Managing Risks in Agile Methods: a Systematic Literature Mapping

Fernando Vedoin Garcia
Department of Informatics and Statistics
Federal University of Santa Catarina
Florianópolis, Brazil
fevedoingarcia@gmail.com

Jean Carlo R. Hauck
Department of Informatics and Statistics
Federal University of Santa Catarina
Florianópolis, Brazil
jean.hauck@ufsc.br

Fernanda N. Rizzo Hahn
Department of Informatics and Statistics
Federal University of Santa Catarina
Florianópolis, Brazil
fernandanrizzo@gmail.com

***Abstract*—Agile software development methods have been around since at least 2001. They accommodate changing requirements with the flexibility to deal with cost and scope and have increasingly been used. However, explicit risk management is often ignored as agile methods deal with risk intrinsically and focus on rapid value delivery. In certain contexts, explicit risk management practices are needed to complement agile methods. Thus, this paper presents a systematic literature mapping aiming to discover how do software organizations integrate explicit risk management practices into agile methods. As a result we found 23 primary studies that, in majority, applied case studies in the industry, using agile methods such as Scrum, and adapting agile practices such as Daily Meeting and Iteration Planning Meeting to manage risks related to schedule and communication, for example. The selected primary studies raise evidence that the introduction of explicit risk management practices bring benefits to agile methods.**

***Index Terms*—software, risk management, agile methods, agile practices**

I. Introduction

Agile software development methods [1] have been widely used in software organizations due to their ability to accommodate changing requirements and flexibility to handle cost, scope and software quality according to customer needs [2]. One of the main advantages of adopting agile methods is their ability to reduce risks [3], which leads to successful and timely software development and deployment. Projects that apply agile methods usually make use of frequent reviews in each development cycle and cross-functional project teams to accelerate knowledge sharing and ensure that risks are understood and implicitly managed [4]. The implicit ability to reduce risks has also been one of the main reasons for adopting agile methods in software organizations [5].

However, despite its importance, risk management is often overlooked in agile software development methods as its focus is on rapid value delivery [6]. Even with the adoption of agile methods and investments in software development, failure of software projects is still frequent, increasing the importance of the software development risk management [7].

The explicit application of risk management consists of inserting principles and practices of risk management in the already used practices of lifecycle management [7], thus risks can be identified, analyzed and managed during each software development iteration [4].

Complementing agile methods with explicit risk management practices, has attracted recent interest. Esteki et al. [8], integrates Scrum with the PRINCE2 delivery layer; Schön et al. [9] standardizes risk increasing transparency in the context of multidisciplinary projects; Hayat et al. [10] estimate the impact of risk and convert it into risk detection and control actions. Risk management in software projects has even attracted the application of Machine Learning (ML) aiming to identify or predict risks before project development starts [11].

However, the existing Software Engineering literature lacks insights into the extent to which the combination of agile methods and risk management processes is being applied [6]. Thus, this paper presents a Systematic Literature Mapping (SLM) [12] to answer the research question "How do software organizations integrate explicit risk management practices into agile methods?".

The main contributions of this work are twofold: (i) for Software Engineering researchers we present an extensive survey, to the best of our knowledge, of the state of the art of risk management practices in agile methods; (ii) for practitioners that are seeking to include explicit risk management practices in agile methods we present the most used risk management practices and typical managed risks.

II. Related Works

As primary studies have reported the integration of explicit risk management practices into agile methods, some secondary studies have analyzed this phenomenon from different perspectives.

Vieira, Hauck, and Matalonga [13] conducted an SLM in order to understand how explicit risk management is being integrated into agile software development methods. With 18 selected papers, authors found that the results of integrating explicit risk management with agile methods are positive. The secondary study, however, is not focused on empirical primary studies and not addresses which risk management practices have been applied empirically in real environments.

DOI reference number: 10.18293/SEKE2022-123

Chadli and Idri [14] identified risk mitigation strategies that target Global Software Development (GSD) through a Systematic Literature Review (SLR). The analysis of the 24 selected primary studies resulted in 39 risk factors and 58 mitigation strategies. The strategies were classified by areas such as task-actor, task-structure, and task-technology. The secondary study, however, do not analyze risk management practices nor the specific context of use.

Podari et al. [15] conducted an SLR selecting 52 papers that identify the risks and challenges that affect globally distributed projects and how agile methods can be useful in managing these barriers. The selected primary studies are only focused on GSD, not covering other types of projects.

Thus, it was not possible to find so far in the literature a comprehensive analysis of the introduction of explicit risk management practices in agile methods and the specific practices adopted.

III. Methods

In order to analyze the state of the art of the integration of explicit risk practices in agile methods, we undertook a Systematic Literature Mapping (SLM) following the procedures defined by Petersen, Vakkalanka, and Kuzniarz [12], Petersen et al. [16], and Wohlin [17]. Based on the identified research need, the general research question was defined as: "How do software organizations integrate explicit risk management practices into agile methods?". Thus, we derived the main research question in four detailed analysis questions, as presented in Table I.

TABLE I
Research questions

	Description
Q1	What are the studies that deal with the integration of risk management practices in agile methods?
Q2	What is the context of use of risk management practices in agile methods?
Q3	What risk management practices are introduced in agile methods?
Q4	What types of risks are managed?

A. Search strategy

The search string was defined following [18], using the most used agile methods [5] and well known terms as synonyms for "agile methods". The search string was then tested and refined by the authors, using previously known primary studies as a reference, resulting in the following search string:

"risk" AND ("agile" OR "scrum" OR "xp" OR "extreme programming" OR "lean" OR "kanban" OR "scrumban" OR "fdd" OR "feature driven development" OR "crystal" OR "iterative development") AND "software"

The search string was applied to the following digital libraries: IEEEXplore, ACM Digital Library, and Scopus, due to their relevance to the software engineering area [19]. The search string was adapted to the specific syntax of each library and applied to title and abstracts fields. The Snowballing technique [17] was also performed using the selected papers from the automated search as input.

Based on the main research question, the following inclusion criteria (IC) and exclusion (EC) criteria were defined: (IC1) Peer reviewed primary studies; (IC2) Written in English; (IC3) Full papers with at least 4 pages; (EC1) Theoretical work/proposal not empirically applied; (EC2) Duplicate studies; (EC3) No full text available; (EC4) Not focused on software development.

B. Study Selection

The selection of studies was performed from July to December of 2021 in four cycles, as presented in Fig. 1.

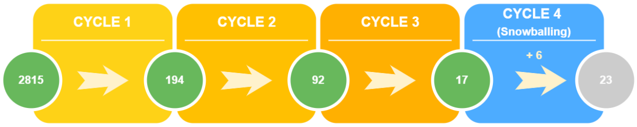

Fig. 1. Number of primary studies by cycle.

In the Cycle 1 the search string was applied to the digital libraries. The resulting list of 2815 primary studies was then divided between the first and third authors, who separately applied the inclusion and exclusion criteria to all paper titles, peer reviewing the results. This initial selection was reviewed by the second author, resulting in 194 selected papers. In Cycle 2, the initial list of papers was filtered by the first and third authors applying the inclusion and exclusion criteria to the papers' summaries, resulting in 92 selected papers, once again reviewed by the second author. In Cycle 3 we merged the lists of papers and filtered the studies on a full-text basis using the inclusion and exclusion criteria, resulting in 17 selected papers after the second author reviewing. Finally, in Cycle 4 the Backward Snowballing technique [17] was applied by the first author using as input the 17 selected papers, resulting in six more papers being selected. After each cycle a meeting was performed with the three authors resolving any possible discordance or inconsistencies. The number of studies for each digital library and cycle is presented in Table II.

TABLE II
Results per digital library and cycle

Digital library	Total	Cycle 1	Cycle 2	Cycle 3
ACM	971	42	15	5
IEEEXplore	957	79	39	5
Scopus	887	73	38	7

IV. Data collection and analysis

The 23 selected primary studies are distributed between the years 2000 and 2020. The concentration of works (12) between 2017 and 2020, and the exponential trend line, shown in Fig. 2, indicate the growing relevance of this topic in recent years.

Next, data collected from selected primary studies are presented and analyzed according to each predefined Analysis Question. Extracted raw data is available at: bit.ly/36i7Wby.

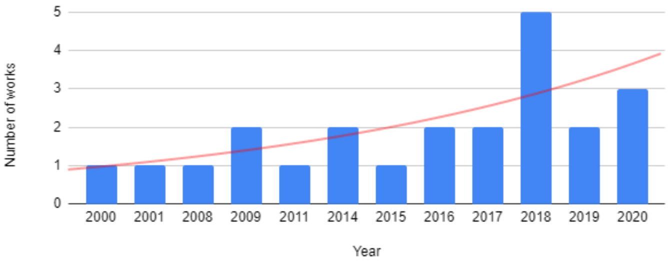

Fig. 2. Distribution of the selected papers per year.

Q1. What are the studies that deal with the integration of risk management practices in agile methods?

The selected primary studies are presented in Table III.

TABLE III
SELECTED STUDIES

#	Title	Ref
S1	Reference Framework and Model for Integration of Risk Management in Agile Systems Engineering Lifecycle of the Defense Acquisition Management Framework.	[20]
S2	A risk management framework for distributed scrum using PRINCE2 methodology.	[8]
S3	A Risk Management Tool for Agile Software Development	[21]
S4	Improving Risk Management in a Scaled Agile Environment	[9]
S5	Risk Assessment Forum	[22]
S6	Agile risk management using software agents	[23]
S7	A risk poker based testing model for scrum	[24]
S8	Agile approach with Kanban in information security risk management	[25]
S9	Integrating Risk Management in Scrum Framework	[26]
S10	Prioritizing and optimizing risk factors in agile software development	[27]
S11	Value-Risk Trade-off Analysis for Iteration Planning in Extreme Programming	[28]
S12	A case study for the implementation of an agile risk management process in multiple projects environments	[29]
S13	A SYSML-Based Approach for Requirements Risk Management and Change Control	[10]
S14	Risk Management for Agile Projects in Offshore Vietnam	[30]
S15	An industrial case study of implementing software risk management	[31]
S16	Characterization of risky projects based on project managers' evaluation	[32]
S17	Characterization and prediction of issue-related risks in software projects	[33]
S18	Outlining a Model Integrating Risk Management and Agile Software Development	[34]
S19	Lightweight Risk Management in Agile Projects	[35]
S20	A risk management framework for distributed agile projects	[36]
S21	Implementation of Risk Management with SCRUM to Achieve CMMI Requirements	[37]
S22	A New Project Risk Management Model based on Scrum Framework and Prince2 Methodology	[38]
S23	Risks to Effective Knowledge Sharing in Agile Software Teams: A Model for Assessing and Mitigating Risks	[39]

Q2. What is the context of use of risk management practices in agile methods?

We define the context of use as: (Q2.1) the type of application environment, (Q2.2) agile method adopted, (Q2.3) type of empirical study and (2.4) number of organizations involved. The context-related data is summarized in Table IV.

The selected studies were applied in two different environments (Q2.1): software industry or academia. 18 (78%) studies were applied in software development organizations and 5 studies (22%) in an academic environment.

Regarding the agile methods adopted (Q2.2), 14 (61%) adopted Scrum, 3 (13%) adopted XP, 2 (9%) cited Kanban, and only 1 (4%) mentioned the Dynamic System Development Method (DSDM), whereas it is not explicit in the search string. Among the selected studies, 7 (30%) did not mention any specific agile method. The total is greater than 100%, as some studies used more than one agile method.

In the industry environment, the agile methods that appeared the most were Scrum (10 - 43%) and XP (3 - 13%). In academia, the predominant method was also Scrum (4 - 17%). Study S3 was the only study applied in academy environment that did not mention any specific agile method.

As for the type of empirical study (Q2.3), 17 (74%) applied case studies, 2 (9%) applied experiments, 3 (13%) applied surveys, and only S8 (4%) applied a proof of concept. It is possible to observe that in the industry most applications were case studies, while in academia there was a balance. The approach proposed in S6 was validated with two case studies.

Most (15 - 65%) of the studies were applied in only 1 organization (Q2.4). Study S17, in turn, was applied in 5 organizations with projects that differ significantly in size, complexity, development process, and community size.

TABLE IV
CONTEXT OF USE

Question	Extracted data	
Q2.1 - Context	Industry	S1, S2, S4, S5, S8, S10, S11, S12, S13, S14, S15, S16, S17, S18, S20, S21, S22, S23
	Academy	S3, S6, S7, S9, S19
Q2.2 - Agile method	Scrum	S2, S5, S6, S7, S9, S10, S12, S14, S18, S19, S20, S21, S22, S23
	XP	S10, S11, S14
	Kanban	S8, S14
	DSDM	S10
	Undefined	S1, S3, S4, S13, S15, S16, S17
Q2.3 - Type	Case study	S1, S2, S4, S5, S6, S7, S10, S11, S12, S13, S14, S15, S16, S17, S18, S19, S20
	Experiment	S3, S9
	Concept proof	S8
	Survey	S21, S22, S23
Q2.4 - Instances	Exactly 1	S1, S2, S4, S5, S6, S7, S8, S11, S12, S13, S14, S15, S16, S18, S19
	Between 2 and 10	S10, S17, S20, S23
	Undefined	S3, S9, S21, S22

Q3. What risk management practices are introduced in agile methods?

Two different strategies were adopted by the studies to integrate risk management into agile methods: using existing agile practices or introduce new risk management practices.

Studies S3 and S9 adopted the first strategy. The most commonly used practices are Brainstorming, Pair Programming, Daily Meetings, Incremental Deliveries and Prototyping.

Adopting the second strategy, the other primary studies have created new agile practices or introduced adapted traditional practices into agile methods to improve risk management. Table V presents the introduced risk management practices, grouped by the existing agile practice impacted (when applicable). Some examples of practices are described below.

TABLE V
PROPOSED PRACTICES

Impacted agile practice	#	Proposed practice
Initial sprint planning	S22	Define obligations of individuals
Sprints	S22	Link processes to sprints
Release	S22	Progress report
Sprint planning meeting	S9	Brainstorming
	S14	Risk register
	S18	Identify the responsibilities of individuals
Daily meeting	S5	Risk Assessment Forum
	S6	Automatic agents
	S12	Impediment matrix
	S19	Automatic agents
	S20	Risk ranking
	S23	Risk list
Sprint review meeting	S6	Automatic agents
	S9	Brainstorming
	S19	Automatic agents
Sprint retrospective meeting	S21	Risk register
	S23	Risk list
Planning meeting	S23	Risk list
User stories	S6	Automatic agents
Continuous integration	S20	Risk ranking
Pair programming		
Face to face communication		
Flexible design		
Customer software demos		
Backlog management		
Iteration planning meeting	S7	Risk Poker
	S11	User stories repository
	S14	Risk register
Risk management meeting	S14	Qualitative risk analysis matrix
		Risk decomposition structure
		Risk cards
Kanban board	S8	Risk distribution
	S14	Risk closing
Work planning (Kanban)	S14	Risk register
		Qualitative risk analysis matrix
		Risk decomposition structure
		Risk cards
–	S1	Feedback loop
	S2	Identify responsibilities
	S3	Practice recommending tool
	S4	Initial meeting
	S10	AR Rank
	S13	Model-driven requirements
	S15	Risks checklist
		Brainstorming
		Analysis charts
		Forms
	S16	Quiz
	S17	Predictive risk identification

In S5, the Risk Assessment Forum (RAF) is proposed to be applied weekly in daily meetings. Thus, the development team and the Scrum Master can increase the identified risks and manage them. Study S9, inserted two brainstorming sessions, after the Sprint planning meeting to identify potential risks and in the Sprint review meeting to risks documentation. The practices proposed in S23 provide heuristics that facilitate risk analysis, prioritizing resolutions, and linking them into an overall plan. The proposed risk management process also involves team members in several informal knowledge-sharing exercises assisting decision-making and forming a risk list with their respective resolutions.

The Risk Analysis practice, proposed in S12, is defined for the XP method to reduce risks of user story overload by providing several alternative plans to improve negotiations between different stakeholders, promoting a deeper understanding and helping to choose a development plan with the greatest chance of being implemented on time.

In S8, authors propose an intervention in the Kanban workflow. In this new practice, identified risks are distributed to team members with defined roles. This provides a clear view of each person's tasks and responsibilities regarding risks. Using selected risk factors, study S17 developed models to predict whether a risk will cause a delay. If so, the model also determines the risk impact and the probability of occurrence.

Q4. What types of risks are managed?

The selected primary studies identify a total of 230 risks. Due to this large number of risks described in different ways, we decided to group them using a well known risks taxonomy [40], [41], [42] that provides three risk classes, its elements and attributes. We have collected all risks reported in the selected primary studies, classified according to the taxonomy, and summarized in Table VI. The complete list of risks, its sources and our chosen classifications is available at: bit.ly/36i7Wby.

The primary studies that reported the highest number of risks were [S2], [S20], and [S23]. Studies [S2] and [S20] were the works that have more risks classified in different attributes (25), followed by [S23], with risks classified in 19 attributes.

The attribute with the most risks occurrences was Schedule, with occurrences in 9 primary studies (39%), followed by other attributes from Budget and Staff (6 - 26%). The element with most risks occurrences was Requirements, with occurrences in 11 primary studies (48%). The class with the greatest number of occurrences was Development Environment, with occurrences in 13 primary studies (57%).

V. DISCUSSION

The results of this secondary study summarize information on the application of explicit risk management practices in agile software development methods.

As the wide majority of the selected studies were applied in industry with reported benefits, this raises evidence of the adequacy of explicit risk management practices in agile methods. All selected primary studies report positive impacts of introducing explicit risk management practices, with 10 studies (43%) [S1, S2, S3, S6, S12, S14, S15, S18, S19, S21] reporting positive impacts without compromising the "agility" of the agile methods.

The vast majority (61%) of the studies applied Scrum, confirming its global trend as the main agile method used

[5]. In contrast, only one study (4%) implemented DSDM, possibly indicating a tendency to disuse of this method.

TABLE VI
CLASSIFICATION OF IDENTIFIED RISKS

Class	Element	Attribute	Studies
Product Engineering	Requirements	Stability	S2, S10, S17, S20, S21
		Completeness	S23
		Clarity	S2, S10, S20, S21, S23
		Validity	S16, S23
		Feasibility	S1
		Precedence	S2, S10, S17, S20, S23
		Scale	S6, S9, S14
	Design	Functionality	S13
		Performance	S13
		Testability	S2, S10, S20
		Hardware Constraints	S9
	Code and Unit Test	Feasibility	S21, S23
		Testing	S2, S10, S20
	Integration and Test	Environment	S1, S14
		Product	S2, S9, S10, S17, S20
		System	S2, S20
	Engineering Specialties	Maintainability	S6, S14
		Security	S9, S13, S21
		Specifications	S9, S16
Development Environment	Development Process	Suitability	S2, S10, S20
		Process Control	S2, S10, S12, S17, S20
		Familiarity	S15, S23
		Product control	S2, S14, S20
	Development System	Capacity	S2, S20
		Reliability	S9, S17
		Familiarity	S6, S9, S15, S23
		Deliverability	S17
	Management Process	Planning	S2, S10, S20, S21, S23
		Project Organization	S6, S16
		Management Experience	S23
		Program Interfaces	S2, S10, S14, S20, S23
	Management Methods	Personnel Management	S1, S2, S6, S23
		Quality Assurance	S12, S14
		Configuration Management	S20, S23
	Work Environment	Quality Attitude	S20
		Cooperation	S2, S6, S10, S17, S20
		Communication	S2, S6, S15, S20, S23
		Morale	S2, S6, S14, S23
Program Constraints	Resources	Schedule	S2, S6, S9, S10, S12, S16, S17, S20, S23
		Budget	S2, S10, S14, S16, S20, S23
		Staff	S2, S9, S14, S16, S20, S23
	Contract	Type of Contract	S2, S20
		Dependencies	S2, S14, S20, S23
	Program Interfaces	Customer	S2, S6, S14, S20, S23
		Corporate Management	S2, S10, S20
		Vendors	S2, S20
		Politics	S2, S14, S23

The existent agile practices most affected by explicit risk management processes were the Daily Meeting and Iteration Planning Meeting, raising evidence that the incorporation of risk management practices especially affects the identification and monitoring of risks, corroborating other results reported in the literature [13].

In addition to the practices, we also have extracted and classified the managed risks. Most studies reported risks related to requirements and communication. The highlight the "delay", which affects 39% (9) of the selected studies.

A. Threats to validity

We have identified potential threats and applied mitigation strategies to minimize impacts on the outcomes following [43].

To reduce the risk of incomplete searches, we have selected, reviewed and tested search terms and also applied the Snowballing technique, which resulted in additional studies.

The number of studies, trend of publishing positive results, and empirical quality of most studies may affect the validity of the conclusions, as we decided to include studies with low empirical evidence to spot trends of topics being worked [12].

VI. CONCLUSION

Considering the lack of risk management processes in agile methods, this paper presents a Systematic Literature Mapping on how software organizations integrate explicit risk management practices into agile methods. We selected 23 primary studies following a defined research protocol.

The data collected indicate that the most used research method is case study. Selected primary studies are mostly applied in the industry using Scrum. The more frequently adapted agile practices are Daily Meeting and Iteration Planning Meeting with the introduction of specific risk management practices such as Risk Poker, Risk Ranking and Risk Register. The risks most frequently identified by studies are related to schedule and communication.

According to the results, explicit risk management practices provided benefits to the agile projects such as the increase in the number of identified risks and the choice of more effective corrective actions, improving team communication and the visibility of impediments, in addition to anticipating problems.

Therefore, the explicit inclusion of risk management practices can help the management of risks in agile projects without hurting the principles of agility, reducing its negative impact, and increasing the chances of success of the projects.

REFERENCES

[1] P. Bourque and R. E. Fairley, Eds., *SWEBOK: Guide to the Software Engineering Body of Knowledge*, version 3.0 ed. IEEE Computer Society, 2014. [Online]. Available: http://www.swebok.org/

[2] F. Hayat, A. U. Rehman, K. S. Arif, K. Wahab, and M. Abbas, "The influence of agile methodology (scrum) on software project management," in *2019 20th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD)*, 2019, pp. 145–149.

[3] 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)*, 2015, pp. 1–6.

[4] Project Management Institute, *A Guide to the Project Management Body of Knowledge (PMBOK® Guide)*, 6th ed. Project Management Institute, 2017.

[5] Digital.ai, *15th annual state of agile report*. Digital.ai, 2021.

[6] 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)*, 2019, pp. 162–1624.

[7] L. Xiaosong, L. Shushi, C. Wenjun, and F. Songjiang, "The application of risk matrix to software project risk management," in *2009 International Forum on Information Technology and Applications*, vol. 2, 2009, pp. 480–483.

[8] 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, 2020.

[9] E.-M. Schön, D. Radtke, and C. Jordan, "Improving risk management in a scaled agile environment," in *Agile Processes in Software Engineering and Extreme Programming*. Cham: Springer International Publishing, 2020, pp. 132–141.

[10] F. Hayat, M. W. Anwar, F. Azam, and A. Kiran, "A sysml-based approach for requirements risk management and change control," in *Proceedings of the 2019 11th International Conference on Information Management and Engineering*, ser. ICIME 2019, 2019, p. 20–24.

[11] A. Sousa, J. P. Faria, and J. Mendes-Moreira, "An analysis of the state of the art of machine learning for risk assessment in software projects," in *Proceedings of the 33rd International Conference on Software Engineering and Knowledge Engineering, SEKE*, 2021, pp. 1–10.

[12] K. Petersen, S. Vakkalanka, and L. Kuzniarz, "Guidelines for conducting systematic mapping studies in software engineering: An update," *Information and Software Technology*, vol. 64, pp. 1–18, 2015.

[13] M. Vieira, J. C. R. Hauck, and S. Matalonga, "How explicit risk management is being integrated into agile methods: Results from a systematic literature mapping," in *19th Brazilian Symposium on Software Quality*, ser. SBQS'20, 2020.

[14] S. Y. Chadli and A. Idri, "Identifying and mitigating risks of software project management in global software development," in *Proceedings of the 27th International Workshop on Software Measurement and 12th International Conference on Software Process and Product Measurement*, 2017, p. 12–22.

[15] Z. Podari, A. F. Arbain, N. Ibrahim, D. N. Abang Jawawi, W. M. Nasir Wan Kadir, and A. M. Fahmi, "Systematic literature review on global software development risks in agile methodology," in *2020 8th International Conference on Information Technology and Multimedia (ICIMU)*, 2020, pp. 231–236.

[16] K. Petersen, R. Feldt, S. Mujtaba, and M. Mattsson, "Systematic mapping studies in software engineering," in *Proceedings of the 12th International Conference on Evaluation and Assessment in Software Engineering*, ser. EASE'08. Swindon, GBR: BCS Learning amp; Development Ltd., 2008, p. 68–77.

[17] C. Wohlin, "Guidelines for snowballing in systematic literature studies and a replication in software engineering," in *Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering*, ser. EASE '14, 2014.

[18] S. M. Al-Saleem and H. Ullah, "A comparative analysis and evaluation of different agile software development methodologies," in *International Journal of Computer Science and Network Security*, vol. 15, no. 7, 2015, pp. 39–45.

[19] M. Turner, "Digital libraries and search engines for software engineering research: An overview," *Keele University, UK*, 2010.

[20] P. Crowe, A. Mostashari, M. Mansouri, and R. Cloutier, "9.2.1 reference framework and model for integration of risk management in agile systems engineering lifecycle of the defense acquisition management framework," *INCOSE International Symposium*, vol. 19, no. 1, pp. 1391–1405, 2009.

[21] 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, 2020.

[22] J. M. P. Carvallo, H. Oktaba, and E. R. Hernández, "Risk assessment forum," in *2018 6th International Conference in Software Engineering Research and Innovation (CONISOFT)*, 2018, pp. 160–164.

[23] E. Odzaly, D. Greer, and D. Stewart, "Agile risk management using software agents," *Journal of Ambient Intelligence and Humanized Computing*, vol. 9, p. 823–841, 2018.

[24] S. Ghazali, S. Salirti, I. Inayat, and S. h. Ab hamid, "A risk poker based testing model for scrum," *Computer Systems Science and Engineering*, vol. 33, pp. 169–185, 05 2018.

[25] V. Dorca, R. Munteanu, S. Popescu, A. Chioreanu, and C. Peleskei, "Agile approach with kanban in information security risk management," in *2016 IEEE International Conference on Automation, Quality and Testing, Robotics (AQTR)*, 2016, pp. 1–6.

[26] M. Hammad and I. Inayat, "Integrating risk management in scrum framework," in *2018 International Conference on Frontiers of Information Technology (FIT)*, 2018, pp. 158–163.

[27] R. Agrawal, D. Singh, and A. Sharma, "Prioritizing and optimizing risk factors in agile software development," in *2016 Ninth International Conference on Contemporary Computing (IC3)*, 2016, pp. 1–7.

[28] X. Dong, Q.-S. Yang, Q. Wang, J. Zhai, and G. Ruhe, "Value-risk tradeoff analysis for iteration planning in extreme programming," in *2011 18th Asia-Pacific Software Engineering Conference*, 2011, pp. 397–404.

[29] L. Ribeiro, C. Gusmao, W. Feijo, and V. Bezerra, "A case study for the implementation of an agile risk management process in multiple projects environments," in *PICMET '09 - 2009 Portland International Conference on Management of Engineering Technology*, 2009, pp. 1396–1404.

[30] L. G. Cuong, P. D. Hung, N. L. Bach, and T. D. Tung, "Risk management for agile projects in offshore vietnam," in *Proceedings of the Tenth International Symposium on Information and Communication Technology*, ser. SoICT 2019, 2019, p. 377–384.

[31] B. Freimut, S. Hartkopf, P. Kaiser, J. Kontio, and W. Kobitzsch, "An industrial case study of implementing software risk management," in *Proceedings of the 8th European Software Engineering Conference Held Jointly with 9th ACM SIGSOFT International Symposium on Foundations of Software Engineering*, 2001, p. 277–287.

[32] O. Mizuno, T. Kikuno, Y. Takagi, and K. Sakamoto, "Characterization of risky projects based on project managers' evaluation," in *Proceedings of the 22nd International Conference on Software Engineering*, ser. ICSE '00, 2000, p. 387–395.

[33] M. Choetkiertikul, H. K. Dam, T. Tran, and A. Ghose, "Characterization and prediction of issue-related risks in software projects," in *2015 IEEE/ACM 12th Working Conference on Mining Software Repositories*, 2015, pp. 280–291.

[34] J. Nyfjord and M. Kajko-Mattsson, "Outlining a model integrating risk management and agile software development," in *2008 34th Euromicro Conference Software Engineering and Advanced Applications*, 2008, pp. 476–483.

[35] E. Odzaly, D. Greer, and D. Stewart, "Lightweight risk management in agile projects," in *Proceedings of the 26th International Conference on Software Engineering and Knowledge Engineering, SEKE*, 2014, pp. 576–581.

[36] S. V. Shrivastava and U. Rathod, "A risk management framework for distributed agile projects," *Information and Software Technology*, vol. 85, pp. 1–15, 2017.

[37] E. Alharbi and M. R. Qureshi, "Implementation of risk management with scrum to achieve cmmi requirements," *International Journal of Computer Network and Information Security (IJCNIS)*, vol. 6, pp. 20–25, 09 2014.

[38] M. Mousaei and T. J. Gandomani, "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, 01 2018.

[39] S. Ghobadi and L. Mathiassen, "Risks to effective knowledge sharing in agile software teams: A model for assessing and mitigating risks," *Information Systems Journal*, vol. 27, no. 6, pp. 699–731, 2017.

[40] M. Carr, S. Konda, I. Monarch, C. Walker, and F. Ulrich, "Taxonomy-based risk identification," Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, Tech. Rep., 1993.

[41] S. Sundararajan, M. Bhasi, and P. Vijayaraghavan, "Variation of risk profile across software life cycle in is outsourcing," *Software Quality Journal*, vol. 27, p. 1563–1582, 12 2019.

[42] H. A. Abdulbaqi, A. S. A. Jabar, and Z. S. A. Jabar, "Integrated software project risks method based on pdf-ann techniques," *International Journal of Civil Engineering and Technology (IJCIET)*, p. 1094–1102, 2018.

[43] X. Zhou, Y. Jin, H. Zhang, S. Li, and X. Huang, "A map of threats to validity of systematic literature reviews in software engineering," in *2016 23rd Asia-Pacific Software Engineering Conference (APSEC)*, 2016, pp. 153–160.