to risk management and is performed in the time of each iteration. The team responds to risks during project execution/replication in accordance with risk recording. Scrum Master/Project Manager keeps a record of any risk in case it is repeated. If a delay happens and is considered as a risk during the iteration, it will be considered as part of the next iteration. Delayed tasks should never be wasted unless it is necessary to address them as part of a review.

There is useful information and other details that can be added to specific tools which can be used to perform identification, rankings (based on custom quantitative and qualitative indicators), and risk tracking during sprinting. An example of such a tool is the Risk Burn Chart that can be created based on counting all risks. Although the use of Agile methods reduces risk in the early stages of the software development process, we must also consider the requirement to study risk management in a more formal way.

4. EVALUATION OF THE PROPOSED FRAMEWORK

In this part, we evaluate our proposed risk management framework. We selected a software manufacturing company that has an Agile methodology for software development in order to evaluate our proposed framework. and we asked 16 experts who work with Agile methods to help us in this assessment. In the first phase of evaluation, experts were asked to plan the risk reaction (risk response) for two projects without our proposed framework, and then twenty days later they were asked to plan the risk reaction for the same two projects using our proposed framework.

We evaluated our proposed framework on risk reaction planning, as this process has the following characteristics:

- How to plan the react to risk is very important in the acceptability of risk management, because it makes it possible to prevent and control risks by implementing appropriate strategies for reaction to risks.
- How to plan the react to risk has a notable effect on the successful finishing of the project.
- How to plan the react to risk is a process that should not be ignored in projects.
- Only a small number of tools were described in the work performed section, effective planning to react to risk.

The evaluation of the framework included the following three steps [13]:

- The first phase includes the selection of contributors in the evaluation, which we tried to select from people with different levels of experience in Agile methods and risk management.
- The second phase, contributors were asked to evaluate both projects and provide risk reaction plans without using our proposed framework.
- The third phase, fifteen days later, the previous contributors were asked to evaluate the same two projects, but this time using our proposed framework for developing risk reaction plans.

4.1. Case study

In this part, we evaluate the proposed framework using a case study. To start the evaluation, we got the help of 16 people, including the customer, the production team, and two Scrum Masters. In choosing these people, we paid attention to the level of work experience in agile development environments, as well as the level of familiarity with the types of risks and how to manage them, as well as the level of individual threshold. Because these factors are very important in our experiment. We defined a threshold for each person to ensure that all contributors have experience working with Agile methods and risk management. In addition, we tried to involve people with different risk thresholds in the evaluation to have a wide range of behaviors when managing risk. Table 2 shows the number of people involved in the test as contributors based on work experience in agile development environments and risk threshold characteristics. We conducted this test for two Agile software development projects at the Smart Land Solution Company, both of which involved more than 5,000 hours of effort and took 5 to 10 months to complete.

Based on the actual events of two projects, we considered the following criteria for selecting project risks for testing [13]: i) the risk has happened and the reaction plan has been helpful in reducing its effect; ii) the risk did not happen because the reaction plan was helpful in reducing its probability; and iii) no risk happened because the reaction plan was effective in eliminating it.

Table 2. The level of experience and risk threshold of contributors

Risk thresholds	Experience	Contributors
Low	High	4
High	Low	4
Low	Low	4
High	High	4

The considered criteria are effective in categorizing the respondents and evaluating the answers provided. Each contributor in the experiment was given information about eight risks (Six risks for project 1 and six risks for project 2). The contributors were then asked to come up with the best reaction plans for each hazard. Scrum masters were responsible for reviewing the risk reaction plans for each project. They also compared the evaluation process of the contributor's reaction with the real risk reaction plans used in the two projects.

For each criterion, we calculated a collective pre-test positive score and a collective post-test positive score in two projects. By comparing these positive pre-test and post-test summary scores, we analyzed the effectiveness of the framework in risk reaction planning. For this purpose, we used IBM SPSS statistical software. First, we created a descriptive analysis to check the average positive score of the risk reaction using the framework and then without using the framework for two projects.

According to Table 3, using the framework, the positive sign of the average risk reaction increased. The rate of improvement using the proposed framework is about 49%. This result shows that this framework has helped the contributors to provide more helpful risk reaction scripts. Table 4 provides a descriptive analysis of the framework for each project. By using the proposed framework, the average positive scores of risk reaction in project 1 increased by 65% and in project 2 by 42%.

Table 3. Descriptive analysis of framework factors

	Average	Minimum positive mark	Maximum positive mark
No framework	9.7	2	17
Using the framework	14.5	5	26

Table 4. Descriptive analysis of framework factors for each project

	Project 1			Project 2		
	Average	Minimum positive mark	Maximum positive mark	Average	Minimum positive mark	Maximum positive mark
No framework	3.7	2.7	4.7	5.9	4.7	7.06
Using the framework	6.1	4.8	7.3	8.3	7	9.6

5. CONCLUSION

In order to design the proposed framework, some previous works were first examined to identify their strengths and weaknesses. In addition, by reviewing the literature, we find that there is no comprehensive risk management model, method, or framework for the Agile method. The proposed framework states the necessary measures for risk management according to the ISO31000 standard at each stage of the Agile method. Subsequently, 16 experts who each had at least six months of experience in using Agile software development methodology were invited to evaluate the proposed framework in the context of two real projects, once without using the framework and again using the framework. Each project involved around 5,000 hours of work and had a team of 16 people. The effectiveness of the risk reaction provided by the framework was evaluated using IBM SPSS software and good results were obtained. The framework was also evaluated in terms of its time overhead for software development projects with the Agile method and the results showed it imposes no time overhead for such projects. The following is suggested for future work: i) development of a risk management framework for each of the Agile methods such as feature-driven development (FDD) and Kanban; ii) evaluating the advantages and disadvantages of the proposed framework in different projects; and iii) ssess the experience of person involved in risk management tasks when using the proposed framework.

APPENDIX

Table 1. Reviewed papers and research

Research	Problems or Shortcomings
A predictive optimization of risk management model [17]	Not designed for Agile software development
Risk management in Scrum-based projects [18]	Limited to Scrum methodology
Agile risk management book [19]	Model adaptation to Agile XP, Scrum and DSDM methodologies
Risk based Scrum framework [20]	The model is limited to the general model not Scrum
A risk testing model for Scrum [21]	Limited size of the team Limited to Scrum methodology
Risk identification and mitigation framework for Scrum methodology [22]	Limited size of the team Limited to Scrum methodology
Risk management in Agile method [23]	
A case study of risk management in Agile method [24]	Case study
A survey on risk management in Agile method [25]	The result was obtained as a poll and was not presented
Requirements risk management in Agile method [26]	Limited to project requirements risks
A risk management tool for XP methodology [27]	Limited to XP methodology
Risk management framework in Scrum methodology [28]	Limited to Scrum methodology
A risk management framework for XP methodology [29]	Limited to XP methodology
A risk management model in the hybrid Scrum and XP [30]	Limited to XP and Scrum methodology
A risk management framework for Agile method in distributed software development environment [31]	Designed only for distributed Agile software development environment
Agile development simulation to model risks of project [32]	Limited to data from project management tools as Jira

REFERENCES

[1] B. W. Boehm, "Software risk management: principles and practices," *IEEE Software*, vol. 8, no. 1, pp. 32–41, Jan. 1991, doi: 10.1109/52.62930.

[2] D. Crnković and M. Vukomanović, "Comparison of trends in risk management theory and practices within the construction industry," *Elektronički časopis građevinskog fakulteta Osijek*, pp. 1–11, Dec. 2016, doi: 10.13167/2016.13.1.

[3] S. M. Arikan and O. Yurekten, "Software risk management process improvement experience in enterprise projects," in *2020 Turkish National Software Engineering Symposium (UYMS)*, Oct. 2020, pp. 1–5, doi: 10.1109/UYMS50627.2020.9247026.

[4] B. Verma and M. Dhanda, "A review on risk management in software projects," *IJIRST –International Journal for Innovative Research in Science & Technology*, vol. 2, no. 11, pp. 499–503, 2016.

[5] M. Kajko-Mattsson and J. Nyfjord, "State of software risk management practice," *IAENG International Journal of Computer Science*, vol. 35, no. 5, 2008.

[6] J. Nyfjord, "Towards integrating agile development and risk management," Institutionen för data-och systemvetenskap (tills m KTH), 2008.

[7] M. N. Sarpiri and T. J. Gandomani, "A case study of using the hybrid model of scrum and six sigma in software development," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 11, no. 6, pp. 5342–5350, Dec. 2021, doi: 10.11591/ijece.v11i6.pp5342-5350.

[8] ISO, "ISO31000 standard in risk management," ISO standard. https://www.iso.org/iso-31000-risk-management.html (Accessed: Jan 23, 2022).

[9] A. El Yamami, S. Ahriz, K. Mansouri, M. Qbadou, and E. Illoussamen, "Representing IT projects risk management best practices as a metamodel," *Engineering, Technology & Applied Science Research*, vol. 7, no. 5, pp. 2062–2067, Oct. 2017, doi: 10.48084/etasr.1340.

[10] Financial Services Agency, "Principles for model risk management." White paper, Financial Services Agency of Japan, 2021.

[11] 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 (BEEI)*, vol. 9, no. 3, pp. 1299–1310, Jun. 2020, doi: 10.11591/eei.v9i3.1905.

[12] 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.

[13] B. G. Tavares, M. Keil, C. E. Sanches da Silva, and A. D. de Souza, "A risk management tool for Agile software development," *Journal of Computer Information Systems*, vol. 61, no. 6, pp. 561–570, Nov. 2021, doi: 10.1080/08874417.2020.1839813.

[14] A. Albadarneh, I. Albadarneh, and A. Qusef, "Risk management in Agile software development: A comparative study," in *2015 IEEE Jordan Conference on Applied Electrical Engineering and Computing Technologies (AEECT)*, Nov. 2015, pp. 1–6, doi: 10.1109/AEECT.2015.7360573.

[15] M. Pilliang and M. Munawar, "Risk management in software development projects: A systematic literature review," *Khazanah Informatika: Jurnal Ilmu Komputer dan Informatika*, vol. 8, no. 2, Sep. 2022, doi: 10.23917/khif.v8i2.17488.

[16] C. Roos, "Governance responses to hacking in the banking sector of South Africa: An exploratory study," Doctoral Thesis, University of Johannesburg (South Africa), 2021.

[17] S. Firdose and L. M. Rao, "PORM: Predictive optimization of risk management to control uncertainty problems in software engineering," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 8, no. 6, pp. 4735–4744, Dec. 2018, doi: 10.11591/ijece.v8i6.pp4735-4744.

[18] M. Mosaei, T. J. Gandmani, and M. S. A. Kopaei, "Risk assessment and management in Agile software development and presenting a new strategy for improving risk management of scrum-based projects," in *2nd International Conference and 3rd National Conference on Application of new technologies in engineering sciences*, 2015.

[19] A. Moran, *Agile Risk Management*. Cham: Springer International Publishing, 2014, doi: 10.1007/978-3-319-05008-9.

[20] N. Uikey and U. Suman, "Risk based scrum method: a conceptual framework," in *Proceedings of the 9th INDIACom; INDIACom-2015*, 2015, pp. 4.120-4.125.

[21] S. Noor Hasanah Ghazali, S. Salwah Salim, I. Inayat, and S. Hafizah Ab Hamid, "A risk poker based testing model for scrum," *Computer Systems Science and Engineering*, vol. 33, no. 3, pp. 169–185, 2018, doi: 10.32604/csse.2018.33.169.
[22] E. Hossain, M. A. Babar, H. Paik, and J. Verner, "Risk identification and mitigation processes for using scrum in global software development: A conceptual framework," in *2009 16th Asia-Pacific Software Engineering Conference*, Dec. 2009, pp. 457–464, doi: 10.1109/APSEC.2009.56.
[23] E. Odzaly, D. Greer, and D. Stewart, "Lightweight risk management in agile projects," in *26th Software Engineering Knowledge Engineering Conference (SEKE)*, 2014, pp. 576–581, doi: 10.13140/2.1.4681.0882.
[24] S. Coyle and K. Conboy, "A case study of risk management in Agile systems development," in *17th European Conference on Information Systems, ECIS 2009, Verona, Italy*, 2009.
[25] M. Hammad, I. Inayat, and M. Zahid, "Risk management in Agile software development: A survey," in *2019 International Conference on Frontiers of Information Technology (FIT)*, Dec. 2019, pp. 162–1624, doi: 10.1109/FIT47737.2019.00039.
[26] H. Puttonen, "Requirements risk management in Agile software development projects," Master's Thesis, University of Jyväskylä, 2018.
[27] H. Mathkour, G. M. R. Assassa, and A. Baihan, "A risk management tool for extreme programming," *IJCSNS International Journal of Computer Science and Network Security*, vol. 8, no. 8, pp. 326–333, 2008.
[28] S. Chaouch, A. Mejri, and S. A. Ghannouchi, "A framework for risk management in Scrum development process," *Procedia Computer Science*, vol. 164, pp. 187–192, 2019, doi: 10.1016/j.procs.2019.12.171.
[29] A. R. Kashanaki and E. Farahani, "A framework for risk management in XP development process," in *13th International Conference on Information Technology, Computers and Telecommunications*, 2021.
[30] M. Afshari and T. J. Gandomani, "A novel risk management model in the Scrum and extreme programming hybrid methodology," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 12, no. 3, pp. 2911–2921, Jun. 2022, doi: 10.11591/ijece.v12i3.pp2911-2921.
[31] E. Khanna, R. Popli, and N. Chauhan, "Artificial intelligence based risk management framework for distributed Agile software development," in *2021 8th International Conference on Signal Processing and Integrated Networks (SPIN)*, Aug. 2021, pp. 657–660, doi: 10.1109/SPIN52536.2021.9566000.
[32] M. I. Lunesu, R. Tonelli, L. Marchesi, and M. Marchesi, "Assessing the risk of software development in Agile methodologies using simulation," *IEEE Access*, vol. 9, pp. 134240–134258, 2021, doi: 10.1109/ACCESS.2021.3115941.
[33] A. Ceric, "A framework for process-driven risk management in construction projects," PhD Thesis, University of Salford, 2021.
[34] J. Šimíčková, K. Buganová, and E. Mošková, "Specifics of the agile approach and methods in project management and its use in transport," *Transportation Research Procedia*, vol. 55, pp. 1436–1443, 2021, doi: 10.1016/j.trpro.2021.07.130.

BIOGRAPHIES OF AUTHORS

Mohammad Hadi Zahedi received his Ph.D. in computer engineering from Ferdowsi University of Mashhad. He is a faculty member of the Department of Information Technology, K. N. Toosi University of Technology. He has published many papers in national and international conferences and journals. His research interests include software methodologies, big data, data mining, and soft computing. He can be contacted at email: zahedi@kntu.ac.ir.

Alireza Rabiei Kashanaki received his MS degree in computer engineering in software in 2022 from Iranian University. He received a bachelor's degree in computer software from the University of Science and Culture in 2013. His research interests include project management, risk management, and Agile methodologies. He can be contacted at email: alireza.rabiei.kashanaki98@gmail.com.

Elham Farahani received her Ph.D. in computer engineering from Sharif University of Technology. She has published many papers in national and international conferences and journals. Her research interests include software methodologies, software product lines, risk management, and software architecture. She can be contacted at email: elham.farahani@sharif.edu.