A Risk Management Framework for Scrum Projects

Samuel de Souza Lopes[1][a], Rogéria Cristiane Gratão de Souza[2][b], Allan de Godoi Contessoto[2][c], André Luiz de Oliveira[3][d] and Rosana Teresinha Vaccare Braga[1][e]

[1] *Institute of Mathematical and Computer Sciences, University of São Paulo, São Carlos-SP, Brazil*

[2] *Department of Computer Science and Statistics, São Paulo State University, São José do Rio Preto-SP, Brazil*

[3] *Department of Computer Science, Federal University of Juiz de Fora, Juiz de Fora-MG, Brazil*

Keywords: Agile Software Development, Scrum, Product Owner, Project Management, Risk Management.

Abstract: Software changes constantly to suit the market volatility, causing risks to the project. Agile software development approaches, such as Scrum, have been proposed to deal with constant changes in project requirements. In Scrum, the Product Owner (PO) is responsible for managing such changes and ensuring that the developed software brings significant value to the customers. However, there are potential risks involved in these responsibilities. If not properly managed, they can lead to project failure. In this paper, we introduce a novel approach to managing risks involving PO's roles. In our work, we tailored the risk management knowledge area from the Project Management Body of Knowledge Guide into the Scrum. We established a framework called RIsk Management PRoduct Owner (RIMPRO), which intends to support project teams to systematically manage risks related to PO activities that may arise during the project. As proof of concept, the processes described in RIMPRO were evaluated by potential users. Through a preliminary assessment, we observed that RIMPRO is promising since it can assist teams in managing risks involving PO in a systematized and effective manner.

1 INTRODUCTION

The traditional waterfall process for software development has been criticized and characterized by disconnecting important life-cycle activities, such as planning, analysis, design, implementation, and testing. A waterfall life-cycle is also characterized by having unpredictable schedules and slower transitions between activities, which may increase project costs. Moreover, there is a risk that the delivered software at the end of a waterfall life-cycle does not meet the customer's expectations since emergent requirements identified after analysis could not be considered throughout the software development process. In the view of this scenario, agile process models (e.g., Scrum) and practices, such as ”release early” and ”release often”, are well established in open source software development and address the limitations of the waterfall model (Sutherland and Sutherland, 2014).

Despite the benefits of agile methods regarding evolution and self-adaptiveness, they were viewed with scepticism for their emphasis on contrary ideas of the traditional software engineering, such as scarce software documentation and prioritization of project changes (Dingsøyr et al., 2012). The uncertainty and active participation of stakeholders on projects contributed to the adoption of agile methods. However, risk management has been neglected or partially supported in agile methods like Scrum, even with risk management gaining importance among organizations, since risks may arise throughout the life cycle of the project (de Godoi Contessoto et al., 2016; Tavares et al., 2019b). In Scrum, the Product Owner (PO) has a major role in defining the requirements to address customer needs and leading the project, but this is also risky as risks regarding its roles may arise throughout the project, then they should be properly managed. Therefore, it is important to incorporate traditional risk management approaches within agile methods (Gold and Vassell, 2015; Tavares et al., 2019b).

In this paper, we introduce a novel approach to managing risks involving PO's roles. A framework called RIsk Management PRoduct Owner (RIMPRO)

[a] https://orcid.org/0000-0003-1721-311X
[b] https://orcid.org/0000-0002-7449-9022
[c] https://orcid.org/0000-0001-6545-5178
[d] https://orcid.org/0000-0003-0564-0034
[e] https://orcid.org/0000-0003-4259-0785

Lopes, S., Gratão de Souza, R., Contessoto, A., Luiz de Oliveira, A. and Braga, R.
A Risk Management Framework for Scrum Projects.
DOI: 10.5220/0010448300300040
In *Proceedings of the 23rd International Conference on Enterprise Information Systems (ICEIS 2021) - Volume 2*, pages 30-40
ISBN: 978-989-758-509-8

has been established to assist the management of such risks in the context of Scrum. As proof of concept, the processes described in RIMPRO were evaluated by potential users. We also present a classification intended to guide RIMPRO users to classify PO risks.

The remainder of this paper is organized as follows. Section 2 introduces the required background for the reader to understand the work contributions. Section 3 presents the RIMPRO and its processes. Section 4 presents the results obtained through RIMPRO assessment. Section 5 discusses the related works. Finally, Section 6 presents the conclusions and proposals for future work.

2 BACKGROUND

Scrum is an agile method for project management with an emphasis on software development projects. Scrum is not characterized as a methodology, but as a framework, since it is possible to employ various processes or techniques (Sutherland and Sutherland, 2014). Scrum emphasizes incremental software development. The set of all requirements of the software is called Product Backlog as well as the set of implemented requirements in each Sprint is called Sprint Backlog. An increment is delivered to customers at the end of each iteration (Sprint). Scrum is an adaptive and self-corrective approach for reviewing the increments implemented in each Sprint and checking possible improvements in the processes used to manage the project in the Sprint Review and Sprint Retrospective meetings, respectively. Scrum is one of the most agile methods adopted in the industry. However, it does not provide support for formal risk management that embraces the main processes, such as planning, analysis, and risk response planning (Tavares et al., 2019b).

Risk management is a set of processes for identifying and controlling areas or events that have the potential of causing unwanted changes. The Project Management Body of Knowledge (PMBoK) Guide defines risk management as a set of seven processes: *Plan Risk Management*; *Identify Risks*; *Perform Qualitative Risk Analysis*; *Perform Quantitative Risk Analysis*; *Plan Risk Responses*; *Implement Risk Responses*; and *Monitor Risks*. Due to the lack of standardization of the term ”risk”, we have used the definition proposed by PMBoK. Risk is an ”event or an uncertain condition that, if it occurs, can result in positive (opportunities) or negative impacts (threats) in one or more project objectives, such as scope, time, cost, and quality” (PMI, 2017).

In agile project management, due to the receptivity to changes, several projects have uncertainties and risks. To ensure that risks are well understood and treated, projects managed through adaptive approaches make use of frequent reviews of work products and multi-functional project teams to accelerate communication and knowledge sharing. These risks can be managed through traditional risk management processes, as long as they are adapted to the context of agile development (Andrat and Jaswal, 2015; Alliance, 2017).

Eventually, several risks remain unknown because they are ignored throughout the project life-cycle. Thus, it is needed to introduce risk management processes within agile development (Andrat and Jaswal, 2015). In this context, the Project Management Institute (PMI), together with the Agile Alliance, developed the Agile Practice Guide. This guide provides tools, situational guidance, and an understanding of the various agile approaches available to enable better results throughout the project. The Agile Practice Guide assists traditional project teams that want to apply agile development concepts to their projects. Even helping traditional teams adopt agile practices, the Agile Practices Guide does not support changes or modifications to PMBoK processes or knowledge areas, such as risk management, justifying the importance of this work (Alliance, 2017).

The PO plays an essential role throughout the project life-cycle, with the responsibility of managing requirements and ensuring that the software brings significant value to customers. Due to her/his importance in the project, the risk identification and analysis associated with PO decisions are needed. The previous knowledge of these risks may contribute to the success of the project (Sutherland and Sutherland, 2014). From the current literature analysis, we can group the potential risks involving the PO decisions into three groups described as follows.

2.1 Requirements Engineering Risks

(Dikert et al., 2016) point out that several risks may arise if the PO does not correctly perform his/her duties in the requirements engineering stage, therefore it is necessary to document and analyze such risks. The risks involving the PO are broken down into four categories:

- **Risks Related to Lack of Requirements Documentation:** The production of lean documentation is one of the main problems in agile development processes. Traditional requirements specification are replaced with stories or backlog items to document the potential interactions of stakeholders with software (Sutherland and Sutherland, 2014). The most effective way to minimize the problems related to lean documentation is through efficient communication since the main problems exposed by the literature related to lean documentation occur due to inefficient communication between the Scrum Team and customers (Inayat et al., 2015; Curcio et al., 2018; Elghariani and Kama, 2016; Fitriani et al., 2016). However, communication failures can also arise when all Scrum Team members are not accommodated in the same work environment because it is more difficult to communicate changes in requirements among Scrum Team members and customers. Thus, the use of more detailed user stories facilitates homogeneous communication between stakeholders, helping developers to make correct choices regarding the implementation of an increment during a Sprint (Inayat et al., 2015; Elghariani and Kama, 2016);

- **Risks Related to Lack of Customers to Take Decisions:** One of the pillars of the Agile Manifesto is the customer's interaction with the project team, which is disseminated in several agile approaches, such as Scrum. Since the PO is responsible for ensuring that the increments have significant value to the customer, it is up to her/him to maintain constant contact with the customer (Sutherland and Sutherland, 2014). Customer availability is important, as changes in requirements being determined directly by the customer can speed up the project, as well as to prioritize requirements by customers creates higher value increments. However, the client's unavailability in taking project decisions is a problem since the continuous contact with the Scrum Team generates costs for the organization that requested the software (Inayat et al., 2015; Curcio et al., 2018; Elghariani and Kama, 2016). A way to solve this problem in practice is assigning customer role to one member of the Scrum team. It may contribute to reducing the need for constant interaction with real customers. However, such an approach is only effective when the Scrum team members have substantial knowledge of the targeted application domain (Inayat et al., 2015). Another issue related to the client is the lack of knowledge to define the requirements. This issue also occurs in projects as there is more than one client, making it difficult to reach a consensus among the group when there are disagreements regarding requirements. That is more evident in projects with shorter Sprints. If this problem occurs, the project's performance is directly affected (Inayat et al., 2015; Curcio et al., 2018);

- **Risks Related to Changes in the Requirements:** The lack of clarity in the definition of the requirements can contribute to project failures, as it impairs the product overview (Alliance, 2017; Sutherland and Sutherland, 2014). To developing increments with significant value to customers, requirements must be documented, prioritized, and, if needed, modified (Inayat et al., 2015; Elghariani and Kama, 2016; Fitriani et al., 2016). The requirements definition process may be complicated, as stakeholders often have different expectations of a requirement. Also, the requirements suggested by some stakeholders may be considered unworkable by the Scrum Team. Another obstacle is when the client considers that most of the software requirements are mandatory and equally important, so it is difficult to use criteria for prioritizing these requirements (Elghariani and Kama, 2016; Alliance, 2017);

- **Risks Related to Budget, Architecture, and Schedule:** Analogously to traditional software development approaches, the design of the software architecture, the estimation of project budget, and the management of the schedule are critical activities for organizations adopting Scrum in their projects. That occurs because, in Scrum, requirements are considered volatile since they may change throughout the project (Inayat et al., 2015; Elghariani and Kama, 2016; Curcio et al., 2018). To reduce the time and effort needed to turn requirements into increments, the requirement can be decomposed into smaller sub-requirements. Requirements decomposition may contribute to improving the accuracy of schedule estimation and minimizing the occurrence of unforeseen events during the Sprint (Alliance, 2017).

2.2 Software Quality Risks

(Asghar et al., 2017) emphasize that the lack of quality and clarity of requirements can contribute to project failures. On the other hand, (Dikert et al., 2016) point out that the quality control functions assigned to Scrum Team members may be adversely affected when ambiguous requirements arise. To avoid ambiguities, the PO can receive specialized training, so that the Scrum Team can establish a structured way to document the requirements.

2.3 Risks Related to Migration of Traditional Teams to Agile

These risks are related to the inherent insertion of agile approaches characteristics in the context of teams that use traditional techniques of software engineering (Gandomani and Nafchi, 2016). The *Agile Practice Guide* provides guidance to support traditional teams to adapt their practices to the agile context (Alliance, 2017). Such adaptation requires sufficient time and effort, but risks can often arise during this change (Gandomani and Nafchi, 2016). So, a risk may or not be related to migration. Even after the migration, risks related to Requirements Engineering (RE) and Software Quality (SQ) can occur, as they are activities inherent to the PO. Thus, the risk can receive more than one classification if it is related to one of the functions of the PO (RE and SQ) and, also, the project team is migrating to Scrum.

In this scenario, we can conclude that the PO has great importance since (s)he is a member of the Scrum Team responsible for crucial stages of the project. Consequently, risks can arise involving her/his decisions throughout the project. The risks related to agile methods, such as Scrum, are scarce, unlike the traditional software development approaches in which the risks are well-known (Andrat and Jaswal, 2015). This leads us to believe that the use of an adapted risk management framework to the Scrum context, focusing on the risks involving the PO, is important to support the documentation of risk analysis and mitigation measures in agile projects. Lessons learned from projects can provide support for decision making by software organizations that use Scrum to manage their projects. For this purpose, such framework should support the classic processes of risk management, such as identification, analysis, and planning of responses to positive and negative risks of the project adapted to the agile context (Tavares et al., 2019b).

3 RISK MANAGEMENT PRODUCT OWNER (RIMPRO)

RIMPRO is a framework that proposes a modification of the risk management processes exposed in PMBoK to managing risks involving PO. In this context, RIMPRO intends to guide traditional teams that need to adopt agile practices in their projects. This is quite common currently so that such teams can combine concepts proposed by Scrum with the structured risk management of RIMPRO, using concepts proposed by the Agile Practice Guide (Alliance, 2017). We consider RIMPRO is a framework instead of a process because each team is free to adapt it according to their needs. We use the word ”process” to define risk management steps to maintain the PMBoK nomenclature standard.

As shown in Figure 1, project information, such as budget, schedule, and stakeholders, is needed to guide the structure established for RIMPRO, which foresees the execution of six processes, described in the following subsections.

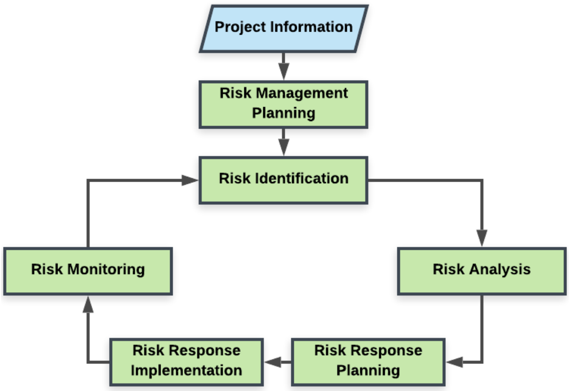

Figure 1: Relationship between RIMPRO processes.

Comparing the structure of RIMPRO with risk management processes established by PMBoK, we observed that Quantitative Risk Analysis provided by PMBoK is not performed due to insufficient numerical data on the identified risks by RIMPRO to carry out such process, considering its agile focus (Alliance, 2017).

For the correct application of RIMPRO, all stakeholders must participate in the proposed processes because the knowledge of them must be gathered during the execution of the risk management processes (Siqueira et al., 2017). Moreover, as risks that involve PO can arise throughout the project, the processes foreseen by RIMPRO must be iteratively executed on all Sprints. Thus, we emphasize that all the documentation provided by the framework must be created or reviewed in a Sprint. The documents generated or updated through the RIMPRO processes are **Risk Management Plan –** which describes how risk management processes are structured and executed; **Project Risk Backlog –** which reports all risks identified for the project; **Sprint Risk Backlog –** which reports all monitored risks of a particular Sprint in which each Sprint has its Sprint Risk Backlog.

3.1 Risk Management Planning

Risk Management Planning is the process of defining how RIMPRO project risk management is conducted to ensure that the degree, type, and visibility of risk management are proportional to the risks and the importance of the project to the organization and other stakeholders (Alliance, 2017). This process must be performed at the beginning of the project, before the first Sprint Planning Meeting, because while the PO performs functions throughout the entire project, risks associated with (s)he may arise. Thus, it is necessary to begin risk management before the beginning of the first Sprint (Sutherland and Sutherland, 2014).

At the beginning of the project, key definitions are established, such as the individual responsible for project risk management. This individual, namely Risk Master, must ensure that all Scrum Team members are performing the risk management processes foreseen by RIMPRO, as well as managing the planning documents. As this framework aims at risk management involving PO, Risk Master should be represented by the PO itself for two main reasons: i) the PO is the most important member of the Scrum Team for risk management (Tavares et al., 2019a); ii) and the risks can be related to the client and the PO is the most suitable member to treat them, since (s)he is the one that has direct contact with the client throughout the project among the other members of the Scrum Team (Sutherland and Sutherland, 2014).

In addition to the Risk Master definition, other definitions must be taken, namely (PMI, 2017):

- **Roles and Responsibilities:** As risk management is a vital process for the success of the project and the Scrum Team is a self-managed team that shares all the knowledge acquired among its members, everyone must participate in the management processes of risks, as this ensures transparency of project information among stakeholders (Sutherland and Sutherland, 2014);
- **Deadlines:** It defines when and how often the risk management processes will be carried out throughout the life cycle of the project and establish the protocols for applying the contingency reserves in the schedule;
- **Stakeholder Risk Appetite:** It is the maximum amount or volume of risks that stakeholders are willing to tolerate, called the maximum overall risk level of the project, to be expressed as limits of measurable risks for each project objective. As the project is divided into Sprints, the overall risk level of the project must be updated with each Sprint;
- **Budget:** It estimates funds based on designated resources for inclusion in the cost baseline, and establishes protocols for applying contingency and management reserves.

At the end of the process, all agreed definitions should be contained in the Risk Management Plan. For Sprint's goal not to be degenerate, changes to the Risk Management Plan must be requested during the Sprint Retrospective, as this meeting makes adjustments to Scrum Team to improve its work (Sutherland and Sutherland, 2014).

3.2 Risk Identification

In this process, risks are identified and their characteristics are documented. All stakeholders, including customers, should be encouraged to suggest new risks at any time. This process must be continuous because the project is susceptible to uncertainties throughout its life cycle, so risks can be identified at any time (Alliance, 2017). To support the Scrum Team and, specifically, the Risk Master, risks related to PO's roles succinctly described in Section 2 are listed and made available. The risks identified are documented with the following attributes:

- **Risk Title:** Short description of the risk;
- **Owner:** If the risk materializes, the owner is a member of the Scrum Team or an external organization. If the implementation of the risk response is outsourced, the owner should be the entity responsible for applying one of the agreed responses;
- **Date of Registration:** Date on which the risk was registered;
- **Type:** Opportunity or Threat;
- **Classification:** According to the analysis exposed in Section 2, PO's risks may be classified into three ways: **Requirements**; **Software Quality**; and **Migration to Scrum**. When the risk does not meet any of the classifications suggested above, it may be defined as **Not defined**;
- **Sprint Affected:** It determines the Sprint that can be affected if the risk materializes;
- **Probability of Occurrence:** It is a float value between zero and one, the higher the value, the greater the probability of the risk to occur. This value must be refined in the Risk Analysis process;
- **Impact:** It is a float value between zero and one, the higher the value, the greater the positive impact, in the case of opportunity, or negative, in the case of a threat, of the risk. This value must be refined in the Risk Analysis process;
- **General Description:** To avoid ambiguities, the following structure should be used to describe the risks identified using the risk specifications: If a CAUSE exists, the EVENT may occur, leading to the EFFECT;

- **Main Causes:** They must be evident, as they can cause one or more identified risks. These causes must be registered to support the future identification of risks for this and other projects;
- **Potential Responses:** During the Risk Identification process, some potential responses to the identified risks can be developed. These responses will be improved in the Risk Response Planning process;
- **Triggers:** Events or conditions that indicate that the risk is about to occur.

The risks present in the Project Risk Backlog will be further analyzed in the subsequent process.

3.3 Risk Analysis

In this process, the risks are analyzed and the Project Risk Backlog is updated for additional action. This analysis is done only in a qualitative way, due to insufficient data to perform a quantitative analysis (Alliance, 2017). As the purpose of this process is to prioritize the risks that will be monitored during the Sprint, the risk analysis must be done during Sprint Planning because in this meeting the Sprint goals are defined (Sutherland and Sutherland, 2014).

Therefore, the risks are analyzed through a technique proposed in this work, called Risk Planning Poker, in which Planning Poker is adapted to risk management (Sutherland and Sutherland, 2014). Based on the Delphi technique which is used to reach consensus among experts while preserving their anonymity (Dalkey and Helmer, 1963), the risk analysis is performed anonymously among Scrum Team members (Alliance, 2017). Risk Planning Poker is divided into two steps.

In Step 1, Scrum Team members empirically choose the risks that can affect the Sprint. These risks will be analyzed in Risk Planning Poker, in which each member has a deck. Each card contains a Fibonacci sequence number, as the human mind has difficulty perceiving faint changes. Thus, the distance between the numbers in this sequence is sufficient for the difference between the probabilities of occurrence or the impacts of the risks to be perceived more easily, intuitively. At each round, the probability of occurrence of risk is analyzed, with each member selecting a card and placing its face down on the table. Each card placed on the table represents the opinion of each participant about the probability of occurrence of the risk analyzed. Then all members cast their respective cards at the same time. If all members' opinions are within a distance of up to two cards from each other, the probability of occurrence of this risk is the arithmetic mean of the card values of all members. Otherwise, members who have selected the highest and lowest card explain their reasoning. After both explanations, a new round is made and the probability of the risk analyzed is the arithmetic mean of the card values (Sutherland and Sutherland, 2014). At the end of all the rounds from step 1, the risks that exceed the established limit value comprise the list of risks that will be analyzed in step 2.

Step 2 is analogous to step 1, but the impact of the risks that exceed the limit value of step 1 will be further analyzed. After completing the two steps, the facilitator determines the highest probability or impact risks that will be monitored throughout the Sprint, so that these risks will compose the Sprint Risk Backlog.

After Risk Planning Poker, the Risk Master creates the Sprint Risk Backlog. Thus, each Sprint has a list of risks that can affect the success of the iteration. Since the Scrum Team has few members, lean documentation, and a limited budget, the Sprint Risk Backlog should contain few risks. Therefore, it avoids a significant increase in additional project work (Sutherland and Sutherland, 2014).

To facilitate the monitoring of the Sprint Risk Backlog, a probability and impact matrix should be used (PMI, 2017). Thus, risks are normalized using min-max normalization in which risks are normalized to values in the range of 0 to 1 (Faceli et al., 2011). Such scaling is performed by the equation in 1, where $risk\_normalized_i$ is the value of $risk_i$ normalized to a value contained in the range [0, 1], $risk_i$ is the probability of occurrence or impact of $risk_i$ calculated on Risk Planning Poker, max and min are the values of the highest and lowest card in the deck used during the analysis of $risk_i$, respectively. For example, consider that an organization uses a deck whose lowest and highest cards range from 3 to 21, respectively. If the probability of occurrence and the impact of a risk is 17 and 15, respectively, then the normalized probability of occurrence and impact of this risk is 14/18 = 0.78 and 12/18 = 0.67, respectively.

After normalization, the Risk Master should define the probability and occurrence ranges for the categories (e.g. very low, low, moderate, high, and very high) and exhibit them in a probability and impact matrix. The responses to the risks that make up the Sprint Risk Backlog are defined in the subsequent process, called Risk Response Planning.

$$risk\_normalized_i = \frac{risk_i - min}{max - min} \tag{1}$$

3.4 Risk Response Planning

Risk Response Planning is the process responsible for developing options and actions to maximize opportunities and minimize threats to project objectives (Alliance, 2017). This process occurs after the Risk Analysis process because the risks of the Sprint Risk Backlog have already been defined, but their respective answers have not yet been elaborated.

Risk responses should be developed with the collaboration of all stakeholders, including customers with knowledge in the application domain and managers. Planned responses should be appropriate to the relevance of the risk, have cost-effectiveness to meet the challenge, be realistic within the project context, be agreed upon by all stakeholders, and have a designated stakeholder. In general, it is necessary to select the best response to risk among the various possible options. The Risk Master should mediate this process before the beginning of the Sprint because the responses to certain risks may vary throughout the project, i.e., risk responses for a given Sprint may not be appropriate for subsequent Sprints (Alliance, 2017).

For each risk of the Sprint Risk Backlog, the strategy or mix of response strategies of greater efficiency should be selected, including major and secondary strategies as needed. If the major strategies do not take effect, the possibility of applying the secondary strategies should be evaluated. Another point to emphasize is the secondary risks. For these risks, a surplus may be allocated for time or cost contingencies, as well as the identification of the conditions that trigger the use of these surpluses (Alliance, 2017). At the end of the process, the Risk Master must update the Sprint Risk Backlog response lists and starting the subsequent process.

3.5 Risk Response Implementation

Risk Response Implementation is the process responsible for implementing the risk response plans that compose the Sprint Risk Backlog to ensure that risk responses are carried out as planned. Attention to this process will ensure that the responses to the agreed risks are implemented. A common problem with risk management is that the Scrum Team devotes efforts to identify and analyze risks and developing risk responses, but no action is taken to manage them (Alliance, 2017; Sutherland and Sutherland, 2014).

Tools and techniques can be used to the implementation of the risk response plans associated with Sprint Risk Backlog, such as (Alliance, 2017):

- **Expert Opinion:** It should be considered by Scrum Team individuals with specialized knowledge to validate or modifying responses to risks, if necessary, and decide how to implement them most efficiently;
- **Interpersonal and Team Skills:** Among the interpersonal and team skills that can be used in this process, the main one is influence. Some risk response actions may be owned by people outside the Scrum Team or who have other conflicting demands. It is necessary, at certain points in the project, that the Risk Master takes influence to encourage the appointed risk owners to take the necessary measures when appropriate;
- **Project Management Information System:** May include schedule, resource, and cost software to ensure that agreed risk response plans and their associated activities are integrated into the project along with other project activities.

If any response is modified during the process, the Sprint Risk Backlog should be updated (Alliance, 2017).

3.6 Risk Monitoring

Risk Monitoring is the process responsible for: (i) monitoring the implementation of the risk response plans contained in the Sprint Risk Backlog; (ii) monitoring risks that may affect Sprint, and (iii) assessing the effectiveness of risk management processes throughout the Sprint. The main benefit of this process is that it allows project decisions to be based on current information about the overall exposure to project risk and individual project risks. This process occurs throughout the Sprint, since risks may arise or materialize throughout the project (PMI, 2017).

The step of evaluating the effectiveness of the risk management processes proposed by RIMPRO is carried out during the Sprint Retrospective meeting, as this meeting aims to identify what worked well, what can be improved and what actions will be taken to improve several aspects that may limit the speed of the project, such as deficiencies in the risk management processes. Such an evaluation must be made with all Scrum Team members at the Sprint Retrospective meeting, as it is the moment when the whole team must present the lessons learned in each Sprint for the benefit of future projects and subsequent Sprints of the current project. In this way, plans to improve risk management processes can be established, to be applied in Sprints and the following projects (PMI, 2017; Sutherland and Sutherland, 2014).

To ensure that stakeholders are aware of the current risks, Sprint must be continuously monitored.

Risk Monitoring uses project information to determine whether (Alliance, 2017): the responses to the implemented risks are effective; the current project risks have changed; the status of individual risks identified in Sprint has changed; new risks of individual projects have arisen; the risk management approach is still appropriate; risk management processes have been followed; the contingency surplices for cost or time require modifications.

Risk reviews are scheduled regularly and should examine and document the effectiveness of the risk responses made in the Sprint Risk Backlog. Risk reviews can also result in identifying newer risks, including secondary risks arising from responses to agreed risks, reassessing current risks, closing out risks that are out of date, identifying problems that arise as a result of risks that have occurred, and identifying lessons learned for implementation in subsequent Sprints or similar projects in the future. The Sprint risk review should be conducted as part of a regular project status meeting, such as the Daily Scrum (Alliance, 2017; Sutherland and Sutherland, 2014).

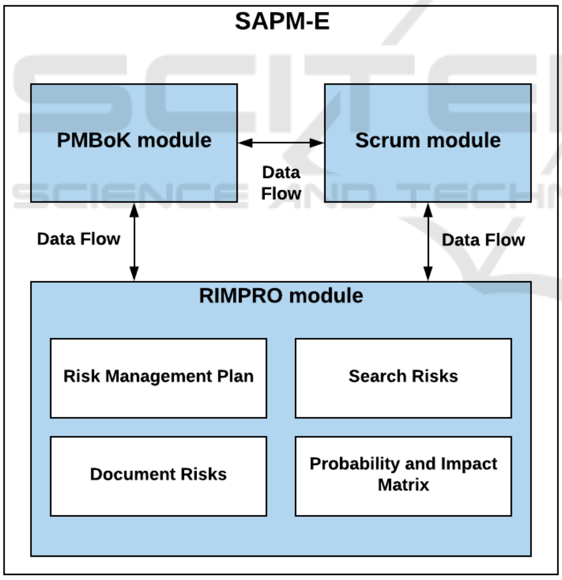

Figure 2: SAPM-Extended Architecture.

4 RIMPRO ASSESSMENT

To evaluate RIMPRO, we automated its processes through a computational module attached to the computational tool called System to Aid Project Managing (SAPM). SAPM has been developed with the participation of one of the authors. The automation of risk management processes is crucial since the lack of knowledge of professionals in this knowledge area, as well as the scarcity of computational tools, are barriers to the execution of risk management processes (Gregoriades et al., 2011), such as those foreseen by SAPM. The development of a computational module facilitates the application of RIMPRO in Scrum projects. SAPM allowed the management of traditional projects, based on the knowledge areas provided by PMBoK, as well as an independent alternative for agile project management through Scrum (Mendonça et al., 2014). The incorporation of RIMPRO into SAPM referred to in this paper as SAPM-Extended has allowed the execution of traditional processes foreseen in PMBoK, in an agile way, according to Scrum. Figure 2 shows the general architecture established for SAPM-Extended.

We performed the RIMPRO assessment from September to October 2018. Students, both undergraduate and graduate who had already completed the discipline of Software Engineering, as well as IT professionals, participated in the assessment process. The assessment process was carried out by 31 participants, of which 12.9% are IT professionals, 22.6% are graduate students, and 64.5% are undergraduate students. Although most participants in the sample are not IT professionals, participating students have contact with projects that use PMBoK and Scrum.

One of the authors invited all participants of the assessment to a meeting, in which RIMPRO and SAPM-Extended were presented and possible doubts were resolved. The participants had the opportunity to simulate a Sprint, in which risks were identified and the probability and impact were measured by the participants. At the end of the meeting, a guide for using the RIMPRO and an assessment form were sent to all participants by email. In the 15 days following the meeting, everyone was free to continue using the RIMPRO on the Web and fill out the assessment form.

We used the Likert Scale (Boone and Boone, 2012) with ”Strongly Disagree”, ”Disagree”, ”Neither Agree Nor Disagree”, ”Agree” and ”Strongly Agree” items, to evaluate RIMPRO under the following three statements (S):

- **S1:** I am satisfied with the ease of learning and use of RIMPRO;
- **S2:** PMBoK’s adaptation by RIMPRO to manage risks involving PO contributes to the management of agile projects as well as managing risks related to PO’s roles;
- **S3:** The risk management divided by Sprints, instead of addressing in a general way as it occurs in traditional projects, contributes to the risk management in agile projects.

The obtained results are illustrated in Figure 3. We also provided two open questions to the participants expressing their opinions about RIMPRO's strengths and weaknesses.

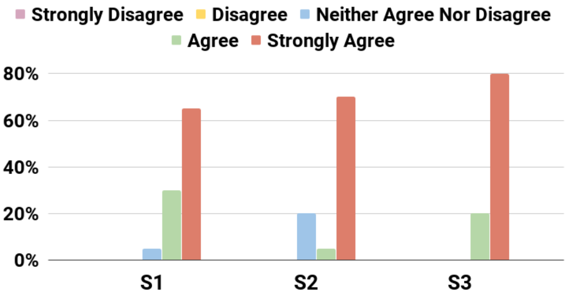

Figure 3: Assessment Histogram of RIMPRO.

From the analysis of the answers obtained from the participants shown in Figure 3, it is possible to verify a satisfactory result, since more than 50% of the participants strongly agreed with all the statements. We can conclude that RIMPRO is easy to learn and use, since the way to classify the risks through Sprints and the prioritization of the risks through the adaptation of Planning Poker, proposed in this work, are analogous to the stage of requirements engineering. In this way, the use of RIMPRO in real projects looks simple and it requires few hours of training because the risks are documented and catalogued in a similar way to the requirements of agile software projects.

As for strengths, the participants concluded that iteratively managing risks is beneficial because the Scrum Team has few members, and the project budget is relatively smaller than traditional projects. Thus, instead of monitoring all risks throughout the project, only risks that can affect a given Sprint are monitored. It is relevant when it comes to risks involving the PO because (s)he is present throughout the project and, so, the probability of occurrence and the impact of the risks involved may vary throughout Sprint. Consequently, risks monitored in a Sprint may not be monitored in subsequent Sprints, and vice versa. As for the weaknesses, even the SAPM-Extended having a previous database containing the risks exposed in Section 2, the participants noted a limited amount of risks involving the PO's roles to guide the execution of RIMPRO through the project, as well as lessons learned from previous projects. They also stressed the importance of assisting the Scrum Team in discussing new risks that may arise.

5 RELATED WORKS

With respect to project risk management, some studies address this topic to increase the success rates in the development of software projects. (Janjua et al., 2016) stated that it is necessary to use risk management practices. Indeed, (Pasha et al., 2018) point out that the risks related to large-scale software development are challenging, and they provide a comparative study of the risks found in the literature related to small and large-scale software development. (Bista et al., 2017) also emphasize the importance of risk management for software projects, to propose a new approach to estimate these risks using the analyzed regression method, which consists of using a collection of statistical techniques for models that describe the reasonable relations between several explanatory variables, the use of which is described quantitatively on a determined process.

Regarding risk management related to the agile approach to software development, some studies highlight the need to identify and analyze such risks. (Khatri et al., 2014) suggested a set of good risk management practices applied to agile methods, but it has not been validated. (Moran, 2014) presented a generic process for performing risk management in agile methods. However, it has not been validated. (Alsaqaf et al., 2017) exposed the main risks related to software quality, emphasizing that agile approaches neglect the software quality requirements. In turn, (Andrat and Jaswal, 2015) warned that, unlike the traditional approach, risks related to the agile approach to software development are little known. Considering Scrum, (Tavares et al., 2019b) warns that such a method does not specify project risk management activities, concluding that classic risk management processes, such as planning, analyzing, and planning responses to risks, must be incorporated and adapted to Scrum. Finally, (Tavares et al., 2019a) consider that the PO plays the most important role and its risk management practices are strongly recommended. This motivated this work, by presenting RIMPRO as a useful framework for managing risks involving the PO.

The adaptation of the PMBoK knowledge area namely risk management exposed in this paper is something that we did not find in any of the analyzed studies. Besides, even showing the importance of PO in Scrum Teams, we have not found studies that have identified the risks related to PO activities or proposed an effective way to manage them. Regarding all researches related to risk management of agile methods, none has any assessment process, highlighting the importance of the RIMPRO assessment carried out.

RIMPRO differs from a general risk management approach because of two features. The classification presented in Section 2 intends to guide RIMPRO users to classify PO risks, as described in Section 3.2. Also, the data obtained by surveying the risks involving the PO was used to create a database for the SAPM-Extended that was used to automate RIMPRO. Thus, during the RIMPRO assessment, users were able to access the risk database through SAPM-Extended. The participants considered this database as a strong point that helped to understand RIMPRO, although it still requires expansion. Another feature that distinguishes RIMPRO from a general risk management approach is the fact that the Risk Master is the PO because the PO is the most important member of the Scrum Team for risk management, and the risks can be related to the client and the PO is the most suitable member to treat them, since (s)he is the one, among the other members of the Scrum Team, who has direct contact with the client throughout the project.

6 CONCLUSION

Although Scrum is one of the most widely used agile methods today, it does not have a structured way of managing risks that may arise throughout the project. This work contributed to structuring the RIMPRO to support the Scrum Team to implement a systematic way of managing risks, based on the PMBoK Guide, without interfering with the agile values of Scrum. Structuring a risk management framework for agile projects contributes to this area maturity and, consequently, promotes the advancement of research aimed at its improvement.

The results obtained through the preliminary RIMPRO assessment are promising. This assessment provides evidence of the feasibility of RIMPRO in supporting the Scrum Team by taking decisions based on the risks related to the PO that may arise in an agile and satisfactory manner. However, we have not applied RIMPRO to real agile projects. Furthermore, another limitation of RIMPRO is the non-use of quantitative techniques to perform risk analysis. Therefore, we intend to evaluate the effectiveness of RIMPRO by conducting experiments and realistic case studies in companies, as well as improving RIMPRO by using quantitative risk analysis without infringing agile manifesto.

ACKNOWLEDGEMENTS

Thanks to God for everything, as well as Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) and Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) for the financial support.

REFERENCES

Alliance, A. (2017). *Agile Practice Guide*, volume 1. Bukupedia.

Alsaqaf, W., Daneva, M., and Wieringa, R. (2017). Agile quality requirements engineering challenges: First results from a case study. In *2017 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)*, pages 454–459.

Andrat, H. and Jaswal, S. (2015). An alternative approach for risk assessment in scrum. In *2015 International Conference on Computing and Network Communications (CoCoNet)*, pages 535–539. IEEE.

Asghar, A. R., Tabassum, A., Bhatti, S. N., and Jadi, A. M. (2017). Impact and challenges of requirements elicitation & prioritization in quality to agile process: Scrum as a case scenario. In *2017 International Conference on Communication Technologies (ComTech)*, pages 50–55. IEEE.

Bista, R., Karki, S., and Dongol, D. (2017). A new approach for software risk estimation. In *2017 11th International Conference on Software, Knowledge, Information Management and Applications (SKIMA)*, pages 1–8.

Boone, H. N. and Boone, D. A. (2012). Analyzing likert data. *Journal of extension*, 50(2):1–5.

Curcio, K., Navarro, T., Malucelli, A., and Reinehr, S. (2018). Requirements engineering: A systematic mapping study in agile software development. *Journal of Systems and Software*, 139:32–50.

Dalkey, N. and Helmer, O. (1963). An experimental application of the delphi method to the use of experts. *Management science*, 9(3):458–467.

de Godoi Contessoto, A., Sant'Ana, L. A., De Souza, R. C. G., Valêncio, C. R., Donegá, G. F., Amorim, A. R., and Esteca, A. M. N. (2016). Improving risk identification process in project management. In *Proceedings of the International Conference on Software Engineering and Knowledge Engineering, SEKE*, pages 555–558.

Dikert, K., Paasivaara, M., and Lassenius, C. (2016). Challenges and success factors for large-scale agile transformations: A systematic literature review. *Journal of Systems and Software*, 119:87–108.

Dingsøyr, T., Nerur, S., Balijepally, V., and Moe, N. (2012). A decade of agile methodologies: Towards explaining agile software development. *Journal of Systems and Software*, 85(6):1213 – 1221.

Elghariani, K. and Kama, N. (2016). Review on agile requirements engineering challenges. In *2016 3rd International conference on computer and information sciences (ICCOINS)*, pages 507–512. IEEE.

Faceli, K., Lorena, A. C., Gama, J., and Carvalho, A. C. P. d. L. F. d. (2011). *Inteligência artificial: uma abordagem de aprendizado de máquina*. LTC.

Fitriani, W. R., Rahayu, P., and Sensuse, D. I. (2016). Challenges in agile software development: A systematic literature review. In *2016 International Conference on Advanced Computer Science and Information Systems (ICACSIS)*, pages 155–164. IEEE.

Gandomani, T. J. and Nafchi, M. Z. (2016). Agile transition and adoption human-related challenges and issues: A grounded theory approach. *Computers in Human Behavior*, 62:257–266.

Gold, B. and Vassell, C. (2015). Using risk management to balance agile methods: A study of the scrum process. In *2015 2nd International Conference on Knowledge-Based Engineering and Innovation (KBEI)*, pages 49–54. IEEE.

Gregoriades, A., Lesta, V. P., and Petrides, P. (2011). Project risk management using event calculus (s). In *SEKE*, pages 335–338.

Inayat, I., Salim, S. S., Marczak, S., Daneva, M., and Shamshirband, S. (2015). A systematic literature review on agile requirements engineering practices and challenges. *Computers in human behavior*, 51:915–929.

Janjua, U. I., Jaafar, J., and Lai, F. W. (2016). Expert's opinions on software project effective risk management. In *2016 3rd International Conference on Computer and Information Sciences (ICCOINS)*, pages 471–476. IEEE.

Khatri, S. K., Bahri, K., and Johri, P. (2014). Best practices for managing risk in adaptive agile process. In *Proceedings of 3rd International Conference on Reliability, Infocom Technologies and Optimization*, pages 1–5. IEEE.

Mendonça, L. T., Esteca, A. M. N., De Souza, R. C. G., Santos, A. B., and Valêncio, C. R. (2014). A scrum support system integrated to a web project management environment. In *29th International Conference on Computers and Their Applications*, pages 111–116.

Moran, A. (2014). Agile risk management. In *Agile Risk Management*, pages 33–60. Springer.

Pasha, M., Qaiser, G., and Pasha, U. (2018). A critical analysis of software risk management techniques in large scale systems. *IEEE Access*, 6:12412–12424.

PMI, editor (2017). *A Guide to the Project Management Body of Knowledge (PMBOK Guide)*. Project Management Institute, 6 edition.

Siqueira, D. L., Fontoura, L. M., Bordini, R. H., and de Lima Silva, L. Á. (2017). A knowledge engineering process for the development of argumentation schemes for risk management in software projects. In *Proceedings of the 29th International Conference on Software Engineering and Knowledge Engineering*.

Sutherland, J. and Sutherland, J. (2014). *Scrum: the art of doing twice the work in half the time*. Currency.

Tavares, B. G., da Silva, C. E. S., and de Souza, A. D. (2019a). Practices to improve risk management in agile projects. *International Journal of Software Engineering and Knowledge Engineering*, 29(03):381–399.

Tavares, B. G., da Silva, C. E. S., and de Souza, A. D. (2019b). Risk management analysis in scrum software projects. *International Transactions in Operational Research*, 26(5):1884–1905.