ICIC Express Letters
Volume **15**, Number **8**, August **2021**

ICIC International ©2021 ISSN 1881-803X
pp. 897–905

BAYESIAN ANALYSIS IN PREDICTING THE SUCCESS RATE OF THE SCRUM-BASED SOFTWARE DEVELOPMENT PROJECT UNDER STOCHASTIC ENVIRONMENT

Tri Handhika[1], Yusuf Sutanto[2], Asep Juarna[3]
and Achmad Benny Mutiara[4,*]

[1]Centre for Computational Mathematics Studies
[2]Doctoral Program in Information Technology
[3]Centre of Study on Applied Combinatorics
[4]Faculty of Computer Science and Information Technology
Gunadarma University
Jl. Margonda Raya No. 100, Depok 16424, Indonesia
{ trihandika; ajuarna }@staff.gunadarma.ac.id; yusuf.sutanto@stie-aub.ac.id
*Corresponding author: amutiara@staff.gunadarma.ac.id

Received November 2020; accepted February 2021

Abstract. *The high risk involved in the Scrum-based software development project comes from the variety of uncertainties that exist in each of its components. Therefore, the success rate needs to be predicted as a basis for the Scrum team to formulate an appropriate management strategy. This stochastic problem is represented formally in the (non-parametric) Bayesian networks model. We then design several scenarios to the generated large-scale Scrum-based development projects with multiple stakeholders and multiple feature teams. We tried to simulate several variables used in this model by using (rank nodes)-based as well as (survey and weight functions)-based algorithms. The experimental results show that the proposed model is running well so that it can be an alternative for Scrum team in predicting the success rate of the Scrum-based software development project.*

Keywords: Software development, Scrum, Success rate, Bayesian networks model

1. Introduction.

There are still gaps on agile improvement solutions and their successful adoption in industry projects [1] such that it is necessary to make adjustments between research and practice [2], e.g., integration between behaviour-driven development and hardware/software codesign [3]. Several frameworks are known in the agile method, e.g., dynamic systems development method, feature-driven design, crystal, agile modelling, and Scrum [4]. In this study, we used Scrum as a framework in the software development project. There are some benefits of Scrum framework as mentioned in [5], such as delighted customers, improved return on investment, reduced costs, fast results, confidence to succeed in a complex world, and more joy.

Risk quantification is an important activity in the Scrum-based software development project management. Risk can be defined as an uncertain event or condition, that if it occurs, has a positive or negative effect on a project's objective [6]. Thus, we need to manage the risk related to the uncertainty of software development project. There are various difficulties during project execution related to the three main objectives, i.e., costs, deadline, and quality [7]. This problem needs to be represented in a mathematical model under stochastic environment, also known as a stochastic model.

Many studies introduced Bayesian analysis using Bayesian networks model in solving problems related to software development project. This model can be used to calculate the uncertainty of software metrics-based models [8]. Moreover, it can also be performed

DOI: 10.24507/icicel.15.08.897

to assist on the interpretation of software metrics such that they can be used to assess or predict the quality of software development project based on the high level of confidence [9], not based on the metrics thresholds as with fuzzy logic method [10].

In addition to software project effort [11] and duration estimation accuracy [12], success rate prediction is an other main issue in software development project. However, as mentioned earlier, the success of this software development project execution needs to fulfill three main objectives which are, in fact, mutually contradictory. Low costs strategies may have an impact on decreasing the quality of the developed software or even delaying project time schedule due to incompetent developers. Meanwhile, the strategy of shortening the project time schedule may have an impact on increasing costs due to overtime or also decreasing quality due to hasty work. Likewise, if the quality improvement strategy is adopted, it can lead to increased costs and delayed project time schedule. We can view these three main objectives as success metrics. In this study, we predict the success rate of the Scrum-based software development project by using non-parametric Bayesian networks model [13]. The term "non-parametric" refers to the method handling discrete-continuous variables which is later used in our problem.

To summarize, our contribution is to build a representative Bayesian network model of the Scrum framework for software development projects. This model can be used to predict the project's success rate. Moreover, the ongoing project performance can be monitored by using this model according to the forecasted data from real time documents.

The rest of the paper is organized as follows. In Section 2, we describe the Scrum framework for software development project. Section 3 generally describes the proposed non-parametric Bayesian networks model. Details of the conducted experiments and its computational results are discussed in Section 4. Finally, in Section 5 we present the conclusion and future works.

2. **The Scrum-Based Software Development Project.** Scrum as one of frameworks in agile methodologies [14] is based on iterative development which comprises three key roles (product owner, Scrum master, team), three artifacts/documents (product backlog, the sprint backlog, the sprint results), and three events/meetings (the sprint planning meeting, the daily Scrum meeting, the sprint review) [15] as illustrated in Figure 1. In the large-scale Scrum-based software development project, multiple teams have different focuses on different features. They are known as feature teams.

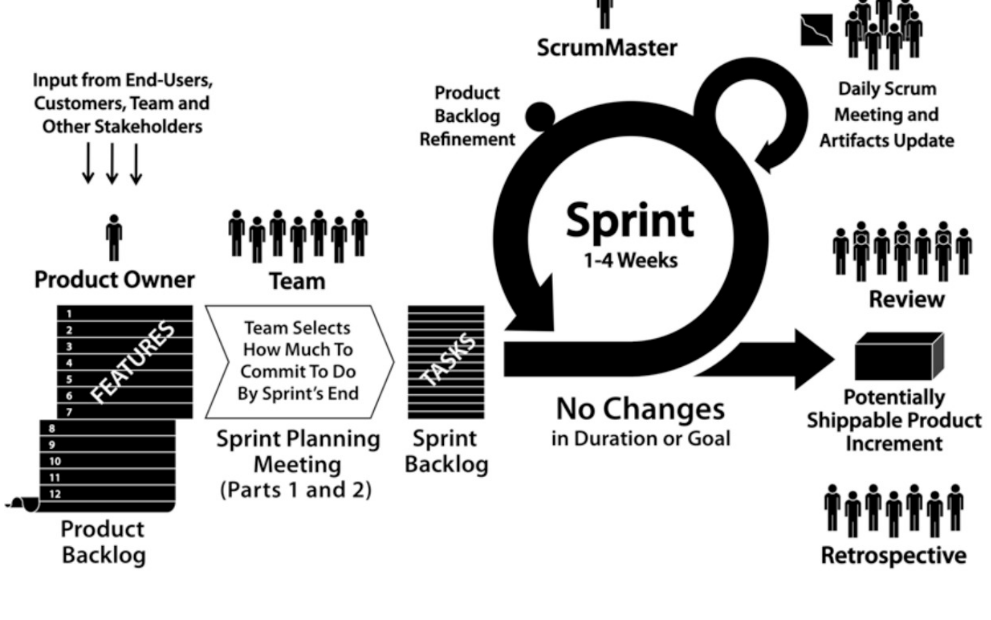
ScrumMaster
Input from End-Users, Customers, Team and Other Stakeholders
Product Backlog Refinement
Daily Scrum Meeting and Artifacts Update
Sprint
1-4 Weeks
Product Owner
Team
Review
FEATURES
1
2
3
4
5
6
7
8
9
10
11
12
Team Selects How Much To Commit To Do By Sprint's End
TASKS
Sprint Planning Meeting (Parts 1 and 2)
Sprint Backlog
No Changes
in Duration or Goal
Potentially Shippable Product Increment
Product Backlog
Retrospective

FIGURE 1. Scrum framework [16]

In the Scrum framework, the iteration is called as sprint. It is time-boxed (fixed deadline). The sprint starts with selecting the features/items/user stories in the product backlog sequentially based on a priority scale. This selection process is carried out together between product owner and the team, led by Scrum master, in the sprint planning meeting part one while part two focuses on detailing tasks of each selected feature. The detailed tasks which have been put in the sprint backlog are then distributed to the workers according to their respective feature teams and need to be completed within a maximum of four weeks. The progress of completing these tasks is reported every day in the daily Scrum meeting for further review when the related sprint ends. The sprint review and sprint retrospective involve inspect and adapt regarding the product and the process, respectively.

In the sprint review, we refer to the sprint results as a potentially shippable product increment, meaning that whatever the Scrum team (all of key roles) agreed to do is really done according to its agreed-upon definition of done as discussed at the beginning of the sprint planning meeting part one. This definition specifies the degree of confidence that the work completed is of good quality and is potentially shippable [5]. There are several criteria that can be used in identifying whether a software development project can be declared done as summarized in [17]. It should be noted that something which prevents the software from functioning as specified is called as defects. It is different with features and refactoring needs as discussed in [15]. In this study, we used one of techniques in conducting a sprint retrospective where the team labels each of the features for at least two questions, i.e., "What working well?" and "What could work better?" with either a caused by Scrum, an exposed by Scrum, or an unrelated to Scrum. See [16] for the detail of Scrum implementation in the software development project or see [18] for brief guide.

The success rate of software development project can be classified into five categories, i.e., (i) significant changes; (ii) over time/budget, (iii) on time, on budget, on scope; (iv) canceled; and (v) postponed [19]. Moreover, in this study, each of success metrics is divided into three classes where the costs consist of lower costs, at costs, and higher costs; the deadline consists of in time, on time, and over time; and the quality consists of aspirational, appropriate, and acceptable. Furthermore, there are three dimensions of the complexity in the software development project, namely requirements, technology, and people [20]. However, in this study, we only considered two dimensions (i.e., requirements and technology) which divide the complexity into four levels, i.e., simple, complicated, complex, and anarchy, as illustrated in Figure 2. We construct a non-parametric Bayesian networks model by the level of complexity, success metrics, and Scrum framework to

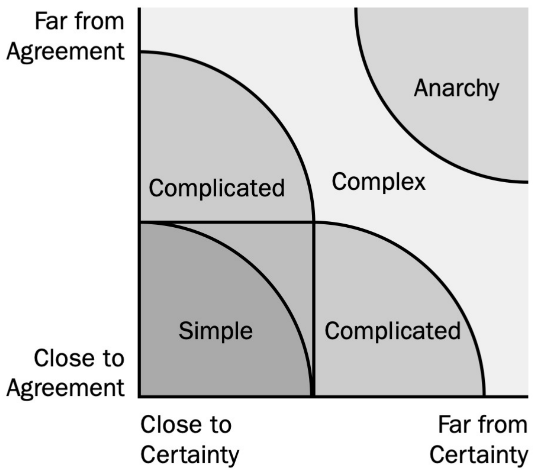

FIGURE 2. Complexity in the software development project based on requirements and technology [20]

represent the success rate of the Scrum-based software development project as discussed in Section 3.

3. The Proposed Non-Parametric Bayesian Networks Model.

In this study, we modify a framework designed by Perkusich et al. [21] to build a non-parametric Bayesian networks model in predicting success rate of the Scrum-based software development project rather than detecting the problems in the project as in [22]. We also defined several required artifacts and metrics as discussed in [23] to build our proposed non-parametric Bayesian network model based on the variables defined in Section 2.

Table 1 shows the operationalization of variables used in the model as described in Section 2. There are seven variables considered as independent variables, i.e., P, H, M, K, N, T_i, and R. As known in each of the artifacts, K, N, T_i and R are determinants in the sprint backlog while P, H, and M are determinants in the product backlog. For monitoring purposes, we used forecasted work remaining on the sprint backlog (X) and product backlog (Y), respectively, where X is a determinant of Y according to the Scrum framework. X is also a determinant for both D and V where they are determinants for Y. S is measured by three metrics, i.e., C, L, and Q, which are determined by X and Y. Based on this rule, we then construct a probabilistic graphical model [24] with mixed – discrete-continuous – variables [13], as known as non-parametric Bayesian networks model. Therefore, the success rate prediction problems in the Scrum-based software development project can be represented as a directed acyclic graph as illustrated in Figure 3 where discrete and continuous

Table 1. Operationalization of variables

No.	Symbol	Type of random variable	Attribute(s)	Description
1	S	Discrete categorical	S_1: Significant changes S_2: Over time/budget S_3: On time, on budget, on scope S_4: Canceled S_5: Postponed	Success rate of the Scrum-based software development project
2	C	Discrete categorical	C_1: Lower costs C_2: At costs C_3: Higher costs	Success metric: Cost
3	L	Discrete categorical	L_1: In time L_2: On time L_3: Over time	Success metric: Deadline
4	Q	Discrete categorical	Q_1: Aspirational Q_2: Appropriate Q_3: Acceptable	Success metric: Quality
5	P	Discrete categorical	P_1: Simple P_2: Complicated P_3: Complex P_4: Anarchy	Level of complexity of the Scrum-based software development project
6	D	Discrete categorical	D_1: Done D_2: Undone	Sprint review
7	V	Discrete categorical	V_{i1}: Caused by Scrum V_{i2}: Exposed by Scrum V_{i3}: Unrelated to Scrum	Sprint retrospective with two dimensions ($i = 1, 2$): V_1: "What working well?"; V_2: "What could work better?"
8	Y	Continuous numerical	$y \in \mathbb{R}^+$	Forecasted work remaining on the product backlog based on the release backlog
9	X	Continuous numerical	$x \in \mathbb{R}^+$	Forecasted work remaining on the sprint backlog based on the daily updates of work remaining on the sprint backlog
10	H	Discrete numerical	$h \in \mathbb{Z}^+$	Number of stakeholders
11	M	Discrete numerical	$m \in \mathbb{Z}^+$	Number of features
12	K	Discrete numerical	$k \in \mathbb{Z}^+$	Number of feature teams
13	N	Discrete numerical	$n \in \mathbb{Z}^+$	Number of workers
14	T_i	Discrete numerical	$t_i \in \mathbb{Z}^+$	Number of tasks for the i-th feature
15	R	Discrete numerical	$r \in \mathbb{Z}^+$	Number of defects

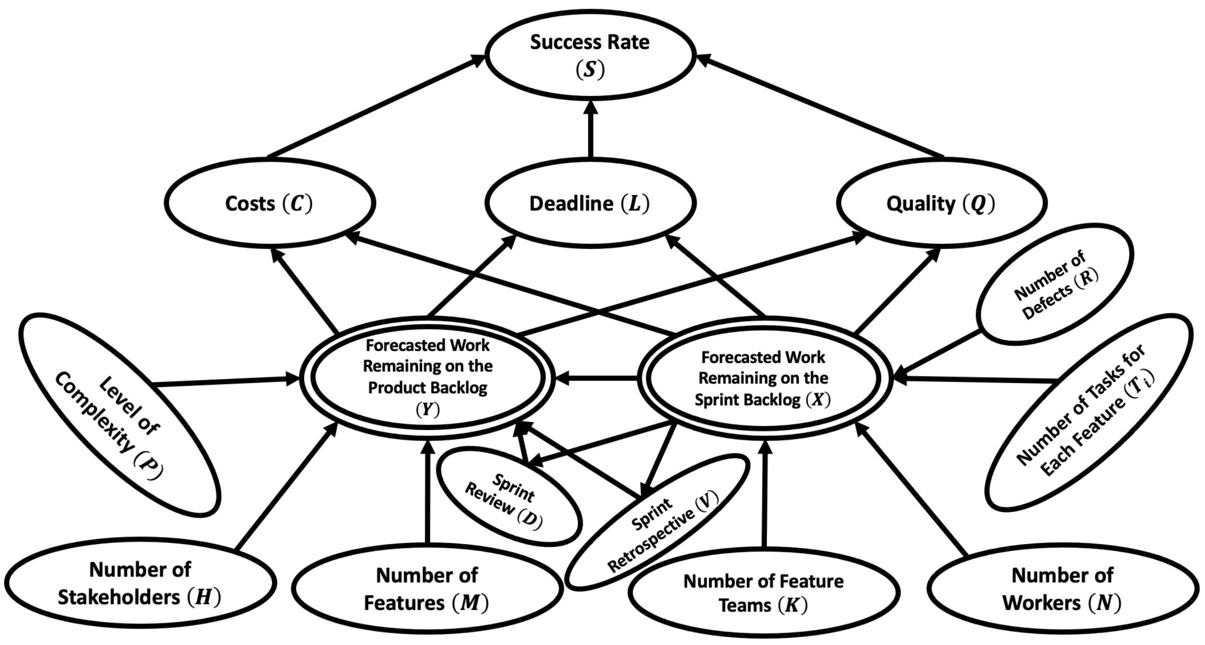
Success Rate (S)
Costs (C)
Deadline (L)
Quality (Q)
Number of Defects (R)
Level of Complexity (P)
Forecasted Work Remaining on the Product Backlog (Y)
Forecasted Work Remaining on the Sprint Backlog (X)
Number of Tasks for Each Feature (T_i)
Sprint Review (D)
Sprint Retrospective (V)
Number of Stakeholders (H)
Number of Features (M)
Number of Feature Teams (K)
Number of Workers (N)

FIGURE 3. The proposed non-parametric Bayesian networks model for Scrum-based software development project

variables are distinguished by single and double ellipses, respectively. Remember that Bayesian (personal) probability has different meaning with classical probability where Bayesian refers to a degree of belief in an event while classical view refers to the true or physical probability of that event [25].

4. **Results and Discussion.** We designed the experiments for large-scale Scrum-based development project with multiple stakeholders and multiple feature teams as described in [26] and [15]. The experiments designed are related to the number of stakeholders (H) who provide input in the implementation of Scrum; the number of workers (N) in the team along with the number of feature teams (K) formed; the number of defects (R); as well as the number of features (M) in the product backlog along with the number of tasks (T_i) in the sprint backlog for each of features (i), which tested on four levels of complexity (P) of the Scrum-based software development project and both of sprint review (D) and sprint retrospective (V) labels. To simplify the experiment, we specify the sprint time duration to be four weeks, five days each, and maximum number of sprints is ten times. Based on the proposed non-parametric Bayesian networks model, we tried to simulate some of variables used in this model to show its effect on the success rate (S) as well as success metrics, i.e., cost (C), deadline (L), and quality (Q). We generate the node probability tables using two algorithms based on: (i) ranked nodes [27]; and (ii) survey and weight functions [28], but in this study, we only used simulated survey for each scenario. Note that we can use a time series model, such as Autoregressive Integrated Moving Average (ARIMA) model, to forecast both of the work remaining on the product backlog (Y) and the sprint backlog (X) in monitoring ongoing project performance according to the forecasted data from real-time artifacts (daily updates of work remaining on the sprint backlog and release backlog).

Table 2 shows the example of simulation results for some specific values of other variables based on Table 1. We can see that the non-parametric Bayesian networks model built is running well in predicting the success rate of the Scrum-based software development project. We attempted to interpret this result as follows. In detail per row, for example the first row, we know that this project simulation has a 5.228762% probability of lower

Table 2. Example of simulation results

P	V		D	C			L			Q		
	V_1	V_2		C_1	C_2	C_3	L_1	L_2	L_3	Q_1	Q_2	Q_3
P_1	V_{11}	V_{21}	D_1	0.05228762	0	0	0.00192319	0	0.02793184	0	0	0
			D_2	0	0	0	0	0.0606877	0	0	0	0
		V_{22}	D_1	0	0	0	0	0	0	0	0	0
			D_2	0	0	0.00997789	0	0.0416214	0	0	0	0
		V_{23}	D_1	0	0	0	0	0	0	0.07067024	0	0
			D_2	0	0	0	0.0329941	0	0	0	0	0
	V_{12}	V_{21}	D_1	0	0	0	0.00819767	0	0.0347645	0	0	0
			D_2	0	0	0	0.05758373	0	0.00543889	0	0	0
		V_{22}	D_1	0	0	0	0	0	0	0	0	0
			D_2	0.0585558	0	0	0	0	0	0.00863618	0.04469239	0
		V_{23}	D_1	0	0	0	0	0.02048782	0	0	0	0.08269081
			D_2	0	0	0	0	0	0	0	0	0
	V_{13}	V_{21}	D_1	0	0	0.03678296	0	0	0	0	0.07513157	0
			D_2	0	0	0	0	0	0	0	0	0
		V_{22}	D_1	0	0	0	0	0	0	0	0	0
			D_2	0	0	0	0	0	0	0.01893567	0	0
		V_{23}	D_1	0	0	0	0	0	0	0.07170185	0.00277691	0
			D_2	0.066677	0.00253745	0	0	0	0	0	0.0713859	0
P_2	V_{11}	V_{21}	D_1	0.06966429	0	0	0	0	0	0	0	0
			D_2	0.02367123	0	0	0	0	0	0	0	0
		V_{22}	D_1	0	0	0	0	0	0	0	0	0
			D_2	0	0	0	0	0.05604965	0	0	0	0
		V_{23}	D_1	0.04924943	0	0	0	0	0	0.02273834	0.03813399	0
			D_2	0.06966803	0	0	0	0	0	0	0	0
	V_{12}	V_{21}	D_1	0	0	0	0	0	0	0	0	0
			D_2	0	0	0	0	0	0	0	0	0.00924269
		V_{22}	D_1	0	0	0	0	0.03413632	0.01550356	0	0	0
			D_2	0	0.0240381	0	0	0	0	0	0.06601211	0
		V_{23}	D_1	0	0	0	0	0	0	0	0	0
			D_2	0	0	0	0	0	0	0	0	0
	V_{13}	V_{21}	D_1	0.02978214	0.01943757	0	0	0	0	0	0	0
			D_2	0	0	0	0	0	0	0	0	0
		V_{22}	D_1	0	0.04104107	0	0.0303406	0.00951046	0	0	0	0
			D_2	0	0	0	0	0	0	0	0	0
		V_{23}	D_1	0	0	0	0.00673605	0	0.04342365	0	0	0
			D_2	0	0.0483437	0.02023798	0	0	0	0	0	0.01054474
P_3	V_{11}	V_{21}	D_1	0	0	0	0.01212	0.01401787	0	0	0	0
			D_2	0	0	0	0	0	0	0	0	0
		V_{22}	D_1	0	0	0	0	0	0	0	0	0
			D_2	0	0	0	0.01056576	0	0.05144046	0	0	0.05083566
		V_{23}	D_1	0	0.01301975	0.04462336	0	0	0	0	0	0
			D_2	0	0	0	0	0	0	0	0	0
	V_{12}	V_{21}	D_1	0	0.06096141	0	0	0	0	0	0	0
			D_2	0	0	0	0	0	0	0	0	0
		V_{22}	D_1	0	0	0	0	0	0	0	0	0.04509297
			D_2	0	0.05647057	0	0	0	0.04446277	0	0	0.08174427
		V_{23}	D_1	0	0	0	0	0	0	0	0	0
			D_2	0.05479252	0.00167993	0	0	0	0	0	0	0
	V_{13}	V_{21}	D_1	0	0.01849757	0	0	0.03084131	0.03177482	0	0.08326134	0
			D_2	0	0	0	0	0	0	0	0	0
		V_{22}	D_1	0	0	0	0	0	0	0	0	0
			D_2	0	0	0	0	0	0	0	0.01854639	0.0339871
		V_{23}	D_1	0	0	0	0	0	0.06349568	0	0	0
			D_2	0	0	0	0	0	0	0	0	0.01300513
P_4	V_{11}	V_{21}	D_1	0	0	0	0.02163572	0	0	0	0	0
			D_2	0	0	0	0	0	0	0	0	0
		V_{22}	D_1	0	0	0	0	0	0	0.02115003	0	0
			D_2	0	0	0	0.00000884	0	0.06207211	0	0	0
		V_{23}	D_1	0.01470765	0.01470765	0.01470765	0	0	0	0	0	0
			D_2	0	0	0	0	0	0	0	0	0
	V_{12}	V_{21}	D_1	0	0	0	0	0	0	0	0.05365212	0
			D_2	0	0	0	0	0.02546178	0.03257175	0	0	0
		V_{22}	D_1	0	0	0	0.05707477	0	0	0	0	0
			D_2	0	0	0	0	0	0	0	0	0
		V_{23}	D_1	0	0	0.00544279	0	0	0	0.0053865	0.00004511	0
			D_2	0	0	0	0.00610765	0	0.04901756	0	0	0
	V_{13}	V_{21}	D_1	0	0	0	0	0	0	0	0	0
			D_2	0.06914046	0	0	0	0	0	0	0	0
		V_{22}	D_1	0	0	0	0	0	0	0	0	0
			D_2	0	0	0	0	0	0	0	0	0
		V_{23}	D_1	0.00929643	0	0	0	0	0	0	0	0
			D_2	0	0	0	0	0	0	0	0	0
Marginal probability				**0.56749259**	**0.30073478**	**0.13177263**	**0.24528809**	**0.29281432**	**0.46189759**	**0.21921881**	**0.45363783**	**0.32714336**

costs (C_1), a 0.192319% probability of in time deadline (L_1), and a 2.793184% probability of over time deadline (L_3) with level of complexity as simple (P_1), a sprint review is done (D_1), and sprint retrospective is done well and could be better because of Scrum (V_{11} and V_{21}). However, we do not know about the quality of the project in this case. We can also summarize the interpretation by adding all the rows for each P_1, P_2, P_3, and P_4, respectively. This project simulation has a 17.752042% probability of lower costs

(C_1), a 0.253745% probability of at costs (C_2), a 4.676085% probability of higher costs; a 10.069869% probability of in time deadline (L_1), a 12.279692% probability of on time deadline (L_2), a 6.813523% probability of over time deadline; a 16.994394% probability of aspirational quality, a 19.398677% probability of appropriate quality, and a 8.269081% probability of acceptable quality with level of complexity as simple (P_1), and so on. If we are only concerned with probability of success metrics, we know from the marginal probability (the last row) that this generated Scrum-based software development project is most likely of spending lower costs ($Pr(C_1) = 56.749259\%$) than it should be, but still produces the appropriate quality ($Pr(Q_2) = 45.363783\%$), even though it will miss the deadline ($Pr(L_3) = 46.189759\%$). The success rate prediction for this example is classified into Over time/budget ($Pr(S_2) =$ 36.93%) (see Figure 4, for $M = 5$). Furthermore, we tried to analyze the sensitivity of this prediction success rate to the one of other variables (e.g., M) as shown in Figure 4. We can see that changes in the number of features (M) in the product backlog have a significant effect on the success rate prediction.

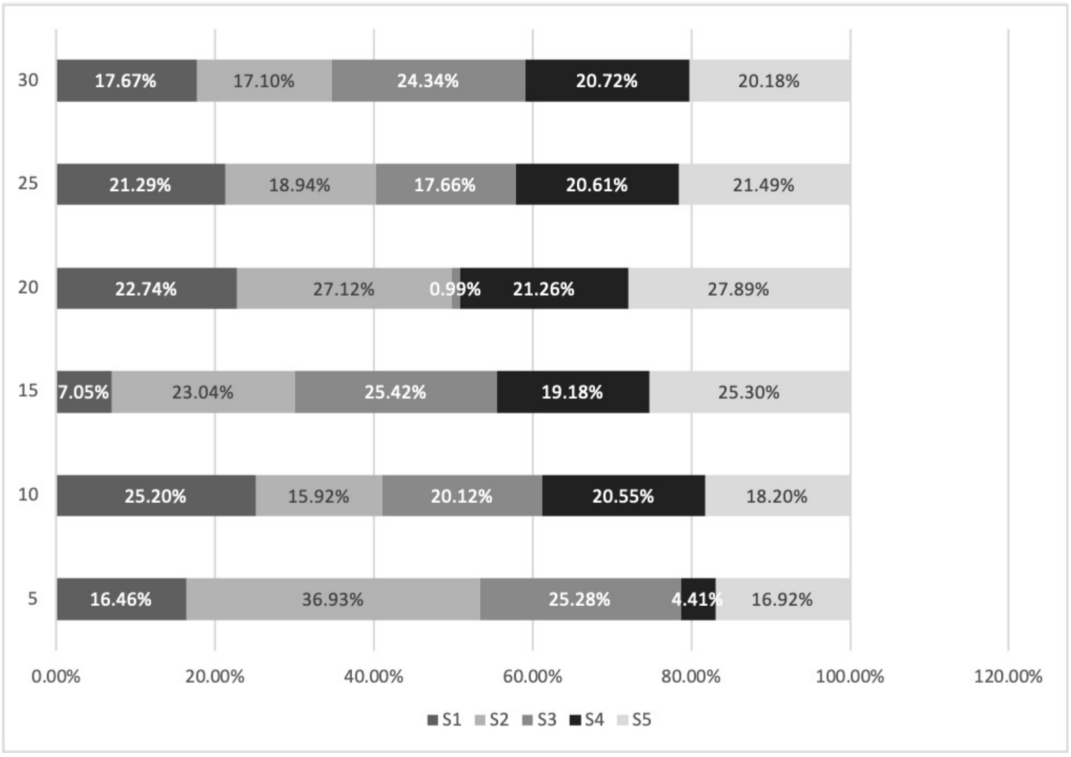

Figure 4. The sensitivity of success rate prediction to the number of features (M) in the product backlog for generated large-scale Scrum-based software development project

5. **Conclusions.** We have successfully modeled the Scrum-based software development project, formally, into a mathematical equation in the form of a non-parametric Bayesian networks model to help the Scrum team predict the success rate of the project to be executed. The Bayesian networks construction we built utilizes artifacts in the Scrum framework to be connected with success metrics of software development project. Thus, the Scrum team, now, have a reference to predicted value of software development project success rate in monitoring ongoing project performance according to the forecasted data from real time artifacts. The model has also been validated through a number of simulated experiments designed based on certain scenarios to be implemented in the generated large-scale software development projects with multiple stakeholders and multiple feature teams. The experiments designed are related to the number of stakeholders who provide input in the implementation of Scrum; the number of workers in the team along with the

number of feature teams formed; the number of defects; as well as the number of features in the product backlog along with the number of tasks in the sprint backlog for each of features, which tested on four levels of complexity of the Scrum-based software development project and both of sprint review and sprint retrospective labels with fixed sprint time duration and fixed maximum number of sprints. In this experiment, we used two algorithms, (rank nodes)-based as well as (simulated survey and weight functions)-based, to generate the node probability tables in the non-parametric Bayesian networks model. The simulation results show that the non-parametric Bayesian networks model built is running well in predicting the success rate of the Scrum-based software development project. The success rate prediction of this generated project is classified into over time/budget where it is most likely of spending lower costs than it should be, but still produces the appropriate quality, even though it will miss the deadline. We also know that changes in the number of features in the product backlog have a significant effect on the success rate prediction. This sensitivity analysis can be developed further for other variables.

Acknowledgment. This document is the result of the research project funded by the Ministry of Research and Technology/the National Agency for Research and Innovation of Indonesia (Grant No. 079/SP2H/LT/DRPM/2020).

REFERENCES

[1] A. Freire, A. Meireles, G. Guimarães, M. Perkusich, R. da Silva, K. Gorgônio, A. Perkusich and H. Almeida, Investigating gaps on agile improvement solutions and their successful adoption in industry projects – A systematic literature review, *Proc. of the International Conference on Software Engineering & Knowledge Engineering*, 2018.

[2] A. Mishra, J. Garbajosa, X. Wang, J. Bosch and P. Abrahamsson, Future directions in agile research: Alignment and divergence between research and practice, *Journal of Software: Evolution and Process*, vol.29, no.6, 2017.

[3] M. Alhaj, G. Arbez and L. Peyton, Approach of integrating behaviour-driven development with hardware/software codesign, *International Journal of Innovative Computing, Information and Control*, vol.15, no.3, pp.1177-1191, 2019.

[4] M. Holcombe, *Running an Agile Software Development Project*, John Wiley & Sons, New Jersey, 2008.

[5] K. S. Rubin, *Essential Scrum: A Practical Guide to the Most Popular Agile Process*, Addison-Wesley, Ann Arbor, 2013.

[6] P. Weaver, The meaning of risk in an uncertain world, *Proc. of the PMI® Global Congress 2008 – EMEA*, 2008.

[7] I. Ancveire, I. Gailite, M. Gailite and J. Grabis, Software delivery risk management: Application Bayesian networks in agile software development, *Information Technology and Management Science*, vol.18, pp.62-69, 2015.

[8] R. M. Saraiva, M. Perkusich, H. Almeida and A. Perkusich, A process to calculate the uncertainty of software metric-based models using Bayesian networks, *Proc. of the International Conference on Software Engineering & Knowledge Engineering*, 2017.

[9] 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*, vol.29, no.6, pp.1-17, 2016.

[10] R. Saraiva, M. Perkusich, H. Almeida and A. Perkusich, A systematic process to define expert-driven software metrics thresholds, *Proc. of the International Conference on Software Engineering & Knowledge Engineering*, 2019.

[11] P. Phannachitta and K. Matsumoto, Model-based software effort estimation – A robust comparison of 14 algorithms widely used in the data science community, *International Journal of Innovative Computing, Information and Control*, vol.15, no.2, pp.569-589, 2019.

[12] P. Pospieszny, B. C.-Chrobot and A. Kobylinski, An effective approach for software project effort and duration estimation with machine learning algorithms, *The Journal of Systems and Software*, vol.137, pp.184-196, 2018.

[13] H. K. H. Lee, *Bayesian Nonparametrics via Neural Networks*, Society for Industrial and Applied Mathematics, Pennsylvania, 2004.

[14] I. Goldstein, *Scrum Shortcuts without Cutting Corners: Agile Tactics, Tools, & Tips*, Addison-Wesley, Crawfordsville, 2014.
[15] T. Stober and U. Hansmann, *Agile Software Development: Best Practices for Large Software Development Projects*, Springer-Verlag, Heidelberg, 2010.
[16] J. Sutherland and K. Schwaber, *The Scrum Papers: Nut, Bolts, and Origin of an Agile Framework*, Scruminc., Paris, 2011.
[17] A. Silva, E. Dilorenzo, M. Perkusich, D. Santos, H. Almeida and A. Perkusich, Definition of done from academy to the industry: An exploratory survey, *International Journal of Engineering Trends and Technology*, vol.58, no.2, pp.72-78, 2018.
[18] K. Schwaber and J. Sutherland, *The Scrum Guide*^{TM} *– The Definitive Guide to Scrum: The Rules of the Game*, Scrum.org, 2017.
[19] B. Davis, *Mastering Software Project Requirements: A Framework for Successful Planning, Development & Alignment*, J. Ross Publishing, Plantation, 2013.
[20] K. Schwaber, *Agile Project Management with Scrum*, Microsoft Press, Washington, 2004.
[21] M. Perkusich, K. Gorgônio, H. Almeida and A. Perkusich, A framework to build Bayesian networks to assess Scrum-based software development methods, *Proc. of the International Conference on Software Engineering & Knowledge Engineering*, 2017.
[22] 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*, vol.42, pp.437-450, 2015.
[23] R. Willamy, J. Nunes, M. Perkusich, A. Freire, R. Saraiva, H. Almeida and A. Perkusich, A method to build Bayesian networks based on artifacts and metrics to assess agile projects, *Proc. of the International Conference on Software Engineering & Knowledge Engineering*, 2016.
[24] D. Koller and N. Friedman, *Probabilistic Graphical Models: Principles and Techniques*, The MIT Press, Cambridge, 2009.
[25] D. Heckerman, A tutorial on learning with Bayesian networks, *arXiv.org*, arXiv: 2002.00269, 2020.
[26] V. T. Heikkilä, M. Paasivaara, K. Rautiainen, C. Lassenius and T. Toivola, Operational release planning in large-scale Scrum with multiple stakeholders – A longitudinal case study at F-secure corporation, *Information and Software Technology*, vol.57, pp.116-140, 2015.
[27] J. Nunes, R. Willamy, M. Perkusich, R. Saraiva, K. Gorgônio, H. Almeida and A. Perkusich, An algorithm to define the node probability functions of Bayesian networks based on ranked nodes, *International Journal of Engineering Trends and Technology*, vol.52, no.3, pp.151-157, 2017.
[28] M. Perkusich, A. Perkusich and H. O. d. Almeida, Using survey and weighted functions to generate node probability tables for Bayesian networks, *Proc. of BRICS Congress on Computational Intelligence and 11th Brazillian Congress on Computational Intelligence*, pp.183-188, 2013.