

# Design Tool to Estimate Effort using Use Case based on Fuzzy Logic and Soft Computing Techniques

Laheeb M. Alzubaidy  
Associate Professor  
Dept. of Software Engineering  
University of Mousel, Iraq

Anfal A. Fadhil  
Assistance Lecture  
Dept. of Software Engineering  
University of Mousel, Iraq

## ABSTRACT

Use case models are used in object-oriented analysis for capturing and describing the functional requirements of a system. Several methods for estimating software development effort are based on attributes of a use case model. This paper designed tool in c# language to estimate effort from XML for use case diagram based on fuzzy logic method in environment factor, and applying these tool on three project ,experimental results support existing claims that use cases can be used successfully in estimating software development effort. The results indicate that the guidance provided by the use case points method can support expert knowledge in the estimation process. ,experimental results show also that the design of the use case models has a strong impact on the estimates

## Keywords

Use Cases, Actor, Diagram, Estimation, Fuzzy logic

## 1. INTRODUCTION

Estimate software project cost and project effort to get an idea of the required amount of work to be done is very important and the related amount to be spent on that particular work during the course of work of software product[1] , Predicting software cost and effort with good accuracy has been a challenge for many project managers and researchers. Several cost estimation techniques have been used for software effort and cost prediction. These techniques include algorithmic models, expert judgment, estimation by analogy and machine learning techniques. And many metrics software size is the most important factor.

The Source Lines of Code (SLOC) is one of the oldest size metrics and has been widely used by models such as COConstructive COst MOdel (COCOMO) [2],[3]. The SLOC metric has been criticized because it cannot be used early and it depends on the programming language and technology used to develop the project, introduced the function points metric to tackle the limitations of the SLOC metric; however, counting function points is sometimes subjective and complicated[4].

Another available size metric is the use case points (UCP) which was proposed by G. Karner [5]. The UCP metric is computed based on the number and complexity of the use cases as well as actors in a use case diagram. Use case models are increasingly being used to capture and describe the functional requirements of a software system. There are different approaches and methods to successfully estimate effort using use cases[ 4].

A few researchers have tested the use case points method and analyzed their findings. The results, though not conclusive, indicate that the use case points method has potential to be a reliable source of estimation, much like the function point method, and it can have a strong impact on estimating the size

of software development projects, especially when it is used along with expert estimates. Also, since use case modeling is increasingly being utilized as the method of choice to describe the software and system requirements and as a basis of design, development, testing, deployment, configuration management and maintenance, it makes sense to have an estimation method that makes use of them[6]

Artificial intelligence (AI) has been widely used in many domains such as medicine, economics, business, and others. Recently, the interest of applying AI in software engineering has been soaring. Conducting software estimation is essential in any project as it helps project managers plan, identify risks, and determine the effort and cost of projects, the machine learning techniques are being used in conjunction or as alternatives to algorithmic models. These techniques include neural networks, fuzzy logic, neuro-fuzzy, genetic algorithm and regression trees[7]

In this work, a fuzzy use case is used to estimate effort because this method more suitable for projects with indistinct and imprecise information.

Fuzzy logic is based on the theory of fuzzy sets and uses the whole interval of real numbers between zero (False) and one (True) to develop a logic as a basis for rules of inference[8].The advantage of fuzzy logic is that it interprets the linguistic values very much similar to the human way of interpretation and the result of estimation effort in this method is good accuracy.

This paper is organized as follows: Section II lists related work, Section III describe fuzzy logic proposed model, Section IV calculate use case point with fuzzy logic ,Section V explain the experimental results and Section VI conclusion and future work.

## 2. RELATED WORK

Many research have worked on predicting software cost and efforts using use case ,some of them proposed tools ,in the following section a brief explanation about their works:

Karner, G in 1993 proposed a method to assigns quantitative weights to actors and weights to Use Cases based on classification as simple, average and complex. In this research the sum of all the weighted actors and the sum of all the weighted Use Cases multiplied a number of technical complexity factors multiplied by the Productivity Facto gives the estimating effort. This method Relatively good accuracy and promising results. More accurate than expert estimates but does not take into account any additional artifacts[9] .

Caroll, E.R in 2005 design a Industrial Use Case Points, the main contribution of this method is the inclusion of the risk factor and additional effort for activities other than the

development of the software product, but add risk factor must be caution and accuracy. [10].

Diev, S. in 2006 design a slight modification of the Use Case Points Method (UCPM), this method stresses more on defining actors and Use Cases comprehensively finally, supplementary effort factor is added to yield the final effort estimate of the software product .In this method effort factor must be determines suspicion [11] .

In 2006, Braz, M.R. use the metric focuses on the internal structures of the Use Cases in depth and hence better captures the functionality. The primary factors considered in this metric are the actors classification, pre-condition classification and post-condition classification, main scenarios, alternate scenarios, exception classification and the adjustment Factor. The sum of all these factors multiplied by the difference of the technical complexity factor and the experience factor [12], if classify of actors not correct the effort not accuracy.

Robiolo, G., et al in 2009, use technique depends on the textual description of a Use Case. A transaction is defined by a stimulus by the actor and response by the system. The sum of all the stimuli is the number of transactions in a particular Use Case. Summing up the transactions for all the use cases in the entire system, the number of transactions is calculated. In order to estimate the final effort but this method deals with the functional requirements only this method depend on textual description and when the description is inaccurate the estimation do not give good results [13].

Wang, F., et al in 2009 proposed method contains three parts; first, refining the Use Case classification with fuzzy set theory. Second, using a learning Bayesian Belief Network (BBN) for getting the Unadjusted Use Case Points (UUCP) probability distribution. Third, using a BBN for generating the effort probability distribution which is derived from UCP. But this method be lengthy because it used fuzzy logic theory with neural network [14].

Ochodek, M., et al., in 2010 explain a method to simplify the use case point method and the process of Effort Estimation in general without certain factors. Metric based on rejection of actor weights and rejection of 9 technical complexity factors and 6 experience factors .but this method does not technical complexity factors and experience factors [15].

Bou Nassifl A,et al, in 2011, proposed effort equation takes into consideration the non-linear relationship between software size and software effort using fuzzy productivity factors ,but this method very complexity [16].

Tahira Mahboob ,et in 2017 proposed determine what factors regarding a project are directly related to the effort estimation. Other than that a model is proposed to predict the effort using minimum number of parameters in software project development[1]

### 3. FUZZY LOGIC

The fuzzy logic model uses the fuzzy logic concepts introduced by Lofti A. Zadeh [17]. Fuzzy logic is a valuable tool, which can be used to solve highly complex problems where a mathematical model is too difficult or impossible to create and enables computers to make decisions. It is also used to reduce the complexity of existing solutions as well as increase the accessibility of control theory [18] therefor fuzzy logic have many application in software engineering . Fuzzy reasoning consists of three main components : fuzzification process, inference from fuzzy rules and defuzzification process. Fuzzification process is where the objective term is transformed

into a fuzzy concept. The membership functions are applied to the actual values of variables to determine the confidence factor or membership value (MV)[19]

In fuzzy logic, a membership function provides a measure of the degree of similarity of an element to a fuzzy set. A membership function represents the degree of truth as an extension of valuation. The membershipfunction fully defines the fuzzy set. Study showed that fuzzy logic model has a place in software effort estimation that use fuzzification technical factor using Trapezoidal membership Fuzzification allows the input and output to be expressed in linguistic terms. Inferencing involves defuzzification of the conditions of the rules and propagation of the confidence factors of the conditions to the conclusion of the rules. A number of rules will be fired and the inference engine assigned the particular outcome with the maximum MV from all the fired rules. Defuzzification process refers to the translation of fuzzy output into objective terms.[20],[18]

### 4. USE CASE MODELING

Use case modeling is an accepted and widespread technique to capture the business processes and requirements of a software application project. Since use cases provide the functional scope of the project, analyzing their contents provides valuable insight into the effort and size needed to design and implement a project. In general, projects with large, complicated use cases take more effort to design and implement than small projects with less complicated use cases. Moreover, the time to complete the project is affected by the following[21]:

- The number of steps to complete the use case.
- The number and complexity of the actors.
- The technical requirements of the use case such as concurrency, security, and performance.
- Various environmental factors such as the development teams' experience and knowledge.

An estimation method that took into account the above factors early in a project's life cycle, to project scheduling, cost, and resource allocation. The Use Case Points (UCP) method. provides the ability to estimate the man hours a software project requires from its use cases. ased on work by Gustav. Karner [22].

The UCP method analyzes the use case actors, and various technical and environmental factors and abstracts them into an equation. Readers familiar with Allan Albrecht's "Function .Point Analysis" [21],[23] will recognize its influence on UCP, The UCP equation is composed of three variables:

1. Unadjusted Use Case Points (UUCP).
2. The Technical Complexity Factor(TCF).
3. The Environment Complexity Factor(ECF).

Each variable is defined and computed separately using weighted values, subjective values, and constraining constants. additionally, when productivity is included as a coefficient that expresses time, the equation can be used to estimate the number of man-hours needed to complete a project. Here is the complete equation with a Productivity Factor (PF) included:

$$UCP = UUCP \times TCF \times ECF \times PF \quad \dots (1)$$

The necessary steps to generate the estimate based on the UCP method are the following:

1. Determine and compute the UUCP.

2. Determine and compute the TCF.
3. Determine and compute the ECF.
4. Determine the PF.
5. Compute the estimated number of hours [23].

#### 4.1 Unadjusted Use Case Points ( UUCPS)

UUCP's are computed based on two computations [21]:

1. **The Unadjusted Use Case Weight (UUCW)** based on the total number of activities (or steps) contained in all the use case scenarios.
2. **The Unadjusted Actor Weight (UAW)** based on the combined complexity of all the actors in all the use cases.

##### 4.1.1 The Unadjusted Use Case Weight (UUCW)

The UUCW is derived from the number of use cases in three categories: simple, average, and complex ,see Table 1[21]. Each use case is categorized by the number of steps its scenario contains, including alternative flows.

**Table 1: Use Case Categories**

Use Case Category	Description	Weight
Simple	Simple user interface. Touches only a single database entity. Its success scenario has three steps or less. Its implementation involves less than five classes	5
Average	More interface design. Touches two or more database entities. Between four and seven steps. Its implementation involves between five and 10 classes	10
Complex	Complex user interface or processing. Touches three or more database entities. More than seven steps. Its implementation involves more than 10 classes	15

The UUCW is calculated by tallying the number of use cases in each category, multiplying each total by its specified weighting factor, and then adding the products

##### 4.1.2 THE UNADJUSTED ACTOR WEIGHT (UAW)

In a similar manner, the actor types are classified as simple, average, or complex as shown in Table 2[21]. The UAW is calculated by totaling the number of actors in each category, multiplying each total by its specified weighting actor, and then adding the products. The UUCP is computed by adding the UUCW and the UAW

**Table 2: Actor Classifications**

actor Category	Description	Weight
Simple	The actor represents another system with a defined application programming interface.	1
Average	The actor represents another system interacting through a protocol, like Transmission Control Protocol/Internet Protocol	2
Complex	The actor is a person interacting via a graphical user interface	3

#### 4.2 TECHNICCAL FACTOR (TCFs)

Thirteen standard technical factors exist to estimate the impact on productivity that various technical issues have on a project see Table 3 , Each factor is assigned a value between 0 and 5 depending on its assumed influence on the project. A rating of 0 means the factor is irrelevant for this project; 5 means it is essential. The Technical Factor (TCF) is calculated by multiplying the value of each factor (T1 –T13) in Table 1 by its weight and then adding all these numbers to get the sum called the TFactor. Finally, the following formula is applied[24]:

$$TCF = 0.6 + 0.1 \times TFactor \quad \dots (2)$$

$$TCF = 0.6 + 0.1 \times Technical TotalFactor \quad \dots (3)$$

**Table 3: Technical Factor[24]**

Technical Factor	Description	weight
T1	Distributed System	2
T2	Performance	2
T3	End User Efficiency	1
T4	Complex Internal Processing	1
T5	Reusability	1
T6	Easy to Install	0.5
T7	Easy to Use	0.5
T8	Portability	2
T9	Easy to Change	1
T10	Concurrency	1
T11	Special Security Features	1
T12	Provides Direct Access for Third Parties	1
T13	Special User Training Facilities are Required	1

#### 4.3 ENVIRONMENTAL FACTOR( ECFs)

The ECF (see Table 4) provides a concession for the development team's experience. More experienced teams will have a greater impact on the UCP computation than less experienced teams. The development team determines each factor's perceived impact based on its perception the factor has on the project's success. A value of 1 means the factor has a strong, 2 negative impact for the project, 3 is average, and 5 means it has a strong, positive impact. A value of zero has no impact on the project's success Each factor's weight is multiplied by its perceived impact to produce its calculated factor. The calculated factors are summed to produce the Environmental Total Factor[7]

**Table 4: Environmental Factor**

Environmental Factor	Description
E1	Familiarity With UML
E2	Part-Time Workers
E3	Analyst Capability
E4	Application Experience

E5	Object-Oriented Experience
E6	Motivation
E7	Difficult Programming Language
E8	Stable Requirements

#### 4.4 FUZZY ENVIRONMENTAL FACTOR (FEFCs)

In this paper use fuzzification Environmental Factor in table 4 using Trapezoidal membership (see Figure 1).They are described by four parameters with the four breakpoints of the trapezium[25].

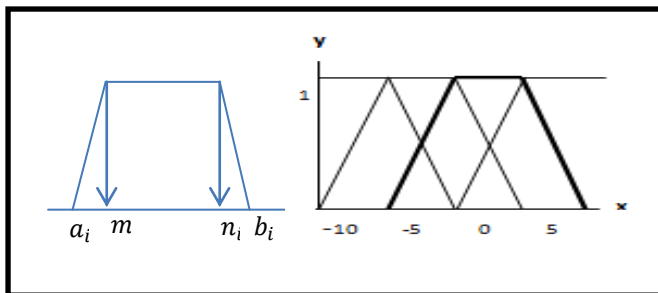


Fig.1 : Trapezoidal membership

Where:

$m_i$ =lower value of linguistic

$$n_i = (m_i + m(i - 1))/2$$

$$a_i = n(i - 1)$$

$$b_i = m(i - 1)$$

the summation Environmental Factor multiplying weight for each factor if less than zero result equal 0.4,if between 1 and 10 the result equal 0.7,if between 11 and 20 the result equal 1,finally if greater than 20 the result equal 1.3 see table 5[7]

Table 5: Fuzzy Environmental Factor

$\sum_{i=0}^a f_i \times w_i$	Productivity Description	Productivity Factor
Less than 0	Very Low	0.4
Between 1 and 10	Low	0.7
Between 11 and 20	average	1
Greater than 20	High	1.3

Calculating the Use Case Point (UCP) As a reminder, the UCP equation is[21]:

$$UCP = UUCP \times TCF \times FEFC \quad \dots (4)$$

Finally determined productivity ,That is the ratio of development man-hours needed per use case point. Statistics from past projects provide the data to estimate productivity[20] . and multiply UCP to calculate effort

$$Effort = Size(UCP) \times productivity \quad \dots (5)$$

Figure 2 explain flow chart of the proposed work ,and (see Figure 3 explain a tool in c# language that show how can any user estimate effort from use case diagram

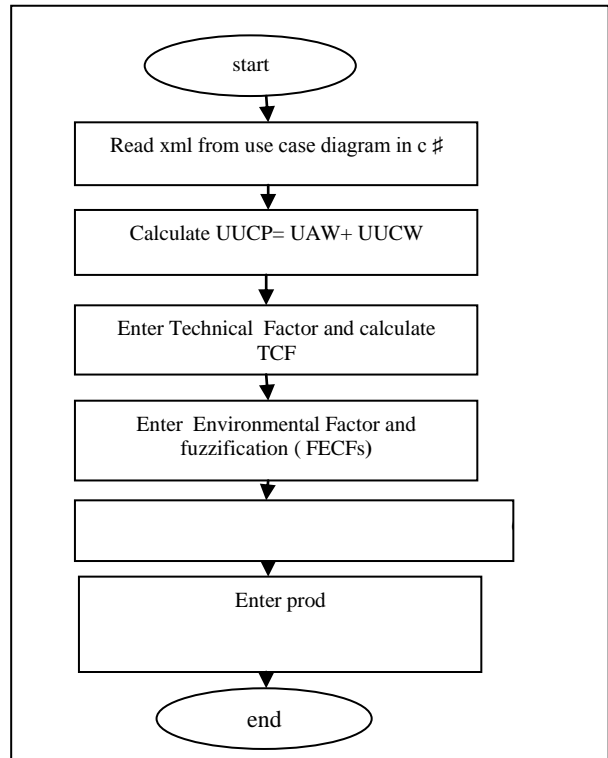


Fig 2: Flow chart of compute effort using use case and fuzzy logic

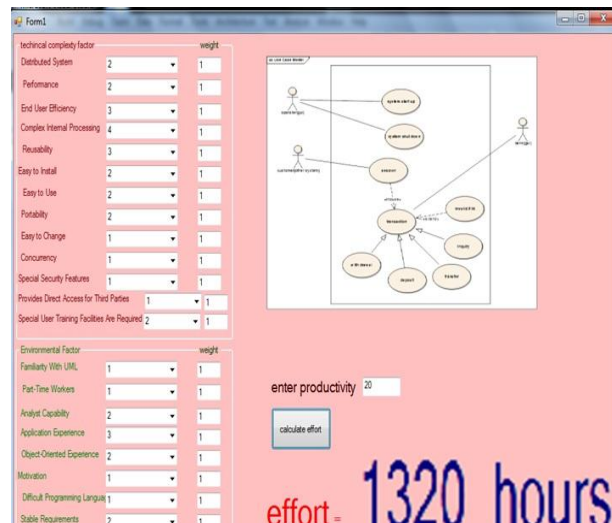


Fig 3: Tool to estimate effort using use case diagram

## 5. EXPERIMENTAL RESULTS

When comparing estimates based on use case points for three development projects(finance ,customer relationship management within banking, Banking support for sale of credit cards) [26] with estimates obtained by experts, and compare result from actual effort , in this case senior members of the development projects ,and actual effort. As that experienced that the guidance provided by the use case points method appears to reduce the need for expert knowledge in the estimation process. The results are shown in Table (6), the use case estimates are fairly close to the estimates produced by the experts

**Table 6. Expert estimate, use case estimate and effort (in hours)**

Project	Expert estimate	Use case estimate	Actual effort
finance	2730	2600	3670
customer relationship management within banking	2340	2170	2860
Banking support for sale of credit cards	2100	2114	2740

## 6. CONCLUSION

When use fuzzification Environmental Factor after The use case point model tending to early software effort estimations and given result near from Experts. The main advantage of this model is that it is simple and can be easily automated.

Experimental results show also that the design of the use case models has a strong impact on the estimates

Future work will focus on two main concerns. First, the proposed model should be tested with projects of larger sizes (greater than 5,000 person-hours) when data are available. Secondly, the Environmental factors proposed in Table 5 should be calibrated using other technique of artificial intelligence.

## 7. REFERENCES

- [1] M. Tahira , S. Bushra ,2017, Predictive Approach towards Software Effort Estimation using Evolutionary Support Vector Machine, IJACSA Vol. 8, No.
- [2] B. W. Boehm, 1981, Software Engineering Economics. Prentice-Hall.
- [3] L. H. Putnam, 1978, A General Empirical Solution to the Macro Software Sizing and Estimating Problem, IEEE Transactions on Software Engineering, vol. 4, pp. 345-361.
- [4] N. B. Ali ,C. F. Luiz, , 2012 ,A Treeboost Model for Software Effort Estimation Based on Use Case Points, Electrical and Computer Engineering Publications.
- [5] G. Karner, 1993, Resource Estimation for Objectory Projects, Objective Systems.
- [6] D.Mel ,N.Aqua, ESTIMATION USING USE CASE POINTS Computer Science Program, University of Houston-Victoria, Texas
- [7] N.B.Ali, H. Danny ,2011, Estimating Software Effort Based on Use Case Point Model Using Sugeno Fuzzy Inference System, IEEE.
- [8] S. Fukami, M. Mizumoto, and K. Tanaka, 1980, Some considerations of fuzzy conditional inference. Fuzzy Sets and Systems, 4:243–273
- [9] Karner, G, 1993, Metrics for Objectory. Diploma thesis, University of Linkoping, Sweden. No. LiTH-IDA-EX-9344:21, December .
- [10] Caroll, E.R, 2005 Estimating Software based on Use Case Points. OOPSLA'05, October 16-20. ACM 1-59593-193-7/05/0010
- [11] Diev, S, 2006 , Use Cases modeling and software estimation: Applying Use Case Points. ACM Software Engineering Notes,
- [12] Braz, M.R., Vergilio, S.R, 2006, Software Effort Estimation based on Use Cases. Proceedings of the 30th Annual International Computer Software and Applications Conference , (COMPSAC'06), IEEE
- [13] Robiolo, G., Badano, C., Orosco, R, 2009, Transactions and Paths: two use case based metrics which improve early effort estimation. 978-1-4244-4841-8/09, IEEE 2009
- [14] Wang, F., Yang, X., Zhu, X., Chen, L, 2009 , Extended Use Case Points Method for Software Cost Estimation. 978-1-4244-4507-0/09, IEEE
- [15] Ochodek, M., Nawrocki, J., Kwarciak, K., 2010, Simplifying effort estimation based on Use Case Points. Journal of Information and Software Technology
- [16] N.B. Ali, Ho D, Capretz L, 2011, Regression Model for Software Effort Estimation Based on the Use Case Point Method, International Conference on Computer and Software Modeling
- [17] Z.A. Lofti., 1965, Fuzzy sets, Info and Control, 8,338-353,.
- [18] K.Ramesh, 2013, Literature Survey On Algorithmic And Non- Algorithmic Models For Software Development Effort Estimation, International Journal Of Engineering And Computer Science ISSN:2319-7242 Volume 2 Issue 3
- [19] P. Y. Man, 2000, Enhanced Software Development Cost and Effort Estimation Using Fuzzy Logic Model. Master's Thesis, University of Malaya.
- [20] S.T.Moon, L.C.Teck, Enhance Software Development Effort And Cost Estimation Using Fuzzy Logic Model, Faculty of Computer Science and Information Technology ,University of Malaya
- [21] R. K. Clemmons ,2006, Project Estimation With Use Case Points, The Journal of Defense Software Engineering
- [22] G. Karner , 1993 , Resource Estimation for Objectory Projects. Objective Systems SF AB
- [23] A.J. Albrecht , Measuring Application Development Productivity. Proc. Of IBM Applications Development Symposium, Monterey, CA, 14-17 Oct.
- [24] J. Botzheim, B. Hámori, L. T. Kóczy , Applying bacterial algorithm to optimise trapezoidal membership functions in a fuzzy rule base, Department of Telecommunication and Telematics, Budapest University of Technology and Economics
- [25] N.B.Ali., L. Fernando Capretz and D. Ho, , 2016, "Enhancing Use Case Points Estimation Method Using Soft Computing Techniques", Journal of Global Research in Computer Science, vol. 1, no. 4, pp. 12-21.
- [26] J. Botzheim, B. Hámori, L. T. Kóczy, Applying bacterial algorithm to optimise trapezoidal membership functions in a fuzzy rule base, Department of Telecommunication and Telematics, Budapest University of Technology and Economics
- [27] Bente Anda, Hege Dreiem, Dag I.K. Sjøberg and Magne Jørgensen, 2001, Estimating Software Development Effort based on Use Cases –Experiences from Industry ,Springer Berlin Heidelberg, Volume 2185, pp 487-502.