Using Bayesian Networks to Support Managing Technological Risk on Software Projects

Emanuel Dantas
emanuel.dantas@virtus.ufcg.edu.br
Intelligent Software Engineering Group - UFCG
Campina Grande, PB, Brazil

Ademar Sousa Neto
ademar.sousa@virtus.ufcg.edu.br
Intelligent Software Engineering Group - UFCG
Campina Grande, PB, Brazil

Mirko Perkusich
mirko@virtus.ufcg.edu.br
Intelligent Software Engineering Group - UFCG
Campina Grande, PB, Brazil

Hyggo Almeida
hyggo@virtus.ufcg.edu.br
Intelligent Software Engineering Group - UFCG
Campina Grande, PB, Brazil

Angelo Perkusich
perkusic@virtus.ufcg.edu.br
Intelligent Software Engineering Group - UFCG
Campina Grande, PB, Brazil

ABSTRACT

Risk management is essential in software project management. It includes activities such as identifying, measuring and monitoring risks. The literature presents different approaches to support software risk management. In particular, the researchers popularly used Bayesian Networks because they can be learned from data or elicited from domain experts. Even though the literature presents many Bayesian networks (BN) for software risk management, none focus on technological risk factors. Given this, this paper presents a BN for managing risks of software projects and the results of a static validation performed through a focus group with eight practitioners. As a result, the practitioners agreed that our proposed to manage technological risks of software projects using BN is valuable and easy to use. Given the successful results, we concluded that the proposed solution is promising.

CCS CONCEPTS

• **Software and its engineering → Risk management**.

KEYWORDS

Risk Management, Technological Risk, Bayesian Network

ACM Reference Format:
Emanuel Dantas, Ademar Sousa Neto, Mirko Perkusich, Hyggo Almeida, and Angelo Perkusich. 2021. Using Bayesian Networks to Support Managing Technological Risk on Software Projects. In ,. ACM, New York, NY, USA, 6 pages. https://doi.org/10.5753/ise.2021.17277

1 INTRODUCTION

A risk is an uncertain event or set of uncertain events that, if occurred, has a negative or positive effect on one or more of the project objectives [15, 29, 36]. Risks can lead to organizations' losses, which may relate to decreased product quality, increased production costs, and not meeting project deadlines [2, 30]

According to ISO 31000 [16], risk management is a set of coordinated activities for the organization to be directed and controlled regarding risk. In general, risk management consists of applying skills and knowledge to reduce threats at an acceptable level while maximizing opportunities [14, 24].

ISE '21, CBSoft '21: Brazilian Conference on Software: Theory and Practice

https://doi.org/10.5753/ise.2021.17277

There are several techniques and tools proposed to support risk management. For instance, graphic methods as cause and effect diagrams [32], SWOT matrix [34], or Intelligent techniques [27], which include Artificial Intelligence and Data Analytics. In special, Bayesian Networks (BN) are widely used to support risk management [10, 20, 37]. Bayesian networks are helpful to aid decision-making due to their ability to support causal reasoning [4]. The technique consists of a probabilistic graph used to represent knowledge about an uncertain domain [4].

Bayesian networks' popularity follows from their ability to deal with uncertainty and because of their ease of understanding by professionals [28]. Besides, they are flexible for learning from data or eliciting knowledge from domain experts [6, 33]. BN-based risk management solutions model the main risk factors of software development, and according to observations of the environment, make inferences to assist the project's risk management [18, 20, 25]. Although the literature presents many BN-based solutions to support software risk management, none focuses on the software projects technological risks. These risks have high volatility and difficult predictability [5, 26].

We built a BN to support risk management in software projects focusing on technological risks within this context. We built the BN thinking as a framework containing cold (i.e., context-independent) and hot spots (i.e., context-sensitive). We called the BN's context-independent fragment its core and developed it by employing a Grounded Theory study [12], in which we collected data from 25 practitioners. However, the details of the employed Grounded Theory (GT) are not within this paper's scope.

This paper summarizes the proposed BN and how users can adopt it by complementing it to fit their needs. Furthermore, it details the applied static validation [13] through a focus group [19] with eight practitioners and discusses its practical utility in light of the Technology Acceptance Model (TAM) [7].

We organized the remaining of the paper as follows: Section 2 summarizes the state-of-the-art of risk management support for software projects. Section 3 presents the proposed BN; Section 4 describes the methods employed to define and validate the BN's use cases and discusses this study's findings and implications. Finally, Section 5 discusses our final remarks and future work.

2 BN FOR SOFTWARE RISK MANAGEMENT

Bayesian networks belong to the family of probabilistic graph models and are used to represent knowledge about an uncertain domain

[3]. A Bayesian network, B, is a directed acyclic graph representing a joint probability distribution over a set of random variables V [11]. The network is defined by the pair $B = \{G, \Theta\}$. G is the directed acyclic graph in which the nodes $X_1, \ldots, X_n$ represent random variables, and the arcs represent the direct dependencies between these variables. Θ represents the set of the probability functions. This set contains the parameter $\theta_{x_i|\pi_i} = P_B(x_i|\pi_i)$ for each x_i in X_i conditioned by π_i, the set of the parameters of X_i in G. Equation 1 presents the joint distribution defined by B over V.

$$P_B(X_1, \ldots, X_n) = \prod_{i=1}^{n} P_B(x_i|\pi_i) = \prod_{i=1}^{n} \theta_{X_i|\pi i} \quad (1)$$

Studies have applied Bayesian networks for supporting risk management for many specific purposes including setting a schedule [25]; budget definition [18]; quality assessment [1]; and defect prediction [20]. Fenton et al. [9] created a Bayesian Network for risk management that supports decision making in various software design activities. For validating BN-based risk management solutions, many studies relied on eliciting knowledge from experts. Nguyen et al. [25] used an expert to create two hypothetical software project situations. Khodakarami and Abdi [18] used a real example lived by one of the authors. Such validation strategy was also applied by Ancveire [1]. Fenton et al. [9] made a simulation with two examples of projects to compare the view of experts with the results of the Bayesian network.

These studies have shown promising results, but they do not focus on risk factors related to the technological aspects of software projects. The risk factors used in their approach include information on requirements, human resources, management knowledge, quality process, and deliveries [1, 9, 18, 25]. Some researchers investigate risk factors related to technological characteristics but simplify the analysis by defining only one risk factor to deal with all the project's complexity related to technologies [5, 26]. When trying to apply these approaches to modeling the technological risks of projects, the models proved to be too generic and not very useful in managing technological risks. Thus, our work is motivated to fill this gap and complement other studies to assist risk management in software projects.

In this study, we constructed a BN to support risk management using risk factors related to the technological characteristics of software projects. The factors are related to the technical aspects of software projects. A previous study [X] found that these factors are related to the development environment, integrations, architecture, and innovative techniques. We validated the BN using simulated scenarios and a focus group with practitioners.

3 PROPOSED SOLUTION

As mentioned earlier, a BN can be used to assist risk management in software projects. A BN can visually construct a cause-consequence relation and provide conditional probabilistic estimations of the software project's risk status. In this study, we propose an approach to support risk management focused on technological risks. The proposed approach involves two stages: BN construction (Section 3.1) and Data Analysis (Section 3.2).

3.1 Bayesian Network Construction

The first step was to construct the BN. Our goal was to build a technology-driven risk assessment model in which the user could calculate the project's overall risk given the technologies adopted. Further, the model should allow the modeling of risk mitigation strategies and minimize the risks' probabilities. Such a structure enables users to assess their project's current risk status and diagnose their mitigation strategies, consequently supporting their decision-making.

Given our solution's requirements, the constructed BN would be context-sensitive, and attempting to construct one with external validity, in other words, ready-to-use by multiple organizations, was doomed to failure.

Thus, we constructed a BN with four fragments: core, technology, strategy, and context-sensitive risk. The Core fragment consists of the kinds of risk factors technological common on software projects. Thus, it should be context-independent. Considering our proposed BN as a framework, the Core fragment would be the framework's cold spot. The Supplementary Material presents the definition for each Core node[1].

We built the core BN through a Grounded Theory study with twenty-five professionals from ten organizations working on software projects. It is out of this paper's scope to describe the research methods applied to build the Core fragment. Figure 1 shows the Core fragment, in which the leaf nodes are the yellow ones. As discussed in what follows, we can think about the Core fragment's leaf nodes as an interface connected to nodes from Technology, Strategy, and Context-Sensitive Risk.

The Technology fragment consists of variables representing the technologies used by the project team, such as JWT, KeyStore, Firebase, and Pandas. For instance, JWT and KeyStore are security technologies; thus, they could be represented, respectively, by the Technology nodes *JWT* and *KeyStore* and connected to the Core fragment's leaf node *Security*. Notice that one Technology node can be connected to multiple Core nodes and vice versa. Figure 2 presents an example in which the Technology nodes *JWT* and *KeyStore* (pink nodes) connect to the Core node *Security* (yellow node). Notice that the *Security* is represented by a dashed line, meaning that it is a leaf node of the Core fragment.

The Strategy fragment consists of variables representing the risk mitigation strategies employed by the project team, such as having offline data persistence, using stateless communication, and reviewing proxy settings. For instance, we could have the Strategy node *Offline data persistance* connected to the Core fragment's leaf node *Security*. Figure 2 complements the example shown in Figure 3 by connecting the Strategy node *Offline data persistance* to the Core node *Security*.

Finally, the Context-Sensitive Risk Fragment consists of specialized risks that further detail the risk of a Core fragment's leaf node. For instance, we could have the Context-Sensitive Risk node *Token decoding failed* connected to the Core fragment's leaf node *Security*. Figure 4 presents an example connecting the Context-Sensitive Risk node *Token decoding failed* to the Core node *Security*. Notice that the Strategy and Technology nodes previously connected to the Core node *Security* are now connected to the newly added Context-Sensitive node.

The nodes on the Technology, Strategy, and Context-Sensitive Risk fragments are defined given the project's mobile, Web, and IoT domains. They can be defined by consulting specialists and the project team or by using data from past projects. Considering our proposed BN as a framework, the Technology, the Strategy, and Context-Sensitive Risk fragments would be the framework's hot spots.

[1] https://doi.org/10.5281/zenodo.4608651

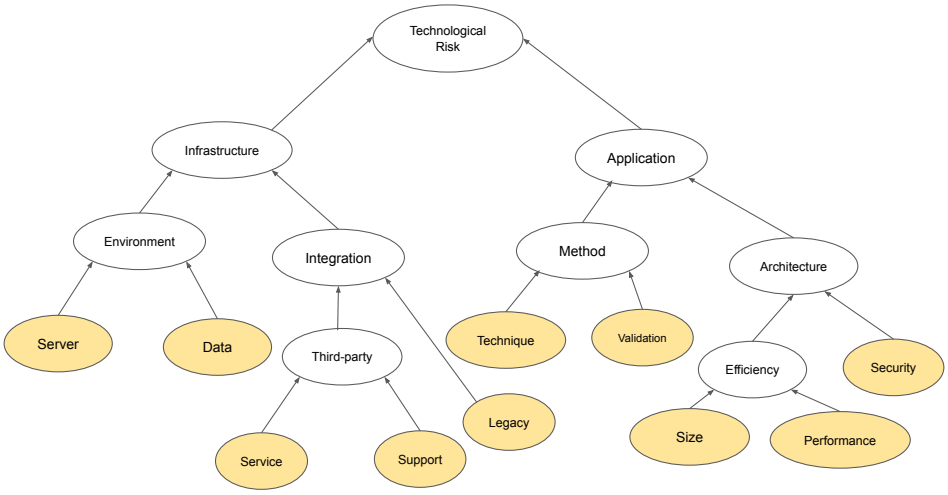

Figure 1: The proposed BN's Core fragment.

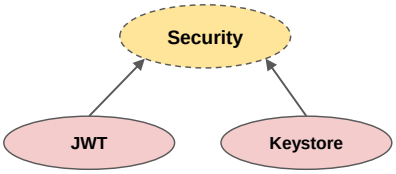

Figure 2: Example of the Technology nodes *JWT* and *KeyStore* (pink) connected to the Core node *Security* (yellow).

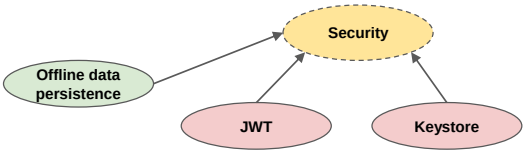

Figure 3: Example of the Technology nodes *JWT* and *KeyStore* (pink) and the Strategy node *Offline data persistance* (green) conneceted to the Core node *Security* (yellow).

By default, we defined all BN's nodes as Boolean, but the users could adapt the nodes' scale to fit their needs. Regarding the probability functions, we defined them by eliciting the knowledge from two domain experts.

3.2 Data Analysis

After identifying the Technology, Strategy, and Context-Sensitive Risk nodes, connecting them to the Core fragment, and making the necessary adjustments to the nodes' scales and probability functions, the users can input data (i.e., evidence) into the Bayesian network. One possible scenario is to fill in the project's technologies to assess the risks. Similarly, in a second scenario, mitigation

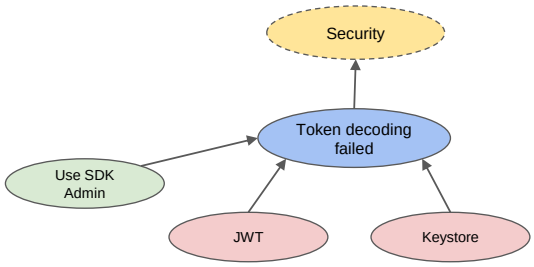

Figure 4: Example of adding a Context-Sensitive Risk node *Token decoding failed* (blue) to a Core node *Security* (yellow) to the example shown in Fig. 3.

strategies are filled and risks assessed. After executing the BN using a BN inference tool, the remaining nodes' probabilities are calculated in both cases. Thus, the users can assess the project's risks given the adopted technologies or employed strategies (i.e., prognosis).

Alternatively, the users can fill in current project risks by entering data into Core nodes. After executing the BN, the probability for the remaining nodes, including the Technology and Strategy nodes, are updated (i.e., diagnosis). In any case, decision-making takes place according to the information provided on the network.

4 EMPIRICAL EVALUATION

We performed a static validation of the proposed BN, presenting it to practitioners and analyzing its acceptance level. According to Gorscheck and Wohlin [13], performing static validation is vital for learning a proposed solution's benefits, to get buy-in from practitioners, and a recommended step before piloting it through dynamic validation (e.g., action research or case study).

For this purpose, following a guideline for the knowledge engineering of BN [21], we developed six simulated scenarios (i.e.,

model walkthrough or use cases) in collaboration with two industry experts (see Section 4.1). Later, we performed a focus group, following the guidelines presented in Kontio et al. [19] to assess the proposed BN's practical utility by analyzing the simulated scenarios (see Section 4.2).

4.1 Simulated Scenarios

According to the Expert-based Knowledge Engineering of Bayesian Networks [22], the first validation step for a BN is to define simulated scenarios. Such a step is essential because before adopting the BN on real projects, it is necessary to obtain a subjective assessment of how well the experts "feel" that the model calculates what it is supposed to estimate, at face value (i.e., face validity) [8]. Simulated scenarios are "what-if" scenarios, establishing a set of inputs and expected outputs and comparing them with the BN's calculated outputs (i.e., similar to software test cases).

We defined the six simulated scenarios focused on mobile application development with a software architect and a technical leader. Both participants were invited, given their knowledge and interest in collaborating with our research. The software architect had over ten years of experience with software projects and five years of experience with mobile application development. The technical leader had more than five decades of experience with software projects and two years of experience with mobile application development.

Each scenario describes a software project with examples of technological decisions to be taken by a development team. For defining each scenario, first, we came up with a product's requirements. We then discussed possible technologies that could be adopted to fulfill the requirements and the risks they triggered. If necessary, we came up with risk mitigation strategies for the given technologies and risks. Due to space limitations, this paper presents one simulated scenario. The remaining ones are made available on a Supplementary Material[2].

The simulated scenario described herein considers the case in which the team is at sprint zero or the initial stages of a project and needs to decide which technologies to adopt. The project has the following technical restrictions defined by the client: user data must be encrypted in the registration, requests during operations cannot take more than 500 ms, the application must work on IOS and Android, and finally, the application must connect with credit platforms in real-time.

Given this context, the experts discussed the technologies that could be used, added the associated Technology nodes, and defined the necessary probability functions. For this first simulated scenario, they came with six Technology nodes: *KeyStore*, *Auth0*, *Electron*, *Web services*, *GRPC*, and *Retrofit*. Since they assumed that they would use these six technologies, they defined each of their associated node's values as *TRUE*.

Then, they executed the BN and visualized the Core fragment's nodes calculated probabilities and noticed that some nodes (i.e.., risks) had a high probability; in other words, they indicated high risk: security flaw with persistence and lack of scalability in the data. The experts evaluated alternative technologies with this information, marking new technologies on the network and replacing some of the initially chosen ones. One alternative analyzed was to swap *Electron* to *Ionic*. After this change in the network, the Core fragment's nodes probabilities, representing the risks mentioned above, significantly decreased.

[2]https://doi.org/10.5281/zenodo.4608684

Figure 5 shows a perspective of the Bayesian network for this scenario. Replacing Electron by Ionic reduced the probability of two context-sensitive Risks: security flaw with data persistence and lack of scalability in the data. These risks are connected to the Core nodes *Security* and *Size*, respectively. This simulated scenario showed the proposed BN's potential to assist software teams in selecting technologies based on the potential triggered risks.

4.2 Focus Group

Our next step was to analyze the practitioner's adoption intention after validating our approach by defining six simulated scenarios with two experts. For this purpose, we followed the Technology Acceptance Model (TAM) [7].

However, before answering TAM's questionnaire, the practitioners needed to have experienced our proposed solution. For this purpose, we conducted a Focus Group [35] with eight professionals from a technological company. All the participants were invited and volunteered to participate in our study with no incentives.

Most participants were graduates of Information Technology courses (75%) with more than two years of experience in software team leadership positions. The remaining participants (25%) were project managers with other degrees but performed a technical leader's profile. The Focus Group session lasted 2 hours and 30 minutes. After reading each of the six scenarios' descriptions, we instantiated the BN nodes and showed how the technology adopted and employed risk mitigation strategies interfered with the project's risks.

After discussing each scenario, we applied a TAM questionnaire. The answers followed a Likert scale [17] with five possible answers ranging from strongly disagree (mapped to number 1) to agree strongly (mapped to number 5).

According to TAM, two variables impact a new technology adoption: perceived usefulness and perceived ease of use [23]. Perceived usefulness refers to the degree to which an individual believes that using a particular technology would enhance their job performance. Perceived ease of use refers to the degree to which an individual believes that using a specific technology would be free of physical and mental effort [38]. Table 1 shows the assigned variables for this study.

Table 2 presents the values for median, mean, Standard Deviation (SD) for the answers for the Perceived Usefulness (PU) and Perceived Ease of Use (PEoU). About PU, all average values were higher than 4.0, indicating that participants generally had positive attitudes toward the approach. In particular, participants reported that the approach is beneficial for identifying risks (median with a value of 5). Regarding the PEoU variables, the assessment of the approach was also positive. Only the variable V5 had an average value of less than 4. We believe that this happens because some participants did not have any previous familiarity with Bayesian networks.

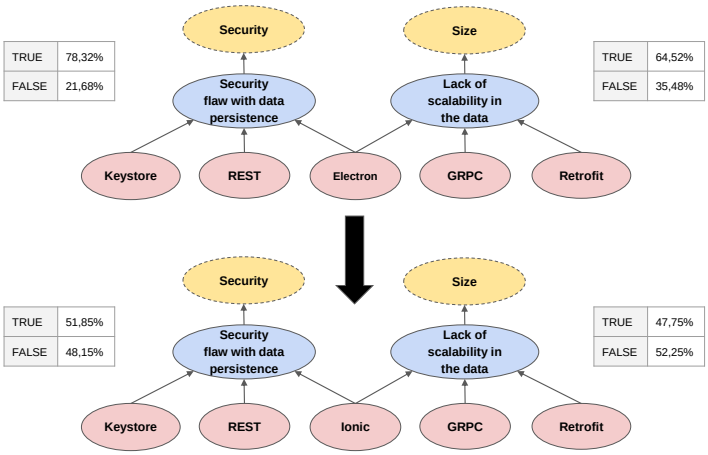

Figure 5: Perspective of the Bayesian network - Scenario 01

Table 1: Questionnaire Statements on: Perceived usefulness and Perceived ease of use

Type	Definition	Variables
Perceived usefulness (PU)	The level at which a person believes that using the tool improves the performance of their tasks.	V1: Using the approach is useful for identifying risks.
		V2: Using the approach is useful for measurement risks.
		V3: Using the approach is useful for monitoring risks.
Perceived ease of use (PEoU)	Level at which the person presents their perception of the tool in terms of ease of learning and operation.	V4: Learning how the approach works was easy for me
		V5: I often get confused in researching and understanding information in the approach
		V6: Understanding the approach is simple

Table 2: Questionnaire results

Variable	Definition	Mean	Median	SD
V1	Using the approach is useful for identifying risks.	4.71	5	0.745
V2	Using the approach is useful for measurement risks.	4.12	4	0.783
V3	Using the approach is useful for monitoring risks.	4.32	4	0.755
V4	Learning how the approach works was easy for me.	4.35	4	0.715
V5	I often get confused in researching and understanding informationin the approach.	3.82	3	0.852
V6	Understanding the approach is simple	4.12	4	0.841

During the Focus Group, participants mentioned suggestions and criticisms of the approach. Below are excerpts from the subjects discussed. Some participants commented on when the approach would be most useful:

> *"I found the approach interesting for risk analysis. I imagine that it would be more useful in the industry in defining the architecture of the project. That is, at the moment of proposal creation or during sprint zero.".* (E02, Project Manager)

> *"I suggest running the Bayesian network at the end of the project to analyze if the decisions made by the team during development were the most correct.".* (E03, Technical Leader)

One of the participants reported a critical factor. It is common for software projects to have technology requirements by the client. Thus, the approach would have less applicability:

> *"I found the decision support of mitigation strategies fantastic, but the choice of technologies is more complicated. Projects are often required to use certain technologies as a customer requirement.".* (E05, Technical Leader)

One suggestion pointed out by a participant is to use the approach to maintain a list of the main risk mitigation strategies. As the objective of the approach is also to assist risk monitoring, it is expected that the mitigation strategies are always updated:

> *"My suggestion is that the approach is always updated with lessons learned from the main mitigation strategies. A kind of forum that is having the processing under the network could facilitate the teams' use.".* (E05, Technical Leader)

Finally, a participant believes that maintaining such a network is too expensive in practice because, in software projects, the risks change frequently.

> *"I believe that the approach would not have a long life in practice. The risks are volatile events. Despite sharing common characteristics, the projects have their particularities. And it is in these more specific things that the most significant risks lie. ".* (E08, Technical Leader)

However, the proposed approach contains different fragments to facilitate the evolution of the network with the particularities of the organizations' projects.

4.3 Threats To Validity

We classified the threats to validity as internal, construct, external, and reliability [31]. The two experts consulted to define the simulated scenarios needed to recall current and past risks in software projects about internal threats. They might have forgotten details and, consequently, built low-quality models. To minimize this threat, we paused whenever necessary and used knowledge elicitation techniques such as asking specific examples of each described scenario. Further, the researchers were available to help them adapt to the BN. Concerning external threats, we validated with professionals from only one organization. Therefore, the outcomes of the study can not be statistically generalized. We intend to conduct a case study with projects from different technological innovation organizations to address this threat.

We used TAM with open-ended questions to collect data from the subjects to minimize construct threats. Finally, we minimized reliability threats by having the data collected reviewed by the study's subjects. Each information added to the Bayesian network, we discussed with the other specialists to avoid bias.

5 FINAL REMARKS

This paper presents an approach based on Bayesian networks to support managing risk focused on technological risks of software projects. To validate our approach, we used six simulated scenarios and a Focus Group with practitioners. The results indicated that the approach supports the different risk management activities, mainly supporting decision-making regarding technology adoption and risk mitigation strategies. Besides, we concluded that most professionals who participated in the study found the approach useful and easy to use.

The Bayesian network is in an early version. In the future, we will use optimization algorithms to populate the probability tables. As a limitation, we have only performed static validation with a few practitioners. Thus, we cannot generalize our findings. We plan to conduct dynamic validations through case studies with real-world projects from multiple companies in the future. Therefore, our next steps with this work include: (a) to assess the approach to validate its feasibility when employed the managers and/or developers; (b) to assess the approach to other software project contexts, (c) the development of a tool the recommendation of risks and strategies.

REFERENCES

[1] Ieva Ancveire, Ilze Gailite, and Gailite. 2015. Software Delivery Risk Management: Application of Bayesian Networks in Agile Software Development. *Information Technology and Management Science* 18, 1 (2015), 62–69.

[2] Henri Barki, Suzanne Rivard, and Jean Talbot. 1993. Toward an assessment of software development risk. *Journal of management information systems* (1993).

[3] Irad Ben-Gal, Ayala Shani, André Gohr, Jan Grau, Sigal Arviv, Armin Shmilovici, and Posch. 2005. Identification of transcription factor binding sites with variable-order Bayesian networks. *Bioinformatics* (2005).

[4] Irad Ben-Gal, Ayala Shani, André Gohr, Jan Grau, Sigal Arviv, Armin Shmilovici, Stefan Posch, and Ivo Grosse. 2005. Identification of transcription factor binding sites with variable-order Bayesian networks. *Bioinformatics* 21, 11 (2005).

[5] John Bowers and Alireza Khorakian. 2014. Integrating risk management in the innovation project. *European Journal of innovation management* (2014).

[6] Saad Yasser Chadli, Ali Idri, José Luis Fernández-Alemán, Joaquín Nicolás Ros, and Ambrosio Toval. 2016. Identifying risks of software project management in Global Software Development: An integrative framework. In *IEEE/ACS 13th International Conference of Computer Systems and Applications (AICCSA)*.

[7] Fred D Davis. 1989. Perceived usefulness, perceived ease of use, and user acceptance of information technology. *MIS quarterly* (1989), 319–340.

[8] Ellen A Drost et al. 2011. Validity and reliability in social science research. *Education Research and perspectives* 38, 1 (2011), 105.

[9] Norman Fenton, William Marsh, Martin Neil, Patrick Cates, Simon Forey, and Manesh Tailor. 2004. Making resource decisions for software projects. In *Proceedings. 26th International Conference on Software Engineering*. IEEE, 397–406.

[10] Norman Fenton and Martin Neil. 2018. *Risk assessment and decision analysis with Bayesian networks*. Crc Press.

[11] Nir Friedman and Daphne Koller. 2003. Being Bayesian about network structure. A Bayesian approach to structure discovery in Bayesian networks. *Machine learning* 50, 1 (2003), 95–125.

[12] Barney G Glaser and Anselm L Strauss. 2017. *Discovery of grounded theory: Strategies for qualitative research*. Routledge.

[13] Tony Gorschek, Per Garre, Stig Larsson, and Claes Wohlin. 2006. A model for technology transfer in practice. *IEEE software* 23, 6 (2006), 88–95.

[14] Kim Heldman. 2010. *Project manager's spotlight on risk management*. John Wiley & Sons.

[15] David Hinde. 2018. *PRINCE2 Study Guide: 2017 Update*. John Wiley & Sons.

[16] ISO Central Secretary. 2018. *ISO 31000: risk management–Guidelines*. Standard. International Organization for Standardization, Geneva, CH.

[17] Ankur Joshi, Saket Kale, Satish Chandel, and D Kumar Pal. 2015. Likert scale: Explored and explained. *Current Journal of Applied Science and Technology* (2015).

[18] Vahid Khodakarami and Abdollah Abdi. 2014. Project cost risk analysis: A Bayesian networks approach for modeling dependencies between cost items. *International Journal of Project Management* 32, 7 (2014), 1233–1245.

[19] Jyrki Kontio, Johanna Bragge, and Laura Lehtola. 2008. The focus group method as an empirical tool in software engineering. In *Guide to advanced empirical software engineering*. Springer, 93–116.

[20] Chandan Kumar and Dilip Kumar Yadav. 2017. Software defects estimation using metrics of early phases of software development life cycle. *International Journal of System Assurance Engineering and Management* 8, 4 (2017), 2109–2117.

[21] Emilia Mendes. 2014. Expert-Based Knowledge Engineering of Bayesian Networks. In *Practitioner's Knowledge Representation*. Springer, 73–105.

[22] Emilia Mendes. 2014. *Practitioner's knowledge representation: a pathway to improve software effort estimation*. Springer Science & Business.

[23] Jislane SS Menezes, Danilo GA Ramos, and Michel S Soares. [n.d.]. On Criteria to Choose a Content Management System: A Technology Acceptance Model Approach. ([n. d.]).

[24] Cinzia Muriana and Giovanni Vizzini. 2017. Project risk management: A deterministic quantitative technique for assessment and mitigation. *International Journal of Project Management* 35, 3 (2017), 320–340.

[25] Ngoc-Tuan Nguyen, Quyet-Thang Huynh, and Thi-Huong-Giang Vu. 2018. A Bayesian Critical Path Method for Managing Common Risks in Software Project Scheduling. In *Proceedings of the Ninth International Symposium on Information and Communication Technology*. 382–388.

[26] V Nikolova, Ju Kuporov, and G Rodionov. 2015. Risk management of innovation projects in the context of globalization. *International Journal of Economics and Financial Issues* 5, 3S (2015).

[27] Mirko Perkusich, Lenardo Chaves e Silva, Alexandre Costa, Felipe Ramos, Renata Saraiva, Arthur Freire, Ednaldo Dilorenzo, Emanuel Dantas, Danilo Santos, Kyller Gorgônio, et al. 2020. Intelligent software engineering in the context of agile software development: A systematic literature review. *Information and Software Technology* 119 (2020), 106241.

[28] Mirko Perkusich, Gustavo Soares, Hyggo Almeida, and Angelo Perkusich. 2015. A procedure to detect problems of processes in software development projects using Bayesian networks. *Expert Systems with Applications* 42, 1 (2015), 437–450.

[29] PMI. 2018. *A guide to the project management body of knowledge (PMBOK guide)*.

[30] Md Forhad Rabbi and Khan Olid Bin Mannan. 2008. A review of software risk management for selection of best tools and techniques. In *2008 Ninth ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing*. IEEE, 773–778.

[31] Per Runeson and Martin Höst. 2009. Guidelines for conducting and reporting case study research in software engineering. *Empirical software engineering* 14, 2 (2009), 131–164.

[32] Robert S Russell and Bernard W Taylor-Iii. 2008. *Operations management along the supply chain*. John Wiley & Sons.

[33] Renata M Saraiva, Mirko Perkusich, Hyggo O Almeida, and Angelo Perkusich. 2017. A Process to Calculate the Uncertainty of Software Metrics-based Models Using Bayesian Networks.. In *SEKE*. 467–472.

[34] Ashish B Sasankar and Vinay Chavan. 2011. SWOT analysis of software development process models. *International Journal of Computer Science Issues (IJCSI)* 8, 5 (2011).

[35] Forrest Shull, Janice Singer, and Dag IK Sjøberg. 2007. *Guide to advanced empirical software engineering*. Springer.

[36] Jeff Sutherland and Ken Schwaber. 2013. The Scrum Guide. http://www.scrumguides.org/docs/scrumguide/v1/Scrum-Guide-US.pdf. Acessado em: 01-06-2020.

[37] June M Verner, O Pearl Brereton, Barbara A Kitchenham, Mahmood Turner, and Mahmood Niazi. 2014. Risks and risk mitigation in global software development: A tertiary study. *Information and Software Technology* (2014).

[38] Linda G Wallace and Steven D Sheetz. 2014. The adoption of software measures: A technology acceptance model (TAM) perspective. *Information & Management* 51, 2 (2014).