Estimating User Stories' Complexity and Importance in Scrum with Bayesian Networks

Janeth López-Martínez(✉), Reyes Juárez-Ramírez, Alan Ramírez-Noriega, Guillermo Licea, and Raúl Navarro-Almanza

Universidad Autónoma de Baja California, Tijuana, Mexico
{patricia.lopezm,reyesjua,alan.david.ramirez.noriega, glicea,rnavarro}@uabc.edu.mx

Abstract. Planning Poker is a light-weight technique for estimating the size of user stories, in face-to-face interaction and discussions. Planning Poker is generally used with Scrum. Planning Poker has a lot of benefits, however, this method is not entirely efficient because the result is always based on the observation of an expert. This paper proposes a new model to estimate the complexity and importance of user stories based on Planning Poker in the context of Scrum. The goal of this work is to facilitate the decision-making of newbie developers when they estimate user stories' parameters. Hence, the decision of each member would be clearer to understand than when the complexity is taken as a whole. We use a Bayesian Network to co-relate factors to have accurate in the estimation. The Bayesian Network gives the complexity of a user story, according to the Fibonacci scale used in Planning Poker.

Keywords: Planning Poker · Complexity · Scrum · User story estimation

1 Introduction

Accuracy in effort estimation is an essential factor for planning software projects to avoid budget overruns and delayed dates of delivery; otherwise, often results in poor software quality [8,9]. Scrum does not provide a unique estimation technique; however, the most used is Planning Poker, which is a light-weight technique for estimating the size of user stories (A User story is an Independent, Negotiable, Valuable, Estimable, Small, Testable requirement), with face-to-face interaction and discussions [4,6,9].

Planning poker has many benefits; however, this method is not always efficient because the result is always based on the observation of an expert and his/her experience. The story-points are a relative value and cannot be easily related to the time duration [4,8,9]. Moreover, the team member decision is unclear, by taking into account only the complexity in general. It is necessary

Á. Rocha et al. (eds.), *Recent Advances in Information Systems and Technologies*,
Advances in Intelligent Systems and Computing 569, DOI 10.1007/978-3-319-56535-4_21

to break down this variable into its elements, in order to clearly establish how developers chose the decision.

Different works have faced this problem [6,9,13] in different ways and having good results, but without considering the uncertainty introduced by person's subjectivity. A good theory for handling the uncertainty is that proposed by Bayesian Networks (BN). A BN expresses the causal relationship between random variables of a domain of knowledge [11]. Throughout a BN, we can show relationships between different variables [12]. With these capabilities, knowledge and experience of people and experts can be represented in a BN. Proposals have been presented to deal this problem using BN, but these are still models [3] or they consider another different approach from ours [14].

This paper proposes a model to estimate the complexity and the importance of user stories based on Planning Poker in the context of Scrum. Five factors were identified for the estimation of user stories: complexity broke down into (1) experience, (2) time and (3) effort, and the importance defined in terms of (4) priority and (5) the value of the user story. The goal of this work is to facilitate the decision-making of newbie developers when they estimate an user story. The decision of each member would be clearer by decomposing a complex decision into simpler and more precise factors.

The model is based on previous work [10], where the proposed factors were validated, but Non-statistical tests were performed. Some changes were made to the model to increase the precision of estimates, such as including more variables and adding more states to the BNs. The linear interpolation formula was used instead of the equation of the previous article, and the Spearman correlation test was used to validate our proposal; this test was used due that the behavior of the data.

This paper is organized as follows. Section 2 contains a background on complexity as the main topic to focus in this paper. Section 3 shows the related work. Section 4 describes how the Knowledge Representation (KR) was done through Bayesian Network (BN). Section 5 shows the experiment made and the results. Finally, conclusions and references are shown.

2 Important Concepts

Complexity is an ambiguous subject with no exact definition agreed upon by software researchers. 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 either to analyze the complexity of problems, code and data [2].

For this work the complexity will be considered for two main aspects, the time and the effort necessary to implement a user story. Time is related to how laborious may be the user story, 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 but could be implemented easily [8].

Effort refers to the amount of cognitive work that is required to analyze, design, implement, and resolve a user story [9].

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. Then experience refers to the knowledge that the person possesses based on similar projects [1,7]. Based on this, having developers more experience then less complexity will be in the project; otherwise, the having developers less experience then greater the complexity will be in the project.

The importance in the user stories is given by the contribution that a user story makes to the project in general. Two main aspects about the importance are considered in this paper: The priority and the value of the user story. 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 the US which has more relationships with others then the US it has higher priority because this kinds of user stories are the pillars of the project [5,13]. 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 have the highest business value must be delivered in the first Sprint [13].

3 Related Work

Next, we describe some studies with a similar focus to our article.

In reference [8] an algorithmic estimation method was proposed. 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 three cases in which different levels of factors are taken into account and compared. The method used is different from a Bayesian Network for estimations. Besides the authors did not decompose the complexity into more variables.

Karna et al. [3] presented a Bayesian model, including the relevant entities that are involved in the formation of the effort estimation. They considered mainly tree entities involved in the estimation process: projects, work items, and estimators. Karna et al. takes the complexity as a single variable, but this variable needs to be broken down into other factors. Even though this is a good proposal, it needs to be tested.

Zare et al. [14] presented a three-level Bayesian network based on COCOMO components to estimate the needed effort for the software development, so that the estimated effort is modified using the optimal coefficient resulted from optimal control designed by a genetic algorithm. They used the attributes defined in COCOMO, the complexity is defined as a single variable. Our approach breaks down the variable complexity and is focused on estimating user stories.

Complexity is a significant variable, but as we can see, most of these works are considering it as a whole. Our proposal considers splitting the complexity into different aspects to be understood more clearly, and contributing to get better estimations.

4 Building the Bayesian Network

As a resume, we can precise the set of variables considered for the estimation: (1) experience, (2) time and (3) effort, (4) priority, and (5) the value of the user story.

This section exposes how the BN is built (Fig. 1) showing its behavior through the variables, and establishing the quantitative and qualitative relationships.

The Bayesian network is built through the following steps: (1) the BN qualitative part (structure) is developed based in the group of variables. (2) The quantitative part is built based in the degree of acceptance of each variable. The quantitative part needs: (a) Get values for the Relationships between variables, (b) Calculate a priori values for the first level variables, and (c) Generate the conditional probability tables.

A BN is formed by nodes, relations and Conditional Probability Tables (CPT). The factors in which the complexity and the importance of user stories were decomposed are represented as BN variables in the nodes. These variables have influence over other; this influence is represented by relations. The variables have possible situations in which they can remain; these situations are called states, it is a way of discretizing the continuous values. Finally, CPT show the probability that an event will occur based on the combination of the variables and the value of their states.

The BN forms a hierarchical structure with variables organized by levels (see Fig. 1).

First level variables: These are nodes that are found in the network extremes. These are evidence collector; team members enter their estimates into these variables. Five states were defined for these variables; which are measured in the scale: *very_low*, *low*, *medium*, *high*, and *very_high*. Second level variables: The network has nodes that group the first level variables. These variables help us to organize the information hierarchically. Besides, 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 user story.

4.1 Obtaining the Quantitative Part

Two statistics aspects were needed to build the quantitative network part:

(1) The values of the Relationships between variables: This value was obtained by the knowledge and experience of the Scrum development team members [10]. These values were calculated through the formulas of the next section. Each variable was validated considering how many are you agree, in using the variable as evaluation criteria in the user story complexity and importance.

(2) A priori values of the first level variables: It is referring how is evaluated (*very_low*, *low*, *medium*, *high*, and *very_high* for Experience, Time and Effort) and (*low*, *medium*, and *high* for Priority and Task Value) other project tasks finalized. According to the first level variables.

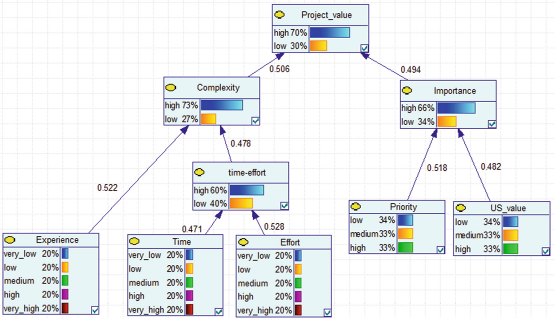

Fig. 1. Bayesian network with planning poker variables

Values for the Relations Between Variables. We used the next formula (1) to establish the relation unnormalized:

$$p = \frac{\sum_i^n (v_i * f_i)}{\sum_i^n f_i} \tag{1}$$

Where v_i represents the i element value of the scale. f_i represents the i element, frequency of the scale. p represents the variable, and the relation unnormalized between 0 and 1. We considered $n = 5$ for the five-point Likert scale. The normalized relationships between 0 and 1 are obtained with the formula 2:

$$rn_i = \frac{p_i}{\sum_{i=1}^m p_i} \tag{2}$$

Where p_i is the i element value of the set P $(P = p_1, ..., p_n)$ unnormalized. p refers to the variable that has a causal influence over another, where all the elements of P have influence over the same variable. In our schema, the *time* and $effort$ variables have influence over the *complexity* variable. Besides, *priority* and *task_value* have influence over the *importance* variable. The variable m refers to the total of variables that influence over another.

The Fig. 1 shows the weight of each relation, calculated with the previous formula, this represent the opinion of students. The network has for the first level variables the following results: *Experience* (0.522), *time* (0.471), $effort$ (0.528), *priority* (0.518), *task_value* (0.482).

The second level variables (*complexity*, *importance*, and $time - effort$) obtain their values through the formula 3:

$$a = \frac{\sum_{i=1}^m p_i}{m} \tag{3}$$

Where m represents the total of variables that have influence over another. The un-weighted value is represented by p. It is calculated with the formula 1. The normalized value is calculated with the formula 2. But, taking into account to the variable a instead of p, we obtain the next variable values: *complexity* (0.506), *importance* (0.494), and $time - effort$ (0.478).

The Fig. 1 shows the values for *complexity*, $time - effort$, and *importance*. Although the network has not evidenced shows priori values. These values are in the middle point between the low value and high value. We consider that any user story needs work to end. So, the minimum value is not 0; each node defines according to the CPT this values. For instance, for the complexity variable, the minimum value is 0.52 and the maximum value is 1.00. The middle point is 0.73 (73%). That represents the value of the complexity variable in the Fig. 1.

A Priori Values for the First Level Variables. We have not 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 maximum value (1) is distributed between the five states (*very_low*, *low*, *medium*, *high*, and *very_high*).

Constructing Conditional Probability Tables. Once we have the network structure and the relations weight, the last step is to assign a conditional probability table to each node in the structure.

We defined a set of the scales that represented as $S = \{s_1, s_2, ..., s_m\}$. The weighted states were allocated as *very_low* (1), *low* (2), *medium* (3), *high* (4), and *very_high* (5) for *experience*, *time*, and *effort* variables. The weighted values were 0.067, 0.133, 0.200, 0.267, and 0.333 respectively. The weighted states to *priority* and *US_value* were *low* (1), *medium* (2), and *high* (3). The weighted values were 0.17, 0.33, and 0.50 respectively. Also, we defined a set to represent the variable weight that has a causal influence over another variable. Its last set is represented as $P = \{p_1, p_2, \ldots, p_n\}$.

We show the process through the matrix shown in Eq. 4. Firstly, a multiplication of values is needed to establish the weighted of the variable relation value with the scale value $(p_1 * s_1)$. Secondly, a combination between the result set in the first step is made. Each combination obtains a unique value adding each element of the combination $((p_1 * s_1) + (p_2 * s_1) + \cdots + (p_n * s_1))$.

$$W = \begin{bmatrix} w_{11} & w_{12} & \cdots & w_{1m} \\ w_{21} & w_{22} & \cdots & w_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ w_{n1} & w_{n2} & \cdots & w_{nm} \end{bmatrix} \tag{4}$$

Where $w_{11} = ((p_1 * s_1) + (p_2 * s_1) + \cdots + (p_n * s_1))$, $w_{12} = ((p_1 * s_1) + (p_2 * s_1) + \cdots + (p_n * s_2))$, $w_{1m} = ((p_1 * s_1) + (p_2 * s_1) + \cdots + (p_n * s_m))$, ..., $w_{n1} = ((p_1 * s_m) + (p_2 * s_m) + \cdots + (p_n * s_1))$, $w_{n2} = ((p_1 * s_m) + (p_2 * s_m) + \cdots + (p_n * s_2))$, $w_{nm} = ((p_1 * s_m) + (p_2 * s_m) + \cdots + (p_n * s_m))$.

After, the maximum value of previous steps is obtained (max). Its value is used to obtain the final values in proportion to the maximum value. So, the matrix in Eq. 4 is divided between the maximum value $(W = W/max)$.

Finally, the CPT is obtained to one variable. The process is repeated for each variable.

5 Testing the Bayesian Network

5.1 Experiment

The study aims is to test the accuracy of the model developed trough a group Software newbie developers. The simulation of the proposal was held with the participation of students of the Computer Engineering undergraduate program. The degree of the participants in the range corresponding 7th and 8th semester. They have experience with Scrum due to projects with software companies. The experiment consisted in a survey with a set of questions for four user stories, estimating the complexity and the user story's value in the project.

Real software projects were considered, consisting in Information systems, such biometric detection, electronic medical profiles, and a sensor-based system in the context of Internet of Things. It works on different programming languages (C#, Java, Python), as well as a series of support frameworks (Entity Framework, Windows Presentation Foundation, NET MVC).

The user stories were described in detail to estimate its complexity regarding the factors proposed (experience, time, effort). Besides, to define other factors (priority and US value) to calculate the US value in the project. The options for each question were in a qualitative way, considering the same states of the first level variables. The answers given by the students were adapted to be input to the Bayesian network system to assess its results. The scale used to assess the complexity of user stories was a modified Fibonacci, consisting of the following series 1, 2, 3, 5, 8, 13, 20, 40, 100.

5.2 Results

We defined a value to represent each factor of the scale. In the case of $experience$, $time$, and $effort$ were defined: very low (1), low (2), normal (3), high (4), and very high (5). For *priority* and US_value were defined: low (1), normal (2), and high (3). All students evaluated each of four user story. In Table 1, Cases 1–6 correspond to US 1, Cases 7–12 correspond to US 2, Cases 13–18 correspond to US 3, and cases 19–24 correspond to US 4. The answer to each question per US is shown in the Table 1.

Also, the results of critical variables (*complexity*, *importance*, and *project_value*) are shown in Table 1 (columns "compl.", "Impor", and "P. Value" respectively). The values returned by the Bayesian Network after the interaction with students (see column complexity) are converted to modified Fibonacci scale (see column BN to Fib.). These values are obtained through the Eq. 5. The student's evaluation on the same scale is presented in the last column.

$$Y = Y1 + [(X - X1/X2 - X1) * (Y2 - Y1)] \tag{5}$$

Where X is the complexity value of the BN, $X1$ is the threshold lower of the BN, $X2$ is the threshold higher of the BN, $Y1$ is the threshold lower of cards position y $Y2$ is the threshold higher of cards position.

Table 1. Results of the user story evaluation

Case	User story	Experience	Time	Effort	Priority	US value	Compl.	Impor.	P. value	BNProb.	StudCplx
1	1	3	3	2	1	3	70%	63%	66%	5	3
2	1	3	4	2	2	3	73%	81%	77%	8	5
3	1	1	1	1	2	2	74%	66%	70%	5	1
4	1	1	3	3	2	3	87%	81%	84%	20	3
5	1	3	4	1	2	3	69%	81%	75%	5	5
6	1	3	4	3	2	2	76%	66%	71%	8	3
7	2	3	3	2	2	1	70%	52%	61%	8	3
8	2	3	2	2	1	2	67%	48%	58%	5	3
9	2	1	2	1	1	1	77%	34%	56%	13	2
10	2	3	1	1	1	2	60%	48%	54%	3	2
11	2	3	4	1	2	2	69%	66%	68%	5	2
12	2	3	4	3	2	2	76%	66%	71%	8	3
13	3	3	2	2	2	1	67%	52%	60%	5	3
14	3	3	2	2	2	2	67%	66%	66%	5	3
15	3	1	2	2	2	2	81%	66%	74%	13	2
16	3	3	1	1	1	2	60%	48%	74%	3	1
17	3	3	3	1	2	2	66%	66%	66%	5	2
18	3	3	4	4	3	2	80%	85%	82%	13	5
19	4	3	2	2	2	3	67%	81%	74%	5	3
20	4	3	2	2	3	2	67%	85%	76%	5	3
21	4	1	3	2	3	3	84%	100%	92%	20	3
22	4	2	2	1	2	2	70%	66%	68%	8	3
23	4	3	4	1	2	3	69%	81%	75%	5	3
24	4	3	4	4	3	3	80%	100%	90%	13	5

The Table 1 in columns BNProb. and StudCplx use modified Fibonacci scale values. However, when the Eq. 5 and the correlation test are employed, the number of the card is used 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 Fibonnaci value 100 corresponds to the card 9. This provides the possibility to adapt to any valuation possible.

5.3 Hypothesis Testing

We used two main variables $BNProb$ and $StudCplx$. $BNProb$ represents the probabilities returned by the Bayesian Network to assign complexity to the user story.

$StudCplx$ represents the values returned by the student to assign complexity to the user story.

Two kinds of tests were realized with Spearman test: (1) Test A where all estimations of the student were used; and (2) Test B where all estimations of the students with low experience were discarded. Low experience means that the student has short time using Planning Poker for estimating (number one in column experience in Table 1). The Hypothesis are H_0: $BNProb$ and $StudCplx$ variables has not correlation and H_1: $BNProb$ and $StudCplx$ variables has correlation.

The same significance level was used to test A and B (0.05). Test A gave a correlation of 0.340. But, $P - value$ was greater than the significance level

(0.104 > 0.05). Therefore, $BNProb$ and $StudCplx$ variables have not correlation (H_0) taking into account all opinions. Test B showed a correlation of 0.446. This case gave a $P - value$ lower than the significance level ($0.049 < 0.05$). Hence, $BNProb$ and $StudCplx$ variables have a correlation (H_1) discarding students' opinions with low experience.

5.4 Discussion

Different ways to define the final value in the Fibonacci scale are defined. This depends on Scrum master and the development team. However, if a simple average of values is defined, a complexity of 8 is obtained for US 1, 5 for US 2, 5 for US 3 and 8 for US 4, on the Fibonacci scale used.

Two tests were analyzed (A and B). Spearman correlation was not accomplished to test A. But, test B gave positive correlation. We noticed that students' estimation with low experience influenced to the final result; so, we did the same test without this kind of students obtaining positive results. A weighted to balance the negative result of students with low experience is necessary, modifying the formula 5 could be obtained. Test B showed a correlation of 0.446. The $BNProb$ and $StudCplx$ variables have moderate correlation because the correlation of test B is between 0.4 and 0.6. This gives the possibility to continue working to obtain a better correlation.

Previously, $project\_value$ was defined as how essential is a US in the project is. The value is obtained as the result of BNProb. The result can be seen in Table 1. Taking into account the same process of the complexity, we obtained a value of 74% for US 1, 61% for US 2, 70% for US 3, and 79% for US 4. These values indicate that the US 4 has more probability to be essential to the project. Therefore, we need to put special attention to this US.

6 Conclusions

This study presents a proposal to assign complexity and importance to User Stories taking into account a set of variables that consider more details of these factors, using Planning Poker. Instead of taking into account the complexity as a unique value, it was divided into three variables. So, the attention is put on each variable at a time, obtaining more precision in developers's decision-making. Moreover, taking into account the priority and User Story's value, more details are considered to estimate a task. This allows advantages such as establishing priority between User Stories in the project.

A structure of knowledge was developed to represent the knowledge of Scrum teams through a Bayesian Network. The conditional probability tables were built with information of Scrum teams from academy and industry.

Applying our proposed model, the Bayesian Network results show an approximate value to the students' estimation. Using the Spearman test, a correlation value of 0.446 was obtained. The best result was obtained when students with low experience are not considered because their estimation are imprecise. This allows to see that considering each variable itself at a time, a more accurate estimation can be performed.

Three main aspects must be worked as a future work: First, Modifying the interpolation formula (Eq. 5) to obtain better accuracy when newbie developers with low experience gives an estimation. Second, Building a Bayesian Network using expert knowledge. Finally, when the accuracy will be improved, we are considering developing a mobile application to automate the estimation process of tasks using Planning Poker.

References

1. Grimstad, S., Jørgensen, M.: A framework for the analysis of software cost estimation accuracy. In: Proceedings of the 2006 ACM/IEEE International Symposium on Empirical Software Engineering - ISESE 2006, p. 58 (2006). http://portal.acm.org/citation.cfm?doid=1159733.1159745
2. Jones, C.: Estimación de Costos y Administración de proyectos de Software. Segunda edn. (2007)
3. Karna, H., Gotovac, S.: Estimating software development effort using Bayesian networks, pp. 229–233 (2015)
4. Mahnič, V., Hovelja, T.: On using planning poker for estimating user stories. J. Syst. Softw. **85**(9), 2086–2095 (2012)
5. Martel, A.: Gestión práctica de proyectos con Scrum: Desarrollo de software ágil para el Scrum Master (2014)
6. 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). http://dx.doi.org/10.1016/j.jss.2008.03.058
7. 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)
8. 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)
9. Raith, F., Richter, I., Lindermeier, R., Klinker, G.: Identification of inaccurate effort estimates in agile software development. In: 2013 20th Asia-Pacific Software Engineering Conference (APSEC), pp. 67–72 (2013)
10. 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: 2016 4th International Conference in Software Engineering Research and Innovation, vol. 1, pp. 167–174 (2016)
11. Santhi, R., Priya, B., Nandhini, J.: Review of intelligent tutoring systems using bayesian approach (2013). arXiv preprint: arXiv:1302.7081, http://arxiv.org/abs/1302.7081
12. Hu, X.-X., Hui, W.H.W., Shuo, W.S.W.: Using expert's knowledge to build bayesian networks. In: 2007 International Conference on Computational Intelligence and Security Workshops (CISW 2007), pp. 220–223 (2007)
13. Zahraoui, H., Abdou, M., Idrissi, J.: Adjusting story points calculation in scrum effort & time estimation (2015)
14. Zare, F., Khademi Zare, H., Fallahnezhad, M.S.: Software effort estimation based on the optimal Bayesian belief network. Appl. Soft Comput. **49**, 968–980 (2016)