

Evaluating a Bayesian Network to Predict Customer Satisfaction in Scrum Software Development Projects: An Empirical Study with One Company

Mirko Perkusich

Intelligent Software Engineering Group
mirko@virtus.ufcg.edu.br

Gleyser Guimaraes

Federal University of Campina Grande
Intelligent Software Engineering Group
gleyser@copin.ufcg.edu.br

Kyller Gorgonio

Federal University of Campina Grande
Intelligent Software Engineering Group
kyller@copin.ufcg.edu.br

Hyggo Almeida

Federal University of Campina Grande
Intelligent Software Engineering Group
hyggo@dsc.ufcg.edu.br

Angelo Perkusich

Federal University of Campina Grande
Intelligent Software Engineering Group
perkusich@dee.ufcg.edu.br

Abstract—Using knowledge-based systems for helping agile teams to improve their performance is not a fact in the industry. In previous work, we have presented Kaizen, a knowledge-based Bayesian network for assisting Scrum teams in diagnosing their value stream in light of the predicted *Customer Satisfaction* and, consequently, improve their performance. This study assesses Kaizen’s accuracy to predict *Customer Satisfaction* using real-world data. We adopted Kaizen for one software development company and collected data from 18 projects using an online questionnaire. We collected two types of data: inputs for Kaizen and the expected *Customer satisfaction*. We used the first type of collected data as inputs for Kaizen to calculate the predicted *Customer satisfaction*. Then, we assessed Kaizen’s accuracy by comparing the predicted (i.e., calculated) and expected (i.e., collected) *Customer satisfaction* using face value and the average Brier score. Considering the face value, Kaizen predicted *Customer Satisfaction* correctly for 14 out of the 18 projects. The average Brier Score was 0.16. The model predicts, with satisfactory accuracy, the *Customer Satisfaction* and systemizes the process for Scrum teams to self-diagnose, enabling for causal analysis and supporting their continuous improvement.

Index Terms—Agile Software Development; Questionnaire Survey; Bayesian Network.

I. INTRODUCTION

Using knowledge-based systems for helping agile teams to improve their performance is not a fact in the industry. In previous work, we have presented a knowledge-based Bayesian network to assist Scrum teams in diagnosing their value stream [14], [13], [12], and, consequently, improve their performance. From here on, we refer to such a Bayesian network as Kaizen. Kaizen has as the “target variable” the *Customer Satisfaction*. Thus, Kaizen assists Scrum teams to improve their efficiency in light of the predicted customer satisfaction. In previous studies, we have validated it with simulated scenarios (i.e., model walkthrough) [13] and by analyzing its ability to identify 14 Scrum anti-patterns [12] described in the literature [4].

Further, we have evaluated its practical utility by using it in two projects from one software company [14]. However, such a case study focused on Kaizen’s ability to support decision-making, not on its predictive accuracy. Thus, at this point, its

accuracy for predicting *Customer Satisfaction* has not been assessed with real-world data.

This study addresses this gap by adopting Kaizen for one software development company. The adoption process included complementing it with a questionnaire to collect projects’ data from their Scrum Masters and customer representatives. As a result, we collected data from 18 projects. We collected two data types for each project: inputs for Kaizen and the expected *Customer satisfaction*. We used the first type (inputs for Kaizen) to calculate the predicted *Customer satisfaction*. Then, we assessed Kaizen’s accuracy by comparing the predicted and expected *Customer satisfaction* using face value and the average Brier score. This paper reports the employed methodology and our results. Further, it shows an example of how Scrum teams can adopt Kaizen.

This paper is organized as follows. Section II presents the necessary background on the Bayesian network and our reasoning for selecting it as the modeling tool for Kaizen. Section III presents an overview of Kaizen. Section IV presents the employed research methodology. Section V discusses our results, their implications, and this study’s threats to validity. Finally, Section VI presents our final remarks and directions to future work.

II. BAYESIAN NETWORKS

Bayesian networks are probabilistic graph models that represent knowledge about an uncertain domain. A Bayesian network, B , is a directed acyclic graph representing a joint probability distribution over a set of random variables V . The network is defined by the pair $B = \{G, \Theta\}$. G is the directed acyclic graph in which the nodes X_1, \dots, X_n represent random variables, and the arcs represent the direct dependencies between these variables. Therefore, a Bayesian network consists of two parts: the directed acyclic graph and the probability distributions. They can be constructed based on domain experts’ knowledge, statistical techniques, or both (i.e., hybrid). Since the data needed to construct Kaizen was not available publicly, we relied on domain experts’ knowledge for both parts of our model [14], [13], [12].

Using Bayesian networks for supporting decision-making in software engineering has the following advantages.

- In a single model, it is possible to maintain observations, statistical distributions, prior assumptions, and expert judgment.
- Allows the encoding of causal relationships among variables for prognosis.
- Enables human diagnosis due to its explanatory nature (in contrast with opaque neural networks that are hard to analyze and test), reducing its risk of adoption.
- Handles missing data [5].

III. KAIZEN OVERVIEW

This section presents an overview of Kaizen. Its purpose is to present a high-level view of its structure. Notice that the methods applied to construct it and its details are presented elsewhere [14], [13], [12].

The reasoning behind the model is based on the Scrum Guide's vision that Scrum is a means for transforming ideas into value [18]. Further, according to the Evidence-Based Management Guide, there are four dimensions four Key Value Areas: Current Value, Time to Market, Unrealized Value, and Ability to Innovate. Kaizen focuses on Current Value; more specifically, it assesses Scrum's adoption in light of the value perceived by customers or the product's users. Its main requirement was to model Scrum's product delivery mechanisms, from conceptualization to release, that affect the probability of delivering products that satisfy customers.

Given this, Kaizen's "target variable" is *Customer satisfaction*. Kaizen's goal is to be used by Scrum teams to improve their product delivery process, having the predicted *Customer satisfaction* levels as the reference. It enables Scrum teams to predict *Customer satisfaction* given their current practices and diagnose themselves, enabling early detection of undesired deviations. Thus, Kaizen's variables model the team's value stream, in other words, its procedures to deliver value to customers.

The process of building Kaizen was top-down, decomposing *Customer satisfaction* into attributes that the team could observe. Since, in Scrum, the ideas are stored in the *Product Backlog*, and the value is delivered through the release of *Increment*, these are the variables to "predict" *Customer satisfaction*.

Figure 1 presents the Kaizen's graph. Figure 1, a white node represents a node not directly observable; an orange node represents a predictor; and a green one, an indicator. A predictor is a factor that can be observed to predict the value of its child node. An indicator is a factor that can be observed to measure the current value of its parent node. The difference between a predictor and an indicator is the time of measurement. For instance, consider *Sprint Planning*. Going into such an event, knowing the *Projected Capacity* and *Past Performance* increases the chances of having good *Sprint Planning*, and its success can be measured by the quality of the resulting artifact: the *Sprint Backlog*.

Notice that Kaizen does not include two of Scrum's key aspects: *Sprint Retrospective* and *Scrum Master*. The reason is that Kaizen was built considering the Scrum Master's perspective and that it should be used during *Sprint Retrospective*

events. Further, it is worthy of mentioning that Kaizen we built by extracting knowledge available on the literature and tacit knowledge from industry experts following the Expert-Based Knowledge Engineering of Bayesian Networks methodology [9]. A file for executing Kaizen in AgenaRisk¹, the Bayesian network inference system in which Kaizen's model executes, is made available online².

IV. RESEARCH METHODOLOGY

This study's goal was to analyze if Kaizen measures what it is supposed to measure (i.e., customer satisfaction level). Further, it is essential to mention that our study's results are limited to the company's context from which we collected data. Given this, we explored the following research question.

RQ What is Kaizen's accuracy for predicting the customer satisfaction level's for the company under study?

As measures for predictive accuracy, we used face value (i.e., outcome adequacy [10]) and the Brier score. Face value measures if the customer satisfaction level calculated by Kaizen provides the highest probability to a state that matches the value expected, given the data collected from the company. For instance, if the expected value for customer satisfaction is *Low*, and the predicted value for customer satisfaction is *Low* = 0.25, *Moderate* = 0.60, and *High* = 0.15, FV is *FALSE* (or 0).

However, using the face value is not enough because one would agree that predicting the correct value with 91% is better than, for instance, 50%. Thus, we also used the Brier score, which is the mean squared difference between the predicted probability and observed outcome. A Brier score of 0 indicates a perfect model, and 1 is the worst score possible. We calculated the Brier score for each project and used the average to assess Kaizen's predictive accuracy.

Next, Section IV-A contextualizes the company for which we collected data. Section IV-B describes the procedures employed to collect data.

A. Context Characterization

This section presents information to characterize the company under study following the context facets described by Petersen and Wohlin [15]: *product, process, practices and techniques, people, organization and market*.

The company under study is a research, development, and innovation center. It supplies services to industry partners in the context of several technological domains, including artificial intelligence, Web systems, and cyber-physical systems. It manages its initiatives through projects lasting around ten and eighteen months.

In general, the projects are executed using agile approaches such as Scrum or Kanban. The development practices and tools follow the guidelines defined by the organization but are adapted given the projects' needs (e.g., programming language and type of system).

¹<https://www.agenarisk.com/>

²<https://doi.org/10.5281/zenodo.4604017>

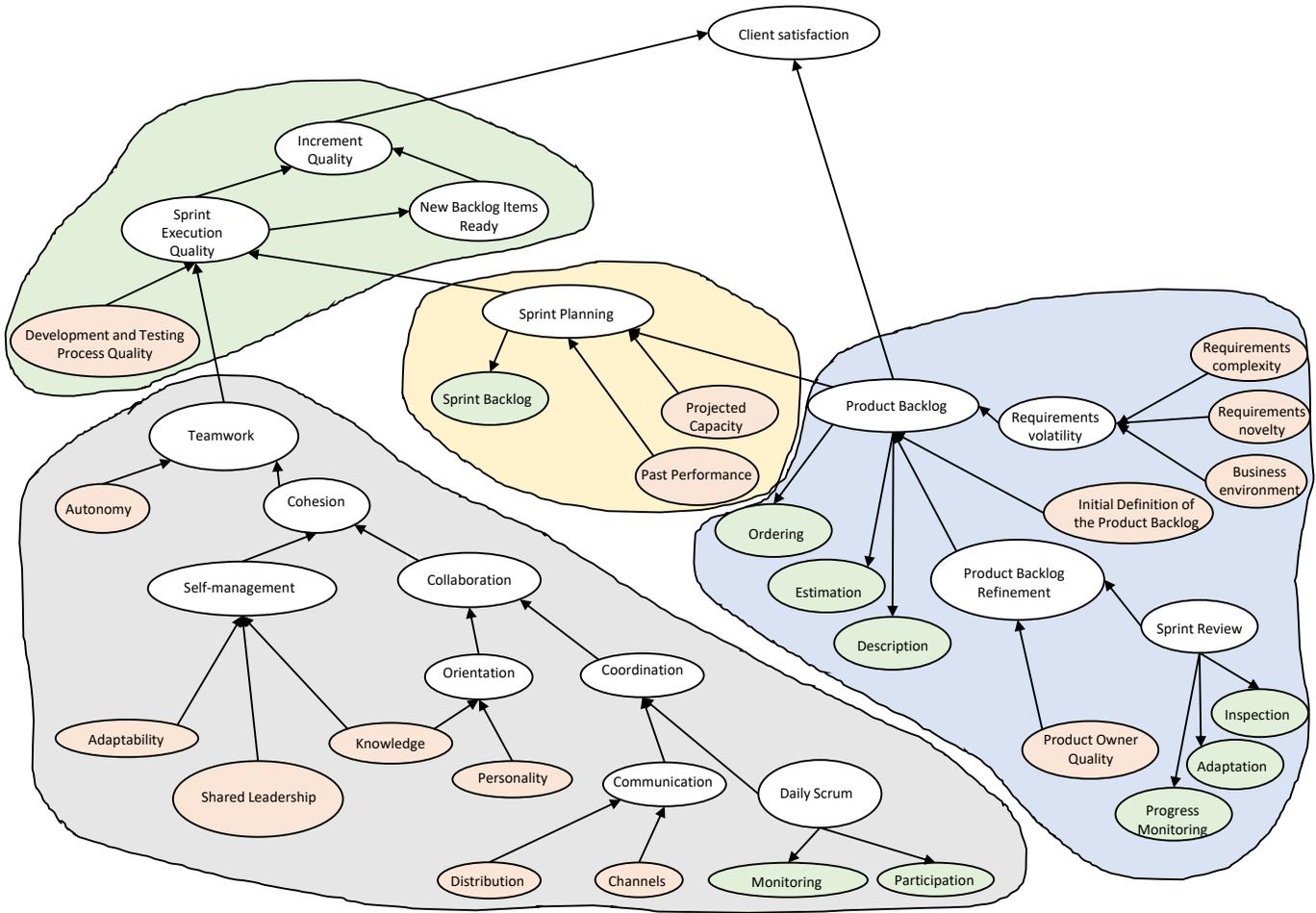


Fig. 1. Kaizen's graph.

B. Data collection Procedure

Kaizen does not claim external validity [12] and adopting it requires the users to adapt it to their context for the same reasons that companies tailor methods (including Scrum) to fit their values, culture, reality, needs, and strategies [7], [1].

One of the steps necessary to adopt Kaizen is to map the variables of interest into data sources so that the users do not need to input data directly into the Bayesian network's variables. Such data sources might be a questionnaire or a tool such as Sonarqube [17]. For instance, the variable *Development and Testing Process Quality* can be mapped to one or more questions on a questionnaire or use the technical debt indicators provided by tools such as Sonarqube.

For our study, our variables of interest are the “target variable”, *Customer support* and the leaf nodes (i.e., the ones marked with the color orange in Figure 1), which we mapped into two questionnaires. One questionnaire was aimed only to collect data for *Customer support*, and answered by the projects' customer representatives. The projects' Scrum Masters answered the other questionnaire to collect data for the remaining variables previously mentioned. For each variable of interest, we defined one question. The questionnaire made available for the Scrum Master also data about his/her profile

and the project's context. Section V-A details the process employed to recruit participants.

We designed and executed our study following the instructions presented by Runeson and Höst [16] and Molléri et al. [11]. Before sending the questionnaire to the customer representatives and Scrum Masters, we evaluated it through a pre-test with eleven participants, including people from the target population and colleagues with Scrum experience. The pre-test's goal was to validate the questionnaire's understandability. To execute the pre-test, each person responded to the questionnaires separately. Afterward, we executed a focus group to define the questionnaire's final version.

As a result of the pre-test, for some questions, we created a glossary to support the respondents of the questionnaire and avoid misunderstandings regarding the used terminology. The questions followed a pattern. For some questions, we directly asked the user to assess each of the target variables' current state using a 3-point Likert scale. We used such a scale because Kaizen's variables use ordinal variables with three states. For instance, for the variable “Complexity”, the question was: “How complex are the requirements?”, with the alternatives “Low”, “Moderate”, and “High”. For others, we presented a sentence and asked their level of agreement (i.e., *Disagree*,

Neutral, Agree). There were also three cases in which we customized the possible answers to ease their understandability; an example is shown in Table I. The Supplementary Material, available online³, presents all questionnaires in their original language (i.e., Portuguese) and their translations into English.

The questionnaires were made available online using a Google Form⁴ and sent by the respondents through e-mail. We gave the respondents one week to answer the questionnaires, and the first author remained available if they had any doubts. All the participants that had agreed to participate in the study answered the questionnaires.

Once the answers were collected, we inputted the collected data into a Google Spreadsheet. Then, manually, we inputted the collected data into Kaizen’s model’s leaf nodes using AgenaRisk. AgenaRisk has a feature named “scenario”, which enables the user to tag a set of inputs for the model’s nodes (i.e., AgenaRisk calls each input into a node as an “evidence”). Thus, we created one “scenario” for each project, containing the set of “evidence” for each of the model’s leaf nodes, which followed directly from the questionnaire’s answers. For instance, the node *Monitoring* was mapped the question *The Developers are monitoring tasks properly.*. Thus, if for a given project, the answer to this question was *Disagree*, we inputted the “evidence” *Low* into the node *Monitoring*. We used the answers related to the variable *Customer Satisfaction* as the reference for calculating Kaizen’s predictive accuracy, discussed in Section V-B.

V. RESULTS AND DISCUSSION

This section describes the units of analysis (i.e., projects) and subjects (i.e., people who answered the questionnaires) (see Section V-A), discusses the results of Kaizen’s predictive accuracy (see Section V-B), presents its implications (see Section V-C), and the study’s threats to validity (see Section V-D).

A. Units of Analysis and Subjects

At the time of the data collection, the company had more than 30 projects being executed. Our goal was to collect data from as most projects as possible that used Scrum and focused on delivering software products or prototypes. The recruiting process started by having the first author directly inviting Scrum Masters and customer representatives from projects with the desired characteristics to participate in this study by explaining its goals and how it could benefit them. The first author explained that the data would be kept anonymous. Further, the first author explained that, after the data analysis was done, the original data, which mapped the project identification with the collected, would be destroyed. Finally, they were asked to sign an informed consent. As a result of the recruiting process, the Scrum Masters and customer representative of 18 projects agreed to participate.

For each project, we collected data regarding the most recently finished Sprint. The questionnaire respondents had an average of 9.8 years of experience in software projects; six years with Scrum. The projects, on average, were composed of 6.7 members, including developers and testers. We also

³<https://doi.org/10.5281/zenodo.4604001>

⁴<https://forms.gle/i9vtsdNJac8vLjfQA>

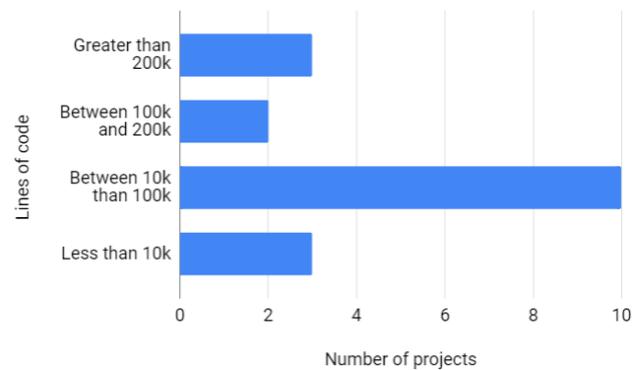


Fig. 2. Distribution of code size for the projects.

collected data regarding the progress of each project, in which the progress was calculated by dividing the number of executed sprints by the number of planned sprints.

Out of the 18 projects, ten were already finished, three with more than 85% of progress, one with 58%, and the remaining one with approximately 50% of progress. To better contextualize the projects, we collected data regarding the size of their codebase, measured in lines of code (see Figure 2).

B. Predictive Accuracy

Table II shows the answers of all respondents given the variable “Customer satisfaction” and the calculated results of the model. The values in bold represent the states with a higher calculated probability for the given project. In Table II, we used a color code to interpret the data given the face value by marking with the color green the correct predictions and red the incorrect ones.

As a result, for the face value, the models’ predictions were correct for 14 of the 18 projects, resulting in 78% of accuracy, which is a reasonable accuracy. The average Brier score was 0.16. Since the average Brier score was close to 0, it means that Kaizen’s predictions were good [3].

The process to adopt Kaizen in this study was not ideal because of the participants’ availability. For instance, we believe that using multiple questions or objective measures as data sources for the variable factors could have improved the answers’ internal consistency and reduced bias. Given this, we believe that Kaizen’s predictions were satisfactory.

We interviewed the respondents of the projects for which the predictions were incorrect, given the face value (projects 2, 8, 12, and 13). For all cases, they argued that their projects were in the product discovery phase and that deliverables were proofs of concepts or MVPs. Thus, the quality rigor was lower than usual. We believe that the incorrect predictions were caused by the lack of proper tailoring of Kaizen, given the company’s context.

A limitation of our analysis for Kaizen’s predictive accuracy is that out of the 18 projects, 16 had an expected *Customer Satisfaction* value of *High*. Ideally, from our study’s perspective, we would have a more uniform distribution of the expected values to test Kaizen with diverse situations. However, when collecting data in the real world, such inconsistencies with the ideal scenarios might happen.

TABLE I
QUESTION AND ANSWERS FOR THE FACTORS *Initial definition of the Product Backlog, Detailed, AND Ordered.*

Factor	Question	Answers
Initial definition of the Product Backlog	What is the level of definition of the Product Backlog at the beginning of the release cycle?	Low - None or a few of item defined Moderate - Main items defined High - Complete definition of all the items
Detailed	Do you agree with the following statement: "The functional and non-functional requirements contain a set of acceptance criteria and are presented in different levels of details given their position on the product backlog (top items are more detailed)"?	Low - Completely disagree Moderate - Neutral High - Completely agree
Ordered	Is the Product Backlog continuously ordered considering attributes such as business value, technical dependencies, effort, cost, and risk?	Low - It is not ordered Moderate - Its ordering is continuously refined using some of the relevant attributes for the project. High - Its ordering is continuously refined using all the relevant attributes for the project.

TABLE II
EXPECTED AND CALCULATED AND EXPECTED VALUES FOR THE VARIABLE *Customer satisfaction.*

Project ID	Expected	Calculated Probabilities		
		Low	Moderate	High
1	High	.130	.429	.441
2	High	.212	.459	.329
3	High	.137	.426	.441
4	High	.121	.405	.474
5	Moderate	.296	.489	.215
6	High	.107	.408	.484
7	High	.129	.411	.460
8	High	.542	.380	.080
9	High	.111	.379	.510
10	High	.120	.417	.463
11	High	.049	.260	.691
12	High	.387	.446	.167
13	High	.388	.455	.156
14	High	.030	.208	.762
15	High	.031	.220	.749
16	High	.023	.190	.787
17	High	.077	.326	.597
18	Moderate	.272	.479	.249

Given our results, one could argue that if we had just guessed that *Customer Satisfaction* was *High*, we would have better accuracy, considering face value (i.e., 16 out of 18). Even though this is true, we believe that this does not mean that such a guess would be better than Kaizen's predictions in general. In previous studies [13], we have demonstrated Kaizen's capability of accurately predicting positive and not positive *Customer Satisfaction* given diverse situations. Clearly, only guessing a *High Customer Satisfaction* is not enough for diverse scenarios. Thus, such an argument is supported only by the coincidence of the data collected in this study being highly skewed toward *High Customer Satisfaction*. Consequently, we believe that the results mentioned above for Kaizen's predictive accuracy is a positive indicator of its reliability, complementing our past results [14], [13], [12].

C. Implications for research and practice

This section discusses our study's implications for research and practice. For research, our results reinforce Kaizen's construct validity by complementing our past studies [14], [13], [12] and presenting its predictive accuracy in an industrial case. On this note, it is worthy of mentioning that Kaizen is mainly based on subjective measures. Thus, our results open

the opportunity for research to investigate valid quantitative measures that would reduce the subjectivity and effort of using it. We believe that a promising way forward is to analyze how to connect its nodes into data collected by tools used by agile teams, including Sonarqube and project tracking software. In other words, Kaizen can be used as a reference to kickoff software measurement programs for Scrum teams focusing on optimizing customer satisfaction.

As previously discussed, Kaizen only focuses on one of the four dimensions of Evidence-Based Management's Key-Value Areas. Since we have had positive experiences with Kaizen using customer satisfaction as the reference for optimizing the Scrum team's performance, we believe that opportunities arise in exploring other Key-Value Areas.

Further, we made available Kaizen's model, and we encourage the research community to investigate its use within a broader context. In terms of industry adoption, we believe that two factors are hindering it. First, there is a need for detailed adoption guidelines and use cases of how Scrum teams can use Kaizen to improve their workstreams. Second, Kaizen relies on AgenaRisk. Thus, there is a need to develop an independent tool that connects with diverse data sources, including the tools used by Scrum teams.

We believe that Kaizen could help immature teams to adopt Scrum by assisting them in detecting deviations or anti-patterns. However, Scrum is a framework, and it does not prescribe tactics since they are context-sensitive. Thus, Kaizen does not inform users about how each of the model's factors should be fulfilled. Even though this characteristic promotes adoption flexibility, it also hinders its usefulness since it does not guide the users on how to solve the detected problems, for instance, by suggesting using Story Points to estimate Product Backlog items. A line of research is to complement Kaizen's current version to model such tactics using the concept of causal intervention [2]. Such a model could be the basis for developing information retrieval or recommender systems to help teams define process improvement action points.

Concerning the implications for industrial practice, we have presented in this paper results the validate Kaizen, which is a tool that can help them, at the project level, by assisting the team on risk management, and also at the organizational level by assisting on knowledge-based process improvement and deployment of a software measurement program.

D. Threats to validity

This section discusses this study’s threats to validity following the classification presented by [16].

Construct validity: we used a 3-point ordinal scale for the factors, which are subjective by nature and susceptible to cognitive bias, the illusion of communication, and invalid inferences [6]. Despite this, similar approaches have been used in other studies with positive results [5], [8], [14], [10].

Reliability: since the respondents of the questionnaire are providing data for their teams, the data might be biased. We minimized this potential effect on our results by guaranteeing the participants that the data would remain anonymous and that their performance was not at stake. Further, we pre-tested the questionnaire to avoid the risk of having respondents misinterpret the questions.

Internal validity: we used a selective sampling approach, in which we collected data from one software development company. Further, the participants from the 18 volunteered to participate in the study during our recruiting process within the company. Finally, we gave the participants one week to answer the questionnaire on their best availability to avoid answering it in a rush.

External validity: the questionnaire defined to operationalize the usage of the causal model is company-specific. Therefore, the results are bound by the company’s context in which the study was performed. Despite this, they might apply to other companies that use Scrum to manage software development projects, even though it is expected that companies that adopt Kaizen would need to tailor it to their context.

VI. CONCLUSIONS

This paper presented the results of an empirical study to evaluate Kaizen [14], [13], [12] with data collected from 18 projects from one software development company.

A limitation of this study is that out of the 18 projects for which we collected data, 16 expected a *High* value for our “target variable” (i.e., *Customer satisfaction*). Ideally, we would have collected data better distributed for the remaining possible states. However, we believe this is a natural risk of collected real-world data. We believe that this aspect does not hinder Kaizen’s evaluation because it has been validated with other datasets previously [14], [13], [12].

Further, this paper presented how to adopt Kaizen by defining a questionnaire as its data source. However, in future work, we will detail an adoption guideline for Kaizen and present use cases of how it can support Scrum teams to identify process improvement opportunities and define action points.

Kaizen’s industry adoption potential is currently limited from an operational perspective because it relies on Age-naRisk, making it challenging to input data into the model since it is a manual endeavor. Currently, we are working on a tool that implements the required algorithms for Bayesian network inference and enables us to connect external data sources to variables in the model using REST-based endpoints. This feature enables, for instance, connecting the variable *Increment* with metrics collected on SonarQube [17]. Furthermore, it enables the registration of corrective (or preventive) actions coupled with the factors. As a result, a knowledge base could

be created for the organization and used by project managers for data-driven risk management through information retrieval, case-based reasoning, or recommender systems. Additionally, we intend to explore evolving the model through the use of dynamic Bayesian networks to handle the iterative nature of Scrum.

REFERENCES

- [1] A. S. Campanelli and F. S. Parreiras. Agile methods tailoring—a systematic literature review. *Journal of Systems and Software*, 110:85–100, 2015.
- [2] A. C. Constantinou, N. Fenton, W. Marsh, and L. Radlinski. From complex questionnaire and interviewing data to intelligent bayesian network models for medical decision support. *Artificial intelligence in medicine*, 67:75–93, 2016.
- [3] T. Corke. Is that a good probability score - the brier score edition. <http://www.matterofstats.com/mafl-stats-journal/2013/12/22/is-that-a-good-probability-score-the-brier-score-edition.html>, 2013. Accessed in: 03-10-2021.
- [4] V.-P. Eloranta, K. Koskimies, and T. Mikkonen. Exploring scrumbut—an empirical study of scrum anti-patterns. *Information and Software Technology*, 74:194 – 203, 2016.
- [5] N. Fenton, M. Neil, W. Marsh, P. Hearty, Ł. Radliński, and P. Krause. On the effectiveness of early life cycle defect prediction with bayesian nets. *Empirical Software Engineering*, 13(5):499, 2008.
- [6] D. Hubbard and D. Evans. Problems with scoring methods and ordinal scales in risk assessment. *IBM Journal of Research and Development*, 54(3):2–1, 2010.
- [7] Z. Masood, R. Hoda, and K. Blincoe. Real world scrum a grounded theory of variations in practice. *IEEE Transactions on Software Engineering*, 2020.
- [8] E. Mendes. Using knowledge elicitation to improve web effort estimation: lessons from six industrial case studies. In *Proceedings of the 34th International Conference on Software Engineering*, pages 1112–1121. IEEE Press, 2012.
- [9] E. Mendes. *Practitioner’s knowledge representation: a pathway to improve software effort estimation*. Springer Science & Business, 2014.
- [10] E. Mendes, M. Perkusich, V. Freitas, and J. Nunes. Using bayesian network to estimate the value of decisions within the context of value-based software engineering. In *Proceedings of the 22nd International Conference on Evaluation and Assessment in Software Engineering 2018*, pages 90–100. ACM, 2018.
- [11] J. S. Molléri, K. Petersen, and E. Mendes. An empirically evaluated checklist for surveys in software engineering. *Information and Software Technology*, 119:106240, 2020.
- [12] M. Perkusich, K. Gorgonio, H. Almeida, and A. Perkusich. A framework to build bayesian networks to assess scrum-based software development methods. In *28th International Conference on Software Engineering and Knowledge Engineering, SEKE 2017*, pages 1 – 7, Pittsburgh, USA, 2017.
- [13] M. Perkusich, K. C. Gorgônio, H. Almeida, and A. Perkusich. Assisting the continuous improvement of scrum projects using metrics and bayesian networks. *Journal of Software: Evolution and Process*, 29(6):e1835, 2017.
- [14] M. Perkusich, G. Soares, H. Almeida, and A. Perkusich. A procedure to detect problems of processes in software development projects using bayesian networks. *Expert Systems with Applications*, 42(1):437–450, 2015.
- [15] K. Petersen and C. Wohlin. Context in industrial software engineering research. In *2009 3rd International Symposium on Empirical Software Engineering and Measurement*, pages 401–404. IEEE, 2009.
- [16] P. Runeson and M. Höst. Guidelines for conducting and reporting case study research in software engineering. *Empirical software engineering*, 14(2):131–164, 2009.
- [17] SonarQube. Sonarqube: Code quality and security. <https://www.sonarqube.org/>, 2019. Accessed in: 08-23-2019.
- [18] J. Sutherland and K. Schwaber. The 2020 scrum guide. <https://scrumguides.org/scrum-guide.html>, 2020. Accessed in: 03-10-2021.