

# User Story Estimation Based on the Complexity Decomposition Using Bayesian Networks

M. Durán<sup>a,\*</sup>, R. Juárez-Ramírez<sup>a,\*\*</sup>, S. Jiménez<sup>b,\*\*\*</sup>, and C. Tona<sup>a,\*\*\*\*</sup>

<sup>a</sup>Autonomous University of Baja California (UABC) Avenida Universidad no. 14418, UABC, Parque Internacional Industrial Tijuana, Tijuana, 22424 B.C., México

<sup>b</sup>Tijuana Institute of Technology (Instituto Tecnológico de Tijuana) Tomas Aquino, Tijuana, 22414 B.C., México

\*e-mail: mayra.duran@uabc.edu.mx

\*\*e-mail: reyesjua@uabc.edu.mx

\*\*\*e-mail: samantha.jimenez@tectijuana.edu.mx

\*\*\*\*e-mail: tona.claudia@uabc.edu.mx

Received April 13, 2020; revised May 25, 2020; accepted July 18, 2020

**Abstract**—Currently, in Scrum, there are different methods to estimate user stories in terms of effort or complexity. Most of the existing techniques consider factors in a fine grain level; these techniques are not always accurate. Although Planning Poker is the most used method in Scrum to estimate user stories, it is primarily effective in experienced teams since the estimation mostly depends on the observation of experts, but it is difficult when is used by inexperienced teams. In this paper, we present a proposal for complexity decomposition in a coarse grain level, in order to consider important factors for complexity estimation. We use a Bayesian network to represent those factors and their relations. The edges of the network are weighted with the judge of professional practitioners about the importance of the factors. The nodes of the network represent the factors. During the user estimation phase, the Scrum team members introduce the values for each factor; in this way, the network generates a value for the complexity of a User story, which is transformed in a Planning Poker card number, which represents the story points. The purpose of this research is to provide to development teams without experience or without historical data, a method to estimate the complexity of user stories through a model focused on the human aspects of developers.

**Keywords:** user stories, fine-grained complexity, estimation, Bayesian network, expert judge

**DOI:** 10.1134/S0361768820080095

## 1. INTRODUCTION

Scrum is an agile development method that is defined as an iterative, incremental and empirical process to manage and control development work [1–3]. It is a framework that presents a set of practices to maintain visibility, inspection, and adaptation of software development projects [4, 5].

A key element in the Scrum framework is the user stories (US). The first version of user stories is a short (a sentence or two), simple, and specific description of an interaction between user and product to develop [6]. The description of a User story typically follows a simple template [7]: *As a < type of user >, I want < some goal > so that < some reason >*. There are no action-items or low-level tasks in the Scrum Product Backlog; however, an entry always add value for the customer.

In practical terms, in the Sprint planning session, the functionality expressed in the coarse-grain version is written down in detailed (fine-grain) user stories

and stored in the Scrum Product Backlog. All the entries in a Sprint are estimated in tasks [8, 9].

A fine-grain version of US defines features and requirements that provide value for the user or client [10]. Also, these requirements should provide information for the developer to do a reasonable estimation [8]. In a fine-grain version of user stories, specific details are expressed, such as acceptance criteria, test cases, sequence flows, between others.

The Scrum Team is also a key element. The main responsibilities are [1–3]: Own story estimates and implementation of value, own emergent design, contribute to intentional architecture, and contribute to backlog refinement and creation of stories.

A software project involves the estimation of the cost, effort, size, and duration [11]. Traditional software teams give estimates in a time format: days, weeks, months. Many agile teams, however, have transitioned to story points, which focus is on complexity. Based on this, when estimating the Product

Backlog, smart teams go for story points rather than hours.

Many factors should be considered while estimating story points [8, 9, 12, 13]: Complexity of the story; risks, such as the team's inexperience with developing this story; implementations issues, deployment requirements; and interdependencies.

The better estimation depends on experience and prior knowledge of the developers which are useful to deduce and consider all possible factors, having a great influence on the accuracy of the estimation. For that reason, the estimation process can be problematic for novice developers, since they do not have enough experience [14]. The uncertainty in requirements also impact negatively the estimate [15].

In Scrum, to estimate a US, several existing techniques can be used such as Planning Poker [8], Wideband Delphi [16], Fist of Five, Affinity Estimation [5], and t-shirt sizes [17, 18]. Planning Poker is the most used method in Scrum to do estimation of US.

Planning Poker is a technique based on expert judgment that consists in assigning story points to each user story to determine its complexity [16]. Estimation is done for each US in terms of story points that represent the complexity for its implementation in the project [16]. Such complexity manifests itself in the fact that software has a large number of parts that have many interactions [19].

Planning Poker has many benefits such as it is easy to use in expert teams, promoted team participation and collaboration and allows the opinion of the team members. However, this method has some disadvantages because the results are always based on developer experience [20], particularly at expert level. The main problem in the use of Planning Poker is the high subjectivity in the US estimation. From that reason, using Planning Poker is complicated and unreliable, especially for team members without experience. Moreover, each team member decision is unclear because they consider the complexity in general terms, when in reality complexity is related to several attributes such as experience in programming language [21, 22], experience in previous projects [23, 24], familiarity with project [25], experience with technology [21, 26], team size [27, 28], project duration [28, 23], number of tasks [29, 30], task dependencies [31, 32], among others [16, 33, 34]. For that reason, it is necessary to break down complexity into its elements to establish how the developer decided [11].

The existing estimation methods take into account aspects of project, technical, and aspects of the company, but very few consider important personal aspects for estimating [20]. The main problem on the existing methods is that they considered the user stories estimation assessing the complexity as a whole [16, 33], and several authors suggested that the complexity estimation is composed of several attributes [8, 15, 34, 35].

In this paper we present a proposal of a set of attributes that influence the complexity estimation in user stories. Part of the attributes have gathered from a Systematic Literature Review (SLR) presented in our previous work [36]. The selected set of attributes was validated by industry practitioners, supporting with their opinion the importance of such attributes in estimates. We propose a Bayesian network model, where each attribute/factor represents a node of the network, with a ponderation in the correspondent loaded edge, extracted from the practitioners' opinion. We considered the following attributes: Developer experience (experience with technology, experience with language, and experience in previous projects), developer skills, knowledge of the project theme, and technical complexity (dependency, size of the user story (connection functions, data functions)). The opinion of practitioners allows taking advantage from such kind expert judgment. Our proposed Bayesian model facilitates the estimation in a significant way; it is used with the Planning Poker technique. During the estimation, developers provide values for the attributes, and the Bayesian model calculates the card value, representing the story points.

In this article we present a model for estimating user stories that mostly consider personal attributes of developers through the breakdown of the experience attribute. The purpose of this research is to provide development teams without experience or without historical data a method that positively favors the accuracy of their estimates.

This article is organized as follows. Section 2 describes the background. Section 3 presents the related work. Section 4 shows the methodology followed by our research. Section 5 describes the attributes used in the model proposed. Section 6 presents the Bayesian network construction. Section 7 presents how Bayesian network was proved. Section 8 presents results and discussion, and finally, Section 9 presents the conclusions.

## 2. BACKGROUND

### 2.1. Scrum

Scrum is the most popular agile methodology in the software industry. By using Scrum practices, several companies have improved their quality and productivity. Scrum is an agile method of development defined as an iterative, incremental, and empirical process to manage and control the development of a project [37, 38]. Scrum has three main roles: Product Owner (PO), Scrum Master (SM), and Scrum Team (ST). The PO represents the voice of the customer; the main responsibility of this role is the creation of the product backlog, which is a list of requirements sorted based on the priority. Scrum starts from the idea of planning based on business objectives prioritizing those that bring more value and waiting to detail

objectives and tasks as the moment of construction of these objectives approaches [39].

In Scrum there are different techniques to estimate, however, Planning Poker is the most common [8], which is described in the next section.

## 2.2. Planning Poker

Planning Poker is a technique mostly used in Scrum to estimate the effort of user stories in a project by members of the development team responsible for implementation. The game utilizes playing cards printed with numbers, these cards are based on a modified Fibonacci sequence (1, 2, 3, 5, 8, 13, 21, 40, 100). Equipped with playing cards, the product owner and all members of the development unit meet to discuss Product Backlog requirements for the purpose of reaching consensus-based estimates [11]. What is sought when using Planning Poker is to assign story points to user stories to determine their complexity [16].

## 2.3. User Stories

User stories are a concise notation for expressing requirements that is increasingly employed in agile requirements engineering and in agile development. Indeed, they have become the most commonly used requirements notation in agile projects and their adoption has been fostered by their use in numerous books about agile development. Despite some differences, all authors acknowledge the same three basic components of a user story [40, 7]:

1. A short piece of text describing and representing the user story.
2. Conversations between stakeholders to exchange perspectives on the user story.
3. Acceptance criteria. These are quality characteristics, functionalities, scenarios, business rules, external interfaces, constraints and data definitions.

The short piece of text representing the user story captures only the essential elements of a requirement: who it is for, what is expected from the system, and, optionally, why it is important [6].

A user story can be estimated at story points, which represent the complexity it will have in its implementation [16, 19].

## 2.4. Complexity

There exist many definitions of complexity. A very basic and simple definition of complexity is the number of different elements in a system alone as its complexity and the number of elements in a system and the possible relations among these elements as complexity [41]. 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 [8].

In the case of user stories, complexity is focused on the different elements of the functionality involved and its implementation aspects. Determining the complexity when estimating a user story is not a simple task since it involves many aspects to consider, however, there are previous works that work with Bayesian networks to address these types of problems, this is because of their probabilistic nature allows us to handle uncertainty [12].

## 2.5. Bayesian Networks

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, Bayesian networks combine principles from graph theory, probability theory, computer science, and statistics [42]. 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 Table (CPT) on  $V$  that shows the causal influence. The CPT is used to express how the potential states of the node's parents affect the posterior probability of the considered node [8]. They consist of two parts:

- The qualitative part is a graphic structure (graph) that describes the possible entities (attributes) and dependencies between them.
- The quantitative part is made up of conditioned probabilities that represent the uncertainty of the problem [43].

Bayesian networks assume that a node depends only on its parents and that each node is associated with a CPT which define the probability of each state in which a variable can be [43]. The structure of nodes is shown in Fig. 1, representing the relationship between the child node and parents' nodes.

In our proposal, the Bayesian network is used to represent the different factors to decompose complexity to estimate user stories.

## 3. RELATED WORKS

Researches have been done and plenty of proposals raised for effort estimation. Some of them have focused on complexity as one of the more important factors for estimation. Different mathematical tech-

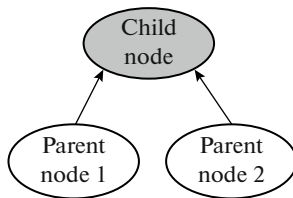


Fig. 1. Relationship between nodes.

niques and algorithms have been used in order to get better estimations. This section describes articles related to the estimation of complexity; some of these studies have a similar approach to ours.

In [12] an estimation based on tasks is presented. The authors considered factors such as complexity of function, complexity of formulary, complexity of reports, type of task, developer's abilities, and quality specification.

More generic methods have been focused on software projects developed with traditional methodologies. In [13], the authors considered factors such as team's experience, task's complexity, task's size, and accuracy in estimation. In [22], a wide set of factors is considered: technical complexity factors, environment complexity factors, stability requirements, team experience in application management, team motivation, and developers of partial time.

Martínez's work [8] proposed a model based on the decomposition of project value into two factors (complexity, importance), and five sub-factors (considered as attributes), which are: time, effort, experience, priority, and value of the user story. A Bayesian network structure is used to decompose project value; each factor represents a node in the network, and the weight of their relationships is represented in the edges. This model is based on the knowledge of experts to estimate user stories, which is used to determine the weights of the edges. In this way, a weighted network is generated to be used as a template for estimations.

In practical terms, this approach is used through questions that represent attributes in a Bayesian network; the values are obtained from the Scrum team members during each Sprint planning session. After a validation in practical cases, this approach obtained good correlations between conventional estimates and those obtained by the model. The main disadvantage of this model is that the authors considered coarse-grain factors, for example, the experience as a unique factor, while each of them should be divided into more sub-factors.

The study conducted by Dragicevic [12] presents a Bayesian network model suitable for predicting effort in tasks. It includes six attributes: new type of task, requirements complexity (form complexity, function complexity, complexity report, and specification quality) and developer skills. The author obtained good results with this approach since the accuracy of the

prediction for all the analyzed data was greater than 90%. The attributes used by the authors focus on aspects of the task and technicians that could be difficult to identify by novice developers; in addition, the authors considered the developer's skills as a single attribute.

The model proposed by Alostad [13] is a model based on fuzzy logic that can improve the estimation of effort in Scrum. The author considered four attributes: the experience of the development team, complexity of the task, size of the task and accuracy of the estimate. An accuracy level of up to 60% was obtained. Due to the improved accuracy in the estimates of the developers in each sprint, the author considers that the use of the model would obtain good results the more it is used, as it becomes more accurate over time and offers a better estimate of effort. This model examined only four elements, however, determining the complexity of a user story implies even more attributes than those mentioned by the author.

As we can see, most of the related work considers important aspects to estimate, however, they present two important aspects that could be disadvantages: 1) there is no consensus on the meaning of each factor and the level of granularity of composition, for instance, the complexity is expressed in a coarse grain level; 2) none one focuses on the personal aspects of the developers in a fine grain level.

Our proposal is inspired by the model proposed in [8]; we are trying to improve the granularity of some attributes, such as the complexity, and considering important attributes for the Scrum context such as personal aspects. Our motivation is to emphasize the personal aspects.

#### 4. METHODOLOGY

This section describes the methodology used to carry out this research.

- Step 1. Selection of model attributes: attributes were identified, analyzed, and selected to integrate the proposed model. Section 5 shows the attribute selection process.
- Step 2. Construction of the Bayesian network: the Bayesian network was built using the selected set of attributes, constructing the qualitative and quantitative part. Section 6 describes in detail how the network was built.
- Step 3. Testing the model: two different experiments were designed and executed in order to test the model, one for students and one for professionals. Section 7 presents the procedure used in each experiment.
- Step 4. Results and validation: Section 8 shows the results obtained. The results obtained were analyzed and compared with other investigations.

**Table 1.** Selected attributes

Attribute	Definition
Experience with technology	The developer knows and dominates the tools, methods, and technologies that will be used
Experience in previous projects	The knowledge that a developer owns based on similar projects and based on the accumulation of knowledge and skills
Experience with language	The developer knows and is able to effectively use the resources and limitations of the programming language
Developer skills	Knowledge and skills that a developer has (problems resolution, their availability, capacity, etc.)
Knowledge of the project theme	The developer knows and dominates the most important nature and concepts about the project theme
Dependence	How important a user story is because other User Stories depend on it
Connection functions	They are the user’s processing requirements such as inputs, outputs, and queries
Data functions	They are the user’s storage requirements, such as internal logical files and external interface files.

### 5. THE PROPOSED BAYESIAN NETWORK MODEL

This section the selection of the attributes of the model of this proposal is described. Currently most of the methods for estimating user stories are mostly based on technical or project aspects [20], due that we proposed this model that mostly considers the personal attributes of developers, without neglecting the technical aspects. Attributes and classifications of the systematic literature review were analyzed [36], which lists more than 50 attributes of different categories that can influence the estimation of user stories. In this article, more than 50 attributes were analyzed and classified, forming 5 categories: personal aspects, technical complexity, project, team, and organizational complexity.

As it is explained in the article, these categories were formed taking into account the opinion of the authors based on the experience of conducting real software projects. Because our model focuses on human aspects, our analysis is addressed on the attributes contemplated in the personal aspect’s category. The most relevant attributes were selected and those that were even suggested by professional practitioners: Experience with technology, experience with language, experience in previous projects, developer skills and knowledge of the project theme. Three of them are referred to experience, so that they are integrated in the experience node.

Although the focus of our model is the personal attributes, some related attributes are added to the technical complexity of user stories. Two more attributes were added, which are dependence and size. These attributes are important to know the technical complexity of a user story, in addition to being attributes commonly used in the estimation. The size node could be broken down further by adding two sub nodes, which are: connection functions and data

functions. The selected attributes to first level are described in Table 1. The selected attributes were organized by their nature and grouped into nodes. The final model is shown in Fig. 2. With this model we intend to improve the accuracy in the estimation of user stories.

### 6. BAYESIAN NETWORK CONSTRUCTION

This section shows how the Bayesian network was built. The qualitative part represents the structure formed by nodes and the relations and dependencies between nodes, this part is developed based on the group of attributes selected from the article [36]. The quantitative part is composed of CPT, for creating this, professionals gave their opinions.

#### 6.1. Qualitative Part

The qualitative part of the Bayesian network is composed of nodes and the relationship between them.

The relationships between the nodes are shown in Fig. 2. The Bayesian network is formed by a hierarchical structure which is organized by levels, each level represents a group of attributes or nodes:

- First level attributes: These nodes are the ones that collect information for the complexity estimation (Experience with technology, Experience in previous projects, Experience with language, Developer skills, Knowledge of the project theme, Dependence, Connection functions and Data functions)
- Second level attributes: These attributes help to organize nodes hierarchically (Experience, Technical complexity and Size).
- User stories complexity attribute/node: This attribute shows the final value to measure complexity estimation.

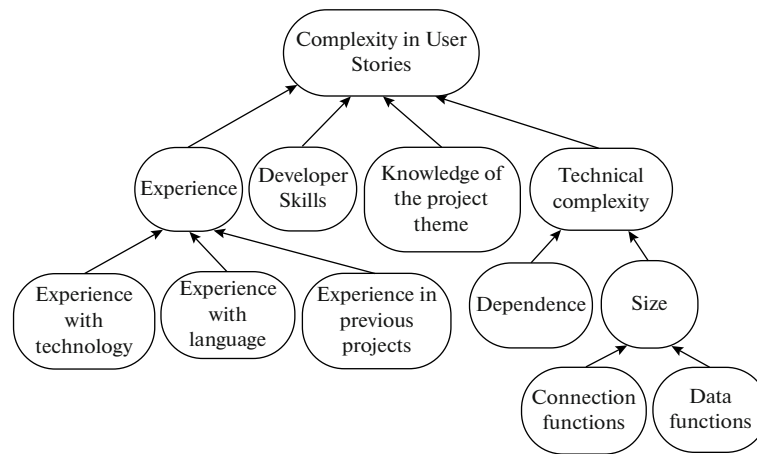


Fig. 2. Selected model.

## 6.2. Quantitative Part

To form the quantitative part of our Bayesian network, it is necessary to cover three important aspects: values between the relationships of the attributes, scales of values for the first level attributes and the construction of the CPT.

**6.2.1. Values of relations between attributes.** To obtain the values of the relationships between the attributes it was necessary to consult the opinion of the Scrum professionals, for which a survey was designed and applied, which is described below.

*6.2.1.1. Survey to determine the importance and value of the attributes within the model.* The objective of the survey was to validate the importance and value of the selected attributes within the model. Respondents were asked the degree of acceptance of the selected attributes. The survey was conducted digitally using google forms. The survey was applied to 21 professionals. Survey question were formulated to know the degree of acceptance of the proposed attributes, and questions to collect information regarding the experience of the respondents. A Likert scale of 5 points was used (Strongly disagree, in disagreement, Neutral, in agreement, totally agree). The survey design was validated taking into account the recommendations of Guidelines for Conducting Surveys in Software Engineering V1.1. [44]. The most relevant questions in the survey are shown below:

- Do you agree that the *experience with the technology* involved in a project is an important factor in determining the *developer's experience*?
- Do you agree that the *experience with the language* involved in a project is an important factor in determining the *developer's experience*?
- Do you agree that *experience in previous project* is an important factor in determining *developer experience*?

- Do you agree that the *connection function* of a user story is an important factor in determining the *size* of a user story?
- Do you agree that the *data function* of a user story is an important factor in determining the *size* of a user story?
- Do you agree that the *size* of a user story is an important factor in determining the *technical complexity* of a user story?
- Do you agree that *dependence* between user stories is an important factor in determining the *technical complexity* of a user story?
- Do you agree that the *developer's experience* is an important factor in determining the *complexity of a user story*?
- Do you agree that *developer skill* is an important factor in determining the *complexity of a user story*?
- Do you agree that *knowledge of the project theme* is an important factor in determining the *complexity of a user story*?
- Do you agree that *technical complexity* is an important factor in determining the *complexity of a user story*?

From the beginning it was contemplated to take into account only the opinion of the professional, though, also it was applied to students to know how they thought and if they differed from the professionals, however, to create the model, we only took account professionals answers.

The survey was applied also to 19 undergraduate students of computer engineering.

The results obtained are shown in Table 2 (Light gray for professionals and dark gray for students). It shows the percentages that each attribute had on each Likert scale. Each attribute was validated considering how much you agree on the use of that attribute. We got the response of 19 students, who have an average of 15 months of experience developing software

**Table 2.** Degree of acceptance of each attribute

	Strongly disagree, %		In disagreement, %		Neutral, %		In agreement, %		Totally agree, %	
Exp. with technology	0	0	0	5.3	4.8	5.3	38.1	10.5	57.1	78.9
Exp. with language	0	0	0	0	4.8	10.5	42.9	15.5	52.4	73.7
Experience in previous projects	0	5.3	0	0	19.0	15.8	47.6	36.8	33.3	42.1
Developer skills	0	0	0	5.3	9.5	15.8	61.9	42.1	28.6	36.8
Knowledge of the Project theme	0	0	0	0.	14.3	26.3	76.2	42.1	9.5	31.6
Dependence	0	0	4.8	5.3	14.3	15.8	57.1	52.6	23.8	26.3
Conection functions	0	0	4.8	0	14.3	21.1	42.9	36.8	38.1	42.1
Data funcions	0	0	4.8	0	23.8	15.8	28.6	52.6	42.9	31.6

projects and approximately 6 months using Scrum. We received a response from 21 professionals, who have an average of 57 months of experience developing software projects and approximately 32 months using Scrum. Both groups of respondents mentioned using Planning Poker as a method for estimate user stories.

We consider that with 19 students and 21 professionals we could obtain a good amount of data, taking as a reference to the sample of López [8] that required only 16 professionals.

*6.2.1.2. Application of equations.* To calculate the weights between attributes we apply the equations and definitions used in [8, 9].

**Definition 1:** Let  $V = \{v_1, \dots, v_n\}$  be the set that denotes the professionals' 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 dues to the number of points. Based on Definition 1, Equation 1 establishes the non-normalized relation of one variable  $p$  (Definition 2).

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

**Definition 2.** Let  $P = \{p1, \dots, pm\}$  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 Equation 1.

$$Pnorm_{p \in P} = \frac{P}{\sum P \in P}, \quad (2)$$

**Definition 3.** Let  $A = \{a1, \dots, aq\}$  the set of second level attributes; experience, size, technical complexity and complexity in user, these attributes have two states: *low*, and *high*. Their values are calculated through Equation 3.

$$a_{a \in A} = \frac{\sum P \in P}{|P|}. \quad (3)$$

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 Equation 2, but, taking into account to the variable  $an$  instead  $p$ . Table 3 shows frequencies organized by factor; weight assigned to Likert scale (strongly disagree=1, in disagreement=2, neutral=3, in agreement=4, totally agree=5). The last row shows the weight of each attribute. We will take as an example the experience node to explain how weights were obtained.

We apply Equation 1 to obtain the non-standardized weight of each first-level attribute: experience with technology, experience with language and experience in previous projects.

$$\begin{aligned} & \text{Exp.withTech} \\ & = \frac{(0x0) + (0.25x0) + \dots + (1x12)}{21} = 0.881, \quad (4) \end{aligned}$$

$$\begin{aligned} & \text{Exp.withlang} \\ & = \frac{(0x0) + (0.25x0) + \dots + (1x11)}{21} = 0.869, \quad (5) \end{aligned}$$

$$\begin{aligned} & \text{Exp. prev. proje} \\ & = \frac{(0x0) + (0.25x0) + \dots + (1x7)}{21} = 0.786. \quad (6) \end{aligned}$$

To obtain the normalized weight of each attribute we will use Equation 2. It is necessary to consider the nodes that are parents of the same child node, in this case, the experience node.

$$\text{Exp.tech}_{norm} = \frac{0.881}{0.881 + 0.869 + 0.786} = 0.347, \quad (7)$$

$$\text{Exp.lang}_{norm} = \frac{0.869}{0.881 + 0.869 + 0.786} = 0.343, \quad (8)$$

$$\text{Exp.proj}_{norm} = \frac{0.786}{0.881 + 0.869 + 0.786} = 0.310. \quad (9)$$

To obtain the weight of the experience attribute and the other second-level attributes we will apply Equation 3.

**Table 3.** Weights and frequencies of attributes

Weights	Likert scale	Frequency							
		Exp. technology	Exp. language	Exp. prev. projects	Developer skills	Knowledge theme	Depen- dence	Conection functions	Data funcions
0	1	0	0	0	0	0	0	0	0
0.25	2	0	0	0	0	0	1	1	1
0.5	3	1	1	4	2	3	3	3	5
0.75	4	8	9	10	13	16	12	9	6
1	5	12	11	7	6	2	5	8	9
Weight		0.347	0.343	0.310	0.253	0.234	0.490	0.504	0.496

$$Experience = \frac{0.881 + 0.869 + 0.786}{3} = 0.845. \quad (10)$$

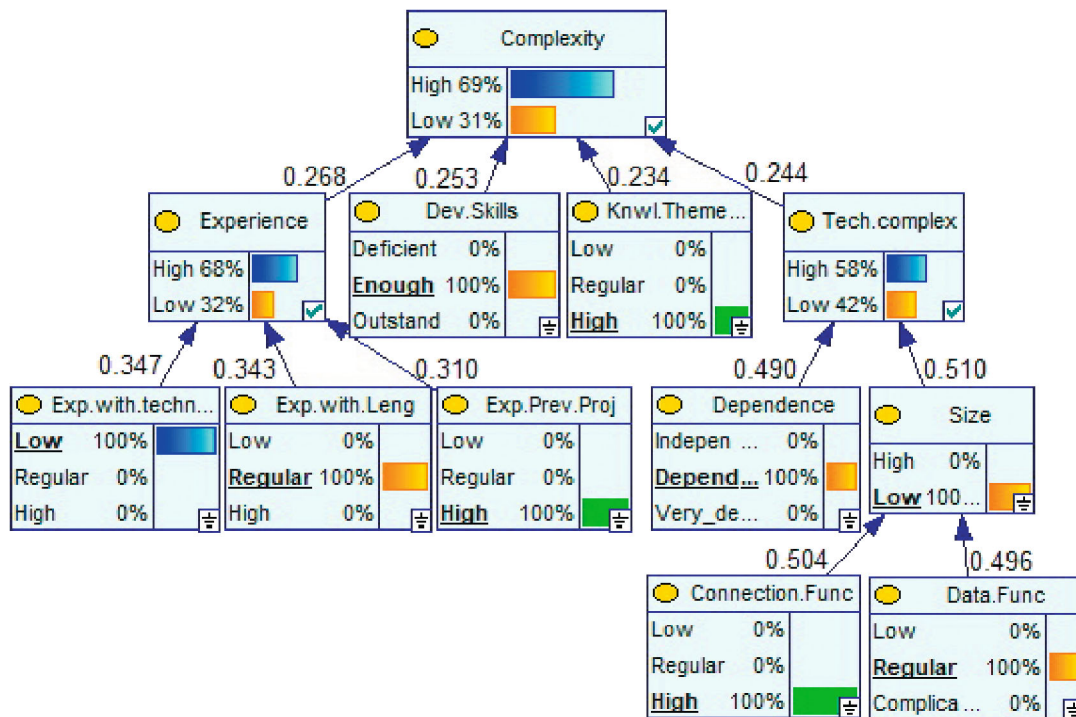
Equation 2 is applied to obtain the standardized weight of the experience attribute. We must also consider the other parents who influence the same attribute and their non-standardized weights. In this case: developer skills (0.798), knowledge of the Project theme (0.738) and technical complexity (0.770).

$$Exp\_norm = \frac{0.845}{0.845 + 0.798 + 0.738 + 0.770} = 0.268. \quad (11)$$

Figure 3 shows the results obtained for the weights of each relationship between nodes in the network.

**6.2.2. A priori values for first level attributes.** Scales of values were chosen for the collecting nodes; that is, what value or state each first level attribute can have. Three scales were used: low, regular and high. A question was made for each collecting node (first level attribute), which should be answered with one of the three scales proposed (low, regular, high). Table 4 shows the questions formulated for each attribute and the interpretation for each scale.

**6.2.3. Construction of Conditional Probability Tables.** A CPT show the probability that an event will occur based on the combination of the attributes and the value of their states [9]. In this step a CPT is defined for each child node of the structure. To obtain the CPT we will apply the equations and definitions used in [8, 9].



**Fig. 3.** Resulting Bayesian network.



**Table 4.** Questions for first level attributes

<b>Experience with technology</b>	
	How much experience do you have with the technology to use in creating the user story?
Low	I have never used the technology with which the user story will be developed
Regular	I know, but I have not yet mastered the technology with which the user story will develop
High	I master the technology with which the user story will be developed
<b>Experience with language</b>	
	How much experience do you have with the programming language to use in creating the user story?
Low	I have never used the language with which the user story will be developed
Regular	I know, but I have not yet mastered the language with which the user story will be developed
High	I master the language with which the user story will be developed
<b>Experience in previous project</b>	
	How much experience do you have in previous projects developing similar user stories?
Low	I have never developed a similar user story
Regular	I have developed a similar user story at some time
High	I have developed several similar user stories
<b>Developer skills</b>	
	How do you consider your skills as a developer to carry out the user story?
Deficient	I can't identify my skills and I have trouble solving problems
Enough	I am aware of the skills I have and sometimes I use them to solve problems
outstanding	I know my skills and I know when to apply them to problem solving
<b>Knowledge of the project theme</b>	
	How much is your knowledge about the project theme to develop the user story?
Low	I don't know what the user story will be about
Regular	I know the basics of the theme from which the user story will be developed
High	I dominate the theme of the user story will be developed
<b>Dependences</b>	
	How dependent is the user story to be developed with respect to others?
Independent	The user story is independent of other user stories
Dependent	The user story depends a bit on other user stories
Very dependent	The user story is the basis for the development of more than several user stories
<b>Connection functions</b>	
	How much difficulty do the connection functions have for the development of the user story?
Low	he user story contains simple input, output and query functions
Regular	The user story contains functions of medium complexity of input, output and queries
High	The user story contains complex input, output and query functions
<b>Data functions</b>	
	How much difficulty do data functions have for the development of user history?
Simple	The user story contains simple functions of internal and external logic
Regular	The user story contains functions of medium complexity of internal and external logic
Complicated	The user story contains complex functions of internal and external logic

**Definition 4:** Let  $S = \{s_1, s_2, \dots, s|S|\}$  be the set of states that a variable can have. The weighted values for attributes with three states were 0.170 (low), 0.330 (regular), and 0.500 (high). The matrix in Equation 12 displays the process to create the CPT. The value of the attribute is multiplied by the value of the scale to

find a weighting  $(p_1s_1, p \in P, s \in S)$ . Secondly, results in the first step are combined, each combination obtains a unique value adding each element of the combination  $((p_1s_1) + (p_2s_1) + \dots + (pns_1))$ .

$$W = (w_{11} \dots w_{1m} \dots w_{n1} \dots w_{nm}), \tag{12}$$

**Table 5.** Combinations of scales and weights of attributes

	P1 Connection Functions (0.504)	P2 Data Functions (0.496)
S1 High (0.5)	0.252	0.248
S2 Regular (0.33)	0.166	0.164
S3 Low (0.17)	0.086	0.084

Where

$$w_{11} = ((p_1s_1 + (p_2s_1) + \dots + (p_n s_1))$$

$$w_{12} = ((p_1s_1 + (p_2s_1) + \dots + (p_n s_2))$$

$$w_{1m} = ((p_1s_1 + (p_2s_m) + \dots + (p_n s_m))$$

$$\dots$$

$$w_{n1} = ((p_1s_m + (p_2s_m) + \dots + (p_n s_1))$$

$$w_{n2} = ((p_1s_m + (p_2s_m) + \dots + (p_n s_2))$$

$$w_{nm} = ((p_1s_m + (p_2s_m) + \dots + (p_n s_m))$$

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 Equation 12, in the form  $W = W/max$ , to obtains the final values. These values represent the state *high* of second level and project value attributes, the state *low* is the complement value of the state *high*. The process is repeated to obtain the CPT of each second level attribute.

$$W = \frac{(w_{11} \dots w_{1m} \dots w_{n1} \dots w_{nm})}{max} \tag{13}$$

We will take as an example the attribute size to explain how the CPT was obtained for that node. The value scales that were used were low (0.17), regular (0.33), and high (0.5). The weights for the attributes connection functions and data functions are 0.504 and 0.496, respectively. To obtain the matrix of Equation 12, it is first necessary to multiply the weight of the

attribute (P) by each value of the scale (S), the result is shown in Table 5.

To form the CPT it is necessary to contemplate all possible combinations  $((p_1s_1) + (p_2s_1) + \dots + (p_n s_1))$  between the attributes that make up the size node. The result is added by each combination and must be divided by the maximum value of the matrix to obtain the final values. These values represent the high state of the second level attributes, the low state is the complementary value of the high state. Table 6 shows the possible combinations that may exist in the size node and the results obtained.

## 7. TESTING BAYESIAN NETWORK

### 7.1. Experiments

To validate the model created, two different experiments were designed, one for beginners and one for professionals.

**7.1.1. Sample of Students.** The objective of this experiment was to obtain estimates of individual user stories using the traditional method (Planning Poker) and the network built as our proposal.

There was a sample of 19 students of the 7th semester of computer engineering at the Autonomous University of Baja California and 112 user stories. The students who participated in the experiment had basic knowledge about Scrum and Planning Poker. To collect the data, two Excel files were created in which the students recorded their estimates individually. One to keep track of conventional estimates and another for estimates made with the network. The experiment was divided into 3 parts:

Part 1: Once the user stories have been described, each member of the Scrum Team estimates each user story individually and conventionally (using Planning Poker).

Part 2: For the second part, we asked to answer the questions listed in Table 4 individually for each user story.

**Table 6.** Conditional Probabilities Table for size node

Combinations			values for node size	
Connec func.	Data Func.	Addition	High	Low
P1*S1	P2*S1	0.252 + 0.248 = 0.500	1.00	0.00
P1*S1	P2*S2	0.252 + 0.164 = 0.416	0.83	0.17
P1*S1	P2*S3	0.252 + 0.084 = 0.336	0.67	0.33
P1*S2	P2*S1	0.166 + 0.248 = 0.414	0.83	0.17
P1*S2	P2*S2	0.166 + 0.164 = 0.330	0.66	0.34
P1*S2	P2*S3	0.166 + 0.084 = 0.251	0.50	0.50
P1*S3	P2*S1	0.086 + 0.248 = 0.334	0.67	0.33
P1*S3	P2*S2	0.086 + 0.164 = 0.249	0.50	0.50
P1*S3	P2*S3	0.086 + 0.084 = 0.170	0.34	0.66

Part 3: As an additional data, it was requested to record a third estimate in a conventional manner. This consisted of re-estimating the user story once it was made.

**7.1.2. Sample of Professionals.** The objective of this experiment was to obtain estimates of individual user stories using the traditional method (Planning Poker) and the network built in this article. There was a sample of 6 professionals in the Scrum area and 10 user stories. This experiment consisted of a digital survey (google forms).

The experiment was divided into 3 parts:

Part 1: A project and 10 user stories written by students were selected.

Part 2: Important information about the project (Description of the project, programming languages to be used, etc.) was provided and the professional was asked to estimate each user story using Planning Poker, simulating that it would be developed, taking into account the information provided and its experience.

Part 3: The professionals were asked to answer the 8 questions listed in Table 4 for each user story considering the information provided and their experience.

### 8. RESULTS AND DISCUSSION

In the case of students, the results obtained were organized by team (4 participating teams), a short part of the results obtained by one of the teams is shown in Table 8. The story points column (initial) represents the estimation in a traditional way (Planning Poker) made by the students before the realization of the user story. The story points (final) column represents student estimates after the end of the user story. The complexity column represents the value obtained by the Bayesian network according to the questions answered by the students for each user story. In order to compare the estimates in story points by the student with what was obtained in the Bayesian network, it was necessary to convert the complexity obtained to a card in Planning Poker, so a card in Planning Poker would be equivalent to a number of story points. The nine most used cards in Planning Poker were selected, Table 7 shows the equivalence we use for each card in Planning Poker. The complexity obtained was converted to a card in Planning Poker using Equation 14. In which, Y is the value we are looking for and it represents the Planning Poker card, X represents the complexity obtained by the

Bayesian network,  $X_1$  represents the lowest value which can be obtained in the Bayesian network,  $X_2$  represents the highest value that can be obtained in the Bayesian network,  $Y_1$  represents the smallest position of the card and  $Y_2$  the greatest position of the card. The result is shown in the card column in Table 8.

**Table 7.** Equivalence between cards and history points.

Card	Story points
1	1
2	2
3	3
4	5
5	8
6	13
7	20
8	40
9	100

$$Y = Y_1 + \left[ \left( \frac{X - X_1}{X_2 - X_1} \right) (Y_2 - Y_1) \right]. \tag{14}$$

The story points (With BN) column represents the equivalent value (shown in Table 7) according to the card obtained as a result of the application of Equation 14.

The results obtained by the professionals are shown in Table 9. The story points column (initial) represents the estimate in a traditional way (Planning Poker) made by the professionals before the user story is initiated. The complexity column represents the value obtained by the Bayesian network according to the questions answered for each user story. As with students, Equation 14 was applied and the equivalences shown in Table 7 were considered to obtain the Story points (with BN) column. To validate the results obtained, we applied Spearman correlation test [45]. In the case of students, the variables were analyzed:

- estimationIni represents the estimation initial before making the user story.
- estimationFin represents the estimation final, once the user story is made.
- estimationBN represents the result obtained by the Bayesian network.

The tests were applied considering the relationships: *estimationIni-estimationBN* which determines the correlation between the estimation initial and the one obtained by the network and *estimationFin-estimationBN* which determines the correlation between the final estimate and the one obtained by the network. The Spearman test for the *estimationIni-estimationBN* relationship, provided us a correlation of 0.659 and a *p-value* of 0.000, while for the *estimationFin-estimationBN* relationship provided us a correlation of 0.799 and a *p-value* of 0.000.

As can be seen, the correlation between the final estimation and the one obtained in the network is greater, this is a good fact because it indicates that what obtained by the network is very related to what the student estimates once made user story.

In the case of professionals, the variables were analyzed:

**Table 8.** Students results

Case	Sprint #	Member Name	User story	Complexity	Card (Planning Poker)	Story points (Initial)	Story points (Final)	Story points (with BN)
1	3	Member 1	1	75	4	5	5	5
2	3	Member 2		67	3	3	5	3
3	3	Member 3		75	4	8	5	5
4	3	Member 4		82	5	5	5	8
5	3	Member 5		75	4	5	5	5
6	3	Member 1	2	73	4	5	8	5
7	3	Member 2		78	5	5	8	8
8	3	Member 3		78	5	5	8	8
9	3	Member 4		82	5	8	5	8
10	3	Member 5		76	4	13	5	5
11	5	Member 1	3	78	5	13	8	8
12	5	Member 2		84	6	20	13	13
13	5	Member 3		84	6	20	20	13
14	5	Member 4		87	6	13	20	13
15	5	Member 5		89	7	40	20	20
16	5	Member 1	4	89	7	20	40	20
17	5	Member 2		90	7	20	40	20
18	5	Member 3		89	7	40	40	20
19	5	Member 4		89	7	20	20	20
20	5	Member 5		93	7	40	40	20

- *estimationIni* represents the initial estimate before making the user story.

- *estimationBN* represents the result obtained by the Bayesian Network.

The tests were applied considering the relationships: *estimationIni-estimationBN* which determines the correlation between the initial estimate and the one obtained by the network. The Spearman test for the *estimationIni-estimationBN* relationship provided us a correlation of 0.887 and a p-value of 0.000.

If we compare the results of students and professionals, we can deduce that the correlation in both cases was good, however, the correlation of the professionals is greater, this is because the professionals have more experience.

Our study proposes a model to estimate the complexity in user stories. Our model estimates around 88% as a professional developer does. The accuracy of our model increases for developers with experience which is a viable result because we can trust in the model as we trust in a professional developer.

The correlation between the model estimation and the student initial estimation is of 65%; and the correlation between the model estimation and the student final estimation (real estimation, because it was conducted after the new programmers develop the user

story) was of 79%. This estimation shows that the model estimates like the real value. If we compare the correlation obtained by professionals (88%) our model proposed is more accurate than the model proposed by Martinez [8] (87%) and Alostad [13] (60%).

Our is more similar to Martinez [8], but model decomposes the developer experience in three sub-categories: experience with technology, experience with the language, and with previous projects. These factors allow deeper analysis of the experience, which one of the most important factors in the estimation process.

On the other hand, Dragicevic [12] reaches 90% of accuracy in the estimation, but we can't compare our model with him model, because that work was focused on effort estimation. Dragicevic's model was focused on decompose the complexity in technical subcategories. The model uses attributes different to ours, doesn't consider the experience of the developer, he collects the skills of the programmers. Nevertheless, our accuracy was near from Dragicevic's accuracy.

Compared to our model, Martínez [8], Alostad [13], and Dragicevic [12] used more generic attributes for estimation, even though these attributes can be broken down. We could notice that if we broke down attributes into sub-attributes, we can cover more details in the estimation.

**Table 9.** Professionals results

Case	Member Name	User story	Complexity	Card (Planning Poker)	Story points (Initial)	Story points (with BN)
1	Person 1	1	77	4	5	5
2	Person 2		67	3	3	3
3	Person 3		77	4	5	5
4	Person 4		56	1	1	1
5	Person 5		75	4	8	5
6	Person 6		81	5	8	8
7	Person 1	2	67	3	3	3
8	Person 2		62	2	2	2
9	Person 3		83	5	5	8
10	Person 4		63	2	2	2
11	Person 5		75	4	5	5
12	Person 6		79	5	8	8
13	Person 1	3	83	5	8	8
14	Person 2		60	1	1	1
15	Person 3		84	6	13	13
16	Person 4		61	1	1	1
17	Person 5		78	5	5	8
18	Person 6		77	4	8	5
19	Person 1	4	78	5	13	8
20	Person 2		60	1	1	1
21	Person 3		94	7	20	20
22	Person 4		60	1	1	1
23	Person 5		75	4	5	5
24	Person 6		73	4	5	5

The model proposed in the present paper will be useful for new developers, and for developers without experience. We can trust that the model's results will be similar as a professional developer.

### 9. CONCLUSIONS

In this study, a method was proposed to estimate user stories through a model composed of attributes emphasizing personal aspects of the developers. With this model, we are trying to improve the accuracy in the estimation of user stories. We take as a basis our previous article [36] of which we analyze attributes to form our model proposed. Because our model focuses on personal aspects, we analyze the attributes that were within this category and select the most relevant: experience with technology, experience with language, experience in previous projects, developer skills and knowledge of project theme. We decided to add some important attributes related to the technical complexity of user stories: dependence and size. We noticed that the size attribute could be broken down into 2 more attributes: connection functions and data

functions. Giving with this a model focused on the personal aspects of the developers, also considering some important technical aspects.

As a technique to represent the model we chose Bayesian networks since these allow us to describe probabilistic relationships between related variables. For the construction of the Bayesian network, it was necessary to obtain the values between the relationships of each attribute, to obtain these data we consulted professionals through surveys. With the values obtained and the application of formulas, we obtained the CPTs for each child node.

To test our model, we used two types of samples, students and professionals. The first experiment was applied to students and consisted of making 3 estimates, the first the traditional way with Planning Poker before developing the user story, the second estimation using the Bayesian network before developing the user story and the third estimate in a traditional way once the user story was developed. In the case of professionals, the experiment consisted of selecting 10 user stories defined by students; these

were estimated by professionals, in a traditional way, and with the Bayesian network.

To validate the results, we applied correlation tests to the estimates obtained. For students, we obtained a correlation of 0.659 between the initial estimate and the estimate obtained through the network, and a correlation of 0.799 between the final estimate and the estimate obtained through the network. These results show us that there is more correlation between the final estimate and the estimate obtained through the network, which indicates that the data obtained through the network are more similar to what the student estimated once the user story was developed. For professionals, we obtained a correlation of 0.887 between the initial estimate and the estimate obtained through the network. We can note that the correlation of the professionals is superior to that of the students, this is because the professionals have more experience. Through the results, we can also highlight that our model estimates around 88% as a professional developer would do, which is a favorable and reliable result.

Comparing our results with some related works such as Martínez [8] who used the same methodology that was used in this study, and Alostad [13], who even uses a different methodology but with the same approach as our research, we can see that our model is more accurate, this could be due to the fact that our model is mostly composed of personal attributes, which have a great impact on the process of the estimate. This is because in our work we break down some attributes into sub-attributes, which allowed us to consider more details and information from the user story and from the developer at the time of estimating.

In our model we contemplate the experience, but not as a single factor. By breaking down an attribute as important as experience, we were able to increase the accuracy in our estimates, in this case, we were able to obtain more information related to the developer's experience. This proves that if the correct attributes are broken down, it is possible to achieve better estimates. The results obtained through the network provide us with more confidence since they were similar to those of a professional developer; this is achieved because the weight of each attribute within the network represents the opinion of the professionals, which helps to manage the uncertainty and inexperience of the developers and improve the accuracy of the estimate. Working with this method will give us confidence, especially in the case of inexperienced developers.

As future work, it is planned to extend the model following the main idea of a model focused on personal attributes in order to increase the accuracy in the estimation; furthermore, to prove this estimation method in more teams to have a larger sample in the experiments in order to obtain more reliable data.

## ACKNOWLEDGMENTS

The authors would like to thank the students of the Software Engineering course enrolled in 2019-1 and 2019-2 of the computer engineering degree at Universidad Autónoma de Baja California who participated in the Scrum projects and the professionals in the area who participated from various companies in the region. Thanks to your participation we managed to collect data and information for the formation of our study. We also have special thanks to Consejo Nacional de Ciencia y Tecnología (CONACYT) for the scholarship to realize the master in science studies, which support this research.

## REFERENCES

1. Schwaber, K.S. and Sutherland, J., *The Scrum Guide: the Definitive the Rules of the Game*, Scrum, Org and Scrum Inc., 2017.
2. Srivastava, A., Sukriti Bhardwaj, and Shipra Saraswat, SCRUM model for agile methodology, *Proc. IEEE Int. Conf. on Computing, Communication and Automation ICCCA 2017*, Greater Noida, 2017.
3. Rubin, K.S., *Essential Scrum: a Practical Guide to the Most Popular Agile Process*, Addison-Wesley, 2013.
4. Pauly, D., Michalik, B., and Basten, D., Do daily scrums have to take place each day? A case study of customized scrum principles at an E-commerce company, *Proc. Annu. Hawaii Int. Conf. on System Science*, Kauai, 2015, pp. 5074–5083.
5. SCRUMstudy, *A Guide to the Scrum Body of Knowledge (SBOK Guide)*, VMEdu Inc., 2016.
6. Lucassen, G., Dalpiaz, F., van der Werf, J.M.E.M., and Brinkkemper, S., Improving agile requirements: the quality user story framework and tool, *Requir. Eng.*, 2016, vol. 21, no. 3, pp. 383–403.
7. Moreno, A.M. and Yagüe, A., Agile user stories enriched with usability, *Lect. Notes Bus. Inf. Process.*, 2012, vol. 111 LNBIP, pp. 168–176.
8. López-Martínez, J., Ramírez-Noriega, A., Juárez-Ramírez, R., Licea, G., and Jiménez, S., User stories complexity estimation using Bayesian networks for inexperienced developers, *Cluster Comput.*, 2017, vol. 21, pp. 715–728.
9. Lopez-Martinez, J., Estimating user stories complexity and importance, *Proc. WorldCIST 2017: Recent Advances in Information Systems and Technologies*, Porto Santo Island, Madeira, 2017, pp. 205–214.
10. Schwaber, K. and Sutherland, J., *La Guía de Scrum, Scrumguides.Org.*, 2013, vol. 1, p. 21.
11. Lopez-Martinez, J., Ramirez-Noriega, A., Juarez-Ramirez, R., Licea, G., and Martinez-Ramirez, Y., Analysis of planning poker factors between university and enterprise, *Proc. 5th Int. Conf. on Software Engineering Research and Innovation CONISOFT 2017*, Mérida, 2017, pp. 54–60.
12. Dragicevic, S., Celar, S., and Turic, M., Bayesian network model for task effort estimation in agile software development, *J. Syst. Software*, 2017, vol. 127, pp. 109–119.
13. Alostad, J.M., Abdullah, L.R.A., and Aali, L.S., A fuzzy based model for effort estimation in scrum proj-

- ects, *Int. J. Adv. Comput. Sci. Appl.*, 2017, vol. 8, no. 9, pp. 270–277.
14. Scott, E. and Pfahl, D., Using developers' features to estimate story points, *ACM Int. Conf. Proc. Ser.*, 2018, no. 106, pp. 106–110.
  15. Lenarduzzi, V., Could social factors influence the effort software estimation?, *Proc. 7th Int. Workshop on Social Software Engineering (SSE)*, Bergamo, 2015, pp. 21–24.
  16. Mahnič, V. and Hovelja, T., On using planning poker for estimating user stories, *J. Syst. Software*, 2012, vol. 85, no. 9, pp. 2086–2095.
  17. Ziauddin, S.Z. and Shahid Kamal Tipu, An effort estimation model for agile software development, *Adv. Comput. Sci. Its Appl.*, 2012, vol. 2, no. 1, pp. 2166–2924.
  18. Peischl, B., Zanker, M., Nica, M., and Schmid, W., Constraint-based recommendation for software project effort estimation, *J. Emerg. Technol. Web Intell.*, 2010, vol. 2, no. 4, pp. 282–290.
  19. Jennings, N.R. and Wooldridge, M., Agent-oriented software engineering, *Artif. Intell.*, 2000, vol. 117, pp. 277–296.
  20. Ramirez-Noriega, A., Juarez-Ramirez, R., Navarro, R., and Lopez-Martinez, J., Using Bayesian networks to obtain the task's parameters for schedule planning in scrum, *Proc. 4th Int. Conf. on Software Engineering Research and Innovation CONISOFT 2016*, Puebla, 2016, no. 1, pp. 167–174.
  21. Zare, F., Khademi Zare, H., and Fallahnezhad, M.S., Software effort estimation based on the optimal Bayesian belief network, *Appl. Soft Comput. J.*, 2016, vol. 49, pp. 968–980.
  22. Rath, S.K., Acharya, B.P., and Satapathy, S.M., Early stage software effort estimation using random forest technique based on use case points, *IET Software*, 2016, vol. 10, no. 1, pp. 10–17.
  23. Hsu, S.C., Weng, K.W., Cui, Q., and Rand, W., Understanding the complexity of project team member selection through agent-based modeling, *Int. J. Proj. Manag.*, 2016, vol. 34, no. 1, pp. 82–93.
  24. Malgonde, O. and Chari, K., *An Ensemble-Based Model for Predicting Agile Software Development Effort*, Empirical Software Engineering, 2018.
  25. Sholiqa, Dewi, R.S., and Subriadi, A.P., A comparative study of software development size estimation method: UCPabc vs function points, *Procedia Comput. Sci.*, 2017, vol. 124, pp. 470–477.
  26. Qazi, A., Quigley, J., Dickson, A., and Kirytopoulos, K., Project complexity and risk management (ProCRiM): towards modelling project complexity driven risk paths in construction projects, *Int. J. Proj. Manag.*, 2016, vol. 34, no. 7, pp. 1183–1198.
  27. Drahý, C. and Pastor, O., Relationship between project complexity and risk kinds identified on projects, *Econ. Manag. Innov.*, 2016, vol. 8, no. 1, pp. 42–50.
  28. Abdou, S.M., Yong, K., and Othman, M., Project complexity influence on project management performance – the Malaysian perspective, *MATEC Web Conf.*, 2016, vol. 66, p. 00065.
  29. Singh, R., A MCDM approach for measuring project complexity, *Archit. Eng. Design Manag.*, 2016, vol. 2, no. 3, pp. 187–202; no. 1, pp. 294–302.
  30. Yong, A.S.M.K. and Othman, M., Categorization of project complexity factors in Malaysian construction industry, *Int. J. Prop. Sci.*, 2018, vol. 7, no. 1, pp. 1–15.
  31. Kayes, I., Sarker, M., and Chakareski, J., Product backlog rating: a case study on measuring test quality in scrum, *Innov. Syst. Software Eng.*, 2016, vol. 12, no. 4, pp. 303–317.
  32. Bosch-Rekveltdt, M., Bakker, H., and Hertogh, M., Comparing project complexity across different industry sectors, *Complexity*, 2018, no. 2, pp. 1–15.
  33. Zahraoui, H. and Janati Idrissi, M.A., Adjusting story points calculation in scrum effort & time estimation, *Proc. 10th Int. Conf. on Intelligent Systems: Theories and Applications, SITA 2015*, Rabat, 2015.
  34. Mendes, E., Knowledge representation using Bayesian networks a case study in web effort estimation, *Proc. World Congress on Information and Communication Technologies WICT 2011*, Mumbai, 2011, pp. 612–617.
  35. Silhavy, R., Prokopova, Z., and Silhavy, P., Algorithmic optimization method for effort estimation, *Program. Comput. Software*, 2016, vol. 42, no. 3, pp. 161–166.
  36. Durán, M., Juárez-ramírez, R., Jiménez, S., and Tona, C., Taxonomy for complexity estimation in agile methodologies: a systematic literature review, *Proc. 7th Int. Conf. in Software Engineering Research and Innovation (CONISOFT)*, Chetumal, 2020, pp. 1–10.
  37. Usman, M., Mendes, E., Weidt, F., and Britto, R., Effort estimation in agile software development: a systematic literature review, *Proc. Int. Conf. on Predictive Models in Software Engineering – PROMISE 2014*, Turin, 2014.
  38. Georgsson, A., Introducing story points and user stories to perform estimations in a software development organisation – a case study at Swedbank IT, *Master's Thesis*, Umeå Univ., 2011.
  39. Popli, R. and Chauhan, N., A sprint-point based estimation technique in Scrum, *Proc. Int. Conf. on Information Systems and Computer Networks*, Mathura, 2013, pp. 98–103.
  40. Pandit, P. and Tahiliani, S., AgileUAT: a dramework for user acceptance testing based on user stories and acceptance criteria, *Int. J. Comput. Appl.*, 2015, vol. 120, no. 10, pp. 16–21.
  41. Cristóbal, J.R.S., Complexity in project management, *Procedia Comput. Sci.*, 2017, vol. 121, pp. 762–766.
  42. Jensen, F.V., Bayesian networks, *Wiley Interdiscip. Rev. Comput. Stat.*, 2009, vol. 1, no. 3, pp. 307–315.
  43. Rodríguez, D. and Dolado, J., Redes bayesianas en la ingeniería del software, *Proc. 15th Workshop de Investigadores en Ciencias de la Computación*, Parana, 2013.
  44. Linåker, J., Muhammad Sulaman, S., Maiani de Mello, R., and Höst, M., *Guidelines for Conducting Surveys in Software Engineering v. 1.1*, 2015, pp. 0–63.
  45. Sirois, L.M.M.J., *Spearman Correlation Coefficients, Differences between*, John Wiley and Sons2006.