How Explicit Risk Management is Being Integrated Into Agile Methods: Results From a Systematic Literature Mapping

Marcel Vieira
Department of Informatics and Statistics, Federal University of Santa Catarina (UFSC), Florianópolis, SC, Brazil
marcel.fv70@gmail.com

Jean C. R. Hauck
Department of Informatics and Statistics, Federal University of Santa Catarina (UFSC), Florianópolis, SC, Brazil
jean.hauck@ufsc.br

Santiago Matalonga
University of the West of Scotland, Glasgow, United Kingdom
santiago.matalonga@uws.ac.uk

ABSTRACT
Agile software development methods exploit software development characteristics to minimize risks. Although Agile practices tend to mitigate risks, they are often not sufficient to manage project risks in certain scenarios such as in highly regulated domains. In order to understand how explicit risk management is being integrated into Agile software development methods, a Systematic Literature Mapping (SLM) was conducted, which identified 18 studies. Results indicate that the most used Agile method for integrating explicit risk management practices is Scrum. Ten studies present evidence for all PMBOK risk management processes, with the risk identification being the only process covered by all studies. In general, the selected studies indicate that the results of integrating explicit risk management with Agile methods are positive, encompassing improved communication, improved product quality, increased risks visibility, cost reduction, improved team efficiency and time-to-market reduction. Based on the results, we claim that Agile methods are enablers of risk management practices. However, a comprehensive guide that allows for the incorporation of risk management practices into any Agile method still seems to be needed. The results presented in this paper can be used as a steppingstone achieving this.

KEYWORDS
Risk management, Agile methods, Literature mapping

1 INTRODUCTION
Agile methods in software engineering were born as a response to the traditional software development methods, known for being rigorous and plan-driven methodologies [1][2][3][4], proposing a way to develop software that allows delivering quickly and change often [5]. The Agile Manifesto [6] formalized these aspirations by defining a set of values and principles that characterises these Agile methods [7]. Since then, several new Agile methods have been proposed, both for managerial and technical aspects of software development, varying in practices and techniques but sharing the common values described in the manifesto [4]. Many benefits have been observed with the use of Agile methods in the most varied software development scenarios, such as: costs reductions, improved teamwork, enhanced trust [8], increased productivity [9], improved project management [10], work visibility, better communication [11], among others. Agile methods claim to be risk-driven and typically address risk management implicitly [12][13].

Software development projects are intrinsically subject to several risks, such as time-to-market, budget and schedule estimation, technology evolution, and stakeholders’ expectations [14]. Risk management during the software development process has been carried out to manage risk impact, especially in some highly regulated software development domains, such as healthcare [15] and automotive [16], for example. Software development risk management is often conducted in a traditional way, following practices provided by standards, guides and specific knowledge bodies for each software engineering domain, such as ISO 14971 [17], PMBOK [18], ISO 31000[19], IEC 80001 [20] and IEC 62304 [21].

Agile methods use an iterative and incremental approach to optimize predictability and control risk [22] which consists of an inherent risk-driven approach that implicitly supports risk management [23]. Common Agile practices such as small increments, work visibility and expectations management tend to be mitigation strategies for risks [12]. However, in many respects, Agile development does not differ significantly from other traditional models [23][24] and in certain scenarios the implicit and inherent risk management of Agile methods may not be sufficient [25][26][27]. Seeking solutions to this apparent dichotomy, many initiatives proposing the integration of explicit risk management practices into the Agile software development context have been proposed [28][29][30][31].

These scenarios become the motivation to better understand “How risk management is being explicitly integrated into Agile software development methods?”. To answer this question, this paper presents a Systematic Literature Mapping (SLM) of primary studies that report experiences and models related to risk management in the Agile software development context.

Thus, the contribution of this paper is twofold: first, to present a comprehensive and structured view of the state of the art on risk management in Agile software development context; second, SLM results can be used by practitioners regarding the possibilities of
integrating risk management with their existing Agile methods, as well as by researchers as a basis for future research on more comprehensive approaches.

The remainder of this paper is organized as follows: section 2 presents the theoretical background on software development risk management; section 3 presents related work; section 4 presents the methodological approach; section 5 presents search execution and section 6 presents data analysis and discussion, followed by section 7 where the conclusions are summarized.

2 BACKGROUND

Risks are events or uncertain conditions that, if occur, affect at least one project objective [18]. The software development process intrinsically involves risks, that can be related to time-to-market, budget and schedule estimation, technology evolution, and stakeholders’ expectations [14], among other possible sources of risks that can be identified [52]. Risk management is rooted in project planning and controlling activities, which are the major sources of risks in software development [33].

Although software project risks cannot be completely eliminated, project managers can reduce the impact of these risks through adequate risk management [34]. This has led research in Software Engineering to propose solutions for the management of risks in software projects since its beginnings [35], increasing the role of risk management within software development processes over time [36].

Software project risk management is a software engineering practice that proposes, at least, the identification of risks, their estimation, mitigation of impacts and monitoring, which tends to lead to a disciplined scenario for efficient decision-making to manage problems in software development [34]. Several standards and guides have proposed risk management practices applicable to software development, both generic (ISO 14971 [17], ISO 31000 [19], IEC 80001 [20]) and for specific software development domains, such as for healthcare (MedITNet [15], IEC 62304 [21]) or automotive (Automotive SPICE [16]). One of the most widely accepted approaches to risk management in software project management is proposed in PMBOK [18], where risk management processes are defined as:

- Planning: definition of how to conduct risk management activities, such as schedule, budget, and resources
- Identification: discovery of which risks may affect the project and document its relevant characteristics
- Qualitative and Quantitative analysis: assessment of the risk exposure to prioritize risks defining its probability and impact
- Plan risk responses: definition of the strategies and options for dealing with risks through mitigation plans
- Implement risks responses: planned responses to risks are implemented, ensuring its effective and efficient conduction
- Monitoring risks: monitor and control risks during the project’s life cycle

Agile methods typically do not include explicit risk management practices [30]. Risk management in Agile methods is performed implicitly [23], given the nature of its iterative and incremental practices that tend to mitigate risks [12][22].

However, software projects that use Agile methods also fail [37] for several reasons such as team capability, customer involvement [38], organization inappropriate size, lack of project management competence [39] and also the absence of risk management [40] where the implicit and inherent risk management has not been sufficient [25][26][27]. As, there is a recognized relationship between project success and risk management, adding explicit risk management techniques to Agile methods can be enriching and useful [41].

3 RELATED WORK

According to Kitchenham [42] it is a good practice, as one of the first steps in a secondary study, to carry out initial searches for possible existing secondary studies. This section presents the identified secondary studies dealing with risks in agile methods.

The literature is rich in secondary studies that address risk management aspects related to software development, such as [25][43][44][45][46]. However, these cover a wide range of software development methodologies, not specifically focusing on Agile methods. Thus, in our initial research, we found three directly related secondary studies.

Tavares, da Silva & de Souza [34] analyzed 34 studies on Agile methods and identified 127 risk management practices. The study categorizes and ranks practices using a multi-criteria method with the participation of experts. The study identifies risk management practices for daily meetings, increment, prototype, product backlog and Sprint planning as the most important for the risk management effectiveness. However, the study focuses only on the identification of practices and classifies them according to Scrum, without using other Agile methods, and also does not include other aspects of the context of use or the results of its adoption. In addition, the search protocol includes other sources such as books and general guides without an explicit quality criterion.

Rafeek, Arbain & Sudarmilah [47] present a Systematic Literature Review (SLR) searching for risk mitigation techniques in Agile methods in the context of Global Software Development (GSD). The secondary study found 40 papers citing risk management but only 5 dealt specifically with Agile GSD and risk mitigation. The analysis of the results of the article is superficial, only covering a direct count of the primary studies found, and only focuses on the context of GSD.

Similarly, Hossain, Babar & Paik [48] present a SLR to identify key challenges in Agile software project risk management. Authors included 20 articles on risk management with Scrum in the context of Global Software Development (GSD), selecting and compiling the best practices for future proposing a model [49]. However, the study only applies to risk management in Scrum, not including other agile methodologies like Kanban and XP and also only focusing on the GSD context.

It was not possible to find in these related works a comprehensive secondary study that focuses on how risk management is being integrated into Agile software development methods in general. Also, the small set of related works found motivates the realization of a SLM on this subject.
4 METHODS
To understand how to integrate risk management phases, processes, and practices into agile development processes, we undertook a Systematic Literature Mapping study (SLM). To accomplish this, we followed the procedures proposed by Petersen, Vakkalanka & Kuzniarz [50], Petersen et al. [51] and Wohlin [52]. A SLM is a literature review method used to structure a field of interest, providing an overview of a research area, and identifying the quantity and type of research and results available within it [51]. In SLMs, articles with no empirical evidence, which would not be included in a Systematic Literature Review (SLR) [53], are included to spot trends of topics being worked on [50]. Therefore, we chose to adopt SLM as we suspected that it would not be possible to carry out a deeper meta-analysis, since the initial investigations pointed to a possible low empirical quality of most studies (see section 6.1).

Following the SLM method, a series of activities were organized into four main phases: (i) protocol definition, (ii) execution of searches, (iii) study selection and snowballing and (iv) data extraction, classification, and analysis.

Research Questions
From the identified research need, the general research question was defined as: “How risk management is being explicitly integrated into Agile software development methods?” Thus, to answer the research question and characterize how risk management has been carried out in Agile software development methods, we derived the following analysis questions:

- **Q1.** What are the Agile methods in which explicit risk management has been integrated?
- **Q2.** What risk management practices are implemented?
- **Q3.** What are the main results of risk management and Agile integration?
- **Q4.** What contextual information is included in the source?
- **Q5.** What are the roles of team members in risk management?

We note that the Agile method/approach (Q1) can be understood within the overarching word context (Q4), however, we decided to extract the agile method/approach from the rest of the contextual information as we had preconceived notions regarding the Agile/methods we were expecting to identify.

4.1 Search Strategy
To carry out the SLM, we ran two complementary strategies: an automated search using scientific databases and a manual search using the Snowballing [52] technique.

Automatic searches were carried out in digital libraries: IEEEXPlore (http://ieeexplore.ieee.org), ACM Digital Library (http://dl.acm.org), Science Direct (http://www.sciencedirect.com), Springer (http://www.springer.com), Wiley Online Library (https://onlinelibrary.wiley.com/) and Scopus (https://www.scopus.com/). IEEEXPlore and ACM Digital Library covered the most important journals and conferences in the field of software engineering [54] and the others, as large indexing services, completed expanding the scope of the research.

The Snowballing technique was carried out after the automated search strategy. We used as input the selected papers from the automated search.

The following inclusion and exclusion criteria were used to determine the pertinence of a paper to our research. Papers were filtered in three filtering stages: first by reading title, then by title and abstract and finally by a full-text reading. The inclusion and exclusion criteria presented in this section applied to the three filtering stages.

**Inclusion criteria**
Studies that explicitly mention the application of an Agile method for software development; Studies presenting approaches directly or indirectly related to risk management in Agile software development; Studies from 2001 to 2020; Studies published in conferences or scientific journals.

**Exclusion criteria**
Studies non-peer reviewed; Studies not accessible in full text; Duplicated studies; Studies not published in English

From the main research question and the analysis questions, the search terms are derived, including synonyms found in the literature. From the defined Search Terms, we defined an initial search string. This string was piloted in some search attempts and was then defined as a generic search string. The search string is defined as broadly as possible, trying to ensure the widest search possible. The search terms are linked by ‘AND’ clauses, and each of the search terms is grouped with its synonyms using ‘OR’ clauses. The generic search string was then adapted to each of the digital libraries’ specific syntax. The final search string was executed on the digital libraries on April/2020.

\[ (<"Risk management"> OR <"Risk administration"> OR <"Risk analysis"> OR <"Risk mitigation"> OR <"Risk assessment">) AND (<"Agile"> OR <"Scrum"> OR <"XP"> OR <"Extreme Programming"> OR <"Lean"> OR <"Kanban"> OR <"Scrumban"> OR <"TDD">) AND (<"Software"> OR <"Software development"> OR <"Software project management">) \]

5 SEARCH EXECUTION
After applying the search strings in the selected databases, 5557 studies in total were returned (Figure 1). Thus, a first iteration for filtering was carried out, in which the studies were selected from the reading of their titles. Studies considered irrelevant for the research were directly discarded. In a second iteration, the title and abstract were carefully analyzed, based on the inclusion and exclusion criteria. Finally, all studies that passed through the previous iterations were filtered again, from the full reading of the texts.

**Snowballing Execution**
As seen in Figure 1, 16 studies were used as input for the snowballing process. After the third iteration the backward snowballing technique [52] was manually applied to select studies from selected articles references. The sources identified by the backward snowballing procedure were firstly selected by title and after filtered by abstract. Two articles considered relevant were selected.

**Study selection summary**
As can be seen in Figure 1, of the 5557 studies, after conducting the initial search, 86 (1.5%) articles were selected from reading its titles. The number was due, in large part, to many articles that did not deal specifically with software development (IC3) as we chose to make the search string as wider as possible, to minimize
the risk of not including relevant studies. After the first iteration, the second stage considered the inclusion and exclusion criteria to abstract, in which the number of articles was reduced from 86 to 49 (57%) selected. Most papers excluded in this iteration presented risks management or Agile methods separately, not integrating the two.

Data Extraction

Data extraction was then completed for the 18 sources in the sample (Table 1). Extracted raw data is available at: http://dx.doi.org/10.17632/mb4z3s465f.1.

5.1 Data analysis and discussion

The data extracted from the selected studies are presented in Table 2, classified according to the analysis questions. To classify the extracted data, we applied the narrative synthesis technique [55]. Thus, Table 2 indicates which studies contain information that classifies them in a given category of each analysis question. The categories are not mutually exclusive.

The selected studies are distributed from 2003 to 2018, with a concentration in 2008-2009 (5 studies) and 2016 (4 studies), indicating two phases of greatest interest in the subject. Most studies are from conferences (10 studies), but the frequency of publications in journals has increased since 2016 (4 studies), indicating a possible maturation of this research topic.

Q1. What are the Agile methods in which explicit risk management has been integrated?

Regarding the Agile method/approach that is the context for adding explicit risk management, of the selected studies, about 44% (8) use Scrum. Only one study (~6%) uses Kanban with risk management. The study S6 that used Kanban, chose this methodology as authors considered the Kanban board an important way to carry out risk management, especially because it was considered secure for the case study that was carried out in the government context. Also, only one selected study (~6%) uses XP as the exclusive context for risk management. It is also important to note that about 39% (7) of the studies used other Agile methods in general or did not explicitly identify one.

Q2. What risk management practices are implemented?

As there is no consensus in the nomenclatures used for the steps/phases/activities/practices proposed in the studies for risk management, we choose to use the processes described in PMBOK [18], which are widely known and accepted in the software industry. The risk management processes in PMBOK include risk management planning, risk identification, risk analysis (qualitative and/or quantitative), risk response planning and risk monitoring and control (M&C). Each selected study practices were grouped according to these processes and tabulated with their respective characteristics. To assess whether the practices reported in the studies cover the PMBOK processes, we assessed whether they meet at least the general objectives of each process, without considering a strict analysis of all the details of expected inputs and outputs.

Most studies (55% - 10) meet the general objectives of all the five processes (Figure 2). However, the way the studies solve each process varies in terms of the level of detail and granularity. In Figure 2 the “y” axis shows the selected studies and the “x” axis shows the total coverage level of the five PMBOK processes.

The only process explicitly covered by all studies is the Risk Identification process (Figure 3), in which brainstorming sessions or meetings (specific or using another ceremony already existing in the chosen Agile method) were used to identify project risks (S1, S3, S11, S15). Another technique used are checklists, in which risks are assessed using ready-made models based on previous experiences (S7, S15). In S6, a person with experience in risk management is assigned to identify threats early in the project.

Approximately 94% (17) of the selected studies carry out explicit Qualitative Risk Analysis, using risk scoring and classification techniques according to severity and probability of occurrence, such as (S4, S6, S8, S9, S10, S11, S12, S13, S14, S15,S16,S17, S18). Studies S1 and S15, on the other hand, explicitly analyses together with the client/sponsor which risks are worth managing depending on their
<table>
<thead>
<tr>
<th>#</th>
<th>Reference</th>
</tr>
</thead>
</table>

The Risk Planning process is explicitly carried out by about 89% of the studies. The most used technique among these studies is the development of a mitigation plan with the respective actions for risk occurrences. A highlight for articles S4, S5, S9, S11, S15, S16, and S18 that, based on assessment techniques carry out the risk mitigation plan, initiating with risks with higher exposure factor and later addressing risks with lower probability or impact.
Table 2: Data extraction and classification

<table>
<thead>
<tr>
<th>Classification</th>
<th>Studies</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q1. Agile method used in the study</td>
<td>Scrum: S1, S2, S7, S8, S9, S11, S17, S18, Kanban: S6, XP: S14, Other: S3, S4, S5, S10, S12, S13, S15, S16, S17, S18</td>
</tr>
<tr>
<td>Q2. Risk management processes</td>
<td>Identification: S1, S2, S3, S4, S5, S6, S7, S8, S9, S10, S11, S12, S13, S14, S15, S16, S17, S18, Analysis: S1, S2, S3, S4, S5, S6, S8, S9, S10, S11, S12, S13, S14, S15, S16, S17, S18, Planning: S1, S3, S4, S5, S6, S8, S9, S10, S11, S12, S13, S14, S15, S16, S17, S18, Response: S1, S2, S3, S4, S6, S7, S8, S10, S11, S13, S15, S18, Monitoring: S1, S2, S3, S4, S5, S6, S8, S10, S11, S12, S14, S15, S16, S17, S18</td>
</tr>
<tr>
<td>Q3. Result of integrating risk management and agile methods</td>
<td>Practices: S1, S2, S3, S4, S5, S6, S7, S8, S9, S10, S11, S12, S13, S14, S15, S16, S17, S18, Model: S1, S2, S3, S4, S5, S6, S8, S10, S11, S12, S13, S15, S18, Tool: S12, S14</td>
</tr>
<tr>
<td>Q4. Contextual information included</td>
<td>Industrial: S4, S6, S8, S10, S13, S14, S17, S18, Educational: S1, S10, S11, S12, S14, No application: S2, S3, S5, S7, S9, S15, S16</td>
</tr>
<tr>
<td>Q5. Risk management roles</td>
<td>External: S3, S4, S6, S8, S11, S18, Development team: S1, S2, S3, S4, S6, S7, S8, S9, S10, S11, S12, S13, S14, S15, S16, S17, S18</td>
</tr>
</tbody>
</table>

Figure 3: Number of studies per process

Figure 4: Practical results

For the Risk Response Planning process, approximately 72% (13) provide explicit strategies for possible risks materialization. The most used response technique among the studies is the application of the mitigation plan when a risk occurs. However, only studies S1, S6, S10, S13 and S15 explicitly allocate the risk to a team member to mitigate it. Studies S11, S16, S17 and S18, on the other hand, use the strategy of adding risk mitigation tasks to the development tasks Backlog, that are assigned to team members.

Q3. What are the main results of risk management and Agile integration?

We classify the results of applying risk management in Agile methods in two categories: practical results and academic results. The practical results refer to those studies that, after being applied in practice in industrial or academic contexts, demonstrated observed results. The academic results refer to the sets of practices, models and/or tools that were produced as a result of the study.

For the analysis of practical results, only primary studies that reported some type of practical application are considered (see details in Q4). In general, the practical results of adopting risk management in the context of Agile methods are quite positive (Figure 4). Most of the primary studies applied in practice, naturally report an improvement in the efficiency of risk management, such as in S4, S6, S8, S10, S13 and S18. In addition, primary studies also report: improved communication S4, S10, S13 and product quality S13, S17, increased visibility of risks S10, S13; cost reduction S13, S14; improved team efficiency S6; solution of impediments S4; and also a reduction in time-to-market S13.

In addition to the positive results presented by the studies, some concerns regarding the integration of explicit risk management in Agile methods are also discussed. In S11, authors warn that explicitly engaging in risk management on Agile projects involves extra time and structure, which can reduce the project’s agility, indicating that the use of a small team software risk assessment is more appropriated. The same study S11 also draws attention to the issue of mitigation triggers that can hinder the closed scope of a Sprint, by instantiating new tasks during an iteration, for example. Study S8 raises the question that risk management unified with an Agile method may not be applicable to all types of projects. Despite
this, S13 argues that its developed model is applicable to any type of Agile method.

We classify academic results produced by the studies into three categories: sets of practices, models and/or tools.

Sets of practices: As a result of the integration of explicit risk management practices into Agile methods, all the selected studies have developed at least a set of practices to manage risks integrated with the adopted Agile method. Among the various presented practices, we highlight S1 that develops a matrix to decide the value and the exposure to project risks, defining probability and impact, as well as the study S4 that besides the risk matrix, assesses the cause and effect relationships of risks. S6 developed a customized Kanban board to specify the stages of risk management, with the appropriate tasks and their progress. In S11 through a case study, raises the importance of defining the roles of team members for risk assessment and mitigation, in addition to proper documentation to track risks. It is worth also highlighting S16 which, to support the proposed practices, conducted interviews with project managers in the software development area to identify the main gaps in risk management in Agile, indicating techniques to fill them, performing threat management more robustly.

Model: As a result of the integration of an Agile method and risk management, about 72% (13) of the selected studies have grouped proposed practices in some kind of model. S4 proposes a model named GARA (Agile Management of Environmental Risks) in which the project must be centered on people and communication, with a life cycle based on the APAM (Agile Project Management) method. S1 and S2 propose models in which, in addition to the regular Scrum Sprints, there are also specialized Sprints to manage risks. S3 presents a model where risks are managed in different levels of detail depending on the project risk-level. S13 presents an empirically grounded model for risk assessment and mitigation using Agile development principles. Finally, S17 proposes an additional model to Scrum with a series of risk management techniques to make Scrum more compatible with CMMI level 3.

Tool: Only 2 studies (11%) presented a software tool developed to support risk management. S12 presents a tool in which intelligent agents identify, assess and monitor risks. Ideally, these agents should be able to react autonomously to changes in the environment and to the risks that occur in the project. Another tool was proposed in S14, which manages risks in the XP method. The proposed tool provides an easy to use interface, reports, online tutorials and scenario analysis, in which it is possible to identify which tasks have the greatest impact on the project, adding integration with other tools and an organizational database to store risk information, such as risk name and priority.

Q4. What contextual information is included in the source?

Among the selected studies, 7 (39%) do not report practical application S2, S3, S5, S7, S9, S15, S16. Although these studies do not add empirical evidence to this secondary study, we decided to include them in the other analyses, since they are important to spot trends of topics being worked on [50]. The other 11 studies (61%) S1, S4, S6, S8, S10, S11, S12, S13, S14, S16, S18 were applied in two different contexts: industrial or educational, and some studies were applied in both contexts.

Industrial context: 8 (44%) of the selected articles carried out a case study in software development organizations to evaluate the proposed models, tools or set of practices. The GARA model proposed in S4 was evaluated in a case study involving three simultaneous evolutionary projects of three different software products from the same company, with 4 participants involved in the application, one of whom assumed the role of Risk Team Master. The model was successfully executed in a multiple-projects environment. Study S6 was applied together with the Agile Kanban method in a project to develop an e-commerce system with criticality risks being tracked monthly. No other information about the context of the case study is presented. Study S8 was evaluated by 10 practitioners from different companies through semi-structured personal interviews, where the model was presented to the evaluators. The model was also applied in a case study in a Swedish company with a development team of 8 employees, which develops web systems and uses Scrum complemented with XP practices. In some studies it was not possible to clearly identify whether there was really a practical application, such as for example study S16 where interviews were conducted with project managers to assess whether they approve the model but it was not possible to infer whether these project managers used the model to be able to carry out the evaluation.

Educational context: 5 (28%) of the selected studies carried out a case study in universities with students to evaluate the proposed models, tools or set of practices. Study S1 separated students in two groups, one group controlling risks using their proposed model and the other group controlling risks using Scrum only. The experiment showed a significantly higher amount of critical risks identified in the group that used the model. Study S11 evaluated the proposed model in an educational context and observed several qualities and improvement opportunities. The study reported that mitigation strategies need to be formulated for the risks before starting to mitigate, that is, for critical risks it is important to spend time and effort for identification, analysis, and planning. Also, the evaluation of the tool proposed by study S12 demonstrated the importance of using existing data from previous projects to identify common risks and which mitigation activities were used to support risk management.

Q5. What are the roles of team members in risk management?

Regarding how risk management roles are allocated in the context of Agile methods, we classify the studies into two groups: those that define specific risks-related roles and allocate exclusive or external human resources for risk management tasks and the other group that allocate development team members for risk management tasks. Some studies (S3, S4, S6, S8, S11, S18) used both strategies.

External or exclusive members: 6 of the selected studies defined exclusive roles to manage risks. Studies S4, S11 and S18 propose a Risk Team Master or Risk Manager role who is responsible for ensuring that the risk-related processes and its activities are being carried out by those involved. Similar to Scrum Master, the defined role must seek solutions to risks and report to project managers or equivalent roles. Study S6 defines five roles for managing risks: Risk Identifier, Risk Team (formed by specialists in Risk Management),
Risk Owner (the most impacted by the risk), Risk Treatment Responsible (responsible for implementing the treatment plan according to the risk owner’s decision), Security PME (part of the information security team that provides knowledge and support in the context of risks). Study S8 defines a group of exclusive team members to manage risks, named RMF Members, who are responsible for all the organization’s risks. Their main tasks are to supervise and coordinate risk in the entire organization and make decisions about them. However, they can delegate their management responsibilities to other roles, business levels or to the development team.

Development team: 17 studies define that the development team itself must assess the risks and perform mitigation tasks. We highlight studies S1, S2, S8, S12, S16 and S17 that reuse or adapt the Scrum method with its respective roles of team members to manage risks, without adding other specific roles or members.

6 DISCUSSION

Results of this secondary study synthesize the evidence on integrating explicit risk management practices with Agile software development methods. Confirming the global trend of using Scrum as the main Agile approach in software companies [56], most primary studies (44%) added explicit risk management practices to Scrum. As risk management is often overlooked in many Scrum projects [31][57][58], several studies (S1, S2, S8, S11, S16 and S18) have chosen for adding explicit risk management practices to Scrum’s native practices and ceremonies, which seems to be a good solution, reducing the need for new meetings or interruptions in the development process.

Only one study used Kanban with risk management (S6 - 5.5%) which is consistent with the trend of the software industry since 7% of companies use Kanban as the main Agile method [56]. The study justified the choice of Kanban especially because of the Kanban board that was used to carry out risk management. For companies or organizations that still depend on documentation, visualization of processes and more traditional aspects, Kanban is an interesting option to control project activities [59]. Also, only one selected study reported the use of XP S14 as the main Agile method, which can be justified by the fact that the study was applied in 2008 when XP was the prominent Agile approach [60]. However, currently, only 1% of the software companies use XP [56] as the main Agile method.

The fact that the majority of the studies (55% - 10) meet the general objectives of all the five risk management processes seems surprisingly good, indicating that there are several good options for risk management proposals integrated with Agile methods. However, the level of detail and granularity of each proposal varies considerably, requiring a careful selection depending on the risk management needs of each project [61][62]. The only process covered by all studies is, as expected, the Risk Identification process, as it is not possible to manage risk without knowing what they are, where they are coming from, in terms of what detrimental effects might be experienced, and the mechanisms underlying these effects [18][63].

Overall, all authors report a positive outcome when integrating risk management within Agile methods. However, the reported results need to be analyzed carefully, as the low empirical quality of most studies can impair the validity of the conclusions. As most studies do not adequately follow the accepted guidelines for reporting case studies in software engineering [64], researchers must be wary of generalizing the conclusions of these studies. For practitioners, in particular, care must be taken when deciding to the adoption of the proposals identified in this study.

In short, we argue that risk management in Agile methods should not differ significantly from the traditional methods. Software organizations that adopt Agile or traditional methods both struggle with similar issues that impact in risks, such as change resistance, management gaps when under pressure [24], and continuous changes management [23]. Thus, as can be seen in the selected primary studies, risk management has been enhanced in Agile methods with adapted practices inherited from the traditional software development methods when the inherent risk-driven nature of the Agile methods is insufficient [27].

6.1 Threats to Validity

We identified potential threats and applied mitigation strategies to minimize their impact on the results following Zhou et al. [65].

The main risk in a secondary study is the omission of relevant studies. To mitigate this threat, we have defined a search protocol that has been validated by all authors containing a precise definition of the inclusion/exclusion and quality criteria. The search strings were also defined using the search terms as comprehensively as possible, considering not only the main concepts but also synonyms that could return relevant results. The Snowballing technique was also used to mitigate this threat, from which the references in the primary studies were analyzed as seeds for new searches.

When answering the analysis questions, the data collected from the studies were classified. The form of classification in these categories can represent a bias in the results. To minimize this threat, we used a known methodological approach to support the extraction and classification of the collected data.

Another possible construct validity stems from the analysis of the coverage of PMBOK’s risk management processes (Q2). In order not to bias the analysis by following a traditional risk management approach, the analysis of the processes’ coverage was not carried out strictly. It was not analyzed whether all activities and artefacts for each of the PMBOK processes are completely satisfied in detail, because it would fall outside the scope of an adaptation of project management to Agile methods [66]. Thus, we decided to assess whether the risk management practices presented in the selected studies meet at least the general objectives of each PMBOK process.

7 CONCLUSIONS

This paper presents a Systematic Literature Mapping (SLM) aiming to understand how risk management is being explicitly integrated into Agile software development methods. Following a rigorous research protocol, 18 studies were found that provided relevant evidence to answer the research question.

Many positive results from the integration of explicit risk management practices with Agile methods have been observed from the selected primary studies, such as improved communication, improved product quality, increased risks visibility, improved team
efficiency, cost and time-to-market reductions. In particular, we found evidence of the successful integration of explicit risk management practices in Agile projects following SCRUM, Kanban and XP methods. Furthermore, 10 out of the 18 identified studies provided some degree of evidence in all of the PMBOK’s risk management processes, being the risk identification process the only one covered by all studies. We interpret this as a clear indication that Agile methods can be enhanced to incorporate explicit risk management within their software development lifecycle.

In contrast, as there is a wide variation in the quality of the reports in the selected studies, even though most of the selected studies tend to present a positive outcome of the integration of risk management practices in Agile methods, further study needs to observe the downside or trade-off between these added practices, in particular regarding effort and resources. Though we would not argue for widespread incorporation of explicit risk management practices in all Agile projects, we would claim that this incorporation is feasible in projects that, while need Agile methodologies, their context or application domain requires that risks be explicitly managed.

REFERENCES


