

Improving the Accuracy of COCOMO II Using Fuzzy Logic and Local Calibration Method

Muhammad Baiquni, Riyanarto Sarno, Sarwosri

Department of Informatics Engineering, Institut
Teknologi Sepuluh Nopember
Surabaya, Indonesia
baiquni.8@gmail.com, riyanarto@if.its.ac.id,
sarwosri@if.its.ac.id,

Sholih

Department of Information Systems, Institut Teknologi
Sepuluh Nopember
Surabaya, Indonesia
sholih@is.its.ac.id

Abstract—Software Cost Estimation is a problem that often appears on the project of making a software. A poor estimate will result in a bad project management. To resolve this problem, a lot of the estimation model was introduced. Constructive Cost Model II (COCOMO II) is the most known and widely used model for software cost estimation. To estimate the cost of a software project, COCOMO II uses cost drivers, line of code and scale factors. However, the COCOMO II is still lacking in terms of accuracy. To improve the accuracy of COCOMO II, this research studies the influence of cost drivers and scale factors on improving the accuracy of effort estimation. Then, This Research using Fuzzy Logic and Local Calibration to improve the accuracy of COCOMO II. Fuzzy Logic with Gaussian Membership Functions (GMF) has been applied to redesign the effort multiplier. This research also implements Local Calibration to make a better accuracy of COCOMO II by using the value of the dataset as the input for the calculation of the calibration. This research gives a value effort multiplier that better suited for use in estimating and improving the method for search a new value of parameters to calculation COCOMO II that is more suitable for a category of a dataset used in this research. The Local Calibration result can be used for a future software project with similar category with the dataset used in this research. The result is a proposed model contributes on decreasing error significantly (about only 10% from MMRE COCOMO II or fuzzy model).

Keywords: COCOMO II; Fuzzy Logic; Software Cost Estimation; Gaussian Membership Function; Local Calibration.

I. INTRODUCTION

Cost Estimation of Software is one the main issues in the field of software project management [1]. The accuracy of cost estimation is important for a software developer to making good software development management [2]. Estimation is an important thing; low on estimation will results the software developer will be in a pressure to complete the software with a lack of resources. Then, high on estimation cause too many resources to be used for the project [3]. This issue makes many research in recent decades [2].

In this recent year, the software industry has been introduced with some models cost estimation software. A common model used to calculate cost estimation of software, the algorithm like Constructive Cost Model, IBM-FSD, Software Life Cycle Management (SLIM), and SPQR[3]. Among all the models, the COCOMO is the best known and most commonly used to calculate the estimated cost of the software. Barry Boehm published COCOMO in 1981[1]. This model has been developed from a dataset consisting of 63 data points. Each data point is divided into 16 variables. In 2000, Barry Boehm introduces the COCOMO II, which proved more accurate with the addition of cost drivers as well as some improvement from previous model. COCOMO II uses three aspects for effort estimation, namely an effort Multiplier (EM), lines of code (LOC) and the scale factor (SF) [4]. All these aspects will be calculated by an equation to produce the amount of effort in person months (PM) [2, 4].

However, COCOMO II has a shortcoming, such as the value of the estimate far exceeded the actual value or less than the actual value of a software project. The most common method to improve accuracy of effort estimates COCOMO II is the Genetic Algorithm, Fuzzy Logic and Artificial Neural Network(ANN). The first method to improve the accuracy of COCOMO II is Fuzzy Logic. Fuzzy logic is more appropriate when using the vague value in the calculation [3]. Fuzzy logic will be implemented to recalculate Effort Multiplier. Fuzzy membership functions have a several types such as triangular, Trapezoidal, and Gaussian membership function. Gaussian membership function (GMF) provides better result for accuracy by achieving a good transition in the membership function [3]. This research applies fuzzy logic with GMF as membership function for the cost drivers that have a quantitative value of effort multiplier. COCOMO II will be more accurate when the estimation effort closer to the actual effort.

The second method to be implemented is the Local Calibration. Local Calibration looking for the most suitable parameter value for COCOMO II estimation [5]. Local Calibration will calibrate the right value for multiplier and an exponent parameter of the COCOMO II equation. Local Calibration is implemented by using an equation to get the parameter value COCOMO II. Local Calibration uses dataset

of the software project as input for calibration equation [6]. By getting the right value for parameter estimation of COCOMO II, the estimation of cost software project would be more accurate. The Local Calibration result can be used for a future software project with similar category.

The other section of the paper is explained as follows, Section 2 explains COCOMO II and the methods used in this paper. Section 3 describes the work related to this research, such as the use of the method of fuzzy and local calibration. Section 4 presents the proposed method used in this research. Section 5 gives the experimental results and evaluation. The final part is the conclusion obtained in this study, i.e the increase of accuracy of COCOMO II.

II. LITERATURE REVIEW

A. Software Cost Estimation

Cost Estimation of Software is a calculation of effort value, staffing, and the time required to build software systems[5]. Software cost estimation is the first step for the company in making a project management software. The estimation is very important to predict the amount of effort, time, and staff, it is the estimates made in the early level of a project. However, estimation in software development early level is a difficult thing to do because software development projects have some factors, that factor has affect the most effort of software development is the changes that continue to occur in the software development process, as well as the interaction between developers with stakeholders [1, 2].

Many models have been proposed to be the base for the calculation of the estimated cost, time and effort. The existing model has similarities in concept but different in the use of the parameter value. Algorithmic models use mathematical calculations to do the estimation in software projects. Algorithmic mode is constructed by do research on costs and project attributes and find a suitable equation so that it will be close to the actual value [3].

B. COCOMO II Model

In 1981, Barry Boehms publish COCOMO. The COCOMO is a model obtained by retrieving data from many of software projects. COCOMO is the most widely used and the most logical of all the traditional model of cost estimation. The data is analyzed to find the most suitable equation as observation material. But COCOMO'81 have some difficulties in estimating cost to future software development practices. Then, COCOMO II Model was created to cover the needs of practices of future software development [7]. COCOMO II is a newer model of the original COCOMO and more accurate to the practice of the life cycle of the 21st century [4].

COCOMO II has a multiple attributes for a software project that is 17 Effort Multipliers, 5 Factors Scale, Software Line of Code (SLOC), and Effort estimates. Effort Multiplier categorized into 4 parts [1,8]. Here is the entire Cost drivers and Scale Factor:

A. Cost Driver

1. Software Cost Driver Product (Product)
 - a. Reliability of Software (RELY)
 - b. Size of Database (DATA)
 - c. Complexity of Software (CPLX)
 - d. Reusability Development (RUSE)

- e. Documentation (DOCU)
2. Platform
 - a. Constraint of Execution Time (TIME)
 - b. Main Storage (STOR)
 - c. Volatility of Platform (PVOL)
3. Personnel
 - a. Capability of Analyst (ACAP)
 - b. Capability of Programmer (PCAP)
 - c. Personnel Continuity (PCON)
 - d. Application Experience (APEX)
 - e. Platform Experience (PLEX)
 - f. Language and Tool Experience (LTEX)
4. Project
 - a. Use of Software Tools (TOOL)
 - b. Multisite Development (SITE)
 - c. Development Schedule (SCED)
- B. Scale Factor
 1. Precendentedness (PREC)
 2. Flexibility of Development (FLEX)
 3. Risk Resolution (RESL)
 4. Cohesion of Team (TEAM)
 5. Maturity Process (PMAT)

COCOMO II equation for effort estimation:

$$\text{Effort}(\text{PM}) = A \times (\text{KLOC})^E \times \prod_{i=1}^{17} \text{EM}_i \quad (1)$$

$$E = B + (0.01 \times \sum_{i=1}^5 \text{SF}_i) \quad (2)$$

Where, A = 2, 94; B = 0, 91, EM = Effort Multiplier, and SF = Scale Factor

The effort is obtained based on equation (1). Equation (2) is used to get exponent E which will be used in equation (1). The equation (1) using A parameter with a value of 2.94 and equation (2) using B-value of 0.91. In addition, there is a SF which is value of ScaleFactor from data. Then, EM is the value effort multiplier from 17 Cost Driver from data. This equation linking all the COCOMO II cost driver. PM is Person-Month, unit of calculation results.

C. Local Calibration

Local Calibration is the search for a suitable value in the calculation of effort estimation in COCOMO[10]. Calibration can be performed using a spreadsheet, or special software COCOMO II from Coconut, USC, or Calico [11,12]. This research uses equation for the calibration [6]. The COCOMO II model definition [8] has an explanation about calibration. Calibration in this model definition calibrates the parameter of COCOMO II equation. The COCOMO II can be calibrated for more suitable to some project category or same project developer. Calico is one of implementation from this calibration method [9].

The equation (1) and (2) have an A and B parameters, or multiplier and exponent based. This method will be used in determining the exact values for A and B [10]. Using the following equation to perform calibration :

$$X = \sum_{i=1}^n \log(\text{KSLOC}_i) \quad (3)$$

$$Y = \sum_{i=1}^n \log (ActualEffort_i \div EAF_i) - (SF_i \div 100) \quad (4)$$

$$\times \log(KSLOC_i)$$

$$Z = \sum_{i=1}^n Y_i \times X_i \quad (5)$$

$$A = 10^{((X^2 \times Y) - (X \times Z)) \div ((n \times X^2) - (X^2))} \quad (6)$$

$$B = ((n \times Z) - (X \times Y)) \div ((n \times X^2) - X^2) \quad (7)$$

Where n = Total Project, EAF = effort adjustment factor, A = New Value of A, and B = New Value of B

The equation (3) is used to obtain the total value of log (KSLOC) of n -data, which will be used in equation (7) and (8). The equation (4) calculates the total of the calculation using the Scale Factor, Effort Multiplier, Actual Effort and KSLOC of n -data. Then, equation (5) calculates the total value of (4) multiplied by (3). Then equation (7) and (8) is used to get the new value of the calculation parameters A in equation (1) and B in in equation (2). Result of Local Calibration can be used to estimate for the similar project used in the calibration [6,7].

D. Fuzzy Logic

Zadeh proposed Fuzzy logic in 1965. Fuzzy Logic is a mathematical system that is developed for modeling the word processing of the human brain. Fuzzy logic is created for solving inaccuracy in the measurement process [13]. Fuzzy Logic in the beginning was a concept of fuzzy set theory. Fuzzy set theory is a theory of classes that considered as an extension of the classical set theory [13].

Fuzzy Logic System is a system that has a correlation with fuzzy logic. Pure fuzzy logic systems, Takagi and Sugeno fuzzy system, and fuzzy logic system with fuzzifier and defuzzifier is the classification of a Fuzzy Logic System that has most popular used. Among all, Fuzzifier and Defuzzifier fuzzy logic systemic the most commonly used. Mamdani was the first to proposed Fuzzy logic system with Fuzzifier and Defuzzifier and successfully applied [5]. Fig. 1 shows fuzzy logic systems that have the fuzzifier and defuzzifier.

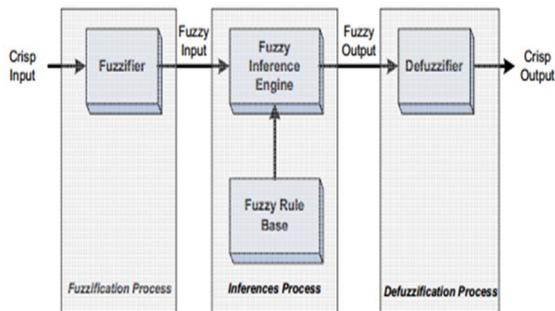


Fig. 1. Fuzzifier and Defuzzifier Fuzzy Logic System

III.RELATED WORK

Mauricio Aguiar [6] applied the calibration on COCOMO II. The Implementation of the calibration is using a Calico tool and then applied it for several categories of datasets. Results show that the value of the calibration results is used for a dataset that included one category. The experimental results of some of the data used to show the level of accuracy in the use of function point and line of code.

R Sarno, et.al [5] improves the accuracy of COCOMO's with the Fuzzy method and Neural Network on COCOMO. They studied how the influence of Gaussian Membership Function to Cost Driver COCOMO. In addition, by applying a multi-layer feed-forward neural network with back-propagation learning algorithm, the study conducted training on the dataset so as to produce more accurate value in the data testing. The Conclusion was accuracy of COCOMO's which could be improved in their research.

IV.PROPOSED WORK

First of all, this research learns about Cost Driver in COCOMO II. Every Effort Multiplier on 17 Cost Driver using linguistic values to represent each character. From the linguistic values ranging from Very Low to Extra High. This research uses a Effort Multiplier that is quantitative and has a major influence on the outcome of the estimated effort. After several experiments, this research wants to implement fuzzy logic on the cost drivers that can be translated into fuzzy membership function. Cost drivers which there are implemented fuzzy was DATA, TIME, STOR, SCED, PVOL, PCON, PCAP, ACAP, LTEX, APEX and PLEX. Fuzzy models used to redesign the Effort multiplier and using the description of Cost Driver as Fuzzy Logic input. For example, effort multiplier PCAP has an interval from very low to very high, the input value for very low is 15%, for low is 35% and so on. The difference for each level is the percentage of programmer capability to work in a team. This study uses a Fuzzy Logic with Gaussian Membership Function (GMF). GMF has the form of equations as follows.

$$\mu_{A_i}(X) = \text{Gaussian}(x, c_i, \sigma_i) = e^{-(x-c_i)^2/2\sigma_i^2} \quad (8)$$

Where C_i is center and σ_i is width of the curve

The equation (8) describes the Gaussian function, which contains the parameters the values center (c_i), and the width of the curve (σ_i). The resulting curve can be seen in Fig. 2.

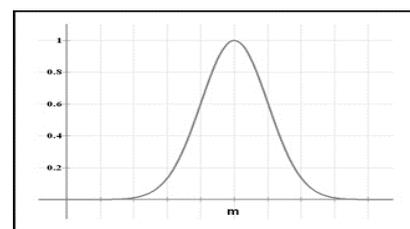


Fig. 2. Representation of the curve of Gaussian Membership Function

Fig. 2 shows the shape of the curve of the Gaussian Membership Function, where the value of m is the center of the curve.

GMF creates a good transition from one level to another. Fuzzy Logic is implemented using fuzzy logic tools in MATLAB software. There are several types of Fuzzy Logic such GMF, Function Triangle Membership, Membership Function trapezoid and other types in the tool box. This tool is named as Fuzzy Inference System Editor (FIS). FIS Editor with GUI, helps to make the input and output with an interval and number of membership functions more easier. With FIS editor, we can estimate the interval from every cost driver that we implemented fuzzy on it. Also, FIS Editor

allows us to create rules from input to output. In this experiment, the following rules will look like this:

- Rule 1: IF Input LTEX is low THEN Output LTEX is increased
- Rule 2: IF Input LTEX is nominal THEN Output LTEX is unchanged
- and ...

Increased and unchanged is the naming to ease in writing the rule on FIS Editor. Rule connects the input to the output. If the input value is on Low, then the output also adapts to the input.

Fig.3. shows the example input from LTEX effort multiplier. Description LTEX at every level translates to GMF. For example, low levels of LTEX have stated description of programmer experience using language and tools, interval for LTEX input is from less than or equal 2 months for very low, 6 months for low, and so on.

Fig.4 shows the output multiplier Membership Function LTEX. GMF value taken from the value of each level in the LTEX EM. For example, a low level has a value of 1.09 so that we set the interval set more accurate new value from 1.09. After making the input and output to GMF, set rules based on the rules that have been declared previously. Then the new value of each level is generated, and apply this to other qualitative EM.

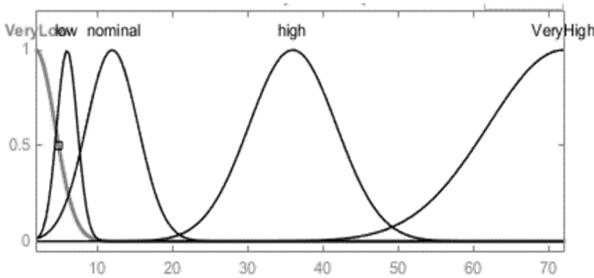


Fig. 3. Input LTEX on FIS Editor Matlab using GMF

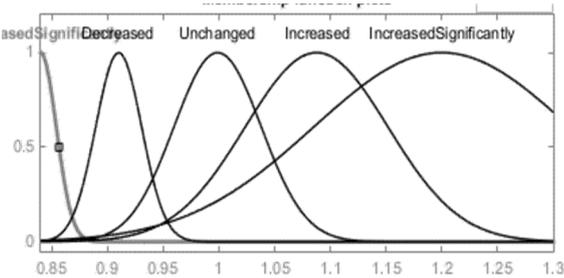


Fig. 4. Ouput LTEX on FIS Editor Matlab using GMF

After getting the new value of the EM, the next step is to change the values in your dataset with the values of effort the new multiplier after applying Fuzzy. Then the results is obtained from calculations of the MRE using the new EM value results from fuzzy.

Then, this research implements local calibration, there are many tools for local calibration, but this research uses CALICO spreadsheets for implementation. COCOMO II model definition manual already describes the calibration method. The model definition explains how to calibrate the COCOMO II parameters by using data from the dataset. This method will provide the parameter value Multiplier and Exponent for the equation calculation of estimated effort.

Using Calico, input for calibration is of the datasets that are used by this research, the equation that is used to calibrate as follows. Equation (3) to (7) is the equation of calculation for calibration, resulting in implementation Local Calibration using the some process to get the result.

Detail of the process using local calibration will describe in this step :

- Step 1
 - a. Calculate $\prod_{i=1}^{17} EM_i$
 - b. Calculate $\sum_{i=1}^5 SF_i$
 - c. $KSLOC_n$
 - d. Actual Effort_n
 Where n = data point n position, SF = Scale Factor, EM = Effort Multiplier
- Step 2 Calculating $X = \log(KSLOC_n)$
- Step 3 Calculating X^2
- Step 4 Calculate $Y = \log(Actual\ Effort_n \div EAF_n) - (SF_n / 100) \times \log(KSLOC_n)$
- Step 5 $Z = Y \times \log(KSLOC_n)$
- Step 6 Perform step 1 to step 5 for the entire n-data point in the dataset
- Step 7 Count total from each step, using equation (3), equation (4), and equation (5).
- Step 8 New calibrate value of A using equation (6).
- Step 9 New calibrate value of B using equation (7).

By using these two methods, this research improves the accuracy of effort estimation by using new parameter multiplication and exponent for estimate effort equation. This parameter value of COCOMO II estimate effort can be used for the similar software project.

V. EXPERIMENTAL RESULTS

This section presents the results achieved by applying fuzzy models are proposed for the dataset. Fuzzy Logic Model is implemented in Matlab while Calibration models implemented by Calico spreadsheet models.

The Turkish Software Industry has 12 data points. Each data point consists of 25 attributes which consists of Project ID, 5 Scale Factor, 17 Effort Multiplier in the value interval from VeryLow to ExtraHigh, Line of Code and Actual Effort. All data points will be used for calibration. Result from calibration can be used for the next project from similar category.

This study attempted to compare the accuracy of estimation of effort with real effort. To evaluate the accuracy of estimates of effort, this study uses the most common evaluation method such as Magnitude Relative Error (MRE).

$$MRE = \frac{|Actual\ Effort - Estimated\ Effort|}{Actual\ Effort} \times 100 \quad (9)$$

$$MMRE = \frac{1}{N} \sum_{i=1}^N MRE_i \quad (10)$$

Where N = total amount of data point

From equation (9) and (10), the value of which has a smaller MRE is closer to the actual effort. MMRE is mean of MRE, MMRE representate mean of MRE in all data point of data set.

Before that, this is the results of the fuzzy implementation, new interval value for effort multiplier. These values can be seen in the Table 1.

TABLE 1 RESULT OF FUZZY IMPLEMENTATION ON EFFORT MULTIPLIER

Cost Driver	Range
LTEX	0.85 – 1.16
DATA	0.90 – 1.24
STOR	1.00 – 1.45
PCAP	0.76 – 1.34
APEX	0.81 – 1.11
PLEX	0.857 – 1.13
ACAP	0.74 – 141
PVOL	0.87 – 1.23
PCON	0.813 – 1.26
TIME	1.00 – 159
SCED	1.00 – 1.31

The results in Table 1 show the range or interval for new effort multiplier. For example, previously LTEX has interval from 0.84 to 1.2, at table 1 changed to 0.85 to 1.16. Change at the interval which gives influence on the accuracy of COCOMO II.

Then, the following is the results from the calibration, where the result is a new value for the parameter A and B in the equation of COCOMO II. The result value of A (Multiplier) is **3.05** and for B (Exponen) is **0.24**. The results are obtained by performing calibration on the dataset.

The Data for calibration was obtained from a Turkish dataset, these data have 12 data points. For example the calculation of calibration used a data point number n, where n is 1, which has the following values: Actual Effort : 1.2, KSLOC: 3, Effort Multiplier : 0.35, and Sum Of Scale Factor : 19.92.

Based on the steps described earlier, it will be calibrated as follows.

- Step 1 Actual Effort : 1.2
KSLOC : 3
Effort Multiplier : 0.35
Sum Of Scale Factor : 19.92
- Step 2 $\text{Log}(3) = 0.477$
- Step 3 $\text{Log}(3) * \text{Log}(3) = 0.227$
- Step 4 $\text{Log}(1.2/0.35) - (19.92 / 100) * \text{log}(3) = 0.439$
- Step 5 $\text{Step4} * \text{log}(KSLOC) = 0.439 * 0.477 = 0.2095$
- Step 6 Perform step 2 to step 5 for the entire data point in the dataset
- Step 7 Furthermore, the total value after performing step 1 to step 5 on the entire data point, obtained:
Total Sum of X is 11.07, X² is 14.46, Y was 8.50, and Z is 8.87
- Step 8 New Calibrate Value of A =
 $10^{(14.46 * 8.50 - 11.066 * 8.87) / (12 * 14.46 - 11.07 * 11.066)} = 3.05$

Step 9 New Calibrate Value of B =
 $(12 * 8.87 - 11.07 * 8.50) / (12 * 14.46 - 11.07 * 11.07) = 0.24$

Step 8 is the result of the calibration of the value of the multiplier is A in (1), an A change from 2.94 into 3.05. Step 9 is the result of the calibration of the exponent value that B, in (2) the value of B changed from 0.91 to 0.24.

So, the result is a suitable parameter to the dataset turkish software industry. After that, it is calculated the value of the MRE by using the new parameter and new effort multiplier. The result can be seen in Table 2.

TABLE 2 COMPARISON OF EFFORT ESTIMATION RESULTS IN MRE

Project ID	MRE (%) COCOMO II	MRE (%) Fuzzy	MRE (%) Fuzzy + Local Calibration
1	190,67	191,75	44,98
2	42,84	45,07	5,41
3	106,76	101,69	20,64
4	1032,58	996,62	143,22
5	1478,89	1475,73	166,35
6	26,05	31,12	88,61
7	14,43	20,89	50,87
8	2841,79	2706,51	186,47
9	1553,36	1545,54	28,65
10	1499,90	1478,73	99,79
11	7,61	3,54	18,92
12	2,75	3,21	27,05
MMRE	733,14	716,70	73,41

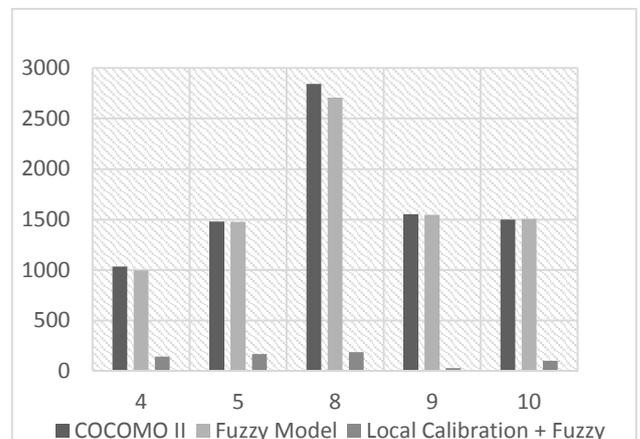


Fig. 5. Bar chart showing effort estimation accuracy in MRE

From Fig.5, the result of calibrating and fuzzy model can decrease the value of MRE, decreased value of MRE due to parameter values from the calculation of COCOMO II has been more suitable for this dataset category. The results of the use of fuzzy can decrease the value of the MMRE down to 16.4% (from 733.14% down to 716.7%). The results obtained because of the effort multiplier that is a result of the implementation has a better compatibility compared with standard effort multiplier values of COCOMO II. The proposed methods, fuzzy and local calibration MMRE, have the smallest value compared with COCOMO II or fuzzy model. For fuzzy and local calibration it has average MMRE

= 73,41%, whereas COCOMO II has MMRE= 733,14% and fuzzy model has MMRE= 716,7% (see Table 2). The results proved that the proposed method is successfully increasing the accuracy of COCOMO II significantly (about only 10% from MMRE COCOMO II or fuzzy model). In addition, the results proved that the method proposed also provide value of effort multiplier which is more suitable for use in doing estimation. Then, values of calibration results will be reusable for a new software development project.

Beside using MRE, to give a comparison with existing methods, carried out using 4 comparison of estimate effort on ERP project. EZERP is an ERP project made by students of ITS, then compared with odoo, Adempiere, and inoERP. The results of estimation using COCOMO II parameters of the comparison with the results of the implementation of proposed method. The results obtained indicated by Table 3. Table 3 describes the results of estimation calculations of COCOMO II with normal parameter and COCOMO II with the proposed method. Those results also use the same values of cost drivers and the scale factor, so that the comparison is done with balanced. This experiment uses nominal range for the entire cost driver so that the value was 0.99 for the effort multiplier and scale factor has a value of 17.87. Using the two values are then calculated using equation (1) and (2) the results obtained in accordance with Table 3.

TABLE 3 COMPARISON OF EFFORT ESTIMATION FROM ERP

Method	EZERP	Odoo	Adempiere	inoERP
COCOMO II(PM)	1596.13	4889.59	29282,64	2038.03
Proposed Method(PM)	34.25	52.68	104.85	37.62

The results of the calculation of the proposed method use the value of the parameter $A = 3.05$ and $B = 0.24$, and the use of the method of fuzzy multiplier effort. Calibration results have a smaller value because it uses the parameter values derived from the calibration of turkish dataset, that have more overestimate so that is considered suitable for use in the calculation of ERP comparison..

VI. CONCLUSION

In the process of making software, a good management is important, because with good management then the project goes well and according to plan. Good management is the result of a precise estimate of the software project will be done. The more accurate estimate effort and time of software project then the software developer can handle software development resources efficiently.

COCOMO is the most famous and most used to calculate the cost of the software. COCOMO II gives a way to be able to estimate the effort or price and time a software project in order to set up a strong and good management. However, the COCOMO II has deficiencies in the accuracy of the estimate, so it needs a method to make a better accuracy of COCOMO II estimation.

The method this research uses Fuzzy Logic and Local Calibration to increase the accuracy of COCOMO II has been proved. Fuzzy logic with a Gaussian Membership Function (GMF) can redesign Effort Multipliers COCOMO II and give a good result. This method can make a smoother transition for better result, which means it can provide a more accurate

Effort Multipliers. Then, Local Calibration will search the value of parameter in the equation, so the estimation effort with that value can improve the accuracy and the result of local calibration can be used to estimate future software project similar with dataset category. This research gives a value of effort multiplier that is better suited for cost estimation software projects as a result. Then, with local calibration this research gives the new value for the parameter calculation of COCOMO II on a dataset. The proposed model has proved by applying in the dataset Turkish Software Industry. Based on the value of the MMRE proved that the proposed method can improve the accuracy of COCOMO II or fuzzy model significantly (about only 10% from MMRE COCOMO II or fuzzy model). In addition, this research method gives results which can be used again to do the estimate on future software project development with the similarity of adataset category as used in this research.

VII. REFERENCES

- [1] B. Boehm, "Software Engineering Economics," New Jersey, USA, Prentice-Hall, Englewood Cliffs, pp. 99-150, 1984. DOI: 10.1007/978-3-642-48354-7_5
- [2] R. Sarno, J. Sidabutar and Sarwosri, "Comparison of different Neural Network architecture for software cost estimation," *International Conference on Computer, Control, Informatics and its Applications (IC3INA)*, pp. 28-33, 2015. DOI: 10.1109/IC3INA.2015.7377748
- [3] C. S. Reddy and K. Raju, "An improved fuzzy approach for COCOMO's effort estimation using gaussian membership function," *Journal of Software*, vol. 4, pp. 452-459, 2009. DOI: 10.4304/jsw.4.5.452-459
- [4] B. Barry, B. Clark and E. Horowitz, "Cost Models for Future Software Life Cycle Processes: COCOMO 2.0," *Annals of Software Engineering*, vol. 1, no. 1, pp. 57-94, 1995. DOI: 10.1007/BF02249046
- [5] R. Sarno, J. Sidabutar and Sarwosri, "Improving the accuracy of COCOMO's effort estimation based on neural networks and fuzzy logic model," *Internation Conference on Information & Communication Technology and System (ICTS)*, pp. 197-202, 2015. DOI: 10.1109/ICTS.2015.7379898
- [6] M. Aguiar, "COCOMO II Local Calibration Using Function Points," in *20th International Forum on COCOMO and Software Cost Modeling*, California, 2005.
- [7] B. Boehm, B. Clark and S. Devnani, "Calibrating the COCOMO II Post-Architecture Model," 2000. DOI: 10.1109/ICSE.1998.671610
- [8] Center of Software Engineering, "COCOMO II Model Definition Manual," in *Software Engineering Departmen of USC*, California, 2000.
- [9] Softstar System, "CALICOCalibration Software,," Softstar System, 2015. [Online]. Available: [Accessed 2015].
- [10] R. Gulezian, "Reformulating and Calibrating COCOMO," *Journal System Computer*, pp. 235-242, 1991. DOI: 10.1016/0164-1212(91)90018-2
- [11] T. Menzies, Z. Chen, J. Hihn and S. Stukes, "Validation Methods for Calibrating Software Effort Models," in *27th International Conference on SoftwareEngineering*, pp. 587-595, 2005. DOI: 10.1109/ICSE.2005.1553605
- [12] T. Lee, D. Choi and J. Baik, "Empirical study on enhancing the accuracy of software cost estimation model for defense software development project applications," in *12th International Conference on Advanced Communication Technology (ICACT)*, pp. 1117-1122, 2010.
- [13] R.R. Putri, R. Sarno, D. Siahaan, A.S Ahmadiyah, S. Rochimah, "Accuracy Improvement of the Estimations Effort in Constructive Cost Model II Based on Logic Model of Fuzzy." *Adv Sci Lett*. 2017;23(3):2478-2480. doi:10.1166/asl.2017.8767.