Cluster Comput (2018) 21:715–728
https://doi.org/10.1007/s10586-017-0996-z

User stories complexity estimation using Bayesian networks for inexperienced developers

Janeth López-Martínez[1] · Alan Ramírez-Noriega[1] · Reyes Juárez-Ramírez[1] · Guillermo Licea[1] · Samantha Jiménez[1]

Received: 31 March 2017 / Revised: 6 June 2017 / Accepted: 15 June 2017 / Published online: 24 June 2017

Abstract Planning Poker is a complexity estimation technique for user stories through cards. This technique offers many advantages; however, it is not efficient enough as estimations are based on experts criteria, which is fuzzy regarding what factors are considered for estimation. This paper proposes a knowledge model to determine two of the most important aspects of estimation, the complexity, and importance of user stories based on Planning Poker in Scrum context. The goal of this work is to model the complex nature of user story estimation to facilitate this task to novice developers. A Bayesian network was built based on the proposed model that considers the complexity and importance of a user story. Students and professionals submitted their estimates to correlation tests to validate the applicability of the proposed model. Based on the results, the proposed model achieves a greater degree of correlation with the estimation from professionals than students, which means that the model includes factors considered in real world application. This proposal could be useful for guiding novice developers to evaluate the complexity and importance of user stories through questions. Students could use the proposal to estimate rather than the traditional Planning Poker.

✉ Janeth López-Martínez
patricia.lopezm@uabc.edu.mx

Alan Ramírez-Noriega
alan.david.ramirez.noriega@uabc.edu.mx

Reyes Juárez-Ramírez
reyesjua@uabc.edu.mx

Guillermo Licea
glicea@uabc.edu.mx

Samantha Jiménez
samantha.jimenez@uabc.edu.mx

[1] Universidad Autónoma de Baja California, Tijuana, Mexico

Keywords Planning Poker · Complexity · Scrum · Bayesian networks · User story estimate

1 Introduction

Scrum is an iterative, incremental and empirical process to manage and control the development of a software project. This framework is one of the agile methods most frequently used in industry [30]. Scrum progresses via a series of iterations called sprints. Each sprint starts with a planning meeting, where the product owner and team estimate work and effort of the sprint [21,29].

Accuracy in effort estimation is an essential factor for planning software projects to avoid budget overruns and delayed dates of delivery. Otherwise, results will show a low quality software [24,25]. Scrum does not provide an estimation technique, however, the most used method is Planning Poker. This technique is a light-weight method for estimating the size of a user story (US) in a face-to-face interaction and discussion between team members [13,16,25]. A user story is an independent, negotiable, valuable, estimable, small, and testable requirement.

In Planning Poker all the team members participate in the estimation process. In other words, every team member has an opinion in estimation process and a greater commitment to the team, resulting in a more enjoyable estimation process [8]. However, this method is not efficient enough because the results are always based on expert observation and experience. Story-points are a subjective value and cannot be easily related to time duration [13,20,24,25]. Moreover, the team member decision is unclear and subjective by taking into account only complexity in general. There is no exact definition of complexity for software projects, which makes the complexity estimation more uncertain and risky.

It could cause additional costs and could negatively impact project performance if participants fail to address it from the planning stage [7]. Therefore, it is necessary to discompose complexity to establish how a Scrum team estimates a US.

Professionals' experience in software development is an important factor to estimate an US. Professionals have a different vision of problems because they have been working for many years with different kinds of projects. Contrary, efficient estimation is not reflected with the inexperienced members like when students begin their venture into the development of Scrum-based projects. Team members with low experience have problems in performing an accurate estimation, due to still lacking an experts vision, and not fully understanding how the complexity of an US is determined. It is necessary to define a strategy where team members with low experience can learn faster and estimate with precision.

Several works [16,25,30] have faced this problem without considering the uncertainty introduced by person's subjectivity. On the other hand, studies [4,20] have contemplated the uncertainty of estimation through Bayesian networks. A Bayesian network (BN) expresses the causal relation between random variables of a domain knowledge [28], and their conditional dependencies via a directed acyclic graph.

A BN shows characteristics that stand out over other similar techniques such as Fuzzy Cognitives Maps [2,4]: (1) forward and backward chaining, (2) efficient evidence propagation mechanism, (3) enough implementation and support tools, (4) mathematical theorems derivable from well-defined basic axiom, and (5) correctness of the inference mechanism is provable. These BN capabilities make them able to represent knowledge and experience. BNs show the best precision effort and cost estimation techniques for agile software development by means of soft computing techniques [1].

Previously, López-Martínez et al. [12] proposed a knowledge structure to estimate the complexity and importance of the USs using Planning Poker. The main proposal was that the decision of each member could be clearer by decomposing a complex decision into simpler and more precise factors. The complexity was discomposed into three factors such as experience, time and effort, and the importance into two factors such as priority and US_value. The structure presented by López-Martínez established weights of relations based on student judge. Students' estimations were made with Planning Poker and using the BN, these results showed a low correlation value of 0.446 due to the students inexperience.

In this new study, we take as basis the BN model described before. In this case, a group of professionals from software development companies validated the proposed factors using a qualitative technique. Based on experts' knowledge, we define new weights of relations between factors; this gives robustness to the proposed BN's structure. In this experiment, a second group of experienced developers made estimations with Planning Poker in a traditional manner and through the BN. Findings show that the BN results and the Planning Poker estimations were strongly correlated, this means that the BN estimation is similar to the experts' estimation. Based on this, we suggest that the BN the estimations could be more accurate because weights in the BN were established based on the experts' judgment. Also, an experiment with students was conducted. Results showed a good correlation between students' estimates when the USs had finished and estimates made through the BN before starting the US.

This paper is organized as follows: Sect. 2 shows related work. Section 3 contains related concepts to this article. Section 4 explains theoretical aspects of this article. Section 5 refers to validation of proposed factors. Section 6 describes how the BN was built based on experts' opinion. Section 7 shows the experiment and results. Section 8 shows a discussion of results. Finally, conclusions and references are shown.

2 Related work

This section describes papers related to complexity estimation, with a similar approach of our study. Although most approaches consider BNs, there are implementations with other methods. Table 1 shows characteristics of related works. It describes the year of publication, whether the article considered the complexity within its estimates and whether the complexity is disaggregated, the approach used, and the objective of estimation. The following paragraphs describe more documents.

Complexity need to be considered in order to reduce the difficulty of estimates. Several authors have proposed different approaches to effort estimation. For instance, Mendes [15] proposed in its study a Bayesian Model for Web effort estimation using knowledge from a domain expert. Their model contained 36 factors identified as influential to software effort and risk management, and also 48 relationships were associated to these factors. The BN presented in this work was built and validated using an adaptation of the knowledge engineering of Bayesian networks.

Nassif et al. [18] developed a regression model for software effort estimation based on use case point model (UCP). Authors employed a Fuzzy Inference System (FIS) to improve the estimation. The main advantage of the UCP model is that it can be used in the early stages of the software life cycle, when the use case diagram is available. They handled complexity as a synonym of technical factor, dividing it into thirteen aspects and assigning weight to each one of them.

Popli and Chauhan [24] proposed an algorithmic estimation method. This approach considered various factors, thereby estimating the more accurate release date, cost, effort, and duration of the project. The effectiveness and feasibility of the proposed algorithm have been shown by considering

Table 1 Comparison of related work

Art.	Year	Complexity				
		No	Yes			
			No disagg.	Disagg.	Approach	Objective
[15]	2011		×		Bayesian network	Web effort estimation
[18]	2011			×	Regression model with Sugeno FIS	Software size and effort
[24]	2014		×		Own algorithm	Cost, effort, and duration of the project
[30]	2015		×		Own algorithm	adjusted story points calculation
[10]	2015		×		Bayesian network	Effort estimation
[31]	2016		×		Bayesian network	Effort estimation (COCOMO)
[20]	2016			×	Own algorithm	Effort, duration and cost estimation
[4]	2017			×	Bayesian network	Task effort estimation

three cases in which different levels of factors were taken into account and compared. The method did not decompose the complexity into more variables.

In order to produce more accurate effort and time estimations, Zahraoui et al. [30] adjusted story points calculation using three factors: priority, size, and complexity. To calculate the total effort of a Scrum project, they proposed to use a new adjusted story point measure instead of story pointing one. Moreover, they gave a way to use the proposed adjusted story point in the adjusted velocity. Nevertheless, they did not decompose the complexity in factors.

In the same way, Karna and Gotovac [10] considered complexity as a single variable, but this variable needed to be discomposed into other factors. They presented a Bayesian Model, including the relevant entities that are involved in the formation of the effort estimation. And they considered mainly three entities involved with the estimation process: Projects, work items, and estimators. This is a good proposal, however, it is only a theoretical model that needs to be tested. The complexity needs to be discomposed to reduce the estimation difficult.

Zare et al. [31] presented a three-level Bayesian network based on COCOMO components to estimate the needed effort (Man-Month) for the software development so that the estimated effort is modified using the optimal coefficient resulted from optimal control designed by the genetic algorithm. The complexity is defined as a single variable.

Owais [20] developed an algorithm for the effort, duration, and cost estimation for agile software development. They considered three types of complexity: Technology, integration, and team. However, story points are not calculated, this metric is an input of the model, our work considers it as the main problem with the calculation of story points.

Dragicevic et al. [4] proposed a fourth-level BN model for task effort prediction in agile software development projects. A set of elements was defined such as: Working hours, complexity requirements, developer skills, tasks, complexity form, complexity function, complexity report, and quality specification. The structure of the model was defined by the authors, while the parameter estimation is automatically learned from a dataset.

All documents presented in Table 1 considered the complexity; this shows the importance of this concept in estimations. Complexity needs to be decomposed to facilitate estimates; however, some authors disaggregate with it [4,18,20] and other do not [10,15,24,30,31]. Our work considers the complexity disaggregated.

BNs have been widely used in this area [4,10,15,31], although authors also have used fuzzy logic [18] and their own methods [20,24,30]. Our work proposes a model based on experts knowledge. By obtaining weights of relations between variables it is possible to construct the CPT for the statistical inference. In addition, the approach is based on Scrum and Planning Poker.

The objective of related work is to estimate the effort (all related work), the cost [20,24], the duration [20,24], and the size [18] of projects. Our work focuses on defining effort through the complexity of USs using planning poker. The Zahraoui's work [30] also focus on defining through its method the complexity of the US, however, the method is only a proposal and has not yet been proven.

None of works cited is focused on education, our research focuses on providing a different view of estimation to students. This through an estimation process based on the decomposition of complexity when Planning Poker is used. Our method is employed through questions that represent variables in a BN. This way the student perceives the complexity, as decomposed elements facilitate their estimation.

3 Fundamentals

3.1 Scrum

Scrum is the most popular agile framework in the software industry. By using Scrum practices, several companies have improved their quality and productivity [17]. It is an agile method of development which is defined as an iterative, incremental and empirical process to manage and control the development of a project.

Scrum has three different levels of planning: (1) the Release Planning, (2) the Sprint Planning, and (3) the Daily Scrum. During the Release Planning, the teamwork discusses the essential strategic aspects like the overall costs or functionality of a development project. Operational details are instead planned from iteration to iteration during the construction project.

Scrum defines these iterations as Sprints, which are supposed to take from two to four weeks. In the according to Sprint Planning meeting, which should take eight hours for a one month Sprint, the requirements and tasks for the next Sprint are fixed. The most detailed level of planning takes place in the Daily Scrum meetings [19,29].

Scrum is based on a structure of the self-organizing team, this not only makes the Scrum projects more transparent and flexible; it also presumes that team members have a strong commitment and sense of responsibility. In addition to team collaboration, the cooperation with the customer also plays a major role in Scrum projects.

A Scrum team is composed of a Scrum Master, a product owner, and the development team (or just, “the team”), with all the skills needed (such as requirements gathering, designing, coding, and testing) to build the software product [5,22].

3.2 Planning Poker

Planning Poker is a technique for sizing Product Backlog Items; it is the most used estimation technique in Scrum projects. This technique is a team-based exercise used for assigning relative estimate values to user stories/requirements to express the effort required to deliver specific features or functionality [20].

The game utilizes playing cards, printed with numbers based on a modified Fibonacci sequence similar to 1, 2, 3, 5, 8, 13, 21, 40, 100 (this study employs a Fibonacci trimmed version). The Product Owner and all members of the development unit meet to discuss product backlog requirements for the purpose of reaching consensus-based estimates.

When the Scrum team finishes discussing the story, each estimator privately estimates the required effort by writing the corresponding number of story points on a paper note card or choosing a card with the corresponding predefined value. If the estimations differ too much, the estimators discuss their proposals. Developers with the lowest and highest estimate justify their proposals; the team repeats the process until reaching consensus [13,30].

Planning poker brings together multiple expert opinions to do the estimation. These experts form a cross-functional team from all disciplines on a software project. They are better suited to the evaluation task than anyone else. Group discussion is the basis of planning poker, and those discussions lead to an averaging of sorts of the individual estimates [3].

3.3 Bayesian network

Bayesian networks belong to the family of probabilistic graphical models. These graphical structures are used to represent knowledge about an uncertain domain. Each node in the graph represents a random variable, while edges between nodes represent probabilistic dependencies among the corresponding random variables. These conditional dependencies in the graph are often estimated by using known statistical and computational methods. Hence, BN’s combine principles from graph theory, probability theory, computer science, and statistics [27].

A Bayesian Network is defined as $RB = (G, P)$. Where $G = (V, E)$ is a directed acyclic graph, V denotes the set of observed variables, and E denotes the set of edges that indicate causal relationships. P represents the Conditional Probability Tables (CPT) on V that shows the causal influence [11]. The conditional probability table is used to express how the potential states of the node’s parents affect the posterior probability of the considered node [6]. This kind of networks have been used today for many fields in the research [28] due to the flexibility to adapt to the resolution of different diagnosis problems and patterns determination.

4 Concepts related to our proposal

This section explains factors considered to develop BN. Factors determined in articles [12,26] are the basis for this study, they did a literature review to identify and relate variables, Fig. 1 represents the structure.

When we talk about complexity in a software context, we can be analyzing the difficulty of the problem that the software application will try to implement, the structure of the code or the relationships between the data elements that will handle in the application. In other words, the term complexity can be used to analyze the complex nature of problems, code, and data [9].

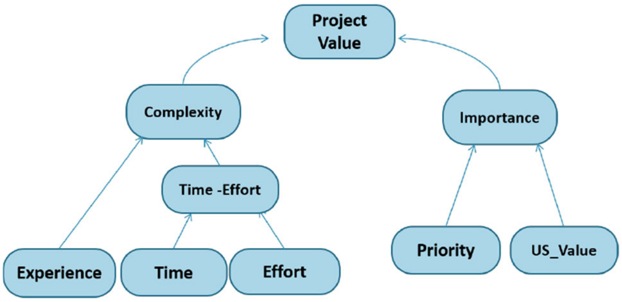

Fig. 1 Relations between variables of the proposal

For this work, the complexity will be considered in two aspects: The time and the effort necessary to implement a user story. Time is related to how laborious the US may be, measured in hours. Even if a user story is simple in nature it could require a lot of time to be implemented, or it could be the opposite, being very difficult but in nature it could be implemented very quickly [24]. Effort refers to the amount of cognitive work required to analyze, design, implement and resolve a user story [25].

On the other hand, in practical scenarios, the experience is not considered as part of the complexity. However, it works as a factor to increase or decrease the complexity. Experience refers to the knowledge that the person possesses based on similar projects [20,23]. Based on this, the more experience a developer has, the easier it will be to estimate and develop an US. Otherwise, the less experience a developer has, the more difficult it will be to estimate and develop an US.

The contribution degree that an US makes to the project represents the importance of that US. This paper considers two aspects of the importance: The priority and the US value. The priority is represented by the dependency about user stories. In practical scenarios, the most important US must be implemented in the first Sprint. This means that an US that has more relation with others has a higher priority because these kinds of user stories are the pillars of the project [14,30]. The US value refers to the amount of revenue that might be generated or lost by a user story. So in Scrum, the user stories that has the highest business value must be delivered in the first Sprint [30].

5 Validation of factors

These factors were validated through professionals in the industry of software development in Mexico. They are 16 Software Engineers from 4 companies, with different roles such as: Scrum masters, product owner, testers, software developer manager, and software architecture. They work with Scrum and use Planning Poker to assign complexity to USs, therefore, they have experience in this area.

Each factor was evaluated considering the sentence: *"I employ the factor X to assign complexity or importance to a user story in Scrum"*, where, the variable *X* represents one proposed factor. The survey used a seven-point Likert scale (strongly disagree to strongly agree) to measure the use degree of the factor.

Survey results are shown in Table 2. We considered that the scale points 1, 2, and 3 are negative aspects, and the scale points 5, 6, and 7 are positive aspects. The neutral point is not located anywhere. The percentages of factors acceptance are found on the positive side in each case: experience (100%), time (87.5%), effort (100%), priority (75%), and US value (81%).

The grade of acceptance of variables in group was obtained taking values of the Likert scale (1–7). Variables were classified in accepted (5–7) or not accepted (1–3), each opinion of respondents is added to obtain the total. We have a group of respondents that accept variables (21–35 points) and another group that does not (5–20 points). There were 100% of opinions to accept the variables.

We concluded that professionals accepted factors either individually or in group.

6 Bayesian network construction

This section display how the BN is built based on [12]. First, the BN qualitative part is designed, this part represents the structure formed by nodes and relations, this part is developed based on the group of variables defined in [26], see Fig. 1. The quantitative part is defined, this part represents to Conditional Probability Tables (CPT), professionals gave their opinions for building CPT.

Table 2 Factors validation by professionals

Factors	Likert scale						
	Strongly disagree			Neutral			Strongly agree
	1	2	3	4	5	6	7
Time (%)	6.25	0.00	0.00	6.25	6.25	31.25	50.00
Effort (%)	0.00	0.00	0.00	0.00	31.00	25.00	44.00
Experience (%)	0.00	0.00	0.00	0.00	6.00	44.00	50.00
Priority (%)	0.00	6.25	0.00	18.75	18.75	12.50	43.75
US value (%)	0.00	0.00	19.00	0.00	31.00	25.00	25.00

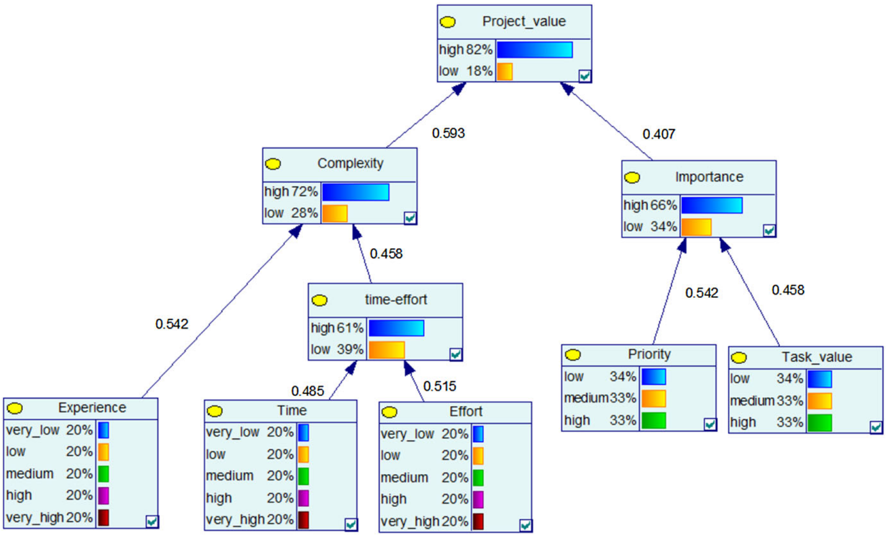

Fig. 2 Bayesian network with planning poker variables

6.1 Bayesian network qualitative part

Nodes and relations compound the qualitative part of BN; this structure is built considering the variables and relations in Fig. 1. The BN forms a hierarchical structure with variables organized by levels, these variables are described below:

- First level variables: These nodes are evidence collectors, Scrum team members enter their estimates by applying mean to these variables.
- Second level variables: The network has nodes that group the first level variables. These variables help to organize information hierarchically. Breaking down the exponential growth that would imply linking many variables to one.
- Project value variable: This variable shows the final value to measure the grade of complexity and importance of the US.

6.2 Bayesian network quantitative part

This study defined the weight of relations between variables with the opinion of Scrum professionals; previous work [12] established relations value based on students opinions. However, domain experts have more knowledge and experience than students. The construction of BN quantitative part needs two statistics aspects: values of relations between variables and a priori values of first level variables.

6.2.1 Values of relations between variables

The following equations and definitions calculate weights between variables:

Definition 1 Let $V = \{v_1, \ldots, v_n\}$ be the set that denotes the experts' opinion. Where the element $v = (weight, frequency)$ represents the weight given to one point to the Likert scale $(weight)$ and frequency of the scale $(frequency)$. The five-point Likert scale defines the variable n in five due to the number of points. Based on Definition 1, Eq. 1 establishes the non-normalized relation of one variable p (Definition 2).

$$\underset{p\in P}{p} = \frac{\sum_{v\in V}(v_{weight}\, v_{frequency})}{\sum_{v\in V} v_{frequency}} \tag{1}$$

Definition 2 Let $P = \{p_1, \ldots, p_m\}$ be the set of values that represent to parents nodes, these nodes have a causal influence over the same child node. Equation 2 obtains normalized relations considering the set of parents and their values calculated with Eq. 1.

$$\underset{p\in P}{p_{norm}} = \frac{p}{\sum_{p\in P} p} \tag{2}$$

The weight of each relation is shown in Fig. 2, this illustrates the opinion of professionals.

Definition 3 Let $A = \{a_1, \ldots, a_q\}$ the set of second level variables; $complexity$, $importance$, and $time\text{-}effort$, these variables have two states: low, and $high$. Their values are calculated through Eq. 3.

$$\underset{a\in A}{a} = \frac{\sum_{p\in P} p}{|P|}. \tag{3}$$

where the set P represents parents of the variable a, the total of variables that have influence over another is represented by $|P|$, and the unweighted value is represented by p. The normalized value is calculated with Eq. 2, but, taking into account to the variable a instead p,

Table 3 Frequencies and Weights of first level variables

Weights	Likert point	Frequency				
		Experience	Time	Effort	Priority	US_value
−1	1	0	1	0	0	0
−0.66	2	0	0	0	1	0
−0.33	3	0	0	0	0	3
0	4	0	1	0	3	0
0.33	5	1	1	5	3	5
0.66	6	7	5	4	2	4
1	7	8	8	7	7	4
	Var. weight	0.561	0.485	0.515	0.542	0.458

6.2.2 Application of equations

This section explains how BN weights were obtained to construct CPT. Table 3 shows frequencies organized by factor, as well as the weight assigned to each point on the Likert scale used, the last row represents the weight of the factor in the BN.

To obtain the non-normalized weight of the variable $time$, we used Eq. 1 based on the frequencies and weights of Table 3:

$$time = \frac{(-1 \times 1) + (-0.66 \times 0) + \cdots + (1 \times 8)}{16} = 0.664 \tag{4}$$

Eq. 2 is used to obtain the normalized weight of the variable $time$. It is necessary to consider nodes that are parents of the same child; the equation considered the non-normalized value of $effort$ (0.706).

$$time_{norm} = \frac{0.664}{0.664 + 0.706} = 0.485 \tag{5}$$

Values of first level variables are represented in Table 3. The second level variables get their weight based on Eq. 3.

$$time\text{-}effort = \frac{0.664 + 0.706}{2} = 0.685 \tag{6}$$

To obtain the normalized weight of the $time\text{-}effort$ variable, we consider Eq. 2 and the other parent that influence the same variable, in this case experience, which should already have its non-normalized weight (0.809).

$$time\text{-}effort_{norm} = \frac{0.685}{0.809 + 0.685} = 0.458 \tag{7}$$

Calculating equations for each first and second level variable, we obtain results of Fig. 2.

6.2.3 A priori values for first level variables

Variables have possible situations in which they can remain; these situations are called states, it is a way of discretizing the continuous values. Firs level variables related to US complexity employ five states: $very\_low$, low, $medium$, $high$, and $very\_high$, and variables related to US importance use three states: low, $medium$, and $high$. We do not have statistics about the proposal factors, however, we defined that each possible state of first level variables will have the same probability to be chosen, this means, the three or five states, depending on the selected variable, distribute the maximum value (one).

6.2.4 Construction of conditional probability tables

The last step is to assign a conditional probability table to each node in the structure. This part constructs a conditional probability table for a given child node, for each child node the process is repeated. A child variable is one that has influence from other variables parents P.

Definition 4 Let $S = \{s_1, s_2, \ldots, s_{|S|}\}$ be the set of states that a variable can have. The weighted values for variables with five states were 0.067, 0.133, 0.200, 0.267, and 0.333 to $very\_low$, low, $medium$, $high$, and $very\_high$ respectively. The weighted values for variables with three states were 0.17, 0.33, and 0.50 to low, $medium$, and $high$ respectively.

The matrix in Eq. 8 displays the process to create the conditional probability table. The value of the variable is multiplied by the value of the scale to find a weighting ($p_1 s_1$, $p \in P$, $s \in S$). Secondly, results in the first step are combined, each combination obtains a unique value adding each element of the combination $((p_1 s_1) + (p_2 s_1) + \cdots + (p_n s_1))$.

$$W = \begin{bmatrix} w_{11} & w_{12} & \cdots & w_{1m} \\ w_{21} & w_{22} & \cdots & w_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ w_{n1} & w_{n2} & \cdots & w_{nm} \end{bmatrix} \tag{8}$$

where

$$\begin{aligned}
w_{11} &=((p_1s_1)+(p_2s_1)+\cdots+(p_ns_1))\\
w_{12} &=((p_1s_1)+(p_2s_1)+\cdots+(p_ns_2))\\
w_{1m} &=((p_1s_1)+(p_2s_1)+\cdots+(p_ns_m))\\
&\cdots\\
w_{n1} &=((p_1s_m)+(p_2s_m)+\cdots+(p_ns_1))\\
w_{n2} &=((p_1s_m)+(p_2s_m)+\cdots+(p_ns_2))\\
w_{nm} &=((p_1s_m)+(p_2s_m)+\cdots+(p_ns_m))
\end{aligned}$$

After, the variable max represents the max value of the matrix, this variable is used to get the final values in a proportional way. The maximum value divides the matrix in Eq. 8, in the form $W = W/max$, to obtains the final values. These values represent the state $high$ of second level and project value variables, the state low is the complement value of the state $high$.

7 Testing the Bayesian network

The experiment consisted on compare the traditional Planning Poker estimations with the BN estimations from two approach: student from real projects and professional from software development companies. Professionals estimations must be more accurate. Therefore, the BN estimations should be similar to the traditional professional estimations. This proves that the decomposition of the complexity through the BN is based on sound theoretical and empirical foundations.

Both types of participants, students and professionals, followed the same process. The process consisted of the following steps:

1. The team was met to estimate USs complexity.
2. Scrum team members received an US.
3. Members individually determined the complexity of the US in a traditional way with cards.
4. Members independently, answered a set of questions related to the BN factors.
5. The team member information was introduced as inputs in the BN to made the estimation.
6. Then the sprint continued its course in a normal way.

Finally, only students made an extra estimation at the end of the US, this last estimation must be more precise than the first traditional estimation, because students acquired experience in that US.

The USs were described in detail to estimate its complexity regarding the factors proposed. For example, to measure experience we used the question: how much experience do you have for this user story? This study considered the states of variables as answers to questions. The BN system received as input answers from professionals and students.

This paper contemplates the modified Fibonacci series to assess the complexity of USs; this series consists of values 1, 2, 3, 5, 8, and 13. The previous study [12] considered nine cards to estimate; however, the enterprise in the second experiment of this study employs only uses six cards. They argued that six cards is enough to estimate, due to one US determined with a card bigger than 20, on a scale of nine cards, represents a lot of complexity.

The Spearman test was used to measure the correlation between students-BN and professionals-BN.

7.1 Estimation by students

The first experiment was held with a group of students from the computer engineering undergraduate program, they are in their final semesters. The experiment considered a short project based on two sprints with nine and five USs.

7.1.1 Students' results

Students made their estimation using Planning Poker with the regular method (throwing a card) and with the BN (using questions to estimate). To estimate through the BN, students estimated an US answering five questions for each first level variable, there was five questions for each USs, the BN received as input the student responses.

The information of each estimated US by students in the sprint one is displayed in Table 4 and the results of the sprint two are represented in Table 5. Both tables represent the output of the BN for each sprint. Second level variables and the variable project value are represented by its name in table columns. The column $complex_{stu}$ represents the estimate made by students in the traditional manner. Values in the column $complexity$ are converted to modified Fibonacci scale based on Eq. 9, results are in the column $complex_{stu\_BN}$. The column $complex_{stu2}$ shows the student's estimate after completing the US.

$$Y = Y1 + \left[\left(\frac{X - X_1}{X_2 - X_1}\right)(Y_2 - Y_1)\right] \tag{9}$$

where X represents the complexity value of the BN, X_1 is the threshold lower of the BN, X_2 is the threshold higher of the BN, Y_1 is the threshold lower of cards position, Y_2 is the threshold higher of cards position.

The result of Eq. 9 is the number of the card instead of the Fibonacci value. That is, the Fibonacci value 1 corresponds to the card 1, the Fibonacci value 2 corresponds to the card 2, the Fibonacci value 13 corresponds to the card 6, this provides the possibility to adapt to any valuation possible.

Table 4 Results of the user story evaluation by students (sprint one)

Case	US	Complexity (%)	Importance (%)	Project value (%)	$complex_{stu\_BN}$	$complex_{stu}$	$complex_{stu2}$
1	US1	55	84	79	1	5	2
2	US1	65	100	86	2	2	2
3	US1	52	84	78	1	2	3
4	US1	59	66	77	2	8	1
5	US1	62	66	78	2	2	1
6	US2	70	100	88	3	3	2
7	US2	73	100	90	3	2	2
8	US2	62	66	78	2	2	2
9	US2	69	66	81	3	3	2
10	US2	72	66	80	3	3	2
11	US3	70	100	88	3	3	2
12	US3	65	82	83	2	3	2
13	US3	67	100	87	3	1	2
14	US3	69	66	81	3	3	2
15	US3	72	66	82	3	1	2
16	US4	62	82	82	2	1	2
17	US4	70	100	88	3	1	2
18	US4	59	100	84	2	1	2
19	US4	58	66	77	2	2	1
20	US4	58	66	77	2	2	1
21	US5	52	82	78	1	2	1
22	US5	59	82	80	2	1	1
23	US5	59	100	84	2	3	1
24	US5	69	66	81	3	3	2
25	US5	58	66	77	2	3	2
26	US6	84	100	94	5	8	5
27	US6	84	100	94	5	13	5
28	US6	69	100	88	3	5	5
29	US6	94	82	94	8	8	8
30	US6	86	82	91	5	8	5
31	US7	52	82	78	1	1	1
32	US7	52	82	78	1	2	1
33	US7	55	100	83	1	3	1
34	US7	55	49	72	1	2	1
35	US7	58	66	77	2	2	1
36	US8	66	100	87	2	5	1
37	US8	62	82	82	2	2	2
38	US8	70	100	88	3	3	2
39	US8	86	100	95	5	3	3
40	US8	86	100	95	5	3	1
41	US9	58	66	77	2	3	3
42	US9	65	82	83	2	1	1
43	US9	59	49	73	2	3	2
44	US9	75	100	90	3	2	2
45	US9	78	100	92	5	5	3

Table 5 Results of the user story evaluation by students (sprint two)

Case	US	Complexity (%)	Importance (%)	Project value (%)	$complex_{stu\_BN}$	$complex_{stu}$	$complex_{stu2}$
1	US1	62	100	85	2	2	2
2	US1	65	82	83	2	2	2
3	US1	62	100	85	2	2	2
4	US1	65	100	86	2	1	2
5	US1	69	100	88	3	3	2
6	US2	88	84	92	5	5	5
7	US2	84	100	94	5	5	5
8	US2	77	100	91	3	5	5
9	US2	94	100	98	8	8	5
10	US2	94	66	91	8	5	5
11	US3	73	100	90	3	1	3
12	US3	62	82	82	2	2	3
13	US3	59	100	84	2	2	3
14	US3	65	100	86	2	3	3
15	US3	55	100	83	1	1	2
16	US4	65	66	79	2	3	2
17	US4	65	82	83	2	1	2
18	US4	69	100	88	3	2	3
19	US4	65	82	83	2	2	2
20	US4	72	100	89	3	3	2
21	US5	52	66	74	1	2	2
22	US5	77	66	84	3	3	2
23	US5	52	49	71	1	3	2
24	US5	52	82	78	1	1	1
25	US5	52	66	74	1	2	1

Table 6 Spearman correlation of estimates of students and the BN

Variables	Sprint one		Sprint two	
	p-value	Correlation	p-value	Correlation
$complex_{stu}$–$complex_{stu\_BN}$	0.004	0.418	0.001	0.643
$complex_{stu2}$–$complex_{stu\_BN}$	0.000	0.605	0.000	0.741

7.1.2 Correlation testing student-BN

We used three variables:$complex_{stu}$, $complexity_{stu\_BN}$, and $complex_{stu2}$. The variable $complex_{stu}$ represents the values returned by students to assign complexity to the US in the traditional manner. The variable $complex_{stu\_BN}$ represents the probabilities returned by the BN to assign complexity to the US based on questions answered by students. The variable $complex_{stu2}$ represents the values returned by students to assign complexity to the completed US in the traditional manner

Spearman's correlation tests considered the relations: $complex_{stu}$–$complex_{stu\_BN}$ to determine the student's correlation before starting the US, and $complex_{stu2}$–$complex_{stu\_BN}$ to calculate the correlation when the US was terminated. Results are displayed in Table 6, where the variable column represents the relations mentioned above. Tests were divided by sprint considering the p-value and the correlation.

7.2 Estimation by professionals

The second experiment was carried out with professionals working in a software development enterprise, they are a Mexican company formed by professionals in the IT field. The company offers customized software development, mobile and web applications, business solutions and database development on various platforms and operating systems.

7.2.1 Professionals' results

Professionals also made their estimation using Planning Poker with the traditional manner and with the BN. To

Table 7 Results of the US evaluation by professionals

Case	US	Complexity (%)	Importance (%)	Project value (%)	$complex_{prof\_BN}$	$complex_{prof}$
1	US1	100	64	92	13	13
2	US1	97	82	95	8	13
3	US1	97	64	91	8	13
4	US1	97	64	91	8	13
5	US2	97	49	88	8	8
6	US2	97	100	99	8	8
7	US2	97	49	88	8	8
8	US2	97	66	92	8	8
9	US3	100	82	96	13	8
10	US3	97	100	99	8	8
11	US3	94	82	94	8	13
12	US3	94	82	94	8	8
13	US4	87	34	81	5	3
14	US4	87	70	89	5	5
15	US4	87	34	81	5	5
16	US4	84	66	87	5	5
17	US5	84	34	80	5	3
18	US5	80	34	78	5	3
19	US5	84	34	80	5	3
20	US5	88	34	81	5	3
21	US6	73	49	79	3	3
22	US6	80	34	78	5	3
23	US6	73	34	76	3	3
24	US6	65	34	73	2	3

estimate through the BN, professionals estimated an US answering five questions for each first level variable, five questions for each US, the BN received as input professional responses.

The information of each estimated US by each professional developer is displayed in Table 7, this table represent the output of the BN. Second level variables and the variable project value are represented by its name in table columns. The column $complex_{prof}$ represents the estimate made by professionals in normal way. Values in the column $complexity$ are converted to modified Fibonacci scale based on Eq. 9, the result are in the column $complex_{prof\_BN}$.

7.2.2 Hypothesis testing professionals-BN

We used two main variables: (1) the variable $complex_{prof\_BN}$ represents the probabilities returned by the Bayesian network to assign complexity to the US based on questions answered by professionals. (2) The variable $complex_{prof}$ represents the values returned by professionals to assign complexity to the US in the normal method.

The Spearman test was applied with results in Table 7 to statistically analyze estimations of professionals. The hypothesis are:

- H_0: variables $complex_{prof}$ and $complex_{prof\_BN}$ do not have correlation.
- H_1: variables $complex_{prof}$ and $complex_{prof\_BN}$ have correlation.

The Spearman test gave a correlation of 0.873 within a confidence level of 95%. The p-value was lower than the confidence level $0.000 < 0.05$. Therefore, variables $complex_{prof}$ and $complex_{prof\_BN}$ have correlation, we accept the hypothesis H_1.

8 Discussion

This study proposed to discompose the complexity estimation in factors such as experience, time and effort, to reduce the subjectivity of Planning Poker evaluation. The process is subjective in nature, but the inexperience of team members make more complex the estimation process. We hypothesized that the BN complexity estimation based on factors such as time, effort and experience is similar to the estimation made by an expert. The findings of this study support this hypothesis.

In a previous work [12], we analyzed student estimations and BN results. BN relations was based on student knowledge, results showed that both estimations had a Spearman correlation of 0.446, a moderate correlation. On the other hand, this study conducted the same experiment but with different students and the BN relations were based on experts knowledge. Student estimations and BN results had a moderate Spearman correlation of 0.418. In both experiments students estimations were similar. This occurs because students have no experience in developing projects and estimating, for that reason their estimation are considered less reliable.

It is a fact that an estimation of a US is more accurate when the US was finished than when it has not yet started. Based on this fact, the estimation made by students prior US and the BN estimation had a correlation of 0.418; and, the estimation made by students post US had a 0.605 of correlation with the BN estimation. For the second sprint, the BN estimation had a correlation of 0.643 with the estimation prior US and a value of 0.741 with the estimation post US (see Table 6). It could be considered that the student estimate post US is more accurate, because their experience developing this US. The results show and increase on correlation value with the estimation post US and BN estimation. This increase indicates that the estimation made through BN is closer to the more accurate estimate. At this point, the BN estimate better than the first complexity estimation of students.

On the other hand, professionals who had more experience estimating US using Planning Poker show a better correlation than students. Most of results of the BN estimation were the same of the professional evaluation. Spearman correlation test obtains a correlation of 0.873, the BN results and the professionals' estimation were strongly correlated. These factors are in expert's mind unconsciously when they make the complexity estimation. A professional with a lot of experience developing software projects estimates more accurately an US because he/she developed it previously and knew how difficult or complex it is. In the same way, experience provides the caution to consider the full complexity of an unknown US.

At the moment, results lead us to establish that the BN estimations are similar to those of the professionals experts. These findings suggest that it could be useful for inexpert members when they need to estimate complexity, this offers the possibility of using the network output to rely on the complexity of an US. Through this process, students will not show a Fibonacci card; they will give answers to questions that feed the network. The proposal is a training process while students acquire the experience and skills to estimate USs.

The network handling two significant aspects, the complexity of an US and the value of the US for the project. The main node of the BN is the US value for the project, considering the estimates of the professionals who have more experience than the students, we obtained USs that contribute more value to the project. These were chosen to consider the complexity and importance of the task. The US 3 is considered the most fundamental with a 96% probability, the US 1 is the next with 92%. These two USs are the most essential of the project because they will give you identity, so, we need to put special attention to these USs. On the other hand, the US 6 is the least necessary; this does not mean that it should not be developed, this US brings less identity to the project, we can focus more resources to other USs before this US.

9 Conclusions

This study presents a knowledge structure to assign complexity and importance to USs based on BNs using Planning Poker. Instead of taking into account the complexity as a unique value, it was divided into three variables. So the attention is given on each variable at a time, obtaining higher precision in developers' decision-making. Considering the complexity and importance of an US we can define the most important US of the project, these tasks give identity to the project, so, they must have special treatments to develop them properly.

This project developed a knowledge structure that helps students to make estimates through the decomposition of complexity. Learning will be reflected when students begin to correlate better with estimates made through BN, so their estimates will be more reliable, and we can affirm that students learned to estimate through Planning Poker. So far, we have promising results that indicate a high correlation of professional estimates in software development, so the student can rely on the results that the BN shows.

Future work considers testing a real project from beginning to end, developing a mobile application to automate the estimation of USs using the proposed BN, and validating that the USs estimated by the network really are essential.

Acknowledgements We appreciate the support of Consejo Nacional de Ciencia y Tecnología (CONACYT) and Universidad Autónoma de Baja California for resources provided to develop this research.

References

1. Bilgaiyan, S., Mishra, S., Das, M.: A review of software cost estimation in agile software development using soft computing techniques. In: 2nd International Conference on Computational Intelligence and Networks (CINE), pp. 112–117 (2016)

2. Cheah, W.P., Kim, K.Y., Yang, H.J., Kim, S.H., Kim, J.S.: Fuzzy cognitive map and Bayesian belief network for causal knowledge engineering: a comparative study. KIPS Trans. Part B **15**(2), 147–158 (2008)
3. Cohn, M.: Techniques for estimating. In: Hall, P. (ed.) Agile Estimating and Planning, 1st edn., chap. 6, pp. 49–60. Prentice Hall, Stoughton (2005)
4. Dragicevic, S., Celar, S., Turic, M.: Bayesian network model for task effort estimation in agile software development. J. Syst. Softw. **127**, 109–119 (2017)
5. Eloranta, V.P., Koskimies, K., Mikkonen, T., Vuorinen, J.: Scrum anti-patterns—an empirical study. In: 20th Asia-Pacific Software Engineering Conference (APSEC), vol. 1, pp. 503–510 (2013)
6. Fenz, S.: An ontology-based approach for constructing Bayesian networks. Data Knowl. Eng. **73**, 73–88 (2012)
7. Floricel, S., Michela, J.L., Piperca, S.: Complexity, uncertainty-reduction strategies, and project performance. Int. J. Proj. Manag. **34**(7), 1360–1383 (2016). doi:10.1016/j.ijproman.2015.11.007
8. Haugen, N.C.: An empirical study of using planning poker for user story estimation. In: Proceedings—AGILE Conference 2006, vol. 23–31 (2006)
9. Jones, C.: Estimating Software Costs: Bringing Realism to Estimating, 2nd edn. McGraw-Hill Education, New York (2007)
10. Karna, H., Gotovac, S.: Estimating software development effort using Bayesian networks. In: 23rd International Conference on Software, Telecommunications and Computer Networks (SoftCOM), pp. 229–233 (2015). doi:10.1109/SOFTCOM.2015.7314091
11. Liu, X., Yang, Y.: A new approach to learn the projection of latent Causal Bayesian networks. In: International Conference on Systems and Informatics, ICSAI 2012, pp. 1083–1087 (2012)
12. López-Martínez, J., Juárez-Ramírez, R., Ramírez-Noriega, A., Licea, G., Navarro-Almanza, R.: Estimating user stories' complexity and importance in Scrum with Bayesian networks. In: Rocha, Á., Correia, A.M., Adeli, H., Reis, L.P., Costanzo, S. (eds.) Recent Advances in Information Systems and Technologies. Springer, pp. 205–214 (2017) (**Chap. 21**)
13. Mahnič, V., Hovelja, T.: On using planning poker for estimating user stories. J. Syst. Softw. **85**(9), 2086–2095 (2012)
14. Martel, A.: Gestión Práctica de Proyectos Con Scrum: Desarrollo de Software Ágil Para El Scrum Master (3ra.). CreateSpace Independent Publishing Platform (2014). Retrieved from https://books.google.com.mx/books?id=nEocjgEACAAJ
15. Mendes, E. Knowledge representation using Bayesian networks—a case study in Web effort estimation. In: Proceedings of the World Congress on Information and Communication Technologies, pp. 612–617 (2011)
16. Moløkken-Østvold, K., Haugen, N.C., Benestad, H.C.: Using planning poker for combining expert estimates in software projects. J. Syst. Softw. **81**(12), 2106–2117 (2008)
17. Mundra, A., Misra, S., Dhawale, C.A.: Practical scrum-scrum team: way to produce successful and quality software. In: Proceedings of the 13th International Conference on Computational Science and Its Applications, ICCSA 2013, pp. 119–123 (2013)
18. Nassif, A.B., Capretz, L.F., Ho, D.: Estimating software effort based on use case point model using Sugeno Fuzzy inference system. In: 23rd IEEE International Conference on Tools with Artificial Intelligence (ICTAI), pp. 393–398 (2011)
19. Overhage, S., Schlauderer, S.: Investigating the long-term acceptance of agile methodologies: an empirical study of developer perceptions in Scrum projects. In: 45th Hawaii International Conference on System Sciences, IEEE, pp. 5452–5461 (2012)
20. Owais, M., Ramakishore, R.: Effort, duration and cost estimation in agile software development. In: 9th International Conference on Contemporary Computing (IC3), pp. 1–5 (2016). doi:10.1109/IC3.2016.7880216
21. Pauly, D., Basten, D.: Do daily Scrums have to take place each day? A case study of customized Scrum principles at an e-commerce company. In: Hawaii International Conference on System Sciences, pp. 5074–5083 (2015)
22. Pham, A., Pham, P.V.: Scrum in Action: Agile Software Project Management and Development, 1st edn. Course Technology, Boston (2012)
23. Popli, R., Chauhan, N.: Agile estimation using people and project related factors. In: 2014 International Conference on Computing for Sustainable Global Development (INDIACom), pp. 564–569 (2014)
24. Popli, R., Chauhan, N.: Cost and effort estimation in agile software development. In: 2014 International Conference on Optimization, Reliabilty, and Information Technology (ICROIT), pp. 57–61 (2014)
25. Raith, F., Richter, I., Lindermeier, R., Klinker, G.: Identification of inaccurate effort estimates in Agile software development. In: 20th Asia-Pacific Software Engineering Conference (APSEC), pp 67–72 (2013)
26. Ramírez-Noriega, A., Juarez-Ramirez, R., Navarro, R., López-Martínez, J.: Using Bayesian networks to obtain the task's parameters for schedule planning in Scrum. In: 4th International Conference in Software Engineering Research and Innovation, vol. 1, pp. 167–174 (2016)
27. Ruggeri, F., Faltin, F., Kenett, R.: Bayesian networks. Encycl. Stat. Qual. Reliab **1**(1), 4 (2007)
28. Santhi, R., Priya, B., Nandhini, J.: Review of intelligent tutoring systems using bayesian approach. arXiv preprint arXiv:1302.7081 (2013)
29. Schwaber, K., Sutherland, J.: The scrum guide. Scrum org, October 2(July), p. 17 (2011)
30. Zahraoui, H., Abdou, M., Idrissi, J.: Adjusting story points calculation in scrum effort & time estimation. In: 10th International Conference on Intelligent Systems: Theories and Applications (SITA), pp. 1–8. IEEE, Rabat, Morocco (2015)
31. Zare, F., Zare, H.K., Fallahnezhad, M.S.: Software effort estimation based on the optimal Bayesian belief network. Appl. Soft Comput. **49**, 968–980 (2016). doi:10.1016/j.asoc.2016.08.004

Janeth López-Martínez has a Degree in Computer Systems at Universidad de Occidente in Sinaloa Mexico. She is currently a Masters student in Computer Science at Universidad Autónoma de Baja California in Tijuana México. She has interest about Estimation in Agile Methodologies in the Industry of Software Development.

Alan Ramírez-Noriega acquired the Master degree in Applied Informatic at Universidad Autónoma de Sinaloa in 2014. He is a Ph.D. candidate at Universidad Autónoma de Baja California from 2014. The main areas of interest are Intelligent Tutoring System, Knowledge Representation, and Text Mining.

Guillermo Licea is a full-time professor at Universidad Autónoma de Baja California in Tijuana México, since 1991. He has a Ph.D. in Computer Science from CICESE Research Center. He is a member of the research and development group "Software technology in education," and his research interests include software engineering and mobile application development.

Reyes Juárez-Ramírez received the Master degree in computer science from the Scientific Research and Higher Education Center in Ensenada in 2000, and the Ph.D. degree in computer science from the Universidad Autónoma de Baja California in 2008. He is currently a Professor and Researcher in the Faculty of Chemical Sciences and Engineering, Autonomous University of Baja California. He has two main areas of interest: Software Engineering and Human–Computer Interaction.

Samantha Jiménez is a Ph.D. student at Universidad Autónoma de Baja California (Tijuana, Mexico). She received the Bachelor's degree in 2011 in Computer Systems and Master's in Engineering in 2013 from University of Colima (Colima, México). Her research interests are in the areas of Human-Computer-Interaction, Dialogue Systems, Affective Computing, Multi-agent Systems and Evolutionary Computing.