International Journal of Computer Information Systems and Industrial Management Applications.
ISSN 2150-7988 Volume 13 (2021) pp. 336-347

Received: 7 Jan. 2021; Accepted: 8 Oct. 2021; Publish: 9 Dec. 2021

Scrum Poker Estimator: A Planning Poker Tool for Accurate Story Point Estimation

Bushra Alsaadi[1], Bashaer Alsaadi[2], Mashaal Alfhaid[3], Athir Alghamdi[4], Nedaa Almuallim[5], and Bahjat Fakieh[6]

[1, 2, 3, 4, 5, 6] Information Systems Dept., King Abdulaziz University
Jeddah, Saudi Arabia

[1]BSADERALSAADI@stu.kau.edu.sa, ORCID: 0000-0001-7028-9725
[2]bsaderalsaadi0001@stu.kau.edu.sa, ORCID: 0000-0002-5074-250X
[3]Msalehalfhaid@stu.kau.edu.sa, ORCID: 0000-0002-5349-0248
[4]aothmanalghamdi0001@stu.kau.edu.sa, ORCID: 0000-0001-7067-2353
[5]Nmalmuallim@stu.kau.edu.sa, ORCID: 0000-0002-8748-9747
[6]BFakieh@kau.edu.sa, ORCID: 0000-0003-2793-1238

***Abstract*: Over the last few decades, the rising popularity of software development projects guided by Scrum methodology led to an increase in the challenges of effective project planning and management. These challenges are more serious for teams which include inexperienced members, as the requirements are constantly changing, and the flow of work is highly dependent on the members' efficiency. Different tools were developed to support teams in managing project phases and whereas the estimation phase plays a crucial role in Scrum software development, it has been given special attention. Planning Poker, considered one of the most common estimation techniques existing in the field, is a gamified technique that helps team members in estimating user stories efficiently. One of the limitations of this technique is that it depends on team experience and judgments. In other words, it does consider the inexperienced teams and guiding them to an accurate estimation. In this research, a Scrum Estimator Poker tool is developed to facilitate the process of estimating user stories, especially for inexperienced members. A set of influential criteria were collected to effectively calculating the proper estimation for each user story based on members' experience, complexity and size of the story, in addition to the time needed to perform that story. The calculation done through a form of questions that filled by members at the beginning of estimation process. The tool also aims to facilitate team negotiation during the estimation by highlighting the state of individuals' estimates as an appropriate estimate, overestimate or underestimate. The developed tool proved its effectiveness during the practical implementation by providing more accurate estimates against previous unstructured estimates regarding a test group's user stories.**

***Keywords*:** Agile, Scrum, Estimation, Planning Poker, Novice Team.

I. Introduction

With the rapid development of information technology and communication, the popularity of software development projects completed by remote teams has raised. This leads to an increase in the challenges of effective communication which became a dilemma for those who are following Agile methodology. Agile is an evolutionary and highly collaborative methodology which adapts to requirements changes that result in increasing the quality of the delivered software [1]. However, there are several Agile methodologies existing in the field such as Extreme programming, Agile Unified Process, and Scrum, which is the most commonly used [1]. Scrum is a framework within which the team can manage and deliver complex projects in incremental development cycles called "Sprints" [2]. At the Scrum planning phase, the product owner, with the help of project stakeholders, has to create a product backlog, which is an ordered list of all project requirements in form of user stories [3]. Scrum teams then estimate the effort of completing each item in the product backlog in terms of story points. For effort estimation, several techniques could be used, such as Delphi, Wideband Delphi, Planning Poker, etc. [4]. Planning Poker got its excellence from integrating expert opinions, the ability of enjoyable cooperation, with easy to implement steps [5]. In Planning Poker, team members participate in the estimation process to reduce uncertainty, which leads to better estimation results. After the user story has been clearly explained by the product owner, each member, including developers, chooses a proper effort estimation poker card for the story; then members expose their cards at the same time. In the case of an estimation variance, the members should explain their viewpoints and the process will be repeated until they reach an agreement [4].

However, Planning Poker has some barriers that could limit the estimation accuracy if not considered properly as it does not provide an assistant and guidance to inexperienced team on estimating the story point accurately. The estimation plays a crucial role in prioritizing user stories as well as giving the team insight into the complexity of each project task. In fact, the estimation of the product backlog is one of the challenges that

face Scrum teams while managing Scrum projects, as inaccurate effort estimation might lead to project failure [6], in addition to its negative impact on project cost and timeline [7]. The lack of team experience in estimating software effort is considered as one of the factors that contributes to unsatisfied estimation [6]. In other words, team members should be aware of experts' perspectives regarding estimation criteria. Another factor is that working from different locations and time zones may increase the risks in team collaboration[8]. In contrast, teams have to cooperatively estimate each story in the product backlog.

The main objective of this research is to develop a Scrum estimation tool based on Planning Poker techniques that contributes to enhance effort estimation accuracy. The tool is called Scrum Estimator Poker and it is based on an estimation algorithm that calculates by weighted mean to measure the proposed estimation story point for each user story, and it relies on a set of criteria that covers four factors, which are complexity, experience, time, and size. The factors that mostly influence the estimation process were obtained by a questionnaire targeting IT project experts.

The rest of this paper is organized as follows. Section II describes some concepts from the literature including the Scrum methodology, some issues related to effort estimation, and the widely used user stories estimation methodologies. Section III shows the experts survey which measures the importance of effort estimation factors. Section IV presents the proposed Scrum estimator poker tool. Section V discusses and analyses the results. Finally, Section VI and Section VII conclude this work and present some recommendations and future work.

II. Literature Review

The following sections start by describing the Scrum methodology and its main components. Then, it explores some significant issues that could affect the progress of the estimation phase. Afterward, the literature presents the most common methodologies used in estimating user stories.

A. Scrum methodology

Scrum is considered one of the most commonly used agile methodologies because of its changing adaptability which reduces time and cost. There are three roles for a Scrum team; each of them has a certain role and responsibilities, which are [9, 10]:

- Scrum master. The one who's responsible for communication, operational activities, and continuous improvements of the project to ensure that all conflicts are being resolved within the appropriate time frame and ensuring that all the roles are being followed.
- Product Owner. Is responsible for ensuring that requirements meet customer needs, managing the product backlog and prioritize these stories to make better project and time management that can effect on the team effort.
- Scrum development team. The Scrum team who should be organized and creative. Also, the team members should have the required skills to deliver the project such as user interface (UI) design, programming, and problem solving as well as collaborating to bring out a high-quality development project result[11].

1) Product Backlog

At the beginning of the Scrum project, all requirements, tasks, and features that contribute toward project success have to be specified by the product owner, then filled in an ordered list called product backlog. Items in the product backlog should be briefly described, estimated and prioritized. The description usually is written as a user story following such a pattern: As a < user role>, I need to <specific requirement>, in order to < user aim> [10]. For effort estimation, Scrum team has to estimate each user story using a measure named story points. These story points will be considered in user stories prioritization. However, the product backlog is a constantly updated list which means that the product owner can add, update, or delete items whenever needed during the project lifecycle.

2) Sprints

In Scrum, the system is broken down into subtasks in order to be delivered within iterations called sprints. Sprints should be a fixed-length duration throughout the project, usually between two to four weeks [10]. By the end of each sprint, the Scrum team must complete sprint tasks and deliver a working piece of software.

Each sprint consists of several events including Sprint Planning, Daily Scrum, Development Work, Sprint Review and Sprint Retrospective [2]. The daily meeting is a short meeting that is conducted to discuss the twenty-four hours of work progress including all team members. The Review meeting is an informal one that is done at the end of the sprint to discuss the product backlog and what are the next steps. The last one is Sprint Retrospective to discuss any failure and how to improve it in the next sprint [1].

B. Story Estimation Issues

Scrum teams continuously seek to have an accurate story estimation. The latter is a reliable way to achieve successful project implementation[6]. In contrast, an opposite outcome may occur if the team fails to estimate project tasks correctly. This section demonstrates some issues that may increase the chance to have an unrealistic story estimation.

- Lack of Team Experience. Inexperienced team members may misestimate product backlog stories, due to their poor background in crucial project aspects such as sprint domain [6]. Teams could increase the time needed to complete a task, which ends up with effort overestimation [12]. Moreover, with the absence of expert members, the decision-making process may be affected by inappropriate members' decisions [12].
- Management Influence. The heavy focus of managers to deliver a project within a specific timeframe and budget will put more pressure on team members [12]. Therefore, members may be afraid of late delivered projects, which could force them to underestimate story effort.
- Unorganized Distributed Team. Workers have to collaboratively share their information, experience, and any supporting material about the project in order to make the estimation phase more useful. In fact, this can be a

challenge while working remotely regarding the multiple time zones and separated physical environments. Thus, having inconsistent and messy distributed teams may have negative consequences on knowledge sharing [13].

C. Story Estimation Methodologies

Owing to the critical role of estimation in Scrum projects' success, many studies have been conducted and plentiful estimation methods were developed over the years [6]. However, just a few methods have become accepted and practiced, while others remained at theoretical levels [6]. Based on the source of inputs the estimation methods can be categorized into:

1) Data-Driven

The estimation is based on analyzing the previous projects' data, then building relationships with the actual project to estimate the expected effort [2], even though it rarely used in agile software development [14]. One example is Case-Based Reasoning, where the estimation is calculated by adapting averages on the current and the most analogous previously existing project [6].

2) Expert-based

It is considerably used in software development [14]. The estimation relies on the expert's understanding of context. The major advantages are the simplicity of application and adjustability to various situations and project models [6]. Some of its widely used techniques are described below.

Delphi Assuring estimator anonymity is what distinguishes Delphi, but that increases structure complexity and overheads [4]. The interaction among estimators is coordinated, and it is only about providing written feedbacks on estimators' estimates. However, the accuracy of estimates is higher and implicates less uncertainty [15].

Wideband Delphi decreases the overhead of Delphi and implicates more interaction while keeping their identities anonymous initially [4]. User stories are provided to the estimators previously to review and contemplate [16]. At the meeting, estimators state their predicts and discuss them; that process is repeated until reaching an agreement [16].

Planning Poker is the simplest technique that offers strong team interaction with limited costs [4]. It is considered as a quick, reliable, and enjoyable technique based on synchronous revealing of estimates [17].

3) Hybrid

This method resulted from repeated endeavors to strengthening existing methods and overcoming limitations [6]. It combines the best of both Expert-Based and Data-Driven techniques in a stand-alone method, although it did not get researchers' attention until recent years such as CoBRA. The main feature is that CoBRA considers effort as two elements - nominal effort (actual project effort) and effort overhead (spent to overcome limitations) [6].

D. Planning Poker Methodology

Planning Poker is considered as one of the most commonly used estimation techniques in Scrum methods [4]. In order to play Planning Poker, user stories should be explained by the product owner [7]. Then, each team member is going to choose a proper estimation value (story points) in the form of cards. These story points can be scaled through different ways, mainly using Fibonacci sequence (1, 2, 3, 5, 8, 13, etc.) where the number is calculated through summing the preceding two numbers [17], or the t-shirt sizing technique, which ranges from XS, S, M, L, XL, ..., XXXL [10]. In this work, the used effort scale is the Fibonacci sequence as it is commonly used in planning poker [18]. After that, the cards must be revealed at the same time. Afterward, the Scrum team should discuss their estimations and re-estimate until they reach an agreement.

E. Related Studies

Previous studies proposed several solutions and algorithms to improve the accuracy of effort estimation. They used different story estimation methodologies and techniques. Some authors followed data-driven approach, which depends on historical data. For example, Zia's work [19] proposes a model that estimates the effort needed to accomplish agile software development. It develops a dataset of 21 projects from different six software houses. The model takes different variables as an input which are: number of user stories in the project, team velocity, sprint size, number of working days, team salary and confidence level in estimation. Another work [20] investigates the use of neural networks algorithms for effort estimation using Zia's dataset. It uses four algorithms: General Regression Neural Networks, Probabilistic Neural Networks, GMDH Polynomial Neural Network and Cascade Correlation Neural Network. Similarly, by using Zia's dataset, [21] validate the use of Decision tree, Stochastic gradient boosting and Random forest for effort estimation.. Another study [22] assessed the effectiveness of four Support Vector Regression (SVR) based models, SVR Linear Kernel, SVR Polynomial Kernel, SVR RBF Kernel and SVR Sigmoid Kernel. However, it conducts an experiment applied on Zia's dataset and finds that SVR RBF Kernel record the best accuracy.

Although authors who followed data-driven approach were able to use advanced predication algorithms such as machine learning, such methods require the existence of an available historical dataset that is related to the company's context. This considered as a limitation as there are limited number of available user stories datasets [23]. Therefore, the use of expert-based methods, such as Planning Poker, is the appropriate solution when there is a lack of available datasets [24].

Planning Poker has some barriers that could limit the estimation accuracy if not considered properly. First, the estimation relies on the team judgments without considering factors that affect the effort, such as project size and team velocity [6, 25]. Further, there is an absence of defining standards that can be reused across similar projects. This lack of standards would lead to reapplication of the entire procedure for each project [6].

Several studies were conducted to examine these limitations and provided different methodologies to overcome them. One research proposed a Bayesian networks model for user story estimation using Planning Poker techniques for inexperienced teams [26] in one of these studies. The Bayesian's paper aims to change their estimation method into a professional factors-based estimation. The model consists of five factors

which are experience, time, effort, priority, and value. However, after testing the model, it has a high correlation to experts' estimation compared to non-experts. At the same time, authors in different research [27] discuss how estimation in Planning Poker based on story point or velocity was not always accurate since they do not consider all the factors that can affect working in the Scrum. They proposed a solution to make it more accurate using a calculation of Adjusted Story Points (ASP) instead of the story point measure. The ASP is based on three factors, which are Priority Factor (PF), Size Factor (SF) and Complexity Factor (CF)[27]. Another study provided an estimation tool to improve the accuracy of story estimation for Scrum methodology in general by utilizing a plagiarism detection method called document fingerprints [28]. The tool aims to extract a previous project that has both successful estimation and similar documentation as an input project, and deal with it as a guideline. The authors prove that this tool has a positive impact on reducing inaccurate estimation by inexperienced members.

F. Recent Studies about Estimation Factors

In order to develop the algorithm and to make the user story estimation more accurate in the proposed tool, a set of common factors was determined. The aim of considering these factors is enabling the team to have a guiding blueprint that would help in better estimation explanation. Those factors are:

- Complexity. This factor [6] refers to how software development is difficult [29]. It is based on the difficulty and challenges levels of several prospects including the required functionalities, algorithms, databases. and the number of interfaces. Increasing the complexity of a system would lead to an increase in the effort the members should make in developing the system [6].
- Experience. The technical experience and knowledge that team members gained from previous projects are important factors that should be considered [30]. This factor can be measured by the background of the team in performing similar functions, in addition to their familiarity with the required programming languages and tools [6].
- Time. In a software context, time refers to the duration a specific work should take to be done. It is not relatively tied with the simplicity of user story; a simple user-story may need a long time to be implemented based on its nature and the converse is also correct [31]. The time should reflect the actual work duration reasonably avoiding any schedule compression or expansion [6].
- Size. The size of the project can be defined as the amount of work required to accomplish project tasks[32]. It may also include the number of members involved in the project. Some techniques determine the size by using the user stories, where such techniques could impact the cost, scheduling, as well as the budgeting [33].
- Priority. Estimation plays a crucial role in prioritizing user stories. This prioritization also depends on several factors, including dependency within user stories [34]. The lack of prioritizing of user stories can result in poor quality of software development. However, in the developed tool in this paper, the product owner is the one who is responsible for prioritizing the stories in product backlog from high to low priority and it will not be included in the prediction algorithm.

In this research, a Scrum Estimator Poker tool based on a prediction algorithm was developed to overcome an unstructured story estimation. The prediction will be based on a set of standard criteria that has had its direct effect on project success proven. The next section explores the criteria that should be considered within the estimation procedure in order to acquire accurate estimates.

III. Experts Survey

The essential factors were determined by analyzing some of the current studies and books in the field, as previously explained in section F of Literature Review.

This section will explore the significance of these selected factors and determine the challenges that might face teams within the story estimation procedure based on field experts' opinions and experience.

A questionnaire was conducted to explore and measure the significance of the four different factors that had been selected and described previously which are: experience, complexity, time, and size (see Section II.F). The questionnaire focused only on the factors that the Scrum team considered while estimating each user story. According to Scrum rules, prioritizing user stories is the responsibility of the product owner[9]. Therefore, the priority factor is excluded in this step. The questionnaire targeted IT project experts and it was aimed to: (a) measure the importance of the selected factors; (b) explore the reasons for their importance; and (c) discover the possible challenges that would face the story estimation process. The questionnaire was distributed online using Google Forms [35], and it received nine responses varied between two IT Managers, five Software Developers and four Academic experts in software engineering. However, some experts play more than one role.

As shown in Figure 1, the questionnaire started by collecting experts' personal information, then indicating the role they played previously regarding IT projects. After that, one closed question posted to rank the importance of each factor, and one open-ended question to justify the reasons for choosing a specific rank. In addition to those, another open-ended question to examine the possible challenges the team may face during estimation.

In terms of experience, 66.7% of participants gave the highest importance rank of five to this factor, while others recognized its importance as well and ranked it with four. However, all participants, regardless of their role, believed that the experience is crucial in story point decision making, and it can lead to a more accurate and responsible estimation phase.

The impact of the complexity factor in user story estimation has an average ranking of 4.2 out of 5. From the Software Developers' perspective, they highlighted that the complexity of a task could affect productivity in short time. They also believe that considering the complexity of a task properly can help to prepare for it by breaking it down into smaller, more specific tasks. Others agree that it plays a crucial role in the

estimation process since hidden complexities would lead to incorrect decisions.

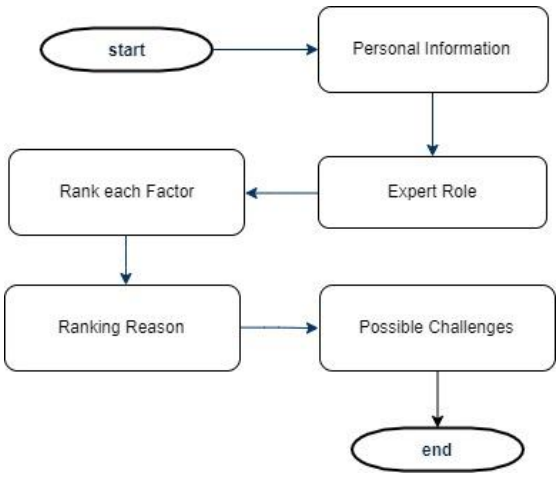

Figure 1 : Conducted questionnaire Workflow

The time factor got an average of 4.1 out of 5. More than half of the questionnaire sample agreed on time consideration importance within the estimation process. They believe that it would help to get better estimation results and to evaluate the work progress effectively, in addition to enabling sufficient assignment of tasks for each sprint. However, two participants ranked the time with 2 out of 5; one of them considered it as an independent factor.

The average impact of the size factor is 3.5 of 5. According to three participants who were questioned, they believe in the significance of the size as an estimation factor. While three out of nine ranked with three, they agreed that this factor depends on other factors such as effort, experience, and the complexity of tasks. Moreover, one of two people who ranked with two explained that the function is supposed to be relevant.

The experts had clarified several challenges regarding to story estimation; one of the challenges is expectations, estimating based only on expectation rather than considering complexity and time. Another challenge is the knowledge. Most people agreed that the lack of knowledge and experience in the past estimation can be a challenge for team members. Also, the project manager's attitude toward the project can play a significant rule by avoiding sudden changes in the project, and also managing team members' relationships by convincing the team regarding the estimated priority and time.

To sum up, the effectiveness of the four selected factors in estimating user stories has been proved by experts in the field of IT projects, even though these factors got different ranking points. The figure below shows a summarization of the factors and their ranks that resulted from the conducted questionnaire (see Figure 2). However, in the proposed Scrum poker tool, these factors will be used in a form of questions as inputs from team members to predict the appropriate story point which will be discussed in detail later.

The Scrum Estimator Poker is a Planning Poker tool targeting novice Scrum teams to assist them in the effort estimation process for user stories. The tool is designed to achieve the following objectives: (a) guide inexperienced team members towards more accurate and structured estimation through estimating user stories based on experts' factors that are presented in the form of questions. Therefore, the team member will answer each one of these questions for a single user story. (b) facilitates Scrum team negotiation and agreement during the estimation phase, by highlighting what is the appropriate estimate, overestimate and underestimate story points for each member.

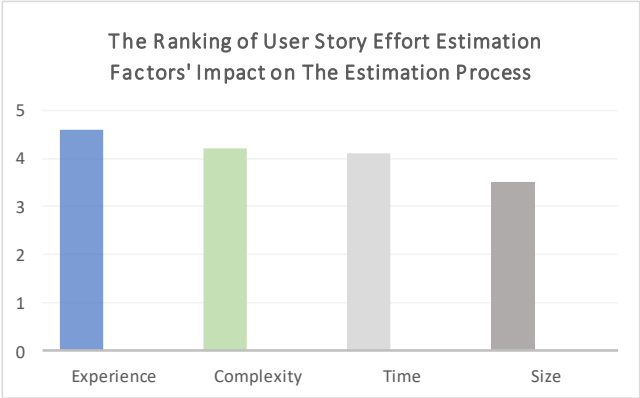

Figure 2: The Ranking of User Story Effort Estimation Factors' Impact on the Estimation Process.

IV. Scrum Estimator Poker Tool

A. Story Points Estimation Algorithm

In order to predict proper estimation values for user stories, an algorithm was constructed to handle this process. The algorithm comprises four steps as summarized in Figure 3. The steps are choose the measurement method; measure the weight of each factor; extract user questions from estimation factors; and determine the nearest Fibonacci number. All those steps will be discussed in detail below.

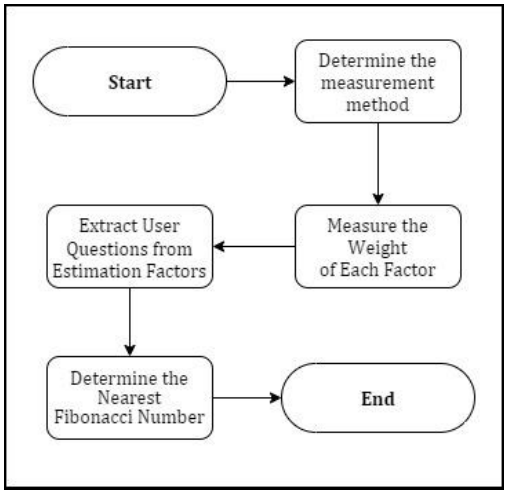

Figure 3: Estimation Algorithm Steps.

Step I: Choose the measurement method. A kind of arithmetic operation is performed on users' inputs regarding factors questions to obtain estimation value for a user story. Due to the unequal significance of estimation factors [14] and with the view of questionnaire results, a Weighted Arithmetic Mean [36]was chosen to calculate the estimation value because it allows unequal consideration of these factors. Weighted

Mean is a statistical method to measure the central tendency of a set of values influenced by its respective weight, where each value has a different weight based on its significance.

$$\sum xw = \frac{\sum_{i=1}^{n}(x_i * w_i)}{\sum_{i=1}^{n} w_i} \quad (1)$$

Where $\sum xw$ = weighted mean, $\sum_{i=1}^{n}$ = the summation, x_i = the values and w_i = the weight of each factor.

The weighted mean is measured by calculation of the sum of quantitative values by their associated weights. Thereafter the result is divided by the sum of weights.

Step II: Measure the Weight of Each Factor. The weight of any factor is determined by its importance[37], and the importance of each factor in the estimation process has been ranked by experts in the same field using a questionnaire (see Section III). The final rank value of a factor is the average of all experts' ranking for this factor. To measure each factor weight, the following equation is used, which states that the factor weight is represented by its contribution to the sum of total factors' ranks [36]:

$$\boldsymbol{Weight} = \frac{Factor\ Rank}{Sum\ of\ Factors\ Rank} \quad \textbf{(4.2)}$$

The table 1 summarizes the calculations performed in order to obtain factor weight, Where:

- Experience weight = 4.6/(4.6+4.2+4.1+3.5) ;
- Complexity weight = 4.2/(4.6+4.2+4.1+3.5) ;
- Time weight = 4.1/(4.6+4.2+4.1+3.5) ;

Size weight = 3.5/ (4.6+4.2+4.1+3.5).

		Factors			
Person	Role	Experience	Complexity	Time	Size
P1	Academic expert	5	5	5	3
P2	Academic expert	5	5	4	5
P3	IT Project Manager	5	2	2	2
P4	Software Developer	4	5	5	3
P5	Software Developer	5	5	5	5
P6	Software Developer IT project manager Academic Expert	4	3	2	2
P7	Software Developer	5	4	4	3
P8	Software Developer	4	3	5	4
P9	Academic expert	5	4	5	5
	Average Rank	4.6	4.2	4.1	3.5
	Weight	0.28	0.26	0.25	0.21

Table 1. Factors Weights.

Step III: Extract User Questions from Estimation Factors. To measure the pre-defined estimation factors per user story from the perspective of a Scrum member, a form of questions was extracted from these factors based on Section II.F, outlining the questions used to facilitate the estimation process for inexperienced members. The member will rank factors' questions using Fibonacci sequence. In the Scrum Estimator Poker tool, the experience factor aims to measure the member's background in performing similar functions as well as in the required programming languages and tools [6]. In addition, a complexity factor is represented by the difficulties a member could face in developing user story tasks [29], while time and size factors measure the duration and the amount of work needed to achieve use story tasks respectively [31, 32]. Questions are weighted based on the factor weight they derived from. If a factor is represented by more than one question, as in the experience factor, the weight will be divided among those questions. Table 2 illustrates the questions formula and its assigned weight.

Estimation Factor	Questions Formula	Weight
Experience	The background I have in performing similar functions as in this user story.	0.28
	The background I have with the required programming languages and tools.	
Complexity	The difficulties I may face in developing these story tasks.	0.26
Time	The time (duration) these story tasks will take to be accomplished.	0.25
Size	The size (amount) of work required to accomplish these story tasks.	0.21

Table 2. Estimation questions formula

Step IV: Determine the Nearest Fibonacci Number. As mentioned in Step III, a member should rank each factor's questions using Fibonacci sequence numbers. After estimating all questions for the selected user story, the question's points will be used as input into the Weighted Arithmetic Mean (Step I). As a result of variant gaps between Fibonacci numbers, the outcome of the Weighted Arithmetic Mean may not always equal an exact Fibonacci number. Therefore, the result is rounded to the nearest Fibonacci number [38]. To illustrate, if 10 is the calculated estimation, it is located between the two Fibonacci numbers 8 and 13. Thus, the nearest Fibonacci number of 10 is 8, so it will be the appropriate poker card for this user story.

B. System Workflow

The system workflow presents the flow of the activities needed towards a system completion. The workflow of the Scrum Estimator Poker is described in a UML activity diagram (see Figure 4).

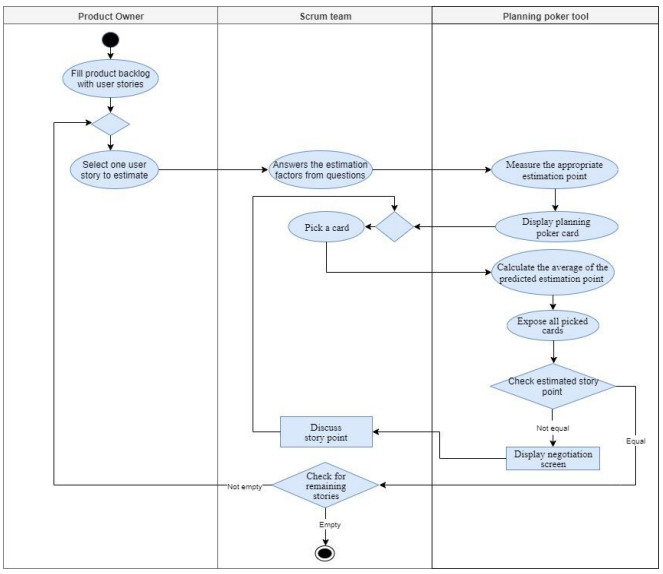

Figure 4: Activity Diagram

The flow of the tool is started when the Scrum team logged in to work on product backlog estimation. Then, the product owner has to fill the product backlog with tasks needed to complete the Scrum project in the form of user stories and order them based on their priorities. The prioritize process of the stories by product owner can help the team to better manage the project, since it can have a high effect on team effort. For example, story that had been first prioritized can bring more effort and need more time to be finished by the team since the resources are not enough yet. While the last story can take less effort because the resources are already available which also effect the final estimation value. After that, the product owner will select the first user story to be estimated by the Scrum team. They must estimate the selected story individually, by estimating a set of questions using the Fibonacci sequence as defined in Step III. Then, the Scrum Estimator Poker will measure the appropriate estimation point using weighted mean that was discussed previously in Step I. After that, cards containing Fibonacci numbers will be displayed for each member. Cards will be classified into three categories, which are: the appropriate card which contains the suggested Fibonacci number by the algorithm, overestimation cards that represent any Fibonacci number higher than the suggested one, and underestimation cards that are less than the appropriate card. However, the Scrum member has the freedom to choose a card from any of these categories.

After cards are picked by the Scrum team, the tool will calculate the average of all members' predicted estimation points to be suggested as the final story point. Then, all selected cards will be revealed at once, in addition to presenting their classification to give the team a clear view of the appropriate estimation point for each one of them, and whether they committed to it or not. If all the members choose the same estimated story point, then the estimation of this story will end, and the team will check for remaining stories to be estimated in the product backlog. In case it was the last story, the workflow of this tool will be ended. Otherwise, the process will be repeated until all stories pass the estimation process.

In case the checked estimated points are different, a negotiation screen will be displayed for the team to discuss their selected estimation story points and to express their viewpoints. After discussion, the team will re-estimate by picking a new card and the process will be repeated until they reach an agreement.

C. System Architecture

The system architecture of the Scrum Estimator Poker tool is defined by a layered architecture diagram (see Figure 5). The model contains three main layers which are the presentation layer, business logic layer, and data access layer. The presentation layer consists of two main packages that represent user interfaces in this tool, which are "Product owner screens" that relate to the managing of the product backlog and monitoring the estimation process, and "Scrum team screens", representing screens that cover the whole estimation process steps. The middle layer involves the system process in the tool, which is product backlog management, estimation management, and negotiation management. Finally, the data access layer contains all data needed by the system and their operations; data in the proposed tool is grouped into user stories data, estimation data, and user data.

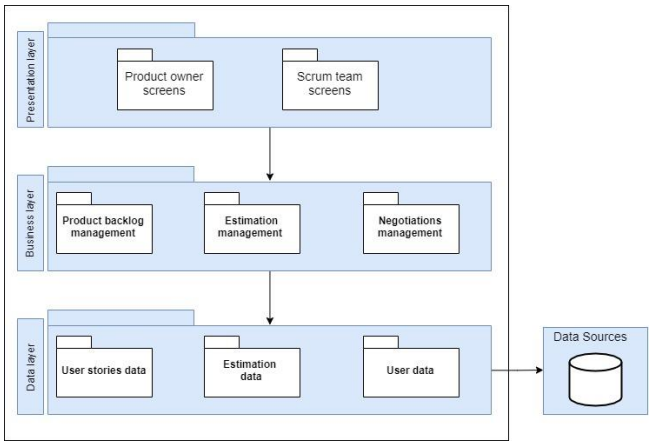

Figure 5: Scrum Estimator Poker Architecture.

D. System Interfaces

The user interfaces of the Scrum Estimator Poker tool are presented using a high-level prototype. Four main interfaces were designed in this prototype which are product owner interface, factors' form interface, and two different story estimation related interfaces with explanation for each interface (see Figure 6, 7,8 and 9).

The product owner must fill the product backlog with stories required to complete the project. These stories can be seen by the team members as well. As shown in Figure 6 each story is prioritized and has its description. The product owner can add new stories or edit an existing one if needed.The status of each user story will be determined as either estimated, in progress, or start the estimation.

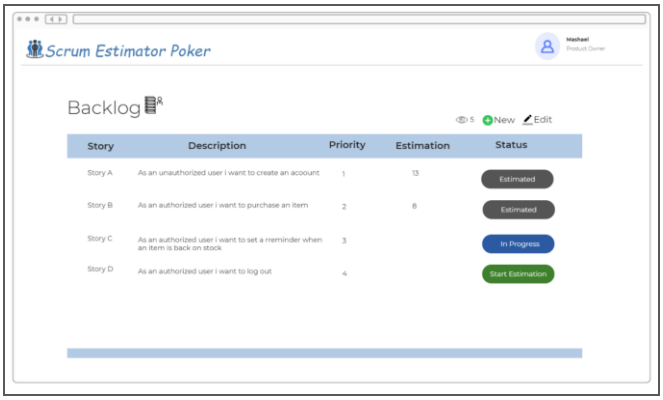

Figure 6: Product Owner page

Figure 7 simulates the factors' form page. This page contains a brief description of the under-processing user story and each team member should individually estimate the selected story through choosing the Fibonacci number that reflects their estimation regarding each factor. Questions included in the form are previously described (see Section IV.A).

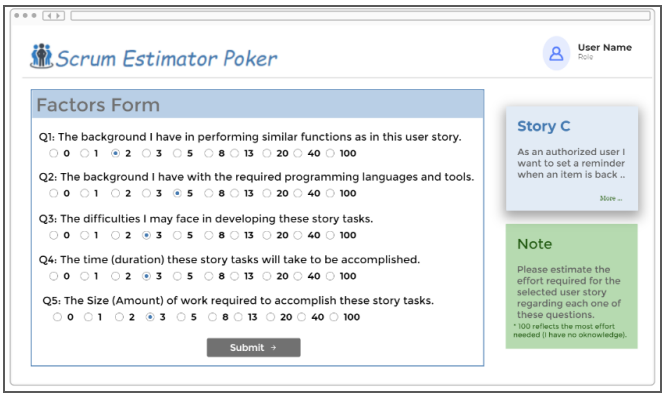

Figure 7: Factors Form Page

Figures 8 and 9 present story estimation interfaces. The team member has to pick a poker card from the colored cards below the page. The color of a card reflects its proximity to the proposed estimation in the following order; green expresses the proposed estimation card; underestimation cards are represented by yellow; red for overestimation cards. After all the team members estimate a user story, the poker cards will be flipped each with its color. As shown in Figure 9, members will negotiate using the negotiation room window in order to reach an agreement. If there is no agreement, members will re-estimate the story.

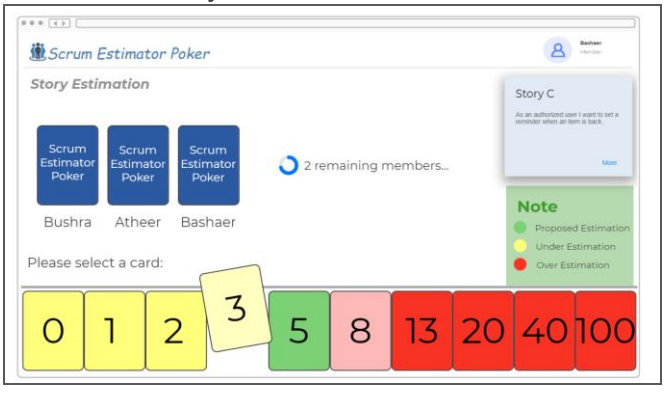

Figure 8: Story Estimation Interface

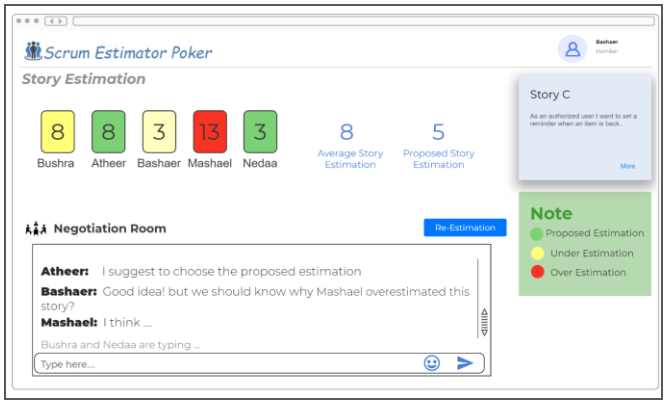

Figure 9: Story Estimation Interface

The next section will examine the testing procedure of the Scrum Estimator Poker tool and discuss its impact on the estimation process, in addition to measuring the proposed estimation accuracy.

V. Results and analysis

A practical implementation was conducted to measure the effectiveness of the developed Scrum Poker tool and its proposed estimation algorithm that aimed to improve the accuracy of user stories estimation for inexperienced teams. The result of this experiment is derived from comparing previous team estimations, a proposed algorithm estimation, and actual team estimations. All of the conducted estimations were based on the form of Fibonacci numbers. Here, the previous team estimation is the story point that team members assigned to each user story based on their own estimation approach. The proposed algorithm estimation, which was discussed in detail in Section IV.A, and the actual team estimation is the story point that team members believe reflects the real effort spent on implementing each user story when done after completing their project.

The experiment was performed on undergraduate students who followed a Scrum methodology in their senior projects. Two teams were initially selected. However, after starting the experiment, it turns out that one of the selected teams did not estimate the effort for user stories; they only relied on prioritizing them due to the lack of knowledge that they have in the estimation process in order to ensure the validity of the conducted test. Therefore, this team was excluded from this experiment and it was conducted only by one group, consisting of four students and five user stories grouped into four sprints. However, the experiments passed through four steps in order to be completed as follows:

- First, team members re-estimate their project user stories using the proposed algorithm individually.
- Then, they were asked to assign the actual effort they spent for these user stories together as a group and justify the reasons behind differences between previous and actual estimation.
- After that, the accuracy of the proposed algorithm was measured and compared to both actual and previous estimations.
- Finally, team members evaluated the effectiveness and

usability of the developed Scrum Estimator Poker tool.

A. Re-estimating User Stories Using the Proposed Algorithm

The team members re-estimated the user stories individually according to the factors' form which was previously described in Section II.F. Then, the appropriate estimation value for each team member is measured using the proposed algorithm (see Section IV.A). Each of the obtained values is converted to the nearest Fibonacci number. The average of the proposed estimation of the team members was also calculated and converted to the nearest Fibonacci number, which represents the final user story estimation, as shown in the Table3.

Story Number	Person	Q1	Q2	Q3	Q4	Q5	Weighted Mean	Nearest Fibonacci	Proposed Average	Nearest Fibonacci
Story A	P1	13	13	21	21	8	16.03	13	30	34
	P2	8	2	8	5	8	6.41	5		
	P3	13	5	8	13	13	10.58	13		
	P4	89	89	89	89	89	89	89		
Story B	P1	2	3	2	3	3	2.6	3	3.25	3
	P2	5	2	3	3	5	3.56	3		
	P3	3	1	2	2	2	2	2		
	P4	3	3	5	5	5	4.44	5		
Story C	P1	2	3	1	3	2	2.13	2	5.25	5
	P2	3	2	3	3	3	2.86	3		
	P3	3	1	3	3	3	2.72	3		
	P4	8	13	21	13	13	14.38	13		
Story D	P1	13	21	21	34	21	23.13	21	13.75	13
	P2	8	5	5	5	5	5.42	5		
	P3	8	5	8	8	3	6.53	8		
	P4	13	13	21	21	13	17.08	21		
Story E	P1	1	5	1	2	5	2.65	3	2.75	3
	P2	2	2	1	2	1	1.53	2		
	P3	1	1	1	1	1	1	1		
	P4	5	5	5	8	8	6.38	5		

Table 3: User stories estimation using the proposed algorithm

Where Qi in the table represents the individual estimates regarding the questions of the factors' form (see Section IV.A).

B. Actual Effort of User Stories

Team members discussed with each other to determine the actual effort of each user story. The members considered the previous effort in similar estimated tasks and what goes out of their expectations, and the reasons behind these differences. The team agreed that collective effort estimation hides individual differences, which make them misestimate the user story. One real example for that is story A, where one team member out of the four had better experience to deal with this story. This made the other members rely on the experience of that member. Therefore, they estimated the effort rank as 3. After they start the sprint, they realized that it required the effort of four members not just one! Therefore, the actual effort of this story jumped from 3 to 21, which resulted in exceeding the expected delivery time from two weeks to around two months. The team argued that if they knew that this story had high estimation, they will reduce the risk of late delivery by dividing the user story among more than one sprint. Stories C and D were also underestimated, but with a difference less than story A. In contrast, the members estimated the correct effort in stories B and E. Table 4 shows the actual effort of each user story.

	Previous Estimation	Actual Effort
Story A	3	21
Story B	3	3
Story C	2	5
Story D	5	8
Story E	1	1

Table 4. Actual Effort of User Stories

C. Algorithm Accuracy

The accuracy of any estimation algorithm is referring to how these estimations are relevant to the actual project effort [6]. Therefore, the proposed algorithm has been evaluated based on the actual effort stated by the group using a burndown chart diagram and an accuracy formula, which will be discussed later in this section. The burndown chart shows the total effort needed in each user story [10] and demonstrates the results of the previous estimation, actual estimation, and the algorithm developed in this research (see Figure 10). Overall, as shown in Figure 10, compared to the previous estimation, the proposed estimation had closer effort values to the actual estimation.

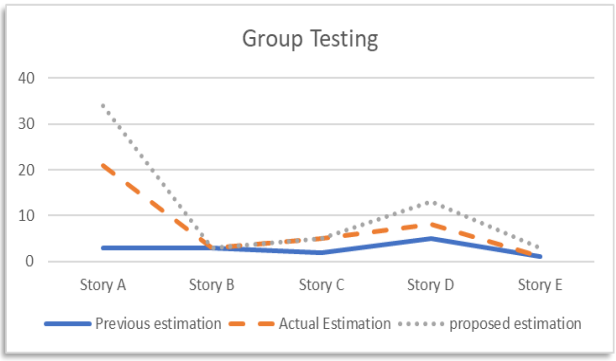

Figure 10: Burndown Chart for Effort Estimation

There was a clear variation in terms of estimating story A. The team agreed that they underestimated it. As a result, their actual effort was 21, which was closer to the proposed estimation that was 34. On the other hand, story B shows how the proposed estimation had exactly the accurate effort needed by the team and was matching to their actual effort and previous estimation. Unlike story C, the proposed estimation had the exact effort of the actual effort comparing to the previous estimation. In terms of Story D, actual effort is closer to the previous estimation comparing to the proposed estimation, while the actual effort of story E had the exact value of team previous estimation.

In order to present the algorithm accuracy in a form of quantitative data, the accuracy formula is used, which represents the ratio of correctly estimated values to the total estimated values, as follows:

$$Accuracy = \frac{Correctly\ estimated\ values}{Total\ estimated\ values} \quad (4.2)$$

The accuracy of the proposed estimation algorithm and the previous estimation were compared and measured first for each user story, and then for all stories as a project estimation. The overall result of this compression stated that the proposed algorithm has proved its efficiency by estimating the user stories more accurately (71.32%) comparing to the team's initial estimation (63.35%), which was higher by around 7.97%. Additionally, the proposed algorithm resulted in a perfect accuracy of 100% for both story B and story C. However, this accurate rate was dropped to around 61% for story A and D, while it got the lowest rate in estimating story E (33.33%), as shown in Table 5.

	Previous Estimation	Proposed Estimation	Actual Effort	Previous Accuracy (In %)	Proposed Accuracy (In %)
Story A	3	34	21	14.28%	61.76%
Story B	3	3	3	100%	100%
Story C	2	5	5	40%	100%
Story D	5	13	8	62.5%	61.53%
Story E	1	3	1	100%	33.33%
Average Accuracy of Initial Estimation				63.35%	
Average Accuracy of Proposed Estimation				71.32%	

Table 5: Proposed and Previous Estimation Accuracy

Finally, based on the test that was done, the estimation of the proposed algorithm was more accurate compared to the initial estimation in four out of five user stories, and almost the same in one user story, while it got a weak result in the fifth remaining story.

D. Effectiveness and Acceptance Evaluation

Regarding the Scrum Estimator Poker tool properties and interfaces, the tool was examined by the testing group from two perspectives: (a) how the tool structured estimation facilities in the estimation process; (b) the acceptance of colored poker cards.

First, the group members were asked about the efficiency of including the factors' form. All group members agreed that following a criteria-based estimation procedure will be more logical, much clearer, and will often lead to a more realistic estimation value. They pointed out that the factors' form makes the estimator aware of different aspects that must be considered within the estimation, unlike in the unstructured estimation. where a member might estimate the user story based on the member's feelings regarding the story difficulty level, or this member might be affected by the discussion and other members' estimations.

Secondly, with regards to the tool appearance, the group members were asked about the proposed estimation poker cards' colors (see Section IV.D). They expressed their admiration. Three of them indicated that visualizing the individual estimates with different colors allows other members to detect their colleagues' capabilities regarding each user story. They believed that this will help to reduce the time needed for discussion within the estimation. One member thought it might help to assign the work tasks more realistically based on each member's skills.

Eventually, the overall accuracy of the proposed tool in calculating the estimation of product backlog user stories was not 100%. However, it presented greater accuracy compared to the testing group's previous estimations. Consequently, the risk of unstructured and inexperienced members' estimation was maintained to be limited. Furthermore, Fibonacci numbers imply increasing gaps between the series' numbers by nature that might affect the reliability of estimation values accuracy. For example, in Fibonacci poker cards the cards 21 and 34 directly come sequent. However, in terms of the common calculation, the difference between these cards is equal to 13. The tool proved its accuracy when applied to the selected project product backlog, but the results may vary when applied on another project due to unrecognized conditions.

The next section concludes the paper by illustrating the challenges faced during the study and the current limitations. It also presents a set of recommendations and detects future work opportunities.

VI. Conclusion

Effort estimation plays a crucial role in the success of Scrum projects since the lack of team experience is considered as one of the reasons that leads to unsatisfactory estimation results. This research proposed an estimation algorithm calculated by weighted mean based on Planning Poker techniques and Fibonacci sequences. A Scrum Estimator Poker tool helps the teams work on Scrum projects by guiding team members and structuring the effort estimation process toward more accurate estimations relying on several standard factors. Moreover, it enhances negotiation across team members. The tool determines the proper estimation value for each story based on different criteria that are collected from each team member as inputs. The criteria rely on several significant factors which are complexity, experience, time, and size. As for the finding of this research, the tool can be used as a guideline for better and more accurate effort estimations in Scrum projects to reduce the effort estimation errors.

VII. Recommendations and Future Work

Since this paper had only tested one group whose members are undergraduate students, which was one of the limitations of the project. moreover, the tool is recommended for inexperience teem who may misestimate product backlog stories, Therefore, one member especially the person with the product owner role should have experience in evaluating user stories. the future work will involve more experiments in order to increase the reliability of the proposed algorithm results and will be tested by real Scrum project development organizations. Also, more experts with different specialties and skills will be involved in order to increase the accuracy of the weighted factors. In addition, it is recommended that further studies explore how to map the desired effort to each Fibonacci number, which will asset novice teams in differentiating between these numbers

and selecting the most appropriate one.

References

[1] K. S. RUBIN, *Essential Scrum : a practical guide to the most popular agile process* 2012.

[2] L. Gonçalves, "Scrum Controlling & Managment Review," pp. 40-42, 2018.

[3] K. Schwaber and J. Sutherland, *The Scrum Guide*. Scrum. org, 2017.

[4] V. M. a. T. Hovelja, "On using planning poker for estimating user stories," *Journal of Systems and Software*, vol. 85, pp. 2086 - 2095, 2012.

[5] A. E. Babiker, A. Mahmoud, and A. Abdalrahman, "Sprint Backlog Estimating and Planning Using Planning Poker Technique in Agile Scrum Framework," *2018*, vol. 8, no. 5, p. 4, 2018-05-30 2018.

[6] R. J. Adam Trendowicz, *Software Project Effort Estimation*. Springer

2014.

[7] N. C. Haugen, "An empirical study of using planning poker for user story estimation," presented at the AGILE 2006 (AGILE'06) Minneapolis, MN, USA, 2006.

[8] M. A. B. Emam Hossain, Hye-young Paik , June Verner, "Risk Identification and Mitigation Processes for Using Scrum in Global Software Development: A Conceptual Framework," presented at the 16th Asia-Pacific Software Engineering Conference, 2009.

[9] H. T. I. Hrafnhildur Sif Sverrisdottira, Haukur Ingi Jonassonc, "The role of the product owner in scrum-comparison between theory and practices," *Procedia - Social and Behavioral Sciences*, vol. 119, no. 1877-0428, pp. 257 - 267, 2014.

[10] J. Evans, "SCRUM REVEALED TRAINING BOOK," *International scrum institution*

[11] O. a. L. Ktata, Ghislain, "Designing and Implementing a Measurement Program for Scrum Teams: What do agile developers really need and want?," pp. 101-107, 2010.

[12] L. M. Marek Majchrzak, "Factors influencing user story estimations: an industrial interview and a conceptual model," *Central and Eastern European Journal of Management and Economics*, vol. 4, pp. 261-280, 2016.

[13] L. B. Helen Sharp, Advait Deshpande, Peggy Gregory, Katie Taylor, "Remote working in an Agile team," *TheAgileResearchNetwork*, 2016.

[14] T. SCHWEIGHOFER, A. KLINE, L. PAVLIC, and M. HERIC, "How is Effort Estimated in Agile Software Development Projects?," presented at the 5th Workshop of Software Quality, Analysis, Monitoring, Improvement, and Applications, Budapest, Hungary, 28-31.8.2016, 2016. Available: http://ceur-ws.org/Vol-1677/paper09.pdf

[15] M. M. Crawford and G. Wright, "Delphi Method," Wiley StatsRef: Statistics Reference Online,, 2016, p. 7.

[16] G. Aiello, M. Cossentino, M. Alessi, and A. Urso, "RTDWD: Real-Time Distributed Wideband-Delphi for User Stories Estimation," presented at the Rapid Integration of Software Engineering Techniques, Geneva, Switzerland, September 13-15, 2006, 2006. Available: https://www.researchgate.net/publication/220872333_RTDWD_Real-Time_Distributed_Wideband-Delphi_for_User_Stories_Estimation?enrichId=rgreq-81e35f3a1e27b47ec4e8b1f245011dc5-XXX&enrichSource=Y292ZXJQYWdlOzIyMDg3MjMzMztBUzo5ODk4NjEyODA1MjIzMkAxNDAwNjExNTQwODIx&el=1_x_2&_esc=publicationCoverPdf

[17] M. Cohn, *Agile Estimating and Planning*, 2 ed. Pearson Education, Inc., 2006, p. 320.

[18] J. López-Martínez, A. Ramírez-Noriega, R. Juárez-Ramírez, G. Licea, and S. Jiménez, "User stories complexity estimation using Bayesian networks for inexperienced developers.," *Cluster Computing*, 2018.

[19] Ziauddin, S. K. Tipu, and S. Zia, "An Effort Estimation Model for Agile Software Development," *Advances in Computer Science and its Applications (ACSA)*, vol. 2, pp. 314-324, 2012.

[20] A. Panda, S. M. Satapathy, and S. K. Rath, "Empirical Validation of Neural Network Models for Agile Software Effort Estimation based on Story Points," presented at the 3rd International Conference on Recent Trends in Computing, 2015.

[21] S. M. Satapathy and S. K. Rath, "Empirical assessment of machine learning models for agile software development effort estimation using story points," *Innovations in Systems and Software Engineering*, 2017.

[22] S. M. Satapathy, A. Panda, and S. K. Rath, "Story Point Approach based Agile Software Effort Estimation using Various SVR Kernel Methods.," presented at the The 26th International Conference on Software Engineering and Knowledge Engineering, Vancouver, Canada, 2014.

[23] M. Choetkiertikul, H. K. Dam, T. Tran, T. Pham, A. Ghose, and T. Menzies, "A Deep Learning Model for Estimating Story Points," *IEEE Transactions on Software Engineering*, vol. 45, pp. 637-656, 2019.

[24] A. Trendowicz and R. Jeffery, *Software Project Effort Estimation*. Springer, 2014.

[25] H. H. O. a. M. E. Musa, "A Survey of Agile Software Estimation Methods," *International Journal of Computer Science and Telecommunications*, vol. 7, no. 3, pp. 38-40, 2016.

[26] A. R.-N. Janeth López-Martínez, Reyes Juárez-Ramírez, Guillermo Licea, Samantha Jiménez, "User stories complexity estimation using Bayesian networks for inexperienced developers," *Cluster Computing*, vol. 21, no. 1, pp. 715–728, 2018.

[27] H. Z. M. A. J. Idrissi, "Adjusting story points calculation in scrum effort & time estimation," 20-21 Oct. 2015 2015.
[28] W. V. Panut Chongpakdee, "Estimating user story points using document fingerprints," presented at the 2017 8th IEEE International Conference on Software Engineering and Service Science (ICSESS), 2017.
[29] N. C. Rashmi Popli, "Agile Estimation Using People and Project Related Factors," 2014.
[30] R. J.-R. Janeth López-Martínez, Alan Ramírez-Noriega, Guillermo Licea, Raúl Navarro-Almanza, "Estimating User Stories' Complexity and Importance in Scrum with Bayesian Networks," presented at the Recent Advances in Information Systems and Technologies. WorldCIST 2017. Advances in Intelligent Systems and Computing, 2017.
[31] N. C. Rashmi Popli, "Cost and Effort Estimation in Agile Software Development," presented at the International Conference on Reliability, Optimization and Information Technology, Faridabad, India, 2014.
[32] E. a. B. Coelho, Anirban, "Effort estimation in agile software development using story points," *International Journal of Applied Information Systems (IJAIS)*, vol. 3, 2012.
[33] M. a. S. Ali, Zubair and Ali, Eaman, "Estimation of Project Size Using User Stories," 2015.
[34] R. P. N. C. H. Sharma, "Prioritising user stories in agile environment," 2014.
[35] T. Weber, "Cloud computing for business goes mainstream," *BBC News Business*, 05/May/2010. Accessed on: 01/March/2013Available: http://www.bbc.co.uk/news/10097450
[36] F. J. Gravetter and L. B. Wallnau, "Statistics for the Behavioral Sciences," *Book*, 2013, 2010 Wadsworth, Cengage Learning.
[37] S. Manikandan, "Measures of central tendency: The mean," *Journal of Pharmacol Pharmacother*, vol. 2, pp. 140–142, 2011.
[38] H. KREBSBACH, "Ideas to optimize your planning poker sessions," June 20, 2016. Available: https://www.atlassian.com/blog/agile/planning-poker-sane-healthy

Author Biographies

Bushra Alsaadi received her master's degree in computer information systems at King Abdelaziz University in Jeddah, Saudi Arabia. She worked as a research assistant and software developer at the Faculty of Computing and Information Technology, King Abdulaziz University. Her research interest focuses on data science, health informatics, human-centered design, and software engineering.

Bashaer Alsaadi received her master's degree of Computer Information Systems at King Abdulaziz University in Jeddah, Saudi Arabia. She is working as a research assistant at the Faculty of Computing and Information Technology, King Abdulaziz University. Previously, Bashaer was working as a software developer at King Abdulaziz University. She is interested in the field of software engineering and improving scrum team experience.

Mashaal is doing a master's degree of computer information system at King Abdulaziz University, she graduated with bachelors degree in computer information system in 2018. Mashaal's undergraduate project was focus on big data analytics, she also interested in software engineering and e-learning field.

Athir Saeed Alghamdi is doing her Master's degree of Information Systems at King Abdulaziz University in Jeddah, Saudi Arabia. She expected to submit her thesis by the beginning of 2021. She is interested in technology adoption and information systems. This published work is part of proposed future research to develop and validate a tool that helps inexperienced scrum team members during estimations toward more accurate user-stories' estimates.

Nedaa is doing her Master's degree of Information Systems at King Abdulaziz University in Jeddah, Saudi Arabia. She expected to complete her degree before the end of 2021.

Bahjat Fakieh is working as assistant professor at King Abdulaziz University in Jeddah - Saudi Arabia. Previously, Bahjat was working as a lecturer in Federation University in association with ATMC college campus in Sydney for 1.5 years to teach cloud computing for a postgraduate program. In addition, he was working as an Information Systems tutor in Macquarie University for 4.5 years.
His research interest is data analysis, digital transformation, information systems, e-business, and e-commerce. Currently, Bahjat is working on data analysis from the medical perspectives and technology adoption for small and medium sized enterprises (SMEs) and information systems projects implementations.
In addition, Bahjat is holding several globally recognized professional certificates in IT service management, as well as working as a coach for business development from the technology perspective.