A Fuzzy based Model for Effort Estimation in Scrum Projects

Jasem M. Alostad
The Public Authority for Applied Education and Training (PAAET),
College of Basic Education
P.O. Box.23167, Safat 13092,
Kuwait

Laila R. A. Abdullah
The Public Authority for Applied Education and Training (PAAET),
College of Business Studies
P.O. Box.23167, Safat 13092,
Kuwait

Lamya Sulaiman AAli
The Public Authority for Applied Education and Training (PAAET),
College of Business Studies
P.O. Box.23167, Safat 13092,
Kuwait

Abstract—**This paper aims to utilize the fuzzy logic concepts to improve the effort estimation in Scrum framework and in turn add a significant enhancement to Scrum. Scrum framework is one of the most popular agile methods in which the team accomplishes their work by breaking down the work into a series of sprints. In Scrum, there are many factors that have a significant influence on the effort estimation of each task in a Sprint. These factors are: Development Team Experience, Task Complexity, Task Size, and Estimation Accuracy. These factors are usually presented using linguistic quantifiers. Therefore, this paper utilizes the fuzzy logic concepts to build a fuzzy based model that can improve the effort estimation in Scrum framework. The proposed model includes three components: fuzzifier, inference engine, and defuzzifier. In addition, the proposed model takes into consideration the feedback that is resulted from comparing the estimated effort and the actual effort. The researcher designed the proposed model using MATLAB. The proposed model is applied on three Sprints of a real software development project to present how the proposed model works and to show how it becomes more accurate over time and gives a better effort estimation. In addition, the Scrum Master and the development team can use the proposed model to monitor the improvement in effort estimation accuracy over the project life.**

Keywords—Scrum; sprint; effort estimation; fuzzy logic; fuzzy inference system

I. Introduction and Problem Definition

Recently, agile software methods have gained a great importance in the field of software projects [1]. Agile software methods provide an excellent solution in the cases of the vague or changing requirements [2]. The software's owner and the development team prefer agile software methods because of their ability to provide a much needed release that has the highest value for business [3]. The most common agile software methods are: eXtreme Programming (XP) and Scrum. In addition, agile software methods include: Feature Driven Development, Adaptive Software Development, Crystal, and Dynamic System Development Methodology [4].

Scrum in the most commonly used agile methods. Scrum is a good method for projects that have critical deadlines, complex requirements, and a significant degree of uniqueness [5]. Scrum is an iterative and incremental approach for managing the software projects in a changing environment. Each iteration aims to produce a potential set of the software functionality [6]. A scrum-based project typically moves forward through a series of iterations called sprints, and each sprint is two to four weeks long.

Before starting any sprint, the Product Owner, Scrum Master, and Development Team hold a meeting which is called "sprint planning meeting". In sprint planning meeting, the attendees decide on a sprint goal that defines what must be achieved in the next sprint [7]. Then, they review the product backlog to select the highest priority items that will be included in the next sprint. The attendees estimate the completion time for each selected item using an estimation technique; such as story points [8]. Each task is estimated in story points based on its complexity.

The estimation process is a very complicated process because it depends on many factors; such as the experience of the developers. The level of experience is different from a developer to another. Some developers are experienced and many tasks are easy for them, while the same tasks are not easy for the others.

In a typical Scrum, there are many factors are not taken into consideration; such as the experience of the developers, effort estimation accuracy, etc. These factors are usually presented using linguistic quantifiers. Therefore, the researcher uses fuzzy logic concepts to build a model that enhances the effort estimation process of Scrum framework. The proposed model depends on: fuzzifier, inference engine, and defuzzifier. In addition, a comparison between the estimated effort and the actual effort is done and useful to evaluate the estimation accuracy. The proposed model is applied on a real software development project to simulate how it works and to present its benefits.

This paper is organized into six sections. Section II introduces a background overview that covers Scrum, effort estimation techniques, and fuzzy logic. Section III provides some significant related work focusing on using fuzzy logic in Scrum. Section IV presents the proposed fuzzy model. Section V introduces how to apply the proposed model on real Sprints of a project. Section VI concludes the paper with final remarks and presents the ideas that are expected to be focused on the future.

II. Background Overview

This section aims to clarify the three basic topics of this paper, which are: Scrum framework, effort estimation techniques, and fuzzy logic concepts. Therefore, this section includes three subsections to provide a brief explanation for these topics.

A. Scrum Framework

Scrum framework is one of the most popular agile methods, used to manage software projects [9]. According to Scrum framework, the team accomplishes their work in software projects by utilizing the improved communication and collaboration among the members and breaking down the work into a series of sprints. Scrum framework includes three main components; Scrum team, events, and artifacts [10]. These components are managed and controlled by explicit rules. Fig. 1 illustrates the components of Scrum framework.

The Scrum team is self-organized and cross-functional in a way that lead to enhanced cooperation, flexibility, creativity, and productivity of the team members. Scrum team has three roles: Product Owner, Scrum Master, and Development Team. The responsibility of the Product Owner is to define the business value and requirements of the project. Moreover the Product Owner also prioritizes the requirements [9]. Scrum Master must ensure that the values, practices, and rules of the Scrum framework are clear to all team and well applied. Scrum events are well-defined and time-boxed to facilitate the work of the Scrum team [10]. Scrum events include: Sprint, Sprint Planning Meeting, Daily Scrum, Sprint Review, and Sprint Retrospective.

Scrum framework includes three main artifacts: Product Backlog, Sprint Backlog, and Increment. Product Backlog includes a refined and prioritized list of tasks [12]. Sprint Backlog is a subset of Product Backlog items that must be in the Sprint to achieve the Sprint goal.

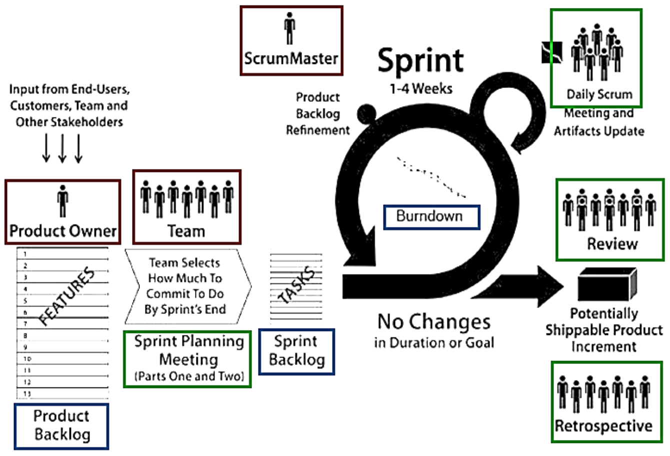

ScrumMaster
Sprint
1-4 Weeks
Daily Scrum
Meeting and Artifacts Update
Input from End-Users, Customers, Team and Other Stakeholders
Product Backlog Refinement
Product Owner
Team
Burndown
Review
FEATURES
Team Selects How Much To Commit To Do By Sprint's End
TASKS
No Changes
in Duration or Goal
Potentially Shippable Product Increment
Sprint Planning Meeting (Parts One and Two)
Sprint Backlog
Product Backlog
Retrospective

Fig. 1. Components of Scrum framework [11].

According to [4], [5], [10], [13], the basic activities that are generally performed in Scrum software projects can be summarized as follows:

- A product owner prepares a list of features that are required in the new software system. Then, this list is validated, prioritized, and put in a product backlog.
- In the sprint planning meeting, the team withdraws a small part from the top of the product backlog and form a sprint backlog. Then, they determine how to achieve those pieces.
- The team members accomplish their work through the Sprint.
- The team meets every day, daily Scrum, to monitor the Sprint progress.
- At the end of the sprint, the team delivers a potentially shippable software piece to the users.
- The sprint ends with a sprint review and retrospective.
- As the next sprint starts, the team selects another part of the product backlog and starts the work again.

B. Effort Estimation Techniques

Effort estimation techniques in the software domain are classified into algorithmic and non-algorithmic models [14]. The most popular non-algorithmic techniques are: Expert Judgment, Delphi technique, Thumbs Rule, Pricing to win, and Parkinson's Law.

The most popular algorithmic models are [14]-[17]: Line Of Code (LOC), KLOC, COCOMO, COCOMO-II, Function Point, and Story Points. Algorithmic models depend on the statistical analysis of historical data. These models require accurate input of specific attributes related to the software project. In this paper, the researcher will use Story Points as a measure of effort estimation in in Scrum projects.

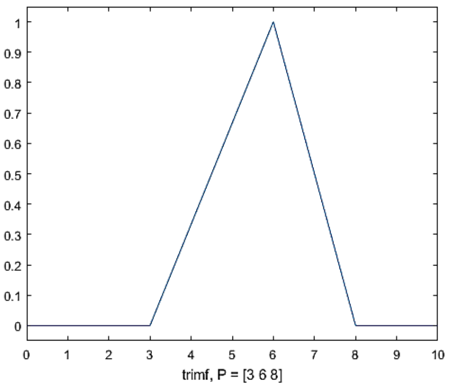
Story Points indicate to an estimate of the relative scale of the work in terms of actual development effort. Story Points are expressed either in numbers that follow the Fibonacci series or T-shirt sizes (XS, S, M, L, XL, XXL) [17]. Effort estimation using Story Points is typically achieved through relative sizing by comparing a story with a sample set of previously estimated stories. In turn, this process is more accurate over a larger sample.

C. Fuzzy Logic

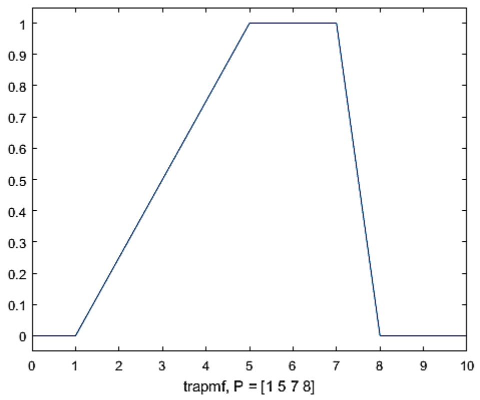
Fuzzy logic is used for solving the problems that are described by linguistic quantifiers or are complex to be understood quantitatively [18], [20]. Fuzzy Logic System deals with fuzzy parameters, which address imprecision and uncertainties using the computing framework called the Fuzzy Inference System. Fuzzy logic is based on fuzzy set theory and introduced in 1965 by Lotfy Zadeh [19].

The fuzzy membership functions are used for fuzzifying the input data, the process of transformation continues variables to [0,1] interval [21]. Fuzzification aims to convert the crisp input data into a fuzzy set. The most common fuzzy membership functions are triangular-shaped, trapezoidal-shaped, PI-shaped, S-shaped, Z-shaped, and Bell-shaped functions. Triangular-shaped Membership Function is characterized with three values representing its vertices [22] while Trapezoidal-shaped

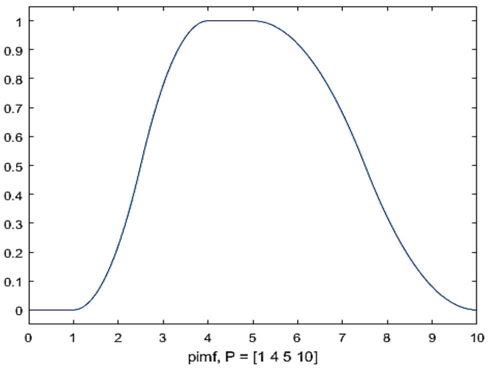
Membership Function is characterized with four values representing its vertices [23]. PI-shaped Membership Function is represented by four values where the first and the last are locating "feet" of the curve, while the others locate its "shoulders" [24]. Fig. 2 illustrates examples of common fuzzy membership functions. The inverse process of fuzzification is called defuzzification that aims to produce crisp values from fuzzy values. The most common defuzzification methods are [25]; bisector of area (BOA), centre of area (COA), etc.



(a) Triangular-shaped membership function



(b) Trapezoidal-shaped membership function



(c) PI-shaped membership function

Fig. 2. Examples of common fuzzy membership functions [22]-[24].

A Fuzzy Inference System (FIS) is a rule based system that consists of four components: Fuzzifier, Fuzzy Rule Base, Fuzzy Inference Engine, and Defuzzification. A Fuzzy Inference Engine is a collection of IF -THEN rules that are stored in the Fuzzy Rule Base. These rules are defined by an expert in the application field or they can be learned from the current data. These rules are useful for decision making depending upon the occurrence of the conditions of IF statements.

III. Related Work

Fuzzy logic is useful for building an expert system when inputs are expressed as linguistic quantifiers. In the following paragraphs, the researcher introduces briefly some important researches related to one or more issue of the domains: fuzzy logic, scrum or agile projects, and effort estimation in software development.

Colomo-Palacios et al., developed a hybrid recommender system for Scrum team roles based on fuzzy logic, rough set theory and semantic technologies. The proposed system provides a powerful tool for project managers to support the development process in Scrum environments and to help them to form the most suitable team for different work tasks. The recommendation of the proposed system is based on the staff available for the project and the competences required for each task. The proposed system has been evaluated on a real data of the software development cycle [26].

Vishal S., et al., proposed an optimized fuzzy logic based framework to estimate efforts in software development process. The performance of the proposed framework is evaluated and validated using live project data of COCOMO public database. Moreover, the proposed framework takes into consideration imprecision and knowledge of experts. The proposed framework explains prediction rationale through rules, offers transparency in the prediction system, and could adapt to changing environments with the availability of new data [27].

Ziauddin A., et al., presented a fuzzy logic based software cost estimation Model. This paper aims to utilize a fuzzy logic model to increase the effort estimation accuracy. This paper aims to fuzzify input parameters of COCOMO II model and then the outcomes are defuzzified to get the resultant Effort. The proposed model is based on Triangular fuzzy membership function to represent the linguistic terms in COCOMO II model [28].

Sedehi H. et al., introduced a short description of two methodologies; Goal Question Metric (GQM) and Practical Software and Systems Measurement (PSM). In this description, the paper focused on selecting a number of "sensible" metrics in agile based software development context. Moreover, the paper focused on the fuzzy set theory, fuzzy logic, with the associated rule based reasoning model, and their implementation on the selected metrics in order to evaluate and monitor an agile based project [29].

Assem H. et al., proposed a fuzzy based framework to calculate the success metrics related to agile software projects. This paper helps in calculating the Success Metric Value (SMV) based on the values of success factors and the importance value of each success factor. The proposed framework helps the stakeholders of the agile based project to represent the values of the success factors in a human-like language [30].

Abeer H. proposed a fuzzy based model for enhancing the accuracy and sensitivity of COCOMO model by fuzzifying the cost drivers. This model was designed and implemented using MATLAB. The dataset was gathered from six NASA centers in a way to cover a wide range of software domains, development process, languages and complexity, culture differences, and business differences. This paper proves that the sensitivity of the proposed fuzzy based model is superior to Intermediate COCOMO [31].

Prasad Reddy et al., utilized Fuzzy Triangular Membership Function and Gaussian Bell Membership Function to predict effort of software development process. The two membership functions are implemented and compared with COCOMO. Moreover, a dataset from NASA93 is used to compare the proposed fuzzy model with the Intermediate COCOMO. It revealed that the Fuzzy Logic Model using Triangular Membership Function lead to better results than the other models [32].

The researcher finds out that most of previous efforts in the domain of effort estimation and Scrum are not enough because they neglect some important factors or they are not directly related to this issue. Therefore, this paper aims to utilize the fuzzy logic concepts to build a fuzzed based model that can improve the effort estimation in Scrum framework.

IV. Proposed Fuzzy based Model

This paper aims to design a fuzzy logic based model which simulates the role of scrum master and development team in effort estimation during the sprint planning phase. Specifically, the proposed model utilizes the fuzzy logic concepts to improve the effort estimation of each task in the sprint planning meeting. To achieve this objective, the researcher tries to make the proposed model is simple, understandable, applicable, and reliable. Therefore, the proposed model takes into consideration the dominant factors that have a significant influence on the effort estimation process. These factors are: Development Team Experience, Task Complexity, Task Size, and Estimation Accuracy. In the following, a brief explanation for them.

- Development Team Experience (TE): It is the amount of experience belongs to each developer. The number of years that were spent in the work is the most suitable measure of experience. As the years pass, the developer gains more knowledge, skills, training, etc. There are three levels of developer's experience: Junior, Intermediate, and Senior.
- Task Complexity (TC): It is influenced by many sub-factors such as; task architecture, the relationships among its components, task regularity, uncertainty, and the required changes [33]. It is described by five linguistic terms; Very Easy, Easy, Moderate, Complex, and Very Complex.
- Task Size (TS): It is the initial size determined by the developer. It is described by four levels as T-shirt size; Small, Medium, Large, and X-Large.
- Estimation Accuracy (EA): It represents a feedback process. It helps the developers to constantly check the accuracy of their estimation and take the result into account in the upcoming estimation process. EA of each developer may vary over time. EA is ranked into three linguistic terms: Over Estimated, Well Estimated, and Under Estimated.

Table 1 illustrates these factors and the levels of each factor. The proposed model includes three components: fuzzifier, inference engine, and defuzzifier as shown in Fig. 3. For each task, the developer submits four inputs; TE, TC, TS, and EA, to the proposed fuzzy model. Each component has inputs and outputs that will be explained in the following sub-sections. At the end, the proposed model produces an Estimated Story Point (Estimated-SP) value as an output that express the effort estimation for each task. Then, the estimated-SP for all tasks of a Sprint are accumulated to produce the estimated-SP for that Sprint.

TABLE I. Effort Estimation Factors

Effort Estimation Factors	**Levels or Categories**
Development Team Experience (TE)	Junior, Intermediate, Senior
Task Complexity (TC)	Very Easy, Easy, Moderate, Complex, Very Complex
Task Size (TS)	Small, Medium, Large, X-Large
Estimation Accuracy (EA)	Over Estimated, Well Estimated, Under Estimated

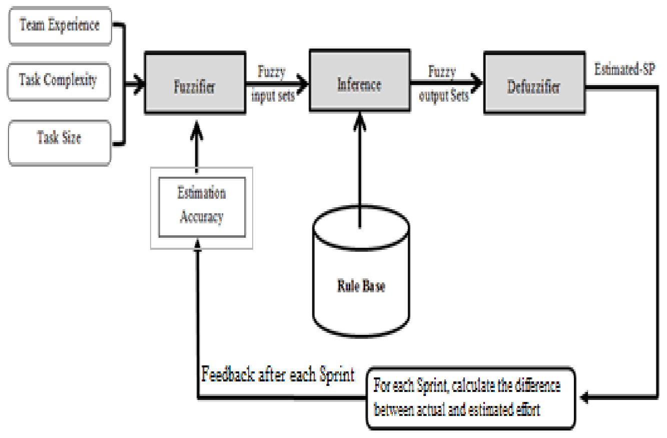


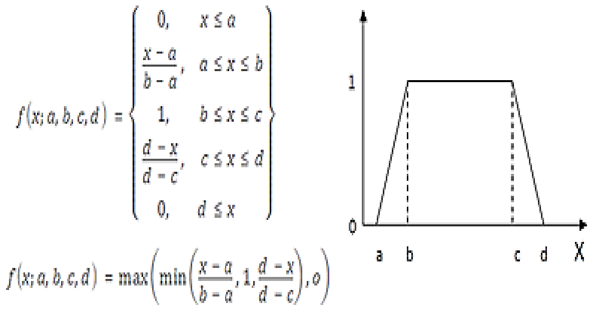
Fig. 3. Proposed fuzzy based model.

A. Fuzzifier

Fuzzifier converts the input data from each developer into a fuzzy set where each input has a membership value according to Trapezoidal Membership Function which is represented in Fig. 4. Each developer should submit the data of TE, TC, TS, and EA that are related to each task in the Sprint. Thus, all developers should participate in this process. All data are manipulated and represented by MATLAB using Trapezoidal MF.

For the first input, TE, the values and representation of levels are shown in Table 2 and Fig. 5, respectively. For each developer, the value of TE is constant for a Sprint or two Sprints because the team experience doesn't significantly change over a few weeks.

For the second input, TC, the values and representation of the levels are shown in Table 3 and Fig. 6, respectively.



$$f(x;a,b,c,d)=\begin{cases}0, & x\le a\\ \frac{x-a}{b-a}, & a\le x\le b\\ 1, & b\le x\le c\\ \frac{d-x}{d-c}, & c\le x\le d\\ 0, & d\le x\end{cases}$$

$$f(x;a,b,c,d)=\max\left(\min\left(\frac{x-a}{b-a},1,\frac{d-x}{d-c}\right),o\right)$$

Fig. 4. Trapezoidal membership function [23].

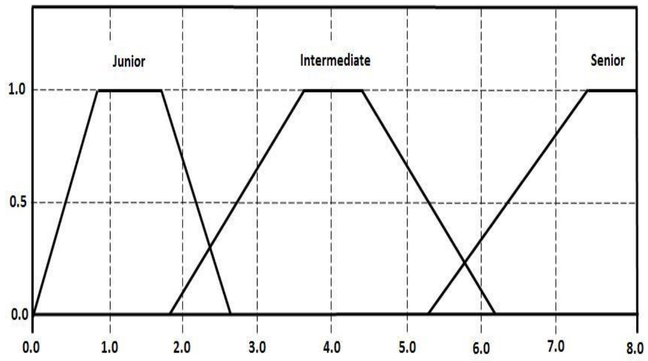


Fig. 5. Representation of TE levels.

TABLE II. DATA OF TE LEVELS USING TRAPEZOIDAL MF

TE Levels	Values
Junior	0, 0.9, 1.8, 2.7
Intermediate	1.9, 3.6, 4.4, 6.2
Senior	5.3, 7.5, 8.4, 11.7

TABLE III. DATA OF TC LEVELS USING TRAPEZOIDAL MF

TC Levels	Values
Very Easy (VE)	0.3, 1.1, 1.3, 2.2
Easy(E)	0.7, 1.7, 2.0, 2.9
Moderate (M)	1.7, 2.6, 2.9, 3.9
Complex (C)	2.8, 3.8, 4.0, 4.7
Very Complex (VC)	4.2, 4.8, 5.1, 6.0

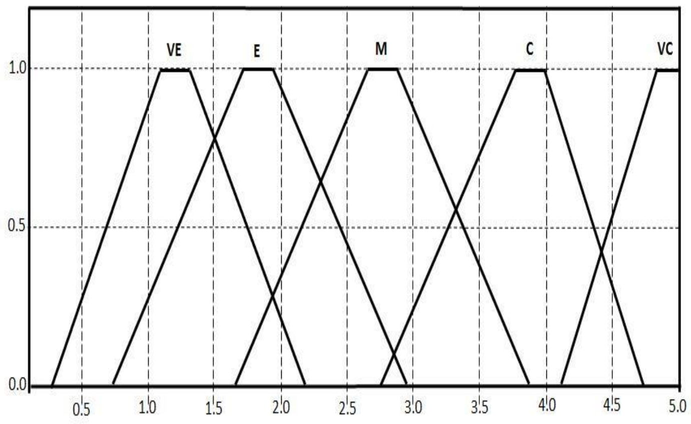


Fig. 6. Representation of TC levels.

TABLE IV. DATA OF TS LEVELS USING TRAPEZOIDAL MF

TS Levels	Values
Small (S)	0.12, 0.2, 0.33, 0.44
Medium (M)	0.21, 0.38, 0.44, 0.6
Large (L)	0.46, 0.6, 0.73, 0.85
X-Large (XL)	0.7, 0.88, 0.99, 1.18

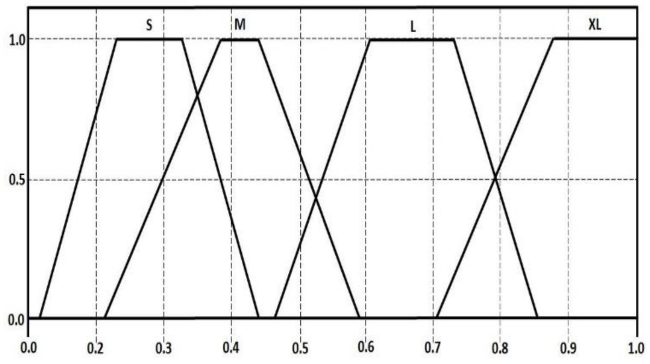


Fig. 7. Representation of TS levels.

TABLE V. DATA OF EA LEVELS USING TRAPEZOIDAL MF

EA Levels	Values
Under Estimated	-1.8, -1, -0.8, -0.17
Well Estimated	-0.36, -0.07, 0.02, 0.36
Over Estimated	0.22, 0.78, 1.2, 1.99

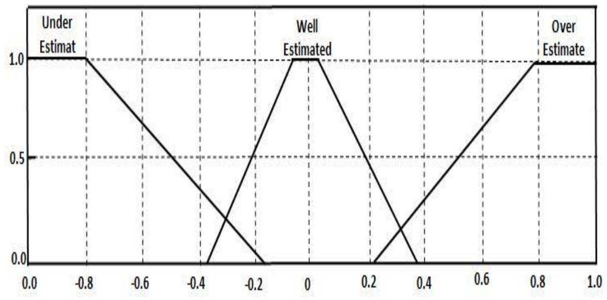


Fig. 8. Representation of EA levels.

For the third input, TS, the values and representation of the levels are shown in Table 4 and Fig. 7, respectively.

For the fourth input, EA, the values and representation of the levels are shown in Table 5 and Fig. 8, respectively. For each developer, the value of EA is constant for all tasks related to a Sprint and it changes from a Sprint to the next one. For facilitating the work of the proposed model, the value of EA is supposed to be "Well Estimated" for all developers in the first Sprint. At the end of the Sprint, the value of EA is produced, as we will explain in Section V, and it will be considered as input to the next Sprint.

B. Inference Engine

A fuzzy inference engine is a collection of fuzzy conditional statements, IF–THEN rules, stored in fuzzy rule base and they are used to make a decision. The fuzzy inference engine combines fuzzy IF–THEN rules into a mapping from fuzzy sets in the input space X to fuzzy sets in the output space Y based on fuzzy logic principles [34].

The Estimated-SP variable describes the output, effort estimation, from the proposed model which can be expressed in a Fibonacci series (0, 1, 1, 2, 3, 5, 8…). The Estimated-SP variable is ranked into four levels; Easy, intermediate, complex, very complex. In order to implement Estimated-SP on the proposed model, the researcher will rescale these levels using a trapezoidal membership function as shown in Table 6. Fig. 9 shows the consequent fuzzy sets parts which are derived using the definition of the Estimated-SP variable.

TABLE VI. ESTIMATED-SP LEVELS USING TRAPEZOIDAL MF

SP Levels	Values
Easy	2, 10, 12, 20
Intermediate	12, 22, 26, 36
Complex	25, 34.8, 40, 49
Very complex	41, 52, 60, 74

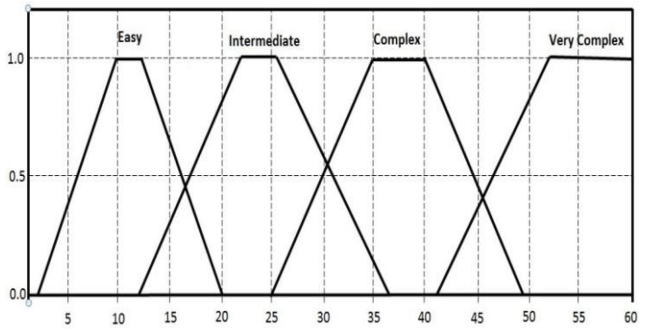


Fig. 9. Representation of estimated-SP levels.

Using the previous inputs and outputs, the researcher designs and builds fuzzy rules-base which includes a set of IF-THEN rules. The recommendations of the Scrum Master are the basic directive that guides the formation of these rules. A sample of the resultant rules is:

1. IF (EA is *Well Estimated*) and (TE is *Intermediate*) and (TC is *M*) and (TS is *XL*) THEN (Estimated-SP is *Intermediate*)
2. IF (EA is *Under Estimated*) and (TE is *Senior*) and (TC is *H*) and (TS is *M*) THEN (Estimated-SP is *Easy*)
3. IF (EA is *Over Estimated*) and (TE is *not Senior*) and (TC is *VC*) and (TS is *X*) THEN (Estimated-SP is *Complex*)
4. IF *(*EA is *Well Estimated)* and (TC is *VE*) and (TS is *S*) THEN (SP is *Easy*)
5. IF (EA is *Well Estimated*) and (TS is X) and (TE is *Intermediate*) and (TC is C) THEN (Estimated-SP is *Intermediate*)
6. IF (EA is *Well Estimated*) and (TE is not *Junior*) and (TC is *VE*) THEN (Estimated-SP is *Easy*)
7. IF (EA is *Under Estimated*) and (TE is *Senior) and (TC is VC) THEN (*Estimated-SP *is Complex)*
8. IF (EA is *Over Estimated*) and (TE is *Senior*) and *(TC is VC)* THEN (Estimated-SP is *Very Complex*)
9. IF (EA is *Under Estimated*) and (TC is *E*) THEN (SP is *Easy*)
10. IF (EA is *Under Estimated*) and (TC is *VC*) and (TS is *XL*) THEN (Estimated-SP is *Very Complex*)
11. IF (EA is *Over Estimated*) and (TS is XL) and (TC is VC) THEN (Estimated-SP is *Very Complex*)
12. IF (TS is XL) and (TE is *Junior*) and (TC is *M*) THEN (Estimated-SP is *Intermediate*)
13. IF (EA is *Under Estimated*) and (TE is *Intermediate*) THEN (Estimated-SP is Complex)
14. IF (EA is *Well Estimated*) and (TC is *M*) THEN (Estimated-SP is *Intermediate*)
15. IF (EA is *Well Estimated*) and (TE is *H*) and (*TC is VC) THEN (*Estimated-SP *is Complex*)

The proposed fuzzy inference system is a Mamdani-type, in which, both inputs and outputs are presented as fuzzy sets, therefore this system is very easy to interpret [35]. For each task, each developer has its own input vector and the inference system evaluates the rules and generates corresponding a fuzzy value and its membership value for each rule and in turn generates many values for each developer. This process repeated for all tasks and results are ready to go to defuzzification component.

C. Defuzzifier

The fuzzy results cannot be used as such to make a decision, hence it is necessary to use the defuzzifier that convert the fuzzy quantities into crisp quantities for further processing [34]. The most common method of defuzzification is a Centroid Method or it is called Center of Area (COA), as shown in (1).

$$Center\ of\ Area\ (COA) = \frac{\sum_{i=1}^{n} x_i \cdot \mu_{A_i}(x_i)}{\sum_{i=1}^{n} \mu_{A_i}(x_i)} \qquad (1)$$

Where; $\mu_{A_i}(x_i)$ is the membership function of the fuzzy set A_i associated with the input x_i

For each task, after applying COA on the values of each developer, a simple average is calculated for the values of all developers. Thus, an estimated value, Estimated-SP, is resulted for this task. Then, the effort for each Sprint can be calculated. Using the result data of defuzzification, it is easy to make a comparison between the actual and estimated effort for each developer, and then calculate the difference between them. The feedback helps to evaluate and improve the developer estimation accuracy over time. Therefore, the proposed model will be stable after few sprints. At the first sprint, the model will assume that all developers have a good accuracy. At the second sprint, the actual accuracy is available to be entered into the estimation process. This process will be repeated until reaching the last sprint in the project. At the start of the next project, a set of trained values will be available to be used in the estimation process.

V. EXPERIMENT AND RESULTS

In this section, a dataset of three Sprints related to a living Scrum project is used to present how the proposed model works. Each sprint consists of ten tasks, as shown in Table 7. The development team includes five developers with different level of experiences in Scrum development projects. There are many methods for assessing and comparing effort estimation models. The most common evaluation methods include; the Magnitude of Relative Error (MRE) and Prediction Level (Pred) metrics. MRE value is calculated for each observation i whose effort is predicted, as shown in (2) [36]. In MRE, Actual Effort is represented by Actual Story Points (Actual-SP) and Predicted Effort is represented by Estimated-SP. The aggregation of MRE over multiple observations (N) can be achieved through the Mean MRE (MMRE) as shown in (3). Another widely used measure is the Pred(i) which is defined in (4) [37]. Pred(i), sometimes is written Pred(MMRE), is the percentage of the number of tasks whose MRE is less than or equal to MMRE.

$$MRE_i = \frac{|Actual\ Effort_i - Predicted\ Effort_i|}{Actual\ Effort_i} \quad (2)$$

$$MMRE = \frac{1}{N}\sum_1^N MRE_i \quad (3)$$

$$Pred\ (I) = \frac{K}{N} * 100 \quad (4)$$

Where, N: is the total number of tasks, and k is the number of tasks whose MRE is less than or equal to (I), and I is the MMRE for each sprint.

TABLE VII. EXPERIMENT RESULTS FOR SPRINTS

Task. No	**Sprint 1**			**Sprint 2**			**Sprint 3**		
	Estimated-SP	**Actual-SP**	**MRE**	**Estimated-SP**	**Actual-SP**	**MRE**	**Estimated-SP**	**Actual-SP**	**MRE**
1	14.40	21.22	0.32	28.56	19.00	0.50	25.40	22.31	0.14
2	36.60	43.12	0.15	32.56	32.00	0.02	31.40	26.29	0.19
3	18.40	12.34	0.49	45.16	39.00	0.16	28.60	27.32	0.05
4	26.80	24.00	0.12	20.96	21.32	0.02	23.00	25.31	0.09
5	30.20	36.31	0.17	25.56	26.03	0.02	30.40	32.16	0.05
6	35.20	29.54	0.19	28.76	20.00	0.44	27.40	29.15	0.06
7	20.60	25.38	0.19	39.56	38.52	0.03	21.80	19.17	0.14
8	28.60	21.39	0.34	37.36	31.00	0.21	29.60	27.70	0.07
9	18.80	14.13	0.33	31.16	28.45	0.10	28.60	29.21	0.02
10	27.60	18.54	0.49	38.16	37.11	0.03	31.40	33.71	0.07
MMRE			**0.28**			**0.15**			**0.09**
PRED (MMRE)			**50%**			**60%**			**60%**

In Table 7, while the value of MMRE is decreased from Sprint to another, the value of Pred(MMRE) is increased; due to the improvement of estimation accuracy of developers over Sprints. If the proposed model is applied on more Sprints, the evolution of MMRE and Pred(MMRE) will be more clear.

Using the proposed model, the estimation accuracy of each developer can be evaluated after achieving each Sprint by calculating the difference between the actual-SP of the Sprint and the accumulative value of the developer's estimations of the tasks of the same Sprint which represents his Estimated-SP for this Sprint. The result may be: zero, negative value, or positive value. As mentioned in Section 4, EA is characterized by: Over Estimated, Well Estimated, or Under Estimated, according to the levels in Table 5 and Fig. 8. The resultant difference shows EA of the developer that should be entered into the proposed model when starting the estimation process of the next Sprint.

VI. CONCLUSION AND FUTURE WORK

This paper aimed to propose a fuzzy based model for effort estimation in Scrum based projects. Therefore, the researchers studied many researches in the domain of Scrum framework, fuzzy logic, and effort estimation. Scrum framework is one of the most popular agile methods, in which the team accomplishes their work in software projects by utilizing the improved communication and collaboration among the members and breaking down the work into a series of sprints. In Scrum, there are many factors that have a significant influence on the effort estimation of each task in a sprint. These factors are: Development Team Experience (TE), Task Complexity (TC), Task Size (TS), and Estimation Accuracy (EA). These factors are usually presented using linguistic quantifiers. Therefore, this paper aimed to utilize the fuzzy logic concepts to build a fuzzy based model that can improve the effort estimation in Scrum framework.

The proposed model includes three components: fuzzifier, inference engine, and defuzzifier. The researcher designed the proposed model using MATLAB. The application of the proposed model on three Sprints of a real software development project is used to present how the proposed model works and to show how it becomes more accurate over time and gives a better effort estimation. In addition, the estimation accuracy of each developer can be calculated and entered as an input to the next Sprint.

In the domain of using fuzzy logic in effort estimation, there are many issues that can be tackled in the future:

- Applying the proposed model on many real Scrum projects.
- Expanding the proposed model to deal with effort estimation process in eXtreme Programming (XP) approach.
- Integrating the proposed model with other estimation techniques; such as COCOMO.
- Studying how to utilize other soft computing techniques to enhance the effort estimation process; such as neural network or genetic algorithms.

REFERENCES

[1] Yang Yong and Bosheng Zhou, "Evaluating Extreme Programming Effect through System Dynamics Modeling", International Conference on Computational Intelligence and Software Engineering (CiSE), Wuhan, China, Dec. 2009.

[2] Giulio Concas, Marco Di Francesco, Michele Marchesi, Roberta Quaresima, and Sandro Pinna, "An Agile Development Process and Its Assessment Using Quantitative Object-Oriented Metrics", 9th International Conference, XP 2008, Limerick, Ireland, Proceedings, June 2008.

[3] A. Qumer and B. Henderson-Sellers, "A framework to support the evaluation, adoption and improvement of agile methods in practice", Journal of Systems and Software Technology Vol. 81, pages 1899–1919, 2008.

[4] Dean Liffingwell, "Scaling Software Agility – Best Practices for Large Enterprises", The Agile Software Development Series, Pearson Education Inc., 2007.

[5] Almseidin, M., Alrfou, K., Alnidami N., and Tarawneh A., "A Comparative Study of Agile Methods: XP versus SCRUM", International Journal of Computer Science and Software Engineering, Vol. 4, No. 5, pages 126-129, 2015.

[6] Jasem M. Alostad, Lamya Sulaiman AlAli, and Laila R. A. Abdullah, “Hybrid Agile Approach for Achieving Higher Quality in Software Development Process”, International Journal of Computer Science and Information Security (IJCSIS), Vol. 15, No. 5, May 2017.

[7] Kenneth S. Rubin, “Essential Scrum: A Practical Guide to the Most Popular Agile Process”, Addison-Wesley, Pearson Education, Inc., 2013.

[8] Cho, J., “Issues and Challenges of agile software development with SCRUM”, Issues in Information Systems, Vol. 9, No. 2, pages 188-195, 2008.

[9] M. Salman Bashir and M. Rizwan Jameel Qureshi, “Hybrid Software Development Approach for Small to Medium Scale Projects: RUP, XP & Scrum”, Sci. Int., Lahore, 24 (4), 2012.

[10] Ken Schwaber and Jeff Sutherland, “The Scrum Guide - The Definitive Guide to Scrum: The Rules of the Game”, Scrum.Org, October 2011.

[11] https://idimension.wordpress.com/2011/05/03/scrum-framework-scrum-roles/ Last visit: 12/6/2017.

[12] Soumyadipta Paul and K. John Singh, “Be Agile: Project Development with Scrum Framework”, Journal of Theoretical and Applied Information Technology, Vol. 40 No.1, 15 June 2012.

[13] https://www.scrumalliance.org/why-scrum Last visit: 12/6/2017.

[14] Ratnesh Litoriya and Abhay Kothari “An Efficient Approach for Agile Web Based Project Estimation: AgileMOW”, Journal of Software Engineering and Applications, VOL.6, 2013.

[15] Ashita Malik, Varun Pandey, Anupama Kaushik, “An Analysis of Fuzzy Approaches for COCOMO II”, International Journal of Intelligent Systems and Applications, Vol.5, 2013.

[16] Komal Garg, Paramjeet Kaur, Shalini Kapoor, and Shilpa Narula, “Enhancement in COCOMO Model Using Function Point Analysis to Increase Effort Estimation”, IJCSMC, Vol. 3, Issue. 6, 2014.

[17] Ziauddin Zia,Shahid Kamal Tipu, and Shahrukh Zia “An Effort Estimation Model for Agile Software Development”, Advances in Computer Science and its Applications (ACSA), Vol.2, 2012.

[18] Anupama Kaushik,A.K. Soni, and Rachna Soni, “A Type-2 Fuzzy Logic Based Framework for Function Points”,

[19] Abeer Hamdy,”Fuzzy Logic for Enhancing the Sensitivity of COCOMO Cost Model”, Journal of Emerging Trends in Computing and Information Sciences, Vol.3, No.9, 2012.

[20] Zadeh, L. A., “Fuzzy logic = computing with words”, IEEE Transactions on Fuzzy Systems, Vol. 4 No. 2, pages 103-111. doi:10.1109/91.493904, 1996.

[21] Zadeh, L. A., “Is there a need for fuzzy logic?”, Information Sciences, Vol. 178, No. 13, 27512779. doi:10.1016/j.ins.2008.02.012, 2008.

[22] https://www.mathworks.com/help/fuzzy/trimf.html Last visit: 15/6/2017.

[23] https://www.mathworks.com/help/fuzzy/tramf.html Last visit: 15/6/2017.

[24] https://www.mathworks.com/help/fuzzy/pimf.html Last visit: 15/6/2017.

[25] Runkler, T. A., “Selection of appropriate defuzzification methods using application specific properties”, IEEE Transactions on Fuzzy Systems, 5(1), 72-79. doi:10.1109/91.554449, 1997.

[26] Colomo-Palacios, R., Gonzalez-Carrasco, I., Lopez-Cuadrado, J. L., & Garcia-Crespo, A., “ReSySTER: A hybrid recommender system for scrum team roles based on fuzzy and rough sets”, International Journal of Applied Mathematics and Computer Science, 22(4), 801-816. doi:10.2478/v10006-012-0059-9, 2012.

[27] Vishal Sharma and Harsh Kumar Verma, “Optimized Fuzzy Logic Based Framework for Effort Estimation in Software Development”, International Journal of Computer Science Issues (IJCSI), Vol. 7, Issue 2, No 2, 2010.

[28] Ziauddin, Shahid Kamal, Shafiullah khan and Jamal Abdul Nasir, “A Fuzzy Logic Based Software Cost Estimation Model”, International Journal of Software Engineering and Its Applications (IJSEIA), Vol. 7, Issue 2, 2013.

[29] Sedehi, H., and Martano, G., “Metrics to Evaluate & Monitor Agile Based Software Development Projects-A Fuzzy Logic Approach”, Seventh International Conference on Software Process and Product Measurement, 2012 Joint Conference of the 22nd International Workshop on, 99-105. IEEE, 2012.

[30] Mohammed, A. H., & Darwish, N. R. (2016). A Proposed Fuzzy based Framework for Calculating Success Metrics of Agile Software Projects. International Journal of Computer Applications. 137(8), 17-23. doi:10.5120/ijca2016908866 .

[31] Abeer Hamdy, “Fuzzy Logic for Enhancing the Sensitivity of COCOMO Cost Model”, Journal of Emerging Trends in Computing and Information Sciences, Vol.3, No.9, 2012.

[32] Prasad Reddy P.V.G.D., Sudha K. R and Rama Sree P, “Application of Fuzzy Logic Approach to Software Effort Estimation” International Journal of Advanced Computer Science and Applications(IJACSA), 2(5), 2011.

[33] Jutta Eckstein, “Architecture in Large Scale Agile Development”, Springer, International Publishing Switzerland, pp. 21–29, 2014.

[34] S.N. Sivanandam , S. N. Deepa, and , S. Sumathi, “Introduction to Fuzzy Logic using MATLAB”, Springer,2007.

[35] F. Martin McNeill and Ellen Thro, “Fuzzy Logic A Practical Approach”, AP professional ,1994.

[36] Iman Attarzadeh and Siew Hock Ow, “A Novel Algorithmic Cost Estimation Model Based on Soft Computing Technique”, Journal of Computer Science, Vol. (6), No. (2): 117-125, 2010.

[37] Abeer Hamdy, “Genetic Fuzzy System for Enhancing Software Estimation Models”, International Journal of Modeling and Optimization, Vol. 4, No. 3, June 2014.